Convolutional neural networks for illumination estimation in complex illumination environments

Domislović, Ilija

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Ilija Domislović

CONVOLUTIONAL NEURAL NETWORKS FOR ILLUMINATION ESTIMATION IN COMPLEX ILLUMINATION ENVIRONMENTS

DOCTORAL THESIS

Zagreb, 2023

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Ilija Domislović

CONVOLUTIONAL NEURAL NETWORKS FOR ILLUMINATION ESTIMATION IN COMPLEX ILLUMINATION ENVIRONMENTS

DOCTORAL THESIS

Supervisor: Professor Marko Subašić, PhD

Zagreb, 2023

Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Ilija Domislović

KONVOLUCIJSKE NEURONSKE MREŽE ZA PROCJENU OSVJETLJENJA U SLOŽENIM SVJETLOSNIM UVJETIMA

DOKTORSKI RAD

Mentor: Prof. dr. sc. Marko Subašić

Zagreb, 2023.

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

Supervisor: Professor Marko Subašić, PhD

The thesis has 127 pages.

Thesis number: _____

About the Supervisor

Marko Subašić was born in 1976 in Zagreb, Croatia. He graduated from the Faculty of Electrical Engineering and Computing (FER) in Zagreb in 1999. He received his master's degree in electrical engineering from FER in Zagreb, in 2003. He received his Ph.D. in electrical engineering from FER in Zagreb in 2007. In 2022, he was elected full professor at FER. Dr. Subašić is conducting research in the field of digital image processing and analysis with applications in medicine, transport, remote sensing, and industry, as well as neural networks, machine learning, and other methods of artificial intelligence. Dr. Subašić is a member of the group for digital image processing at FER. He is a member of the following professional organizations: IEEE (Institute of Electrical and Electronics Engineers) and IEEE Computer Society, the Scientific Center of Excellence for Data Science and Cooperative Systems, the Center of Excellence for Computer Vision, and the Croatian Society for Medical and Biological Engineering. Dr. Subašić has actively participated in the organization of several international conferences. He has participated in several scientific projects of the Croatian Ministry of Science, the Croatian Science Foundation, competitive EU projects, and commercial projects.

O mentoru

Marko Subašić rođen je 1976. u Zagrebu, Hrvatska. Diplomirao je na Fakultetu elektrotehnike i računarstva (FER) u Zagrebu 1999. godine. Magistrirao je u polju elektrotehnike na FER-u u Zagrebu 2003. godine, a doktorirao u polju elektrotehnike na FER-u u Zagrebu 2007. Godine 2022. izabran je u zvanje redovitog profesora na FER-u. Dr. Subašić provodi istraživanja u području digitalne obrade i analize slike s primjenama u medicini, prometu, daljinskim istraživanjima i industriji, kao i istraživanje neuronskih mreža, strojnog učenja i drugih metoda umjetne inteligencije. Dr. Subašić je član grupe za digitalnu obradu slike na FER-u. Član je sljedećih strukovnih organizacija: IEEE (Institute of Electrical and Electronics Engineers) i IEEE Computer Society, Scientific Center of Excellence for Data Science and Cooperative Systems, Centra izvrsnosti za računalni vid i Hrvatskog društva za biomedicinsko inženjerstvo i medicinsku fiziku. Dr. Subašić je aktivno sudjelovao u organizaciji nekoliko međunarodnih skupova. Sudjelovao je u nekoliko znanstvenih projekata Ministarstva znanosti RH, Hrvatske zaklade za znanost, konkurentskih EU projekata i komercijalnih projekata.

Preface

Thanks to everyone who helped me on this journey. I am grateful to my supervisor, Prof. Marko Subašić who guided me during the creation of my thesis. I would also like to thank my other colleges, especially the ones present in d159, with whom I discussed various topics that led to much of my research.

I also need to thank all my friends who listened to my ramblings about topics unknown to them. I would like to thank my mother, Jadranka, for being supportive throughout my education. Finally, I would like to give a big thanks to my girlfriend, Lea, for being there and supporting me.

Abstract

The Human Visual System is an interesting part of the human body that allows us to perceive the world around us. One of its most fascinating parts is its adaptability. The color of illumination has a significant effect on the color of an object illuminated by said illumination, but we will perceive the color of the object as relatively constant regardless of whether the illumination color is blue, orange, or yellow. This ability of the Human Visual system to ignore the illumination color when looking at an object is called color constancy. Unlike humans, cameras cannot automatically remove illumination color from an object, which results in unnatural-looking images. This creates a need to develop a method that emulates the adaptability of the Human Visual System. Color is also an important object feature in many computer vision tasks such as object tracking, which relies on the constancy of an object's color. In a camera image processing pipeline, the removal of illumination chromaticity is called white balancing. Many different methods have been developed ranging from simple methods that use image statistics to complex learning-based methods. Recent research has shown that learning-based methods achieve the best results. The methods that achieve state-of-the-art results most commonly use convolutional neural networks. For the proper training of learning-based methods, a large number of samples is needed. This presents a problem because the creation of quality images with multiple illuminants is a very laborious process so there are not many multi-illuminant datasets. The primary focus of current research has been the simplest most common variation of the white-balancing problem, where the image contains only one uniform illuminant. This thesis explores both the development of methods for illumination estimation as well as methods for the creation, labeling, and validation of images with multiple illuminants. For illumination estimation five methods, one for single-illuminant and four for multi-illuminant estimation were developed. The proposed single-illuminant method is a lightweight convolutional neural network that achieves state-of-the-art results on several existing datasets. The multi-illuminant methods were created for three different situations, one method for multi-illuminant estimation when the number of illuminants is known apriori, one method for illumination estimation is performed on a patch-by-patch basis, and two different methods for illumination estimation for each pixel separately. In addition to estimation methods two dataset creation methods were developed, one for the creation and labeling of real-world two-illuminant images and the other for the automatic creation of images with a variable number of non-uniform illuminants. The performed experiments show that all proposed methods achieve comparable or better results than methods from the literature.

Prošireni Sažetak

Ljudski vizualni sustav zanimljiv je dio ljudskog tijela koji nam omogućava da vidimo svijet oko nas. Jedna od najinteresantnijih dijelova ljudskog vizualnog sustava je njegova prilagodljivost na promjene u okolini. Boja osvjetljenja značajno utječe na boju predmeta osvijetljenog tim svijetlom, ali ljudi percipiraju boju predmeta kao relativnu konstantu bez obzira na to je li ta boja plava, narančasta ili žuta. Ova sposobnost ljudskog vizualnog sustava da zanemari boju osvjetljenja kada gleda predmet naziva se postojanost boja. Kamere takvo što ne mogu to učiniti automatski, što rezultira slikama koje izgledaju neprirodno. To stvara potrebu za razvijanjem metode koja oponaša tu sposobnost ljudskog vizualnog sustava. Boja je važna komponenta predmeta te se mnogi problemi računalnog vida, poput praćenja predmeta, oslanjaju upravo na konzistentnost boje predmeta. U postupku obrade slike u kameri uklanjanje kromatičnosti osvjetljenja naziva se bijelo balansiranje.

Prvo poglavlje rada uvod je u tematiku disertacije u kojem će se predstaviti kratki opis problema kojim se disertacija bavi. Osim toga, u ovom poglavlju daje se pregled znanstvenih doprinosa ove disertacije. Doprinosi se mogu podijeliti u dvije kategorije metode bazirane na dubokom učenju za procjenu boje osvjetljenja slike te metode za izradu i označavanje slika osvijetljene s više izvora svijetla.

Drugo poglavlje daje pregled ljudskog vizualnog sustava. U ovom poglavlju opisuju se glavni dijelovi ljudskog vizualnog sustava i koju ulogu svaki dio ima kako bi čovjek imao sposobnost gledanja. U ovom poglavlju opisuje se fascinantna sposobnost ljudskog vizualnog sustava da se prilagodi promijeni jačine i promjeni boje osvjetljenja. Fokus ove disertacije upravo je prilagodljivost ljudskog vizualnog sustava na promjenu boje koja se zove postojanost boja, a u ovom poglavlju opisuje se kako se naše znanje postojanosti boja razvijalo kroz povijest i do kojih smo zaključaka došli.

Treće poglavlje opisuje računalnu postojanost boja. U ovoj poglavlju dan je fizički model koji pokazuje o čemu sve ovisi boja objekta kojeg percipirano. Osim toga, dana je i matematička definicija digitalne slike te definicija piksela. Na kraju poglavlja dan je opis metode kojom se može emulirati postojanost boja ljudskog vizualnog sustava. Uz pomoć te metode može se ukloniti kromatski utjecaj svijetla na scenu tako da slika izgleda kao da je fotografirana pod savršeno bijelim svijetlom.

Četvrto poglavlje opisuje metode dubokog učenja. Svi doprinosi ove disertacije koje se koriste za imitiranje postojanosti boje bazirane su na dubokom učenju. Osim toga, u poglavlju se daje kratka povijest metoda dubokog učenja. Značajan dio metoda dubokog učenja je backpropagation algoritam koji je nastao 1986. godine. Backpropagation se koristi kako bi metoda dubokog učenja "naučila" kako riješiti problem. Pomoću njega se računaju gradijenti kojima se mijenjaju vrijednosti parametra koje metode dubokog učenja koriste kako bi riješili zadani

problem. Postoje mnogo različitih metoda dubokog učenja, a metode iz ove disertacije koriste neuronske mreže. Neuronske mreže su kompleksni algoritmi koji se sastoje od mnogo slojeva. Skup raznih slojeva te njihov redoslijed određuju arhitekturu neuronske mreže. Istraživanja su pokazala da se precizniji rezultati za zadatke iz računalnog vida postižu iz neuronskih mreža koje koriste konvoluciju. Opis često korištenih slojeva, među kojima je i konvolucijski sloj se nalaze u ovom poglavlju.

Peto poglavlje opisuje razne skupove podataka. Na početku poglavlja se opisuju razni skupovi trenutno dostupni na internetu. Postoje nekoliko velikih skupova slika koji se koriste za evaluaciju novih metoda za emuliranje postojanosti boja. Ti skupovi imaju tisuće slika, ali su namijenjeni za jedan specifičan slučaj postojanosti boja, a to je kada slika sadrži samo jedan izvor svijetla koji jednoliko osvjetljuje cijelu scenu. Postoje skupovi slika s više izvora osvjetljenja, ali ti skupovi uglavnom sadrže malen broj slika. To predstavlja problem jer su istraživanja pokazala da najbolje rezultate za procjenu osvijetljena postižu metode dubokog učenja, a kako bi metoda dubokog učenja dala kvalitetnu procjenu osvjetljenja potreban je veliki skup slika s raznolikim bojama svjetlosti. S obzirom na to da nije postojao veliki skup slika s više osvjetljenja na početku istraživanja, za ovu disertaciju napravljene su dvije metode za sakupljanje, označavanje i procjenu kvalitete slike scene osvjetljenja s više izvora svijetla. Veliki problem kod označavanja slike s više osvjetljenja je da je svakom pixelu u toj slici potrebno pridijeliti boju osvjetljenja. Kad slika ima više osvjetljenja ta osvjetljenja se mogu stopiti i stvoriti nove boje osvjetljenja ovisno o jačini i poziciji izvora u sceni. Prva razvijena metoda za izradu slika s više osvjetljenja pojednostavljuje postupak pridjeljivanja osvjetljenja svakom pixelu. Slike u skupu podataka koja je napravljena uz pomoć ove metode sadrže dva različita osvjetljenja i dobro definirane granice između regija koja su osvijetljena jednim svijetlom i regija koja su osvijetljena drugim svijetlom. Za izradu takvih slika potrebne su dvije vrste izvora svijetla. Prvi tip osvjetljenja je ambijentalno svijetlo koje jednoliko osvjetljuje cijelu sliku. Drugi izvor svijetla mora osvjetljavati samo dio scene i mora biti jačeg intenziteta nego ambijentalno svijetlo. Zbog razlike u jačini izvora, utjecaj ambijentalnog izvora u regijama gdje postoje oba svjetla postane zanemariv. Time dobijemo sliku s dvije regije osvjetljenja koje imaju dobro definarane granice. Jednostavan primjer takve situacije je sunčan dan gdje je dio regije u sjeni. U ovom slučaju ambijentalno svijetlo je nebo, a jače svijetlo je sunce. Takva situacija znatno olakšava označavanje slika, ali se svaka slika svejedno mora ručno označiti. U sklopu izrade skupa podataka napravljena je i metoda za verifikaciju točnosti boje osvjetljenja slike. Boja osvjetljenja se računa pomoću kalibracijskog objekta koji je prisutan u svakoj slici. Ti kalibracijski objekti sadrže sive i bijele plohe, a boja osvjetljenja se računa tako da se izračuna prosječna boja jedne od sivih ploha. Točnost tih kalibracijskih objekata je provjerena na način da se u scenu stavilo više kalibracijskih objekata te da se iz svakog izračunala boja osvjetljenja. Te boje su se usporedile kako bi se saznala njihova sličnost. Istraživanje je pokazalo da se

izvučene boje ambijentalnog svjetla mogu znatno razlikovati. Isti eksperiment je napravljen za jače svijetlo i pokazalo se da kalibracijski objekti daju neznačajno različite boje u istoj sceni. Zbog toga se za izračun ambijentalnog svijetla u svakoj slici koriste dva kalibracijska objekta a za izračun jačeg svijetla se koristi jedan kalibracijski objekt. Još jedna važna stvar za skup slika za procjenu boje osvjetljenja je raznolikost kamera s kojima se sakupljaju slike. Razni proizvođači kamera koriste razne kamera senzore. Ako poslikamo istu scenu s dva različita kamera senzora dobit ćemo dvije slike koje imaju vrijednosti piksela koje se malo razlikuju. Korištenje jedne kamere pri izradi skupa predstavlja problem metodama dubokog učenja jer će se metoda naučiti procijeniti boju osvjetljenja samo za slike te kamere. Zbog toga za izradu ovog skupa podataka korišteno je 5 različitih kamera, Canon EOS 5D Mark II, the Canon EOS 550D, Panasonic DMC-FZ1000, Sony DSLR-α300 i Samsung ISOCELL Plus GW1 1/1.72" senzor u Motorola One Fusion+ mobitelu. Još jedan važan faktor pri izradi skupa slika je izvor svijetla. Izvori svijetla mogu se podijeliti u dvije kategorije, prirodni izvori i umjetni izvori svijetla. Kako bi skup bio što generalniji potrebno je sakupiti slike s raznolikim izvorima svijetla. Iz tog razloga, u skupu postoje slike napravljene po danu, noći, vanjskom i unutrašnjem prostoru. Skup podataka sadrži 2500 različitih slika, a sa svakom je kamerom poslikano oko 500 slika. U disertaciji definirano je kako se pravilno moraju obraditi i koristiti slike da bi se skup mogao koristiti za usporedbu metoda. Ovaj skup izvrstan je za usporedbu metoda, no on ima i svojih ograničenja obzirom da svaka slika sadrži točno 2 osvjetljenja i u slikama nema regija gdje se dva ili više osvjetljenja preklapaju. Zbog toga je razvijena još jedna metoda za izradu slika s više izvora osvjetljenja koja je opisana u ovoj disertaciji. Glavni doprinos ove metode je taj da omogućuje izradu proizvoljnog broja svjetlosnih okolina za jednu scenu. Uz ovu metodu nije potrebno ručno označavati boju svijetljenja svakog piksela. To se postiže tako da se uzme više slika iste scene. Prije slikanja scene, pred leću kamera stavlja se svjetlosni filter. S filterima raznih boja dobivaju se slike iste scene s raznim osvjetljenjima bez mijenjanja sadržaja scene. Kako bi scenu mogli koristiti u ovoj metodi ona mora biti osvijetljena samo s jednim svijetlom koje ju uniformno osvjetljuje. U takvoj situaciji poznata je boja osvjetljenja svakog piksela. Te razne slike iste scene možemo kombinirati da dobijemo novu sliku scene, ali ovaj puta s više izvora osvjetljenja. Zbog fizičkih svojstva svjetlosti to možemo postići jednostavnim zbrajanjem tih raznih slika iste scene. Boja predmeta osvijetljenog s više svijetla je jednostavno linearna kombinacija boja kada je predmet osvijetljen svakim izvorom zasebno. Pri izradi ovog skupa pazilo se na to da se postigne što veća raznolikost scena kao i za prvu metodu. To znači da je i ovaj skup stvoren uz pomoć tri različite kamere te da slike sadrže razne scene po danu, noći, u vanjskom i unutrašnjem prostoru. Također se pazilo da izračunata osvjetljenja budu što točnija. Sakupljeno je 7800 slika od 300 različitih scena uz pomoć 25 svjetlosnih filtera i sa svakom kamerom poslikano je oko 100 scena. Za spajanje slika postoji beskonačno mogućnosti. U ovoj disertaciji predstavlja se jedna metoda razvijena za ovaj skup

podataka. Metoda kreira dva tipa maski. Svaka maska dijeli sliku u nekoliko regija. Prva maska dijeli sliku u regije tako da stvori nekoliko nasumičnih pravaca koji dijele sliku u regije. Druga maska dijeli sliku tako da iz konteksta scene izvuče superpiksele. Svaki superpiksel predstavlja jednu regiju. Regije obje maske se zatim nasumično spoje i svakoj regiji se pridijeli jedna slika scene. Tijekom izrade skupa slika napravljena je analiza kako svjetlosni fliteri utječu na izgled slike te kako izgleda korigirana filter slika. Istraživanje je pokazalo da postoji nekoliko filtera koji imaju negativan utjecaj na sliku. Slike koje su napravljene s tim filterima ne mogu se pravilno korigirati. Neki od tih filtera uzrokuju da slike izgube žarkost boja dok neki filteri uklanjaju svu boju stvarajući crno bijelu sliku. Iz tog razloga napravljena su dva skupa slika s više osvjetljenja, jedna koja koristi loše filtere i jedna koja ih ne koristi. U oba slučaja za svaku scenu izrađeno je 20 slika koje mogu imati od 1 do 10 različitih osvjetljenja.

Šesto poglavlje sadrži sve metode razvijene za procjenu osvjetljenja. U ovoj disertaciji opisano je 5 različitih metoda od kojih je jedna za procjenu osvjetljenja u slikama s jednim osvjetljenjem i četiri metode za procjenu osvjetljenja u slikama s više izvora osvjetljenja. Prva opisna metoda je metoda za procjenu osvjetljenja u slikama s jednim izvorom svijetla. Istraživanja su pokazala da najbolje rezultate za procjenu osvjetljenja postižu metode bazirane na učenju, konkretno metode koje koriste konvolucijske neuronske mreže. One su toliko precizne da postižu rezultate koji su bolji ili usporedivi s ljudskim vizualni sustavom. No, problem kod tih metoda predstavlja činjenica da su izuzetno računalno zahtjevne i neprikladne za slabija računala, kao na primjer mobitel. Zbog toga je u ovoj disertaciji predstavljena jednostavna konvolucijska neuronska mreža koja može raditi u stvarnom vremenu. Metoda je testirana na nekoliko poznatih baza slika s jednim osvjetljenjem. Iako metoda nije računalno zahtjevna s malim brojem parametra, ona postiže rezultate sumjerljive ili bolje od ostalih metoda baziranih na učenju. Uz tu metodu u disertaciji je opisana nova metoda za augmentiranje slika kojoj se smanjuje utjecaj kamera senzora na točnost metode. Metode za procjenu više svjetla napravljene su za specifične podskupove problema procjene više osvjetljenja. Prva metoda za procjenu više osvjetljenja opisana u ovoj disertaciji radi procjenu osvjetljenja kada je poznat broj osvjetljenja u slici. Metoda je bazirana na postojećoj metodi za procjenu jednog osvjetljenja iz literature. Ta metoda je izabrana zato što postiže dobre rezultate i koristi mehanizam pozornosti za pronalazak regija koje sadrže korisne informacije za procjenu svjetla. Regije koje sadrže malen broj ploha nisu korisne za procjenu. Primjerice, za jednobojni zid ne možemo znati je li boja na slici stvarna boja zida, boja osvjetljenja ili kombinacija boja. Kad slika sadrži više osvjetljenja, regije koje sadrže osvjetljenje koje se trenutno ne procjenjuje, ne sadrže korisne informacije i trebale bi se ignorirati. Metoda je testirana na skupovima predstavljenim u ovoj disertaciji te postiže rezultate sumjerljive ili bolje od ostalih metoda. Mana ovog pristupa je ta da je potrebno znati koliko osvjetljenja slika ima. Postojeće metode dijele sliku na mnogo malih sličica te koriste pretpostavku da zato što su isječci tako mali one sadrže samo jedno

svijetlo. Metoda predstavljena u ovoj disertaciji također dijeli sliku na male isječke. Problem s tim pristupom je taj da ti mali isječci često ne sadrže dovoljno informacija za točnu procjenu osvjetljenja. Metoda rješava ovaj problem tako da uz lokalne informacije iz isječka koristi i globalne informacije iz cijele slike za procjenu osvjetljenja jednog isječka. Provedeni su i eksperimenti kojima se dokazalo kako dodavanje globalne informacije slike značajno povećava točnost metode. Metoda je testirana na skupovima predstavljenim u ovoj disertaciji te postiže sumjerljive ili bolje rezultate od ostalih metoda baziranih na učenju koje dijele sliku na male isječke. Nedostatak tog pristupa je da metode koje ga koriste zapravo rade procjenu jednog osvjetljenja i kad isječak sadrži više osvjetljenja one neće davati točne rezultate. Zbog toga su u ovoj disertaciji opisane još dvije neuronske mreže koje rade procjenu osvjetljenja za svaki piksel. Metode imaju sličnu arhitekturu i obje su bazirane na metodi za procjenu jednog osvjetljenja opisanoj u ovoj disertaciji. Prva metoda dijeli sliku na male isječke i za svaki piksel u svakom isječku radi procjenu boje osvjetljenja, dok druga metoda obrađuje cijelu sliku odjednom i procjenjuje osvjetljenje za svaki piksel. Svaka metoda ima svoje prednosti i nedostatke i nijedna se nije pokazala kao apsolutno bolja od drugih. Metode su također uspoređene s postojećim metodama iz literature i obje postižu usporedive ili bolje rezultate od metoda iz literature.

Sedmo poglavlje predstavlja zaključak ove disertacije. Naglašavaju se sva postignuća ostvarena s novo razvijenim metodama za procjenu osvjetljenja i s novo izrađenim skupovima podataka.

Ključne riječi: strojno učenje, neuronske mreže, obrada slika, analiza slika, skupovi podataka, postojanost boja

Contents

Chapter 1

Introduction

1.1 Overview

In today's world, humans create large amounts of data on a daily basis. One of the most common types of data we create are images. With mobile smartphones, it is easier than ever for each person to create hundreds of images daily. With so many images, the need for fast and accurate image processing is needed. The camera processing pipeline is multi-faceted with many important steps. One of the first steps is white-balancing. White-balancing is the process of removing the chromatic effect the scene illumination has on the colors of objects in the scene that is being photographed. The color of an illuminant has a significant effect on the color of an object. The Human Visual System (HSV) has adapted itself to ignore the illuminant color so that regardless of the situation, we can perceive an object's color as relatively constant. This color is present when an object is illuminated by a perfectly white light, also known as the canonical illuminant. This HSV phenomenon is also known as color constancy.

To emulate the color constancy of the Human Visual System researchers have proposed many different methods that perform white-balancing. From the older simpler methods that use image statistics to the more recent more complex learning-based methods. Recent research has shown that the best results are achieved using artificial intelligence, more specifically convolutional neural network (CNN) methods, which are often complex and computationally intensive. The problem of white-balancing can be divided into two subproblems, single-illuminant and multi-illuminant. The main focus of existing research has been single-illuminant white-balancing. Here the scene is illuminated by a single illuminant which uniformly illuminates each region of the scene. Single-illuminant is a simpler problem, as no spacial information knowledge is needed for white-balancing. Multi-illuminant is the generalization of the single-illuminant as in this situation the scene can be illuminated by an arbitrary number of illuminants where regions of the photographed scene can have non-uniform illumination.

A problem with the development of a multi-illuminant white-balancing algorithm is the need

for labeled high-quality multi-illuminant images. This need arises from the fact that the best results are achieved by learning-based methods, which require thousands of samples to produce accurate results. The problem with multi-illuminant image labeling is that to properly label a multi-illuminant image one would need to know the illumination of each pixel in the image. Modern images have high dimensionality, which would mean one would need to manually label millions of pixels which is a nearly impossible task.

1.2 Scientific contributions

This thesis tackles the problem of computation color constancy in all its different forms. From the simplest already thoroughly researched area of single illumination estimation, through several variations of multi-illuminant estimation, to the creation of datasets that contains images with an arbitrary number of illuminants. All of the proposed estimation methods are deep-learning-based methods. Different methods are proposed for different estimation subproblems. The first is a neural network architecture for single illuminant estimation. The novelty of this method is that it is a lightweight computationally effective convolutional neural network. The next problem is illumination estimation when we know the number of illuminants in the image. The novelty of our approach is that it can be applied to existing single-illuminant methods so that they can perform multi-illuminant estimation. The third method performs multi-illuminant estimation regardless of the number of illuminants in the image. It does this by dividing the image into many small patches and estimating the illumination for each patch separately. The novelty of this method is that in addition to the local patch information the method uses global image information to estimate patch illumination. The next subproblem is illumination estimation on a pixel-by-pixel basis. For this, two different methods were developed, one that does pixel illumination estimation for the entire image and one that performs pixel illumination estimation on a patch-by-patch basis. Both methods use similar neural network architectures with their novelty being the low number of parameters they use for illumination estimation.

The other main contribution of this thesis are two different ways to create and label a multi-illuminant image. Using the first method a large two-illuminant multi-camera dataset was created. The novelty of the dataset is the size of the dataset, the diversity of the dataset, and the simplicity of labeling the illumination in an image. The second method allows us to create a large number of images with a variable number of the illuminant. The novelty of the dataset is the usage of lighting filters which allows us to automate the creation and labeling of images with multiple illuminants. In conclusion, the scientific contributions can be summarized as:

1.Method for single illuminant estimation using lightweight convolutional neural network
2.Framework for creation and validation of images captured in complex lighting environments
3.Method based on convolutional neural network for multi-illuminant estimation in single image

1.3 Thesis structure

This thesis is divided into seven chapters. Chapter 1 gives an overview of the problem, its causes, and its effect. The scientific contributions are also presented in this chapter. Chapter 2 is an introduction to Color Constancy, how the Human Visual System works, and the history of Color Constancy research. Chapter 3 is an overview of computational color constancy, or what needs to be done to emulate the Human Visual System and digitally remove the effect of illumination color from an image. Chapter 4 is an introduction to learning-based methods. Here a brief history of deep learning methods is given as well as the basic components of a proper neural network. Chapter 5 presents the two methods developed for the creation of multi-illuminant datasets. In addition to the methodology used to create the dataset, an overview of existing datasets is given. Chapter 6 introduces the four methods developed for illumination estimation. The first method explained is the single illuminant estimation model. The next method is a multi-illuminant estimation model that requires the apriori knowledge of the number of illuminants in an image. It is followed by the multi-illuminant estimation method that performs illumination estimation on a patch-by-patch basis. Finally, the two methods that perform illumination estimation on a pixel-by-pixel basis are presented. In addition to the model architectures, the motivation for their creation, their evaluation, and the results they achieve are presented in this chapter. Finally, Chapter 7 contains the conclusion of this thesis.

Chapter 2

Color Constancy

To start things off, color, how humans perceive color, and how the human visual system performs chromatic adaptation will be defined in this chapter. Color is a fairly difficult concept to define. It is the brain's interpretation of stimuli received and processed by the eye and brain. The color stimuli the eye receives are dependent on two things, the color of the light source that illuminated the object we are looking at, as well as the physical and chemical properties of the object itself. The color of the radiation source is defined by the electromagnetic radiation the source emits. More specifically, the visible spectrum, or electromagnetic radiation with a wavelength between about 380 nm to 750 nm also known as visible light. The intensity of the radiation at those wavelengths determines the color the light source will have. These light rays then travel to the observed object. Depending on the properties of the object, the rays will be absorbed or reflected or a combination of the two. The light can be reflected in two ways, specular reflection, and diffuse reflection. Specular reflection also known as mirror-like reflection is the reflection where the light source is reflected in a single outgoing direction with the same reflection as the ingoing rays but from the opposing side of the surface. The other type of reflection is diffuse reflection, where the ingoing light ray is reflected at many angles. This is the more common type of surface reflection. The reflections are visualized in Figure 2.1.

Afterward, the light rays travel and reach the human eye. The human eye is a complex organ made out of many parts. The first part the light rays reach is the cornea. It shields the rest of the eye from dust and other harmful materials. It also acts as the eye's outermost lens that refracts light so that we can focus on an observed object. After the cornea, the light passes through aqueous humor. Its main function is to ensure the flexibility of the eye and is made of a substance that is similar to water. The next two steps are the pupil and the iris. The pupil is a hole in the center of the eye through which light enters the eye. The bigger the pupil, the more light enters the eye. The iris is a pigmented muscle that controls the size of the pupil. Behind the pupil is the lens. Like the cornea, it is used to refract light. It is a flexible structure that is used to change the focal point of the eye, so we can focus on nearby and far away objects. After the

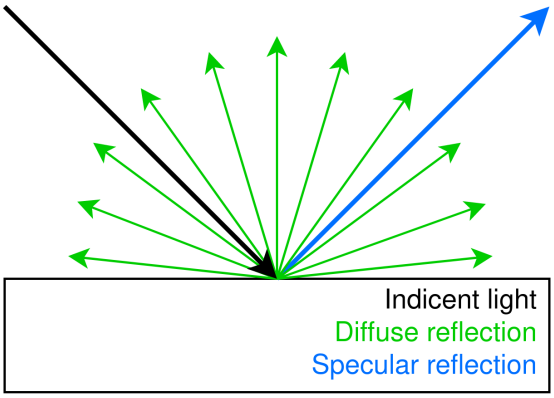


Figure 2.1: Diffuse and specular reflection of a light ray on a surface

lens, the light passes through another humor, the vitreous humor. It has the same function as the aqueous humor but is more viscous. Finally, the light rays reach the retina. The retina is a thin layer of photoreceptors called rods and cones. The photoreceptors are neurons that are a part of our nervous system. They are used for initial signal processing. There exist three types of cones, long-wavelength, middle-wavelength, and short-wavelength which can also be referred to as L, M, and S cones. The three types of cones serve our color vision. An important part of the retina is the fovea. The fovea is the area of the retina with the best color and space vision. Half of the information that is carried by the nerves comes from the fovea. The last important part of the eye is the optical nerve. It takes the information generated by the photoreceptors and transfers the information from the eye to the brain for further processing.

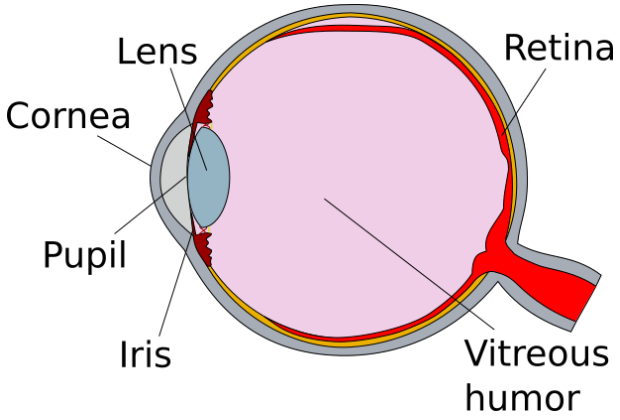


Figure 2.2: Schema of a human eye, showcasing some important eye components

2.1 Adaptation

One of the most important functions of the human visual system is its ability to change stimulus sensitivity when the stimulation conditions change. This is called adaptation. The Human Visual System has three important types of adaptations, dark, light, and chromatic adaptation.

Dark adaptation occurs when there is a decrease in the luminance level of what we are perceiving. A simple example is a room at night with a light bulb that has just been turned off. Immediately after turning off the light bulb, the room will look completely dark. After some time, the objects in the room will gradually start to become visible. This is possible because the Human Visual System gradually increases its sensitivities through the mechanism of dark adaptation. This is a relatively slow process that can take up to 20 minutes to reach maximum sensitivity when the decrease in lumination is massive. The cones and rods play a role in this. The cones respond to higher levels of lumination and are less sensitive, while rods are more sensitive and responsive to lower-level lumination. Since cones respond to higher levels, their sensitivity increases fairly rapidly. After about 10 minutes, the cones become more sensitive than the rods. The sensitivity of the rods starts increasing, and 20 minutes after the lumination change they reach maximum sensitivity.

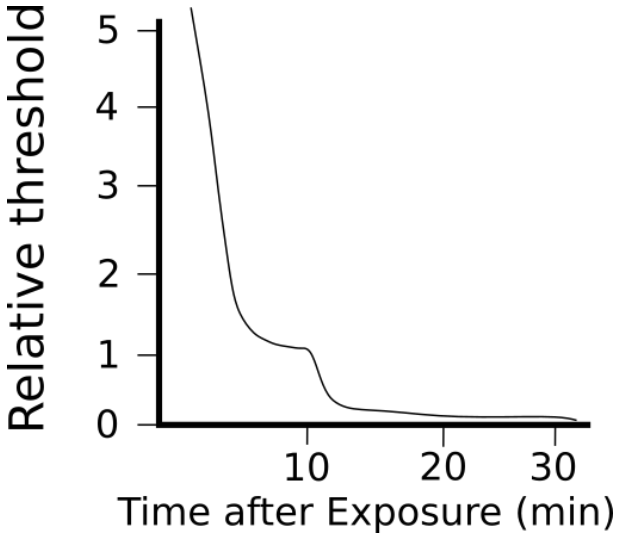


Figure 2.3: A dark adaptation curve after a strong exposure

Light Adaptation can be looked at as the inverse of dark adaptation. The major difference is the speed at which the Human Visual System adapts to an increase in lumination. While dark adaptation can take up to 20 minutes, light adaptation takes only several seconds. Here the cone and rod sensitivities need to be decreased. The Humany eye adaptation curve can be seen in Figure 2.3.

While dark and light adaptations change our color perception, they are adaptations that allow us to perceive objects. After our receptors' sensitivity has calibrated and we do not just perceive everything as completely white or completely black. After that is when the final adaptation, chromatic adaptation, comes into play. Chromatic adaptation is the ability of the Human Visual System to adjust to the change in scene illumination. This means that we will perceive an object's color as consistent regardless of the color of the light that illuminates the object. A banana will have yellow color regardless of whether it is looked at noon or at sunset.

This feature of The Human Visual System is also known as color constancy. An example of how illumination affects object color can be seen in Figure 2.4.

Figure 2.4: Four images of the same flower under different illumination conditions

The mechanics of chromatic adaptations can be divided into two mechanisms groups [1], sensory and cognitive. Sensory mechanisms are automatic responses to stimuli. The response is represented by pigment depletion or regeneration with the change in luminance level. An increase in luminance level breaks down pigment molecules. This decreases the responsiveness of the photoreceptors. This is referred to as Gain Control. The Gain is decreased when there are many photons and increased when there are few photons. An important attribute of Gain Control is that the sensitivities of the three types of cones change independently. This idea was first proposed by von Kries (1902) who stated: ". . . the individual components present in the organ of vision are completely independent of one another and each is fatigued or adapted exclusively according to its own function." Cognitive mechanics rely on the human experience. Since we have seen the object so many times, we tend to remember the color instead of observing it. This is an interesting area of research, but it falls out of the scope of this work. For those interested, there are many existing works that tackle the problem [1].

A color appearance model is a mathematical model used to describe perceptual aspects of color vision. There are many color appearance models since this is a complex problem. In this work, a couple of influential models are described. For a chromatic adaptation model to be valid, it must work with cone responses. This means that for any colorimetry application, the color values must be transformed into cone responses. Luckily, cone responses can be accurately represented using a linear transformation of the CIE tristimulus values (XYZ).

The oldest and most influential chromatic adaptation model is the model proposed in 1902 by Johannes von Kries. The model is extremely simple, but even a century after its proposal it is still used. Johannes von Kries did not provide a set of equations instead, MacAdam's translation of what he said is: "This can be conceived in the sense that the individual components present

in the organ of vision are completely independent of one another and each is fatigued or adapted exclusively according to its own function."

This hypothesis can be expressed as three simple equations.

$$\begin{aligned} L_a &= k_L L \\ M_a &= k_M M \\ S_a &= k_S S \end{aligned} \tag{2.1}$$

L, M, and S represent the initial cone responses. k_L, k_M and, k_S are the scale coefficients used on the initial cone responses. L_a, M_a, and S_a are the adapted cone signals. The 2.1 equation represents a simple Gain Control model. The main focus of the chromatic adaptation model is how the k_L, k_M and, k_S coefficients are calculated. A common way to calculate the coefficients is to take the inverse L, M, and S cone responses of the maximum stimulus values. This allows us to determine the corresponding colors between two viewing conditions.

$$\begin{aligned} L_2 &= (L_1/L_{max1})L_{max2} \\ M_2 &= (M_1/M_{max1})M_{max2} \\ S_2 &= (S_1/S_{max1})S_{max2} \end{aligned} \tag{2.2}$$

Here $_{max1}$ and $_{max2}$ represent the maximum cone response in each viewing condition. This model can be shown in matrix form and combined with the CIE tristimulus values to relative cone responses transformation matrix M.

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} L_{max2} & 0 & 0 \\ 0 & M_{max2} & 0 \\ 0 & 0 & S_{max2} \end{bmatrix} \begin{bmatrix} 1/L_{max1} & 0 & 0 \\ 0 & 1/M_{max1} & 0 \\ 0 & 0 & 1/S_{max1} \end{bmatrix} \mathbf{M} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} y \tag{2.3}$$

The results the model provides can be seen in Figure 2.5. It shows us that even though the model is simple, it performs chromatic adaptation extremely well. There are discrepancies between the true values and the von Kries model, which has led to further research and the creation of other chromatic adaptation models.

Many enhancements to the von Kries model have been proposed. An important proposition that enhances the von Kries model is the Retinex theory [2]. An important aspect of the Retinex theory is the fact that it takes into account the spatial distribution of scene colors for

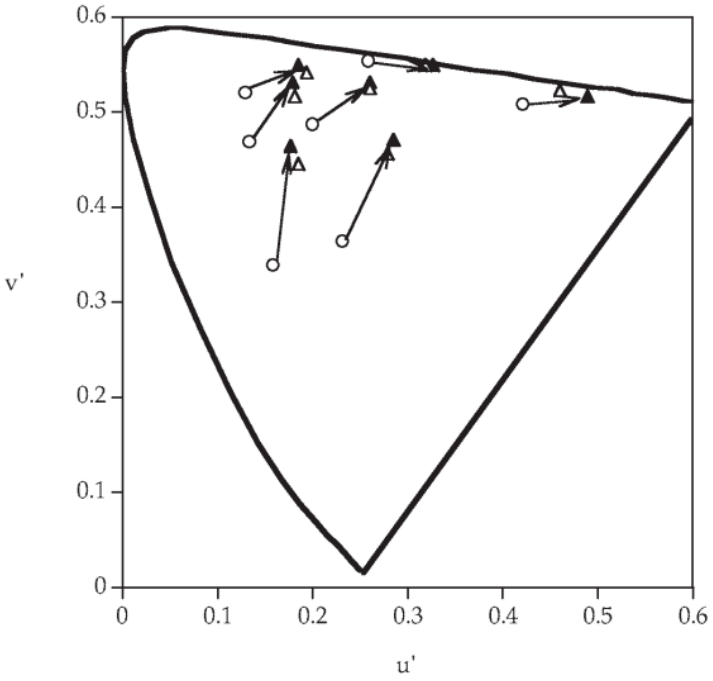


Figure 2.5: Some examples of corresponding color data using von Kries model. Black triangles represent model predictions and white triangles represent correct data.

better visual perception modeling. The theory also states that color is not controlled by reflected light distribution. Instead, it is controlled by surface reflectance. Land proposed three color mechanics that have the spectral responses of the cone photoreceptors, which he called retinexes. The three retinex map to the long-wavelengths, middle-wavelengths, and short-wavelengths. The retinex output is calculated by taking the ratio of the signal at any point and normalizing it with the average signal of that retinex in the scene. It also takes into account the influence of the changes in the background on the color. Changing the spatial distribution of the retinex signals used to normalize a point in the scene can vary the influence of the background. To reduce Retinex theory to the von Kries model one only needs to use the scene average for signal normalization. Retinex theory provides a good interpretation, but there are some flaws. Many other chromatic adaptation models that enhance the von Kries model have been proposed [3], but they fall outside the scope of this work.

Chapter 3

Computational color constancy

The previous chapter explained the mechanics the Human Visual System uses to perform chromatic adaptation so that the objects we see retain their color regardless of the color of the illumination. Computers are unable to achieve this, and the need for a method that emulates color constancy arises. The removal of illuminant chromaticity from a digital image is known as computational color constancy. The term white-balancing is also used, especially in digital photography. In this chapter, a quick overview of digital image formation and the basic steps that a computational color constancy method performs to remove the chromatic effect of an illuminant are provided. Concrete examples of how some methods work will be given in a later chapter.

A digital image is made out of pixels. A pixel is a vector that contains three values $p = (p_R, p_G, p_B^T$. R, G, B represent the red, green, and blue intensities of the pixel. The values of a pixel can be calculated using the Dichromatic Reflection Model [4].

$$p_c = m_b(\mathbf{x}) \int_\omega I(\lambda) S(\mathbf{x}, \lambda) p_c(\lambda)\, d\lambda + m_s(\mathbf{x}) \int_\omega I(\lambda) p_c(\lambda)\, d\lambda \tag{3.1}$$

Pixels RGB intensities depend on illumination color $I(\lambda)$, surface reflectance $S(\mathbf{x}, \lambda)$ and the camera sensitivity function $p(\lambda) = (p_r(\lambda), p_g(\lambda), p_b(\lambda))$ of the three channels. $\mathbf{x}$ represents the spatial coordinates, λ represents the light wavelength. m_b is the relative amount of body reflectance, m_s is the relative amount of spectral reflectance, and $c = R, G, B$ the red, green, or blue intensities.

This model can be simplified using the Lambertian assumption. Under this assumption, the spectral reflection m_s is ignored. This assumption creates a physically unfeasible model, but this assumption creates a satisfactory approximation of many real-world surfaces.

$$p_c = m_b(\mathbf{x}) \int_{\omega} I(\lambda) S(\mathbf{x}, \lambda) p_c(\lambda)\, d\lambda \tag{3.2}$$

For computational color constancy, we can further simplify the formula by removing the surface reflectance $S(\mathbf{x}, \lambda)$ as the objects in the scene do not affect the illumination color.

$$\mathbf{l} = \begin{bmatrix} l_R \\ l_G \\ l_B \end{bmatrix} = \int_{\omega} I(\lambda) p(\lambda)\, d\lambda \tag{3.3}$$

When the scene is illuminated by one illuminant, we can color the light source $\mathbf{l}$ as a *RGB* vector that depends on the illumination color $I(\lambda)$ and the camera sensitivity function $p(\lambda)$.

A major problem with computational color constancy is there is often no apriori knowledge of $I(\lambda)$ and $p(\lambda)$. This makes illumination estimation an under-constrained problem. To remove the chromatic effect of the scene illumination, various assumptions that simplify the problem need to be applied. This fact can be exemplified in an image with a few surfaces. As an example, we can use an image that only contains a yellow wall. Without additional information, we cannot tell whether the wall is yellow and the scene illumination is a white light, whether the wall is white and the illuminant is yellow, or if another wall color/illumination color combination has created this particular image of a yellow wall.

There are two main approaches to perform computational color constancy. The first approach divides the process into two steps. In the first step, the scene illumination is estimated. In the second step, the estimated illuminant is used to remove the illuminant chromaticity from the image. This is the more popular approach used by a majority of methods.

This thesis will focus on this approach, as it is the approach used in the methods proposed in this thesis. Existing approaches for the first step will be presented later in the paper, as there is a large variety of different methods. The second step is relatively simple and will be presented here.

The removal of illumination chromaticity in digital images is based on the von Kries model [5]. The idea is to take the image illuminated under the unknown illuminated and transform it into an image illuminated by the canonical illuminant. The canonical illuminant is usually perfectly white light $\mathbf{l} = (1,1,1)^T$. The von Kries model [6] is a diagonal transformation that uses the assumption that the red, green, and blue camera sensor responses are independent. The assumption is the same as the one proposed by von Kries for the Human Visual System where the L, M, and S cone responses are replaced with the camera sensor responses.

$$P^c = \Lambda^{u,c} * P^u \tag{3.4}$$

P^c is the image taken under the canonical illuminant, $\Lambda^{u,c}$ represents the von Kriss diagonal transformation, and P^u is the image taken under an unknown illuminant. The $\Lambda^{u,c}$ can be represented using a diagonal matrix.

$$\Lambda^{u,c} = \begin{bmatrix} \frac{l_R^c}{l_R^u} & 0 & 0 \\ 0 & \frac{l_G^c}{l_G^u} & 0 \\ 0 & 0 & \frac{l_B^c}{l_B^u} \end{bmatrix} \tag{3.5}$$

where l_R^u, l_G^u, l_B^u are the red, green, and blue values of the unknown illuminant and l_R^c, l_G^c, l_B^c are the red, green, and blue values of the canonical illuminant. The von Kries model is a Chromatic Adaptation Transform (CAT), and there exist several more recent CAT computational techniques. Some examples are Bradford [7] and CIECAT02 [8]. The methods proposed in this thesis use the von Kries model, as the focus of this research was illumination estimation, and because it was shown the von Kries model is sufficient for illuminant chromaticity removal [5]. An example of a von Kries corrected image can be seen in Figure 3.1.

Figure 3.1: Image before and after being chromatically adapted using the von Kries model

The other approach performs computational color constancy in a single step. In this approach, the illumination is not directly estimated instead, the corrected image is directly estimated. Many methods based on Retinex [9, 10] use this approach.

Chapter 4

Neural networks

All the methods proposed in this work are based on neural networks, so in this chapter, a quick overview of what a neural network is, what a neural network consists of, and how a neural network is trained is presented.

Neural networks have recently become a popular tool for solving many different tasks, from natural language processing to pattern recognition. Even though the neural network popularity explosion was recent the first research for neural networks was developed in 1943 by McCulloch & Pitts [11]. A little later in 1949 Hebb proposed Hebbian learning [12]. In "The Organization of Behaviour", he proposed that pathways that connect different neurons get strengthened with repeated use. If multiple neurons are activated at the same time their connection gets stronger. This theoretical research was first realized in 1958 [13]. This implementation was called Mark I Perceptron. In essence, the Mark I Perceptron is a binary classifier that inputs a real value vector $\mathbf{x}$ and outputs a single binary value $f(\mathbf{x})$.

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & otherwise \end{cases} \tag{4.1}$$

$\mathbf{w}$ is a vector of real numbers known as weights, $\cdot$ is the dot product and b is the bias that is input independent which shifts the decision boundary.

The next major breakthrough was backpropagation [14]. Backpropagation is a powerful algorithm used for gradient calculation. Neural networks are made out of numeric values called weights. Using weights the neural network computes the output for a given input. Backpropagation is used to calculate the value of each weight in a neural network. The algorithm is so powerful because it does not directly compute the gradient for each weight. It uses the calculated gradients of the previous network layer to calculate the next layer. The algorithm is called backpropagation because it calculates the gradient of the final layer first and with each step calculates the gradients of the previous layer, with the first layer gradients being calculated last. Its efficiency allows for the effective use of gradient methods, such as Stochastic Gradient

Descend (SGD) [15] and Adaptive Moment Estimation (Adam) [16].

The gradients of the loss function with respect to the weights are calculated using back-propagation. A loss function is used to compute how accurate the neural network is, in other words, how similar the predicted value is to the expected value. Some common loss functions are Mean Square Error, Mean Absolute Error, and Cross Entropy loss.

There is a large number of different types of Neural Networks, such as Feedforward neural networks [17], Recurrent neural networks[18], Generative Adversarial neural networks[19], and Convolutional neural networks[20]. The focus of this work are Convolutional neural networks since all the proposed methods are Convolutional neural networks. A Convolutional neural network is a type of neural network and as the name suggests, a Convolutional neural network uses convolutions to produce an output. A convolution is a mathematical operation that takes two functions f, g and creates a new function $f * g$. It is used to express the amount of overlap one function g has with another as it is a shifter over the other function f.

$$\begin{aligned}(f*g)(t) &= \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\,d\tau \\ (f*g)[n] &= \sum_{m=-\infty}^{\infty} f[m]g[n-m]\end{aligned} \tag{4.2}$$

The formulas 4.2 show two variants of the convolution definition. The first one shows the integral form that is used with a continuous function. The second one shows the definition of when discrete functions are used. The second form is the more useful definition, as images are made out of discrete pixels.

It was shown [21, 22, 23] that the use of convolutions greatly improves results for computer vision tasks. In Convolutional neural networks, they use multidimensional discrete convolutions as building blocks. The building blocks of a neural network are also called layers. The number of layers, the layer parameters, and the arrangement of layers in a neural network are defined as its architecture. In addition to the convolutional layers, there are also pooling, fully connected, and activation layers.

The layers in a Convolutional neural network can be divided into three groups, the input layer, the output layer, and the hidden layers. The input layer is the first layer of a neural network and it feeds the input data to the next layer. It does no processing and contains no weights. The output layer is the final layer of a neural network and it outputs the prediction of the information we are trying to extract from the input. The hidden layers are the most important part of the neural network. They are called hidden layers as they are located between the input and output layers and their outputs are feature maps. Their output is fed to the next layer and is not visible outside the network.

The fully-connected layer is made out of neurons. A neuron is simply another name for the perceptron. A fully-connected layer can have an arbitrary number of neurons. Each neuron is

connected to all the neurons of the previous layer. The outputs of the neurons from the previous layer are multiplied by the weights vector of the neuron. After that, the bias is added. Finally, an activation function is applied.

Activation functions are important because they introduce non-linearity into the neural network. A network that contains multiple fully-connected layers and no activation functions can be reduced to a network that contains only one fully-connected layer. There are many activation functions. Some popular activation functions are, the sigmoid activation function, the hyperbolic activation function, Rectified Linear Unit (ReLU) activation function. The ReLU activation function is a very popular activation function. Its definition is $f(x) = x^+ = \max(0,x)$,, and it was shown to be quite effective as the vanishing gradient problem is much smaller when compared to other activation functions such as the sigmoid activation function.

The convolutional layer is the most important part of the network. It takes an input of shape of $n \times h \times \times w \times c$. n is the number of input samples, h is the height of each sample, w is the width of each sample, c is the number of channels in each sample. Unlike the neurons in the fully-connected layers, the neurons of a convolutional layer have a limited receptive field. This significantly reduces the number of trainable parameters the layer has. The number of parameters is depended on the kernel size of a filer and the number of filters in the layer. Unlike fully connected, the number of parameters is not dependent on the image size, for example, if we create a convolutional layer with 5 3x3 filters it will always have 45 parameters. Convolutions are also great for images as spatial relations between different features are taken into account.

The final important type of layer in a convolutional neural network are pooling layers. They are used to reduce the dimensionality of the feature maps by clustering the outputs of multiple neurons of the previous layer. With a pooling layer, the features extracted by the previous layer are summarized which reduces the required computation for data processing. There are several different types of pooling layers. Two popular types of pooling are max-pooling and average-pooling. The parameter of a pooling layer is the kernel size or the size of the window that is used to calculate the maximum value for max-pooling or the average value for average-pooling.

After you have defined the architecture, the neural network needs to be trained. For proper training of a model, we need data, a cost function, and an optimizer. For the problem of computational color constancy, the data is natural images. There are many cost functions. Some popular regression losses are Mean Squared Error and Mean Absolute Error. For classification, there is the Categorical Cross Entropy cost function. There are also many different optimizers, some popular optimizers are SGD [15], RMSprop [24], and Adam [16]. To properly use optimizers, we need to define their hyperparameters. An important hyperparameter is the learning rate. It defines the step size that the optimizers use to minimize the cost function at each step.

For a neural network to accurately solve a problem, all of these factors need to be selected. We need to define an architecture that has a small memory footprint and is computationally

inexpensive. We need to define a cost function so that the model properly learns to solve the problem. We also need to define the proper optimizer and its hyperparameters so that the model minimizes the cost function as fast as possible.

Chapter 5

Datasets: creation, labeling and proper usage

To create, analyze, and evaluate a method to solve a problem, a good set of problem examples is needed. This is especially true for learning-based methods, as their accuracy is very dependent on the size, diversity, and quality of samples of the dataset. For the problem of computational color constancy, the dataset is made out of minimally processed images that have not been chromatically adapted. Such datasets also need to contain the color of the illumination for each image.

For a computational color constancy dataset to be diverse, it needs a large set of diverse scenes, a large set of different natural and artificial illumination sources between the images, and a diverse set of cameras used to capture the images in the dataset. A large number of scenes is required to capture the diverse set of scene geometries and diverse sets of surfaces. The diverse set of illuminants is required to capture the effects different illuminants have on different surfaces. Different cameras are needed since each camera uses a different camera sensor, and if you capture the same scene with two different cameras, the images will have slight variations in pixel values because of the physical characteristics of the camera sensor.

To extract the illumination from an image, a calibration object is used. The color of a calibration object is known apriori. When the calibration object is illuminated by an unknown illuminant, its color will change. Since the object's actual color is known, one can extract the illuminant color by comparing the actual object's color with the color it has under the unknown illuminant.

5.1 Existing datasets

Since white-balancing is one of the first steps in any camera's image processing pipeline, there are several publically available computational color constancy datasets. In this chapter, we will

give a quick rundown of some existing dataset and their characteristics. Existing datasets can be divided into two categories, datasets with single uniform illuminant images, and images with multiple different illuminants in a single image. To start the single illuminant datasets will be presented first.

5.1.1 Single-illuminant datasets

The first presented single illuminant dataset is ColorChecker [25]. This dataset is the oldest, still widely used computational color constancy dataset. There are 568 indoor and outdoor images captured using 2 different cameras in this dataset. The used cameras are Canon 1D and Canon 5D. In ColorChecker, they used a Macbeth ColorChecker calibration object which contains 24 surfaces of different colors. The authors used the gray surface of the Macbeth ColorChecker to extract the scene illumination. This dataset comes with the problem of unreliable ground-truth illumination. Researchers [26, 27] have shown that multiple different illuminants have been extracted from the same image and different methods were tested using different ground truths of the same image. The researchers also showed that the single illuminant assumption is violated in some of the images in this dataset. These facts make the dataset difficult to use for methods evaluation and comparison.

Figure 5.1: Example image from ColorChecker dataset. For display purposes, the images were tone mapped.

The next important Single illuminant dataset is the NUS-8[28] dataset. This dataset contains 1853 images. The main contribution of this dataset is that it was created using 9 different cameras. The used cameras are Canon 1D Mark III, Canon 600D, Fujifilm XM1, Nikon D5200, Olympus EPL6, Panasonic GX1, Samsung NX 2000, Sony α57 and Nikon D40. This is the largest number of cameras in a single dataset to the best of our knowledge. Around 200 images were captured with each camera. The Macbeth ColorChecker was used to extract scene illumination. The dataset does not contain 1853 different scenes, instead with a few exceptions the cameras were used to capture images of the same scene. This dataset does not contain a large,

diverse set of scenes, but it allows us to compare the accuracy of methods trained on one camera and tested on another.

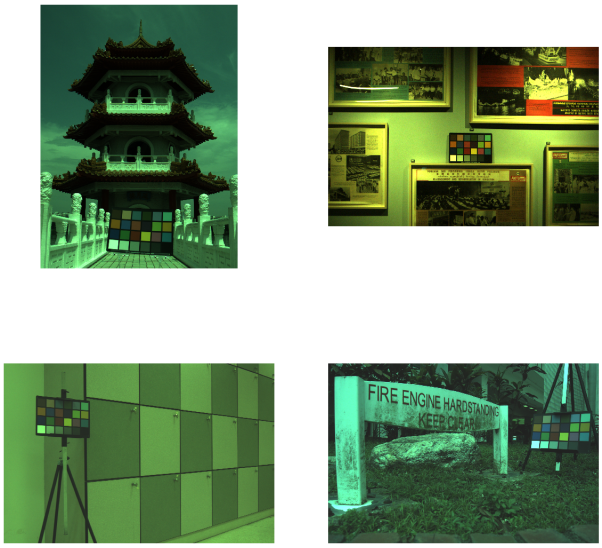


Figure 5.2: Example image from NUS-8 dataset. For display purposes, the images were tone mapped.

There are two more large-scale single illuminant datasets, the Cube+ [29] and the Intel-TAU [30]. They both contain a diverse set of indoor, outdoor, and nighttime images. The Intel-TAU also contains images of laboratory printout scenes and real scenes in a laboratory environment. The Cube+ dataset contains 1707 images captured using The Canon EOS 550D. The Intel-TAU contains 7022 images captured using 3 different cameras. The used cameras are Canon 5DSR, Nikon D810, and Sony IMX135. Intel-TAU is the largest single illuminant dataset used in this thesis. Cube+ used a SpyderCube as the calibration object, while Intel-TAU used X-Rite ColorChecker Passport. Cube+ examples can be seen in Figure 5.3 and Intel-TAU examples can be seen in Figure 5.4.

Figure 5.3: Example image from Cube+ dataset. For display purposes, the images were tone mapped.

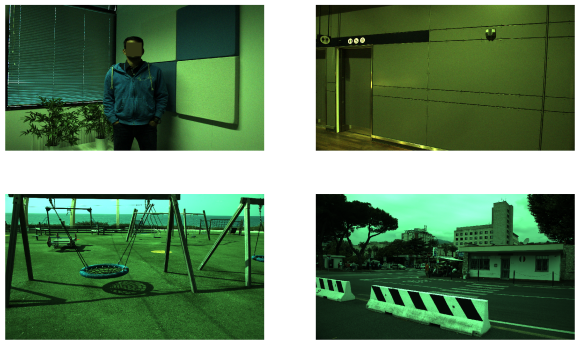


Figure 5.4: Example image from Intel-TAU dataset. For display purposes, the images were tone mapped.

5.1.2 Multi-illuminant datasets

When looking at multi-illuminant datasets, the selection is much smaller. There are not many datasets and most of the dataset contain a fairly small number of images. This can be explained by the fact that ground-truth extraction in multi-illuminant images is much harder when compared to single-illuminant images. Unlike single illuminant images where one illuminant color needs to be extracted, the illumination color of each pixel needs to be extracted in multi-illuminant images.

The first example of a multi-illuminant dataset was introduced in Bleier et al.[31]. This dataset contains only 36 images captured using the Canon EOS 550D. The dataset contains 4 different scenes that were captured under 9 different lighting situations. All the images were captured in a laboratory environment. This means that both the scene and illumination setup were artificially created. The scene was created by placing different objects on a flat surface in a dark room. The illumination was created using two Reuter lamps and the illumination was diversified using LEE color filters. The dataset does not contain enough images to properly train learning-based methods. Examples can be seen in Figure 5.5.

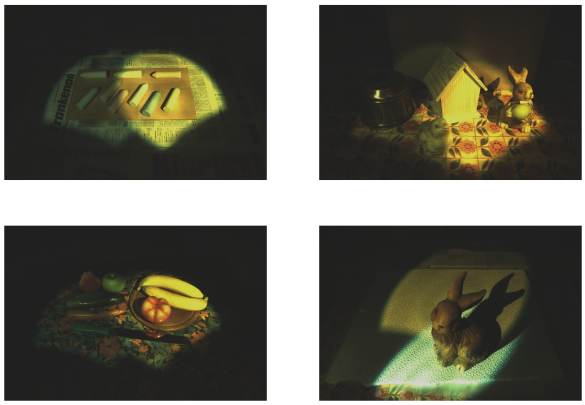


Figure 5.5: Example image from Bleier et al. dataset. For display purposes, the images were tone mapped.

The next dataset is the Multiple-Illuminant Multi-Object [32] dataset. The dataset contains 80 images taken using the Sigma SD10 camera. 60 of the images were created in a similar way

as the images in Bleier et al.[31]. 10 different scenes illuminated in 6 different illumination conditions. The other 20 images real-world images with multiple illuminants. Another contribution of the [32] paper is a method for automatic per-pixel ground truth illumination extraction. They use the linearity of illumination to extract the ground-truth. An image with two illuminations is simply the sum of the two images with only one of the illuminants. Because of the illumination linearity, the effect of each illuminant on each pixel can be extracted. The problem with this labeling method is the fact that for the method to work, you need to be able to remove the effect of the illuminant from the scene. This means turning the illuminant off or obscuring it so that it does not reach the scene. This is oftentimes impossible with a simple example being the sun. Examples can be seen in Figure 5.6.

Figure 5.6: Example image from Multiple-Illuminants Multiple-Objects dataset. For display purposes, the images were tone mapped.

The final and largest multi-illuminant dataset presented is the LSMI [33] dataset. The authors of this dataset use the labeling method introduced in [32]. The dataset contains 7486 images taken using 3 different cameras. The used cameras are Samsung Galaxy Note 20 Ultra, Sony α9, and Nikon D810. The dataset contains images with 1, 2, or 3 illuminants. There are 2762 different scenes in the dataset. The number of scenes is smaller than the number of images because a scene with two illuminants can create three different images, one with both illuminants and two with one illuminant. For scenes with three illuminants, we can create one image with three illuminants, three images with two illuminants, and 3 images with one illuminant. All the images in this dataset are indoor images as the labeling method cannot be used in outdoor images. This is a relatively new dataset that was not available for the longest part of our research period. Examples can be seen in Figure 5.7.

5.2 Created datasets

Since the number of multi-illuminant datasets is limited and the purpose of this research is the creation of a method that performs illumination estimation on images where the number

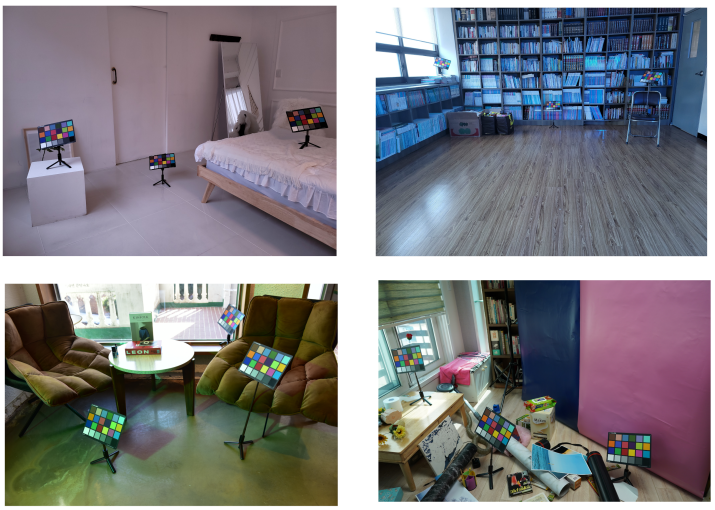


Figure 5.7: Example image from Large Scale Multi-I. For display purposes the images were tone mapped.

of illuminants can vary, and the illumination does not need to be uniform, during research two different large scale datasets were developed for the creation, analysis, and evaluation of different methods. The first dataset called the Shadows & Lumination (Shal) dataset, is a two-illuminant dataset, and the second, called the Filters & Lumination dataset, is a variable number of illuminants dataset. This chapter will go over the motivation, creation, evaluation, and results for each of the datasets. The Shadows & Lumination dataset is presented first and the Filters & Lumination dataset is presented second.

5.3 Shadows & Lumination dataset

The first dataset that was developed during this research is the Shadows & Lumination dataset. This dataset contains 2500 different images taken in a variety of different locations, in a variety of indoor, outdoor, and nighttime illumination conditions. Five different cameras were employed to create the dataset. Around 500 images were taken using each camera. The used cameras were: the Canon EOS 5D Mark II camera, the Canon EOS 550D camera, the Panasonic DMC-FZ1000 camera, the Sony DSLR-α300, and the Samsung ISOCELL Plus GW1 1/1.72" camera sensor in the Motorola One Fusion+ mobile phone. A couple of example images can be seen in Figure 5.8.

5.3.1 Motivation

The motivation for the creation of the dataset was the lack of existing multi-illuminant datasets. The lack of scene diversity and the lack of camera diversity in existing multi-illuminant datasets were also motivations. Before the release of the LSMI [33] dataset, no multi-illuminant dataset

Figure 5.8: A couple example images. The leftmost image was taken by the Motorola camera. The top row images were created using the Canon 5D and Canon 550D cameras. The bottom row images were created using the Sony and Panasonic cameras.

had more than 100 images. Such small datasets are not enough for the training and evaluation of a learning-based method. In comparison with the LSMI dataset, our dataset uses a larger number of cameras and contains nighttime and outdoor scenes with a variety of natural and artificial illumination sources.

Because the labeling of multi-illuminant images is a laborious process, the created two-illuminant dataset has borders between the two illuminants which are easily recognized. The idea here is that each image has one illuminant that is present in the entire image, in other words, that illuminant uniformly illuminates the entire scene. We call this illuminant, the ambient illuminant. The other illuminant, which has a stronger intensity, uniformly illuminates only a part of the scene. We call this illuminant, the direct illuminant. Because the direct illuminant is stronger, it overpowers the ambient illuminant in regions where both illuminants are present. This creates a situation where there are two illuminants with their border clearly defined. A simple example of such a situation is an image of an outdoor scene taken on a sunny day. Here the sun is the direct sunlight and the blue sky can be seen as the ambient light. The sun's rays do not reach the regions where objects cast shadows, and those regions are illuminated by the sky. All the other regions are illuminated by both the sky and the sun, bus since the sun is such a stronger light, the skylight has a negligible effect on those regions.

Because of such a setup, two things need to be extracted from each so that the image can be properly chromatically adapted. The first thing is the illumination, the color of the two illuminants present in the scene. The second thing is an influence mask. A mask that labels which illuminant affects each pixel. To extract the illumination, calibration objects are used. There is no automatic way to create the influence mask, so this was done manually.

5.3.2 Illumination extraction

The calibration object used for this dataset is called the SpyderCube. An example image can be seen in Figure 5.9. The SpyderCube contains four different faces. Two white faces (WL, WR) and two spectrally neutral 18% gray faces (GL, GR). The SpyderCube faces can be grouped into two surfaces, left (GL, WL) and right (GR, WR), which are at an angle. The gray faces are used to extract the illumination and the white faces are used for verification. The illumination is extracted by calculating the average pixel value of one of the gray faces.

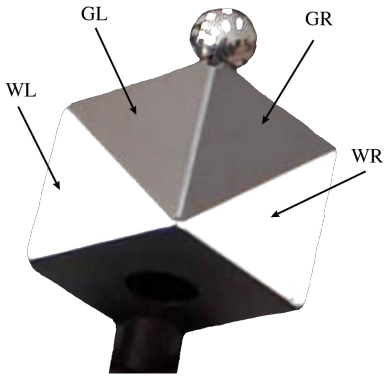


Figure 5.9: A SpyderCube. GL and GR represent the gray faces. WL and WR represent the white faces.

Since the surfaces are at an angle, the extracted illuminants from the two gray faces can significantly differ. The sunlight is cast at a certain angle and in certain orientations, only one of the SpyderCube surfaces will be illuminated by the sun. This is easy to identify by looking at where the SpyderCube casts the shadow. The difference between extracted illuminants is much more visible in the regions illuminated by the ambient illuminant. This is caused by the complex geometry present in each scene. The various surfaces in the scene reflect light at various wavelengths, and there is no clear angle at which the ambient illuminant is cast. This in conjunction with the fact that the SpyderCube is a simple and relatively cheap object raised the question of how accurate the extracted illumination is. Because of this, experiments with SpyderCubes were performed to determine how to extract the illuminant and to determine how accurate the extracted illuminant is.

Two experiments were performed. One to test the accuracy of the extracted direct light source illuminant, and the other to test the accuracy of the extracted ambient light source illuminant. Both experiments were performed by placing multiple SpyderCubes into the illumination region that was being tested. For this, three different SpyderCubes were used. Multiple images with such a setup were created. After collecting the images, the average pixel values were extracted from all the SpyderCube faces in the image. The extracted average face values were grouped into two regions, left and right, based on which surface of a SpyderCube the face was. Then the similarity between the gray faces of each of the groups was calculated. To calculate

the color similarity of the two pixel values, the angular distance metric was used.

$$Angular\ distance = cos^{-1}\left(\frac{\mathbf{L}_1 \cdot \mathbf{L}_2}{||\mathbf{L}_1||_2||\mathbf{L}_2||_2}\right) \tag{5.1}$$

$\mathbf{L}_1$ represents the average pixel vector value of one face, $\mathbf{L}_2$ represents the average pixel vector value of the other face, $\cdot$ is the scalar product, and $||\ ||_2$ is the L2 norm. The angle is calculated in degrees.

The side with the bigger similarity between the gray faces is selected. If the maximum angle between the gray faces is less than 2° degrees, it means the extracted illumination is accurate. The 2° degree threshold was selected based on research done in [34]. This research states that humans cannot distinguish two colors if the angular distance between them is less than 2°.

The experiment on direct light showed that the angle between the gray faces does not exceed 1°. This tells us that the extracted illumination is quite accurate. The experiment on the ambient light gave a different result. It showed that in 20% of the images, the angular distance between the gray faces was over 2° and in those 20% the average angular distance was 3.05°. This means that additional measures need to be taken so that one can confidently say the extracted illumination is accurate. Because of this, each image in the dataset contains three SpyderCubes, one for direct light and two for ambient light.

The dataset contains three types of images, daytime outdoor, nighttime outdoor, and indoor images. For daytime images, the two illuminants can be easily identified, they are the sun and the sky. For indoor and nighttime images, the situation becomes more complex. In nighttime images, the direct light is a sodium, LED, or incandescent streetlamp. The night sky should take the role of the ambient light, but its intensity is too low and the role of ambient light is fulfilled by a combination of streetlamps that are further away and the reflections from surfaces in and outside the scene. This makes the ambient illumination ill-defined and can even cause the creation of regions with non-uniform illumination. Such images cannot be manually annotated, and the multiple SpyderCubes were used to detect such situations. Images, where this situation occurs, were not used in the dataset.

The same problem occurs in indoor images. Here, the direct light is usually LED light close to the scene, while the ambient is room lighting that gets reflected by various surfaces to illuminate the regions not illuminated by the direct light or natural lighting coming from a window. Non-uniformity is also common in such an image. The multiple SpyderCubes were used to detect such images so that they can be excluded from the dataset.

For direct light illumination extraction, two situations can occur. In the first situation, only one gray face is directly illuminated. Here the illuminant is extracted from the gray face that is directly illuminated. The second situation is where both SpyderCube surfaces are about equally

illuminated by direct light. Here the illuminant is extracted from all four SpyderCube faces. To select which gray face will be used, the similarity of the gray and white faces for each side is calculated. The used illuminant is extracted from the gray face, which is more similar to its white face.

To extract the ambient illuminant, a slightly more complex process is used. Here, the illumination from all the faces on both cubes is extracted. Then the extracted illuminant are grouped into left and right based on which side of a SpyderCube they are. The similarity between the gray faces is calculated for each side. The side that is more similar is selected. The gray faces on the selected side are then compared to the white faces on the same surface. The gray face which is more similar to its white face is selected as the illuminant.

In each image, the cubes in ambient light will have an angular distance of less than 2°, and the angular distance between the ambient and direct light is greater than 2°. With this, it is ensured that the illumination regions are uniform and that the two illuminants are different enough from one another.

5.3.3 Influence mask creation

The other information from each image is the per-pixel illumination influence mask. This mask can be looked at as a binary segmentation mask. This is possible because both the illuminants are uniform and there is a clear border between the two illuminants. Such a situation was selected for the dataset, as illumination influence masks are notoriously hard to manually create in a situation with non-uniform illumination. The method proposed in [32] automates the influence mask creation, but to use it you need to be able to turn off the scene illuminants. These conditions greatly limit the scenes that can be used. The influence mask was manually annotated for each image. Since manual annotation is an error-prone process, multiple experts verified each influence mask. A couple of images and their illuminant influence masks can be seen in Figure 5.10.

5.3.4 Dataset statistics

There are a total of 2500 images in this dataset. There are three types of images in the dataset, outdoor, indoor, and nighttime images. Five different cameras were used to create the dataset. There are around 2000 outdoor, around 300 nighttime, and around 200 indoor images. Each camera was used to capture around 500 images. The exact distribution by camera and by type can be seen in Table 5.1.

All the images have been minimally processed and are provided in `png` format. To get an image in `png` format, the `dcraw` was used. `dcraw` is an open-source program that is used to process images in various RAW formats. The `dcraw` program outputs 16-bit images in the

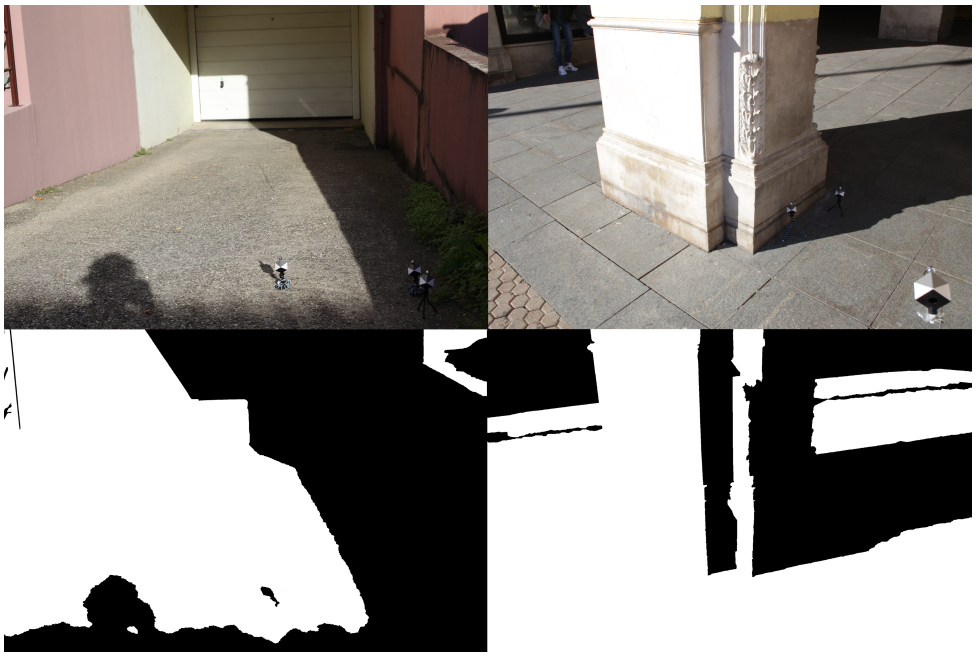


Figure 5.10: Two images and their segmentation masks underneath them.

	Outdoor	Indoor	Nighttime
Canon 5D	395	39	61
Canon 550D	403	44	57
Motorola	400	40	59
Sony	400	38	60
Panasonic	395	39	70

Table 5.1: Table representing the number of images taken by each camera presented by type of image.

image camera's RAW color space. We use simple debayering to debayer the image. The red and blue components are directly taken from the Bayer pattern, while the average value of two green components is used to create the green channel for each pixel. This method was used as it creates practically no artifacts. The width and height of an image are reduced by half, but the images have such a high dimensionality that this has no noticeable effect on how an image looks like. Certain images also contain black boxes over certain regions of the image. These black boxes exist to make the datasets GDPR-compliant as those regions contain sensitive private information such as faces or license plates. Because of this, the dataset contains only the `png` format images and the unedited RAW format images are not provided with the dataset.

The aperture, ISO value, shutter speed, and other image metadata are important settings in photography. Since the original RAW images are not provided, a metadata `txt` file for each image is provided so that anyone who uses the dataset has access to all image information. Because of the diverse set of scenes and diverse set of illumination settings, the camera setting

differs from image to image. The only consistency between the images was the ISO value, which was the smallest possible for each image.

The images in the dataset were created during various times of day and times of the year, there is a diverse set of illuminant present in the dataset. The distribution of illuminant by image type can be seen in Figure 5.11

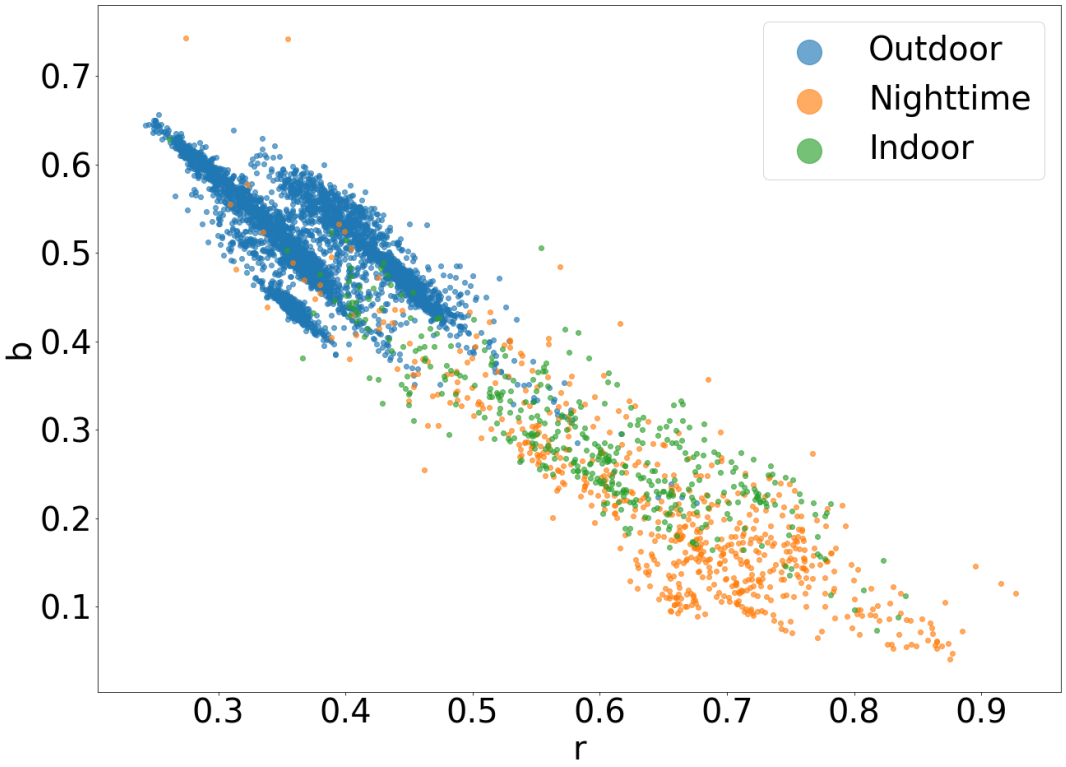


Figure 5.11: Illumination distribution of the dataset by image type.

In Figure 5.11 one can see that the illumination in outdoor images is bluer, the illumination in nighttime images is redder, while the illumination in indoor images contains both reddish and blueish illuminants. This can be explained by the fact that outdoor images contain only natural light which has a bluer hue, i.e., the blue sky. The nighttime contains only artificial illuminant lights that have a warmer, redder color, for example, tungsten lights. Indoor images have both warm and cold colors, as indoor images were taken during both day and night, and they contain natural and artificial lighting.

The illuminant distribution by camera can be seen in Figure 5.12. Figure 5.12 shows us that illumination differs by camera because of the different camera sensors each camera uses. The two Canon cameras have a similar distribution, which is to be expected since they have the same manufacturer. It can also be seen that the Motorola and Sony cameras have a similar distribution. This is an interesting discovery since the Sony camera uses a Sony camera sensor and Motorola uses a Samsung camera sensor. The Panasonic camera has a unique distribution with little overlap with other cameras.

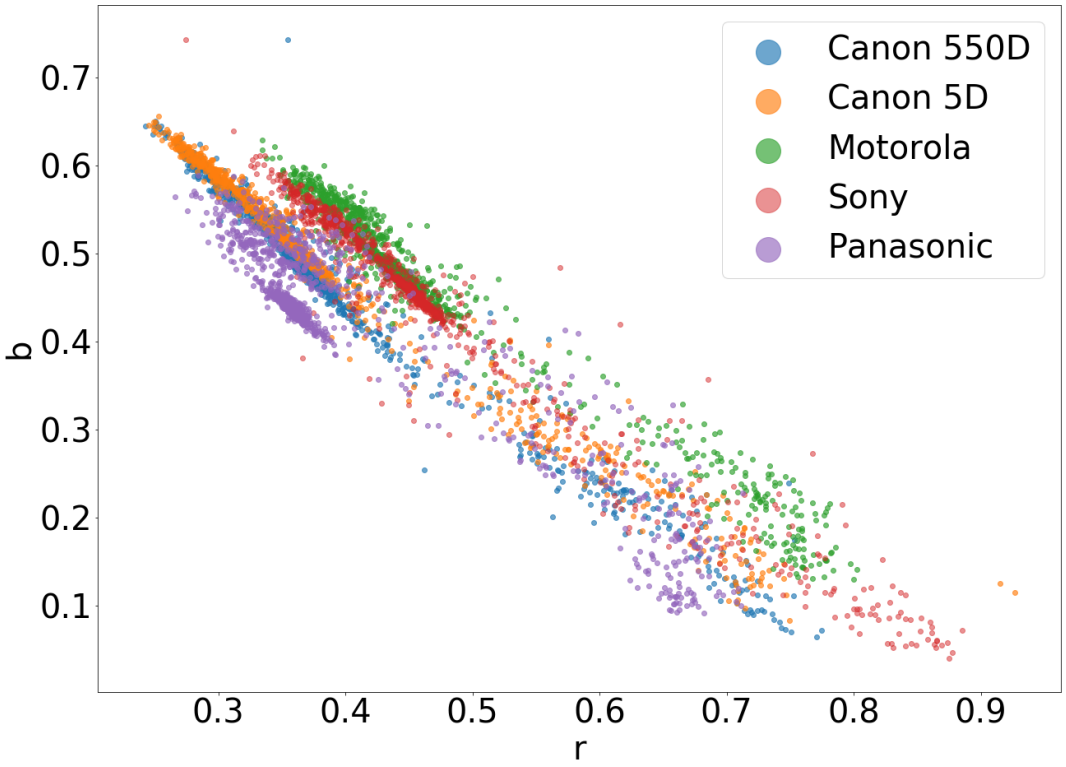


Figure 5.12: Illumination distribution of the dataset by camera.

5.3.5 Dataset evaluation

For datasets evaluation and proper method comparison, we propose three different protocols to evaluate different aspects of illumination estimation methods. The first protocol is a simple protocol used in most other datasets. All images are used. The second protocol is used to see how a method performs when it encounters images from an unknown camera. This protocol was created for learning-based methods, as the results of non-learning-based methods will produce the same results for both protocols. The final protocol is used to evaluate how a method performs on images evaluating only one type of image.

Before using the dataset, images need to be preprocessed. The first preprocessing step is the removal of image blacklevel. Each camera has a different blacklevel. Canon 5D has a blacklevel of 1024, Canon has a blacklevel of 2048, Motorola has a blacklevel of 63, Panasonic has a blacklevel of 127, and Sony has no blacklevel or a blacklevel of 0. Next, the oversaturated pixels need to be set to zero. This is done because oversaturated pixels are all white and contain no actual color information and should be ignored. After that, the calibration objects need to be masked out. This is because calibration objects contain the color of illumination, and the learning-based method can simply be trained to detect the calibration object and extract the image illumination. This would negate the ability of learning-based methods to generalize since normal images do not contain calibration objects. This is done by setting the pixels where the calibration objects are to 0.

We call the first protocol the Use-All protocol. Here the images are divided into 5 folds. Each fold contains images of all types and images from all the different cameras. To evaluate a method with this protocol, five experiments need to be performed. In each experiment, four folds are used for method training and one is used for method evaluation. This needs to be done for learning-based methods. For non-learning-based methods, all the images need to be fed into the method once.

The second protocol is called One-to-Many protocol. Here the dataset is also divided into 5 folds. Here, each folder contains images from only one camera. For this protocol, five experiments need to be performed. In each experiment, one fold is used for training and the other four are used for evaluation. Again, this protocol will produce the same results as the Use-All protocol for non-learning-based methods.

The final protocol is called By-Type protocol. This protocol actually evaluates three different things, how well a model performs on outdoor, indoor, or nighttime images. Here the dataset is divided into three subsets. Each subset contains only one type of image. Each subset is divided into 5 folds. To evaluate each subset, five experiments need to be performed. In each experiment, four folds are used for training and one is used for method evaluation. For non-learning-based methods, the results only need to be divided into three sets. Each set contains images from only one image type.

The metric used to evaluate the illumination estimation methods is angular distance 5.3.2 between the prediction and the ground truth. The metric to evaluate the image segmentation methods is the dice coefficient 5.3.5.

$$Dice = \frac{2TP}{2TP + FP + FN} \tag{5.2}$$

TP are the true positives or pixels that have correctly been labeled as pixels illuminated by the proper illuminant, FP are the false positives or pixels that have been incorrectly labeled as pixels illuminated by the proper illuminant, and FN are false negatives or pixels that have been incorrectly labeled as pixels not illuminated by the proper illuminant.

The dataset was used to test several existing illumination estimation and image segmentation methods. Methods proposed in this thesis were also tested on this dataset. The results obtained on this dataset are not presented in this chapter. Instead, they can be found in later chapters, where the results for the individual methods are presented.

5.4 Filters & Lumination dataset

While the Shadows & Lumination dataset is a large and diverse dataset, it only covers a subset of possible multi-illuminant situations. Because of this, another dataset using a different

methodology for image creation was developed. The dataset created using this method is called the Filters & Lumination datasets (Filters). It contains 7800 images from 300 scenes that were captured using 3 different cameras. The used cameras are, the Canon EOS 550D camera, the Panasonic DMC-FZ1000 camera, and the Samsung ISOCELL Plus GW1 1/1.72" camera sensor in the Motorola One Fusion+ mobile phone. The dataset contains a variety of outdoor, indoor, and nighttime scenes. The idea behind this dataset is to increase the illumination diversity not by increasing the scene diversity, but instead by using lighting filters. The idea takes inspiration from the authors of the Bleier et al.[31] dataset. The main difference between the Filters & Lumination dataset and the Bleier et al.[31] dataset is that instead of placing the lighting filters over the light source, the lighting filters are placed over the camera lens.

5.4.1 Motivation

There were two main motivating factors in the creation of this dataset. The first one are the shortcomings of the Shadow & Lumination dataset and the second one is the LSMI [33] dataset. The Shadows & Illumination contains a diverse set of scenes and illuminants, but each image only contains two illuminants. Also, because of the way the scenes are set up, there are multiple illuminants but they affect only a part of the image, so there is no illumination overlap. There are no non-uniform illumination regions in the image. This dataset cannot be used to evaluate method performance on images that contain a variable number of illuminants with non-uniform illumination regions. The other motivation is the recent release of the LSMI [33] dataset. This dataset contains over 2700 different scenes, images in the dataset contain anywhere from 1 to 3 illuminants, and images with multiple illuminants contain regions with non-uniform illumination. The shortcoming of this dataset is that it only contains images of indoor scenes. The goal of the creation of this dataset was to create a dataset with images that can have an arbitrary number of illuminants from a diverse set of indoor, outdoor, and nighttime images.

5.4.2 Dataset creation

The creation of the dataset is based on two previous works. The first is Bleier et al.[31] and their usage of filters. With lighting filters, one can alter the illumination without changing or editing the scene. The second work is Beigpour et al.[32] where they introduced a method for the automatic creation of an illumination influence mask.

The idea of this dataset is to take multiple images of the same scene, with each image having different illumination. This is achieved by placing a different lighting filter in front of the camera lens. This changes the illumination while leaving everything else the same. When looking at the image formation model 3.2, the addition of a filter adds another variable $F(\lambda)$ into the formula that represents the spectral response function of the filter.

$$\mathbf{l} = \begin{bmatrix} l_R \\ l_G \\ l_B \end{bmatrix} = \int_{\omega} F(\lambda) I(\lambda) p(\lambda)\, d\lambda \tag{5.3}$$

The $F(\lambda)$ and $I(\lambda)$ variables can be combined into $\hat{I}$ that is identical to the original formation model with only the illumination being different.

For this dataset to contain multi-illuminant images another condition needs to be satisfied. The other condition is that the scene being captured needs to have a single uniform illuminant. This goes contrary to the idea of multi-illuminant images, but here the research from Beigpour et al. [32] is employed. In their research, they use the fact that the illumination of an object illuminated by two illuminants is simply a linear combination of the situations when the object is illuminated by each illuminant separately. They have images with multiple illuminants and use illumination linearity to extract the influence of each illuminant in each pixel when both illuminants illuminate the object.

In the created dataset, there are no images with multiple illuminants, but there are multiple images of the exact same scene under different illuminations. By combining the different images of the same scene, one can create an image with multiple illuminants. Only the influence of each illuminant on each pixel needs to be set. There are no conditions when setting the influence coefficient of each illuminant, which allows us to create an arbitrary number of illumination influence masks with the only limitation being the number of filters used while taking images of a scene.

To maximize the scene and illumination diversity, 25 different lighting filters were used while taking images of a scene. The names given to the filters are Night 1, Night 2, Night 3, Night 4, Night 5, Night 6, Day 1, Day 2, Day 3, Day 4, Gray H, Green light H, Yellow H, Beige H, Transparent H, Orange light, Orange dark, Blue light, Purple, Yellow, Gray, Red H, Red dark, Green dark, and Blue dark. This means that there are 26 different images of the same scene because an image without a filter was also taken. An example of how the filters affect the scene can be seen in Figure 5.13

5.4.3 Illumination extraction

To extract the illumination from an image, the SpyderCube calibration object was again used. The number of SpyderCubes used depends on the type of scene being captured. In outdoor images, there is only one SpyderCube present, while in indoor and nighttime images there are two SpyderCubes. This was done to ensure the uniformity of the illumination in the nighttime

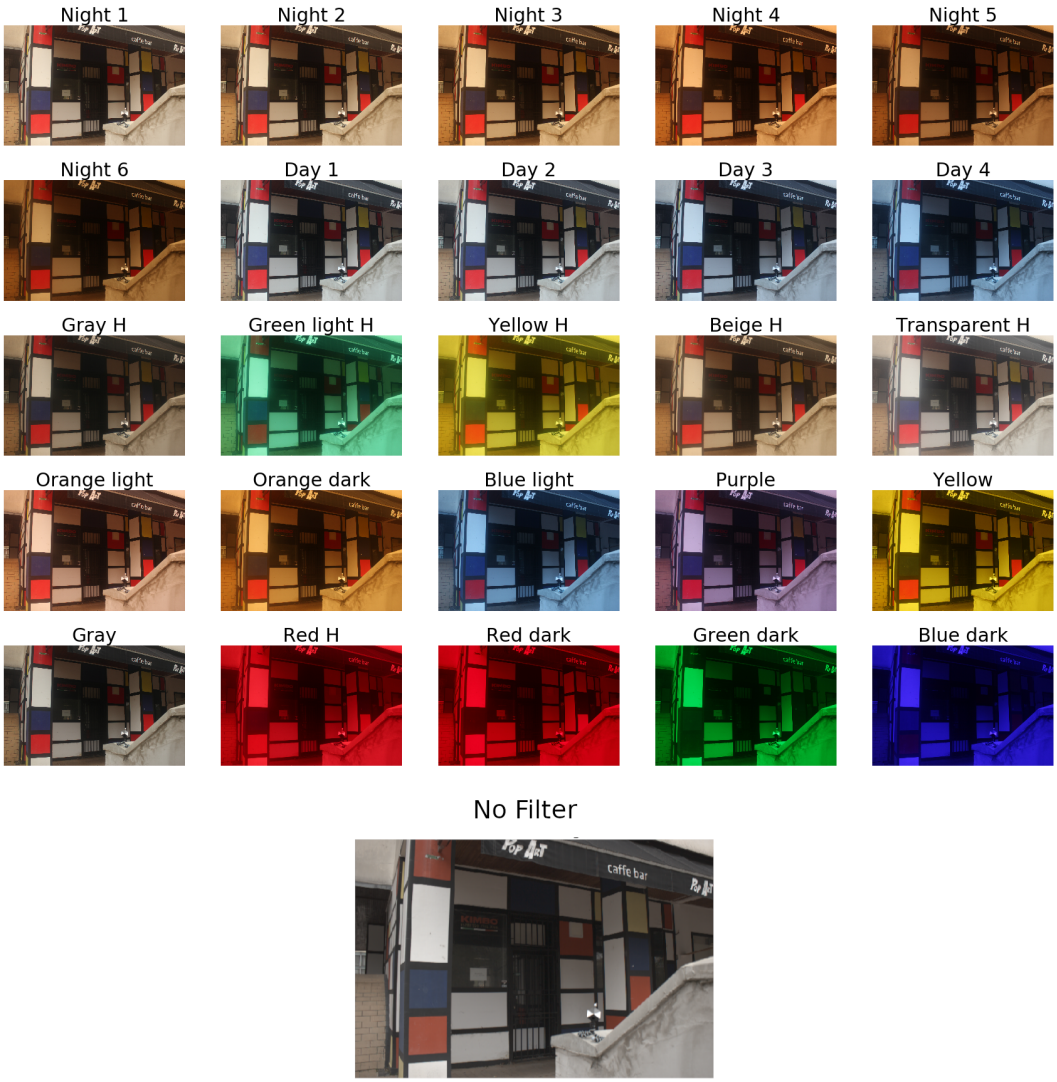


Figure 5.13: Example images of the same scene taken using different filters. The image at the bottom was taken without a filter.

and indoor images. In outdoor daytime situations, it was easy to ensure that a single uniform illuminant is present. All daytime images were taken during cloudy weather where neither the sun nor the blue sky is visible. To ensure this, the illumination was extracted from both gray faces of the SpyderCubes. If their angular distance was greater than 2° the scene was not included. The gray face which is more similar to its white face was selected as the image illumination.

In indoor and nighttime situations one cannot easily create a situation with a single uniform illuminant. This is because in such situations LED lights or streetlamps are used. Unlike the cloudy sky, artificial lights have a single point from which the light rays come. To ensure illumination uniformity two SpyderCubes were placed on the scene. The process of illumination extraction is identical to the process described in Chapter 5.3.3 for ambient light extraction. Again, if the angle between the gray faces on the more similar side is greater than 2°, the

images were not included in the dataset.

5.4.4 Illumination influence mask creation

Since there is no limit on how to set the influence of each illuminant on each pixel, a wide variety of segmentation methods can be used. For the purpose of making the influence masks as realistic as possible, the following algorithm was developed for mask creation.

Algorithm 1 Algorithm used to create an illumination mask

Require: image, illuminant, $n > 0$, $L > 1$
  $CorrImage = ColorCorrect(image, illuminant)$
  $mask1 = CreateEmtpyMask(1)$
  $i \leftarrow 0$
  $N \leftarrow n$
  while $i < N$ **do**
    $mask1 = drawLine(mask_1)$
    $i+ = 1$
  end while
  $mask1 = AssignRegionNumber(mask1)$
  $ruler = random(0.6, 1.0)$
  $regionSize = random(600, 1001)$
  $mask2 = CreateSuperpixelRegions(CorrImage, ruler, regionSize)$
  $M \leftarrow GetNumberRegions(mask2)$
  $C = max(M, N)$
  $i \leftarrow 0$
  $mask3 = CreateEmtpyMask(C)$
  while $i < C$ **do**
    if $i \leq N$ **then**
      $mask3[i][mask1 == i] = 1$
    end if
    if $i \leq M$ **then**
      $mask3[i][mask2 == i] = 1$
    end if
    $kernel = random(5, 301)$
    $mask3[i] = blur(mask3[i], kernel)$
    $i+ = 1$
  end while
  $FinalMask = CreateEmtpyMask(L)$
  $i \leftarrow 0$
  while $i < C$ **do**
    $FinalMask[i\%L]+ = mask3[i]$
  end while
  $FinalMask = L1Normalize(FinalMask)$

The algorithm has two components. Each component creates its own mask. The two created masks are then combined to create the final illumination influence mask. An example of the

created masks can be seen in Figure 5.14.

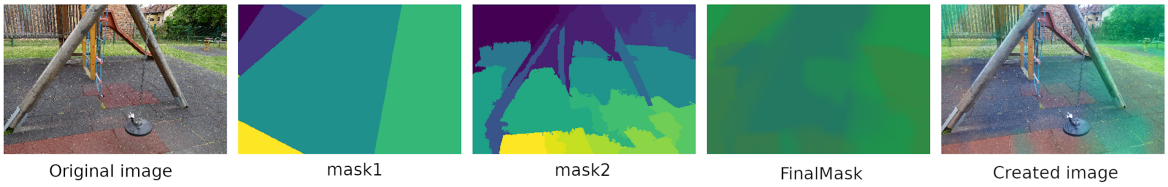


Figure 5.14: An example of the created intermediary masks used to create an illumination map. All images have been tone-mapped for better visualization.

The first component creates a scene-independent mask. This mask is used to simulate the objects outside the frame that influence the illumination mask. The mask is created by dividing the image into a N regions using randomly placed straight lines. The regions get assigned values from 1 to N.

The second component creates a scene-dependent mask. This mask is used to simulate the effect the scene geometry has on the illumination mask. The mask is created by dividing the image into regions using superpixels. The SLIC (Simple Linear Iterative Clustering) [35] method was used to divide the image into M regions. The regions get assigned values from 1 to M.

After both masks have been created, $max(N,M)$ matrices with the same spacial dimensions as the image are created. Each matrix has a value of 1 in pixels that belong to either mask region with the same value as the matrix index. The other pixels in the matrix have a value of 0. An average blur with a random kernel size is applied to each matrix. They are then added to create C matrices. C is the number of images that will be used to create the multi-illuminant image. The matrices are combined modulo C on the matrix indexes. A couple of example multi-illuminant images can be seen in Figure 5.15.

5.4.5 Dataset statistics

The way the images were processed in this dataset is the same as in The shadows & Lumination dataset. The images are in 16-bit png format. The same debayering was used. The camera setting varies from image to image, with the ISO being the smallest possible for each camera. This dataset was also made to be GDPR-compliant. All sensitive private information was masked out.

A couple of experiments were performed to test how similar the filters are to one another, how the filters affect the created images, and how the illumination distribution looks when compared to other datasets.

The desire to examine the filter similarity stems from the fact that several filters that simulate the artificial illumination that is present at night. They look very similar and produce similar-

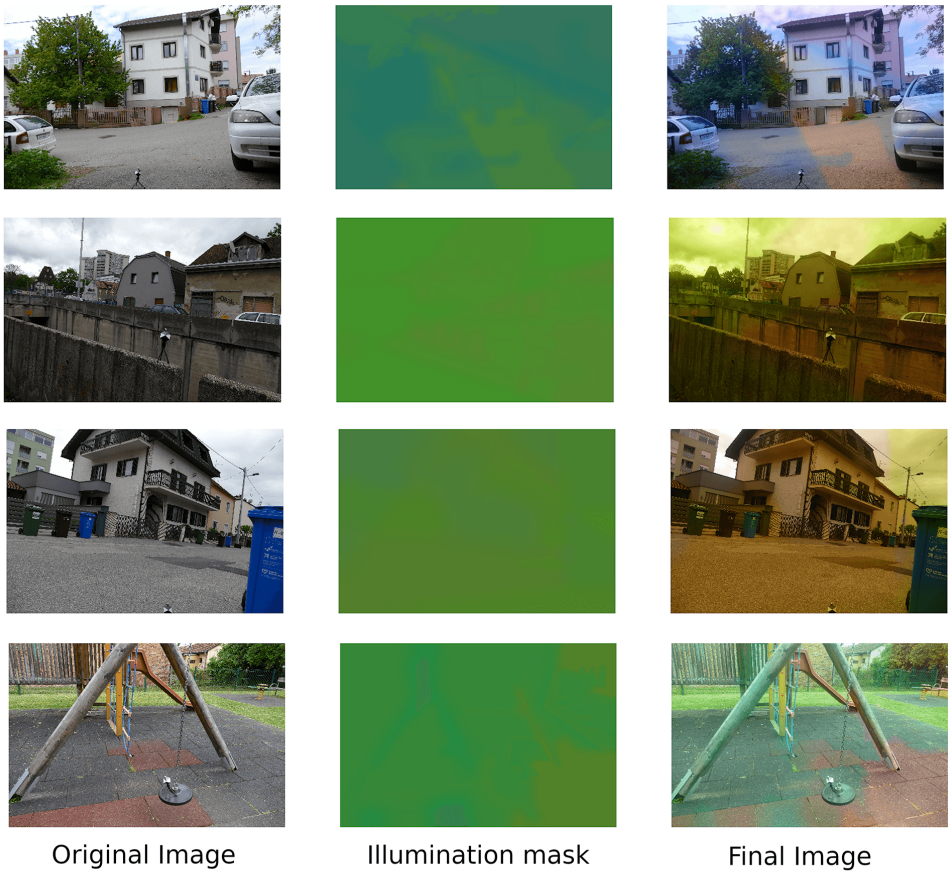


Figure 5.15: A couple of examples of created multi-illuminant images. The first column shows what the original image looks like. The second column shows the illumination mask we created using the different filters. The last column shows how the image looks when the illumination mask is applied. All images have been tone-mapped for better visualization.

looking images. The same goes for the filters that simulate natural daytime lighting. For this, a confusion matrix that shows in how many scenes the two filters produce illuminants whose angular distance is over 2° was created. The confusion matrix can be seen in Figure 5.16.

Figure 5.16 shows that in most situations all the filters produce different illuminants, but there are some situations that produce similar illuminants in a subset of scenes. There are also a couple that produces similar illuminants in almost all situations.

The second thing that was examined was how the filters affect the image and image correction process. Before creating the dataset, images of a couple of test scenes were created. The illuminant for each filter was extracted. The extracted illuminants were then used to correct the images. In most situations, the filter-corrected image looked identical to the corrected image when no filter was used. This was not the case for four filters. The Red H, the Red dark, the Green dark, and the Blue dark filters produced unnaturally colored images when they were corrected. An example scene can be seen in Figure 5.17.

In Figure 5.17 one can see what the corrected image looks like when no filter is used. When

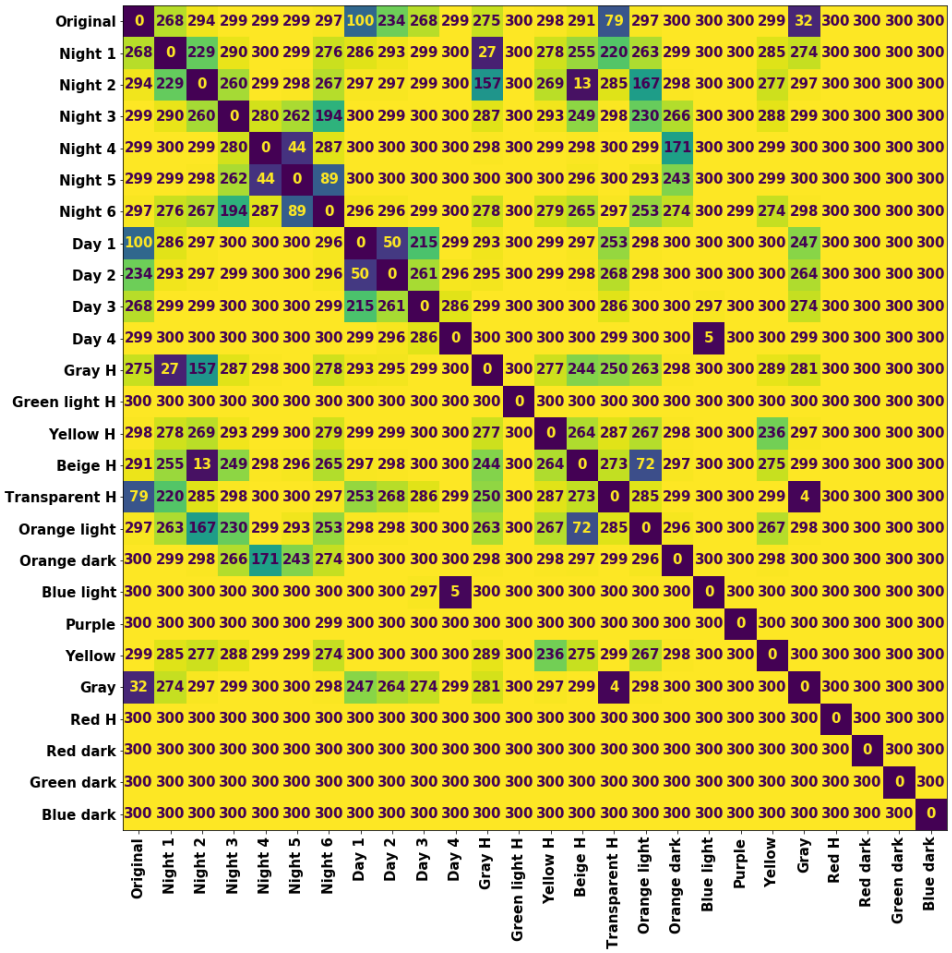


Figure 5.16: Confusion matrix that shows in how many scenes the two filters had significantly different illumination. Significantly different means, their angular error was over 2°.

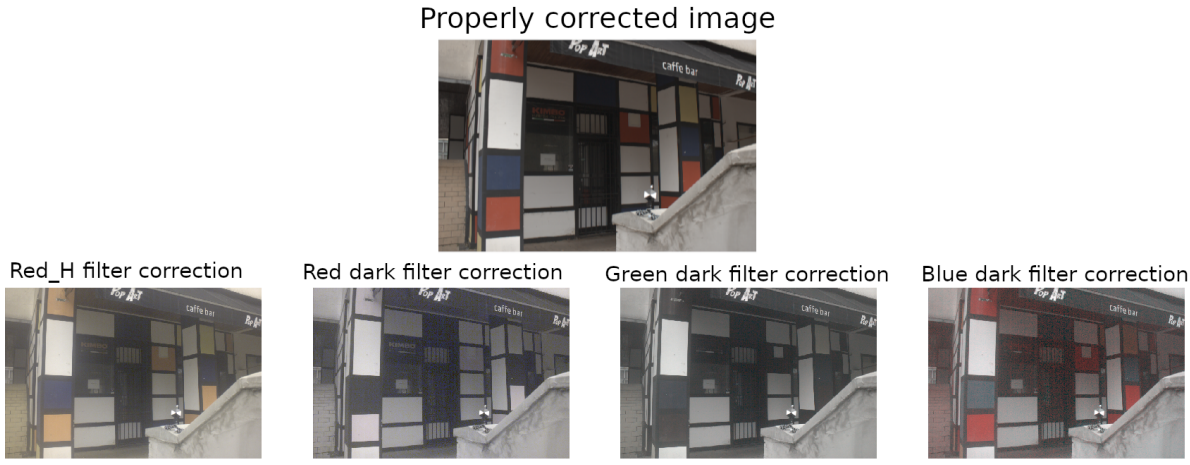


Figure 5.17: A couple of examples where the image cannot be properly corrected because of the extreme filters. The top row shows how the image should look and the bottom row the four situations where the image cannot be properly corrected.

we compare this image to the Red H filter we can see that the yellow and blue patches have the expected color. The red patch on the other hand loses most of its color turning into a light

orange/yellow color. When looking at the Dark blue filter, the red patch retains its expected color. Here the blue and yellow patches lose color. The yellow patch becomes slightly darker, while the blue patch becomes completely gray. The corrected Dark red and Dark green filters create images that look grayscale. The difference between the two filters is the red patch. With the Dark red filter, the red patch becomes white and with the Dark green filter, the red patch becomes black. This phenomenon can be explained by looking at the histogram of the unedited Dark red filter image in Figure 5.18.

Figure 5.18 shows us that since the filter is extremely red, little light in the blue and green wavelength passes it. The histogram of the red channel has a normal-looking distribution of values. In such a situation the green and blue channels contain little useful color information. When the image is corrected the red color is removed resulting in a grayscale image.

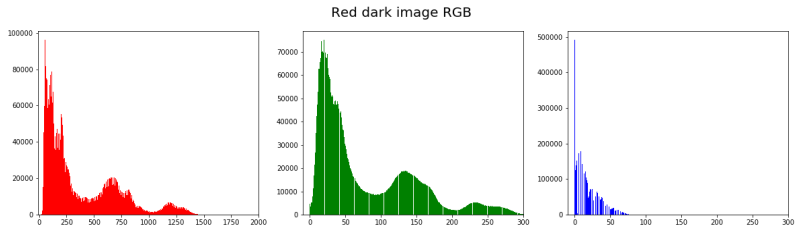


Figure 5.18: Three histograms for the three color channels. The x-axis shows how much more information is stored in the red channel than in the green and blue channels.

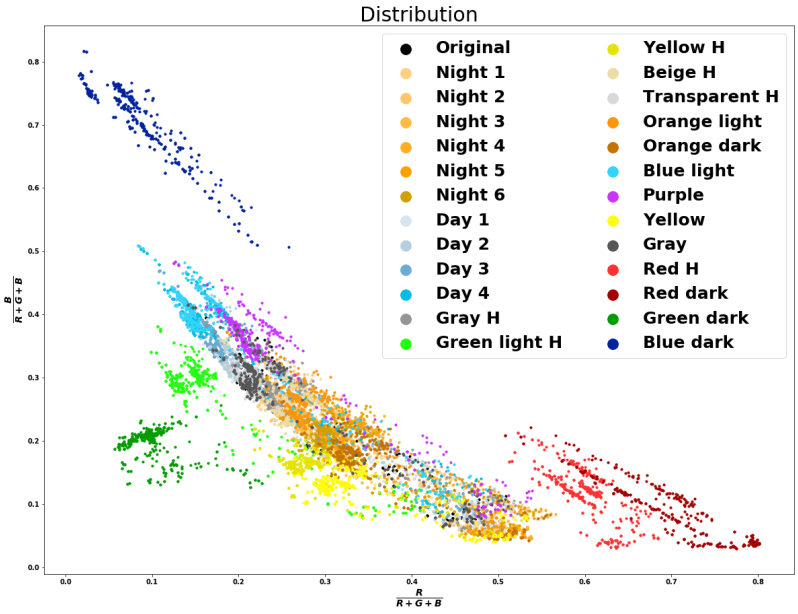


Figure 5.19: The illumination gamut of the images in the dataset. The illuminations are grouped by which filter was used to take the image.

Figure 5.19 shows us that most of the filters are clustered in a line in the lower left corner

of the graph. There are three outlier clusters visible in the graph. An extreme blue, an extreme red, and an extreme green cluster. These three clusters create the endpoints of the triangle of the possible illuminants.

Figure 5.20a shows how the Filters illuminant distribution compares to the LSMI illuminant distribution. Figure 5.20b shows how the Filters' illuminant distribution compared to the Shadows & Lumination illuminant distribution. When compared to the Shadows & Lumination dataset, the Filters dataset has a much larger diversity in illuminant. This is true even though the Filters dataset contains significantly fewer different scenes and fewer cameras were used to create the dataset. When compared with the LSMI dataset, the Filters dataset has more variety in its illuminants. The interesting this to note in this graph is that there is an overlap between the Filters and the LSMI dataset in the extreme red area of illumination.

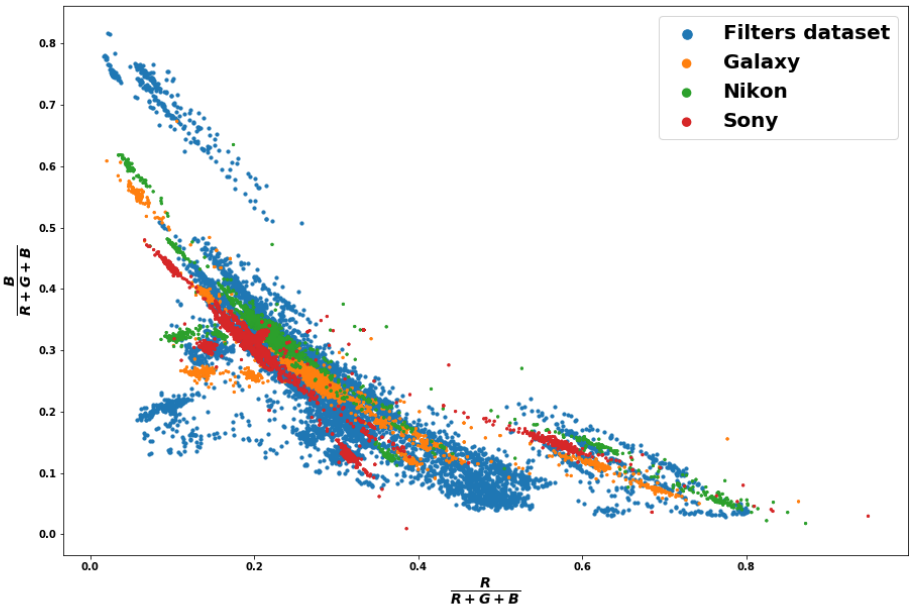
5.4.6 Dataset evaluation

For this dataset, no special evaluation protocols are proposed. If we use the influence mask creation algorithm, the dataset can be used to evaluate how a method performs on multi-illuminant non-uniform illumination images. This is the original idea behind this dataset. Because there are filters that cause images to be unnaturally corrected, two varieties of multi-illuminant non-uniform illumination image sets were created. One that includes all the filters (Full Filters dataset), and one that only contains images that can be properly corrected (Reduced Filters dataset). For each set using the algorithm, 20 different images of each scene were created and for each image, a random number of one illuminant images was used. The number of used images can be anywhere from 1 to 10.

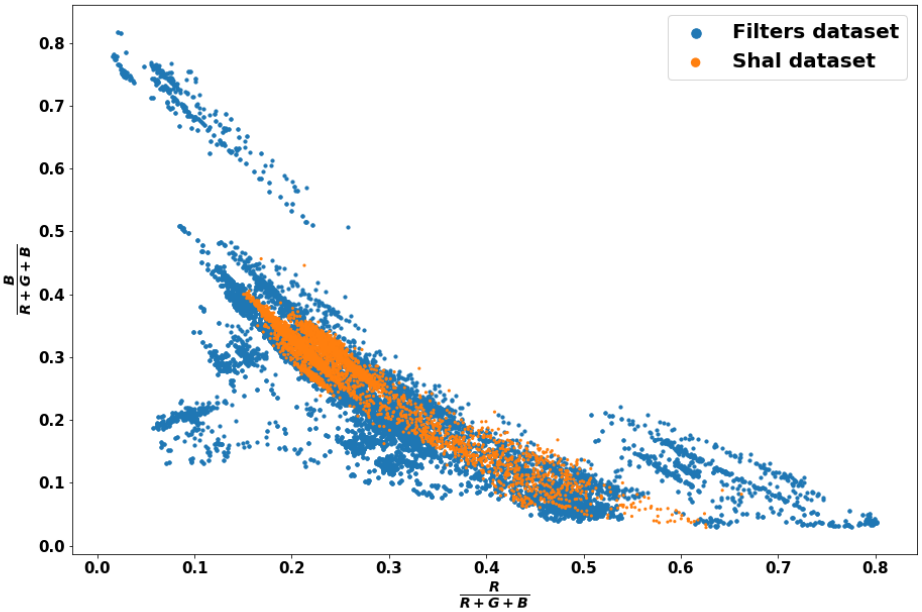
Since the original images in the dataset contain only one uniform illuminant, the unedited images can also be used to compare single illuminant estimation methods and see how they perform on a large diverse set of illuminants and see which illumination situations are harder and which are easier for a method. Here the dataset is split into a set that contains all images (Full Unedited Filters dataset) and a set that contains only images that can be properly corrected (Reduced Unedited Filters dataset).

Here, a 3-fold dataset split is used for both situations. Three experiments need to be performed for each subset. In each experiment, two folds are used for training and one is used for evaluation. For non-learning-based methods, all the images need to be fed into the method. An important thing to note is that the images need to be split based on the scene they are showing. Images of the same scene cannot be present in two folds.

For this dataset, the same preprocessing steps that were used on the Shadows & Lumination dataset were used on this dataset. This means the blacklevel needs to be removed, the oversaturated pixels need to be set to 0 and the SpyderCube calibration object needs to be masked out.



(a) Comparison of the illumination gamuts of the Filters dataset and the LSMI dataset.



(b) Comparison of the illumination gamuts of the Filters dataset and the Shadows & Lumination dataset.

This dataset was tested on several existing multi-illuminant estimation models and our own multi-illuminant estimation methods. The results obtained on this dataset are not presented in this chapter. Instead, they can be found in later chapters, where the results of the proposed methods are presented.

Chapter 6

Illumination estimation: methods, analysis, evaluation, results

The main goal of this research was not the creation of a color constancy dataset that contains images with multiple illuminants with non-uniform illumination. Instead, the main goal was the creation of a method for illumination estimation in an image regardless of what illumination the image contains. Such an image can contain natural, artificial, or both types of lighting. The number of illuminants and the illumination uniformity should not affect the method.

To achieve this, five different illumination estimation methods were developed. The first is the simplest, it performs illumination estimation on images with a single uniform illuminant. The second method performs illumination estimation on images with multiple illuminants. The number of illuminants must be known beforehand for this method to work. The third method is a patch-based illumination estimation method. This method does not need to know the number of illuminants in an image. The final two methods perform per-pixel illumination estimation. In this work, an overview of each of these methods will be given.

6.1 Single illuminant estimation method

In this chapter, an overview of the created single-illuminant estimation method is given. This method is a learning-based method. It is a lightweight convolutional neural network. It has under 22k parameters and uses 5 convolutional layers. The method was tested on several single-illuminant color constancy datasets, and it achieves state-of-the-art results.

6.1.1 Motivation

This method was developed as a starting point for the research. It is the simplest variety of the color constancy problem, and there are many existing methods that achieve results better

than the Human Visual System. The idea was to examine how the existing methods achieve great results. The examination showed that the greatest results were achieved by learning-based methods, more specifically, methods that use convolutional neural networks [21, 36, 37]. Research also showed that many of these methods used complex neural networks with several million parameters. Such methods are very computationally complex, and white-balancing methods need to run in near real-time. These results were the motivation for the creation of a simple and fast single illuminant estimation method.

6.1.2 Model architecture

The proposed method is a simple convolutional neural network. It consists of 5 different convolutional layers. After each convolutional layer, the ReLU activation function was used. An important feature of this neural network is the fact that all the convolutional layers use a kernel size of (1,1). The use of such a small kernel size significantly reduced method complexity. In addition to convolutional layers, the model also has two max-pooling layers, a dropout [38] layer and a global average-pooling layer. The entire architecture can be seen in Figure 6.1.

The model takes images of size 384×384. The first layer of the model is a convolutional layer with 64 filters of size (1,1). Its output is fed into a max-pooling layer with a kernel size of (8,8). The output of the max-pooling layer has a shape of (48,48,64). After that, the max-pooling output is fed into another convolutional, max-pooling layer block. This block has the same parameters as the original block. The created output has a shape of (6,6,64). This output is fed into a convolutional layer with 128 filters with a size of (1,1). After that, the output is fed into another convolutional layer with 64 filters of size (1,1). Its output is fed into a dropout layer with a rate of 0.5. After that comes the final convolutional layer that has 3 filters with size (1,1). The purpose of this layer is to reduce the output dimensionality to 3 so that it has the same number of elements as an RGB vector. The (6,6,3) dimensionality output is then fed into the global average-pooling layer that outputs the predicted RGB values of the image illuminant.

6.1.3 Data preprocessing

Before being fed into the model, several augmentations are applied to an image. Some augmentations are applied only during training, while others are always applied. Random image cropping, image rotation, image flipping, and illumination noise addition are applied during training. These augmentations are applied to increase the variety of images present in the training dataset. Image cropping, image rotation, and image flipping are standard image augmentation used for the training of many neural networks.

The illumination noise addition is an augmentation added specifically for the problem of illumination estimation. The first step for this augmentation is to use the von Kries[5] model

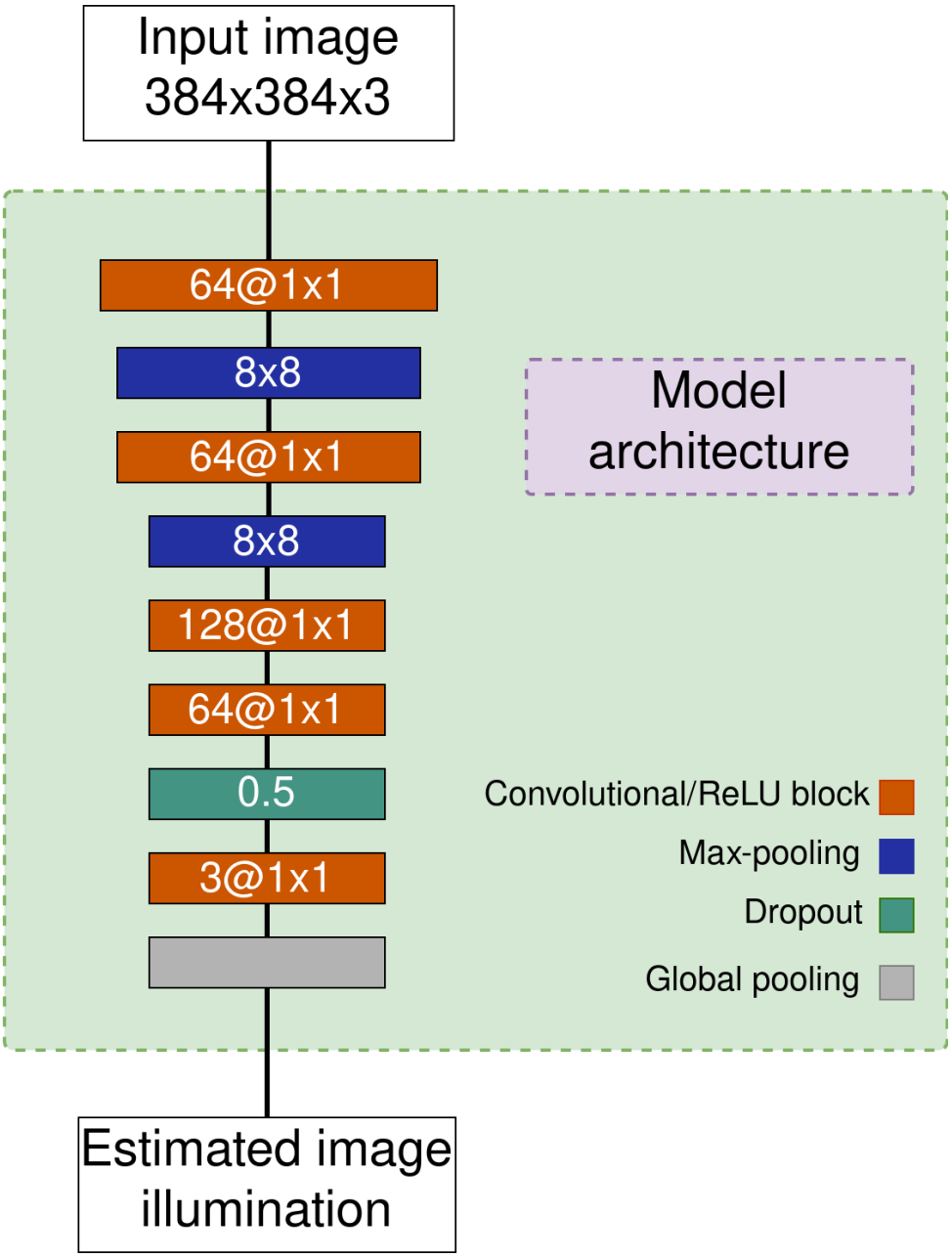


Figure 6.1: Model architecture of the single illuminant estimation method

to correct the image with the ground truth. Then a random noise from a normal distribution is added to the image ground truth illumination. The inverse of the von Kries[5] model is used to apply the new illuminant to the image. There are existing augmentations that produce a similar result, but they scale the original illuminant [21] or add a random natural illuminant [39]. The idea of our augmentation is to increase the ground truth illumination gamut. For example, most of the images in the Cube+ [29] dataset are of outdoor daytime scenes. There are fewer nighttime images and indoor images and their illumination significantly differs from the illumination of the daytime images. Because the added noise is small, the newly created illuminants remain within or very near the realm of possible illuminants for a given image. This

results in images that have realistic illumination even though the illumination is artificial.

Another important problem that this augmentation tackles is camera sensitivity invariance. If a model is trained on a dataset that only contains images from one camera, then the model will learn how to estimate illumination for that particular camera sensor. When such a model is tested on images from a different camera, it will produce less accurate results. The illumination gamuts of two camera sensors will have a similar shape, but they will be slightly translated to a different part of the illumination space. The training illumination gamut will not increase significantly with the added random noise but this will cover more camera sensors without needing the explicit knowledge of which camera sensor is used.

The augmentation uses a normally distributed additive noise. The mean of the noise is set to 0. The standard deviation is a method hyperparameter. Its value depends on the used dataset. Even though the increase in illuminant variety is small, tables in Chapter 6.1.7 show that it increases model accuracy.

Different standard deviations on different datasets were tested. For the Cube+ [29] dataset, the best results were obtained with a standard deviation of 0.01. For Intel-TAU [30] the used value was 0.05, for NUS-8 [28] the used value was 0.02. For the Full filters 0.01 was used and for Reduced Filters 0.01 was used.

Finally, before being fed into the model, the image is standardized. The mean value of the image is 0 and the standard deviation of the image is 1. This is done for all images regardless of whether it is during training or testing The model output and the ground truth illumination are L2 normalized since the intensity of the illumination color is not important.

6.1.4 Training setup

This method was implemented in Python [40] using TensorFlow 2.9 [41] for model implementation. For model training, the AdamW [42] optimizer with a weight decay of $5e^{-5}$ and the loss function introduced in [43] were used. The model was trained for 400 epochs on the Nvidia 2080Ti GPU and AMD Ryzen 7 3700X CPU. After training, when the method is running in single-thread CPU mode, it can process 25 images per second. A mini-batch of 64 was used. For the learning scheduler, the cyclical learning rate [44] with a half cycle of 200 epochs was used. The minimum learning of $1e^{-7}$ and maximum learning rate of $2e^{-3}$ were used.

$$Loss = \left\| \frac{ill_{pred} - ill_{gt}}{ill_{gt}} \right\|_2 \tag{6.1}$$

Equation 6.1 is the Li et. al. loss[43], where the ill_{pred} is the estimated illumination and ill_{gt} is the illumination ground truth.

6.1.5 Evaluation

To evaluate the method, four datasets were used. The Cube+ [29], the Intel-TAU [30], the NUS-8, and the unedited Filters & Lumination dataset.

For Cube+ and the unedited Filters datasets, we used a 3-fold split. The NUS-8 has multiple images of the same scene captured by different cameras so a simple 3-fold would cause data leakage. For that reason, a 3-fold split is used for each camera and the results are combined to get the final results. The Intel-TAU dataset provides two protocols that can be used to evaluate a method. The first one is a 3-fold split, where each split contains images from only one camera. The second is a 10-fold split, where each split contains images from all cameras. Two variants of the Filters & Lumination were tested. The first variant uses all the images from the dataset and the second variant excludes images that were taken with Red_h, Dark_Red, Dark_green, and Dark_blue. For both variants, a 3-fold split was used, where each split has images from all present filters.

To compare the results of the models, the angular error was used. The standard evaluation metrics used in other works were used. This includes mean, median, trimean, Best 25% mean, and Worst 25% mean.

6.1.6 Ablation study

Since the fact that the best results are obtained using only convolutional layers with kernel sizes of (1,1), it was decided to perform an ablation study to see what results are obtained using different filter sizes. The ablation study was performed on the Cube+ dataset. Five different varieties of the model were used to perform the test. The architecture of each model is the same, except for the kernel size used in the convolutional layers. Four of the models use the same kernel size for all convolutional layers. The used kernel sizes were: (1,1), (2,2), (3,3), and (5,5). The final model uses different kernel sizes for each convolutional layer. The used kernel sizes are (1,1), (3,3), (5,5), (3,3), and (1,1) in sequence from the first to the last convolutional layer.

Two additional ablation studies were performed. One was done to see how the model performs when one of the convolutional layers was removed. A test without the final convolutional layer was not performed because it is used to set the dimensionality of the output. The other study was done to see how the model performs with different-sized max-pooling layer kernel sizes. The tested kernel sizes were (4,4), (8,8), and (16,16). The results of all three ablation studies can be seen in Table 6.1.

Table 6.1 shows that the model architecture with convolutional kernel sizes of (1,1), a max-pooling kernel size of (8,8), and 5 convolutional layers achieves the best results for each metric. It can also be seen that the increase in kernel size causes an increase in model complexity and a

Method variant	Mean	Med.	Trimean	Best 25%	Worst 25%	number of parameters
Kernels (2,2)	1.27	0.77	0.90	0.24	3.13	84k
Kernels (3,3)	1.31	0.77	0.90	0.23	3.30	188k
Kernels (5,5)	1.39	0.82	0.95	0.24	3.51	522k
Kernel sequence 1-3-5-3-1	1.27	0.76	0.88	0.22	3.18	316k
Max-pool 4	1.55	0.97	1.12	0.27	3.80	21k
Max-pool 16	1.81	1.09	1.26	0.24	4.61	21k
No Conv1	1.84	1.17	1.35	0.38	4.41	17k
No Conv2	1.95	1.17	1.36	0.32	4.84	17k
No Conv3	1.31	0.80	0.93	0.22	3.27	8k
No Conv4	1.37	0.81	0.93	0.22	3.27	13k
Proposed variant	1.21	0.71	0.84	0.21	3.04	21k

Table 6.1: Results obtained on Cube+ with different kernel sizes, different max-polling kernel sizes, and different number of convolutional layers

drop in model accuracy. The second-best results are obtained with the model that has a kernel size of (2,2) and the model with different kernel sizes in each convolutional layer. Their results are slightly worse than the proposed variant while having 4x and 15x more parameters. The different kernel sizes cause a significant decrease in model accuracy. The removal of convolutional layers decreases the model complexity, but it also decreases the accuracy of the model.

Since the loss introduced in [43] is relatively new, its results were also compared with the results obtained when using Mean Squared Error. Table 6.2 show that the [43] loss achieves better results in all metrics except for the worst metrics, where it has a 2° bigger angular error.

Method	Mean	Med.	Trimean	Best 25%	Worst 25%	Worst
Li et. al. loss [43]	1.21	0.71	0.84	0.21	3.04	11.99
Mean squared error	1.31	0.86	0.97	0.26	3.15	10.07

Table 6.2: Comparison of results for different loss functions

6.1.7 Results

In this chapter, the results obtained by the proposed method on different datasets are presented and they are compared with the results obtained by current state-of-the-art methods. Some qualitative results can be seen in Figure 6.2.

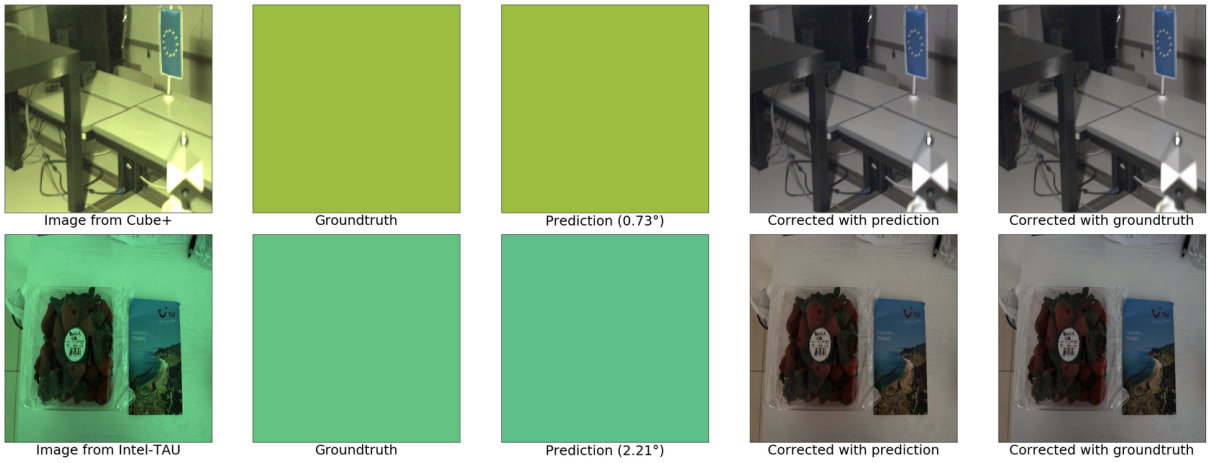


Figure 6.2: Visualization of color constancy results. Images are edited for visualization. Two examples are presented, one from the Cube+ dataset and one from the Intel-TAU dataset. The angular error in degree is also given.

In Table 6.3 the results obtained on the Cube+ dataset can be seen. It can be seen that the proposed method outperforms all other methods in almost all metrics. In the Worst 25% and Best 25% the FFCC [45] and MDLCC [37] achieve the best results. The FFCC method outperforms all methods when looking at the Best 25% but achieves results worse than other learning-based methods when looking at the median and Worst 25%. This shows that the FFCC struggles with outliers, unlike the proposed method.

The MDLCC outperforms the proposed model when looking at the Worst 25%. The unique aspect of MDLCC is that it uses multiple cameras from different datasets for model training. The additional camera sensors allow the method to reduce the effect of Cube+ outlier images.

Table 6.4 shows the results on the NUS-8 dataset. Here it can be seen that the proposed method does not outperform other existing methods. When comparing the proposed method with FFCC, the proposed method achieves slightly worse results in all categories but the Worst 25%. The MDLCC method achieves reasonably better results for all metrics. Even still the proposed method achieves results around the human eye threshold of 2° degrees. This is because research from [34] states that the Human Visual System cannot distinguish colors when the angular error between them is 2° or less. The only metric where this is not true is the Worst 25% where the error is over 2°. The same can be said about the MDLCC method. The proposed model is significantly less complex than MDLCC, having over 50x fewer parameters.

The next dataset is the Intel-TAU dataset. For this dataset, there are two Tables. Table 6.5 for the Intel-TAU camera invariance protocol results and Table 6.6 for the Intel-TAU Cross-validation protocol. In table 6.5 it can see that the proposed method significantly outperforms all other methods in all metrics. This shows us that even though the proposed method is simple, it does not overfit on the camera sensor and can be used with multiple cameras when the illumination noise data augmentation is used.

Method	Mean	Med.	Trimean	Best 25%	Worst 25%
White-Patch [2]	9.69	7.48	8.56	1.72	20.49
Gray-world [46]	7.71	4.29	4.98	1.01	20.19
Double-opponency (max pooling) [47]	6.76	3.44	4.15	0.79	18.54
Using gray pixels [48]	6.65	3.26	3.95	0.68	18.75
Shades-of-gray [49]	2.59	1.73	1.93	0.46	6.19
1st-order Gray-Edge [50]	2.41	1.52	1.72	0.45	5.89
2nd-order Gray-Edge [50]	2.50	1.59	1.78	0.48	6.08
General gray-world [51]	2.38	1.43	1.66	0.35	6.01
Attention CNN [52]	2.05	1.32	1.53	0.42	4.84
FFCC(model J) [45]	1.38	0.74	0.89	**0.19**	3.67
FC4(Squeezenet) [21]	1.35	0.93	1.01	0.30	3.24
[36] (VGG16)	1.34	0.83	0.97	0.28	3.20
MDLCC [37]	1.24	0.83	0.92	0.26	**2.91**
Proposed model (no noise)	1.25	0.74	0.86	0.20	3.17
Proposed model (with noise)	**1.21**	**0.71**	**0.84**	0.21	3.04

Table 6.3: Comparison of results obtained on the Cube+ dataset. The best angular error for each metric is bolded.

In Table 6.6 it can see that the proposed method outperforms all other methods in all metrics except for the Best 25% where the FFCC method achieves the best results. This is the same situation as for the Cube+ dataset, where the Best 25% is better than all other methods but the Worst 25% is significantly worse when compared to the proposed method.

Tables 6.7 and 6.8 showcase the accuracy the model obtains on the Shadows & Lumination dataset. Both tables show that the proposed approach outperforms existing methods in all metrics. It can also be seen that the addition of the extreme filters reduces the accuracy of the models by around 0.1°.

Figure 6.3 shows us how accurately the model can predict illumination with each filter, it shows the mean angular error in the full dataset and the reduced dataset. It can be seen that in all situations the model produces better results when the extreme filters are removed. They are also some of the most difficult to estimate, with the exception of the Dark_Green filter. It can also be seen that Nighttime filters are easier to estimate than Daytime filters.

Tables 6.3, 6.4, 6.5, 6.6, 6.7, and 6.8 showcase how the illumination noise preprocessing affect the method accuracy. For 6.3, 6.6, and 6.8 it can be seen that there is no significant improvement when the illumination noise is added. This is the expected result since the train and

Method	Mean	Med.	Trimean	Best 25%	Worst 25%
White-Patch [2]	9.91	7.44	8.78	1.44	21.27
Gray-world [46]	4.59	3.46	3.84	1.16	9.85
Shades-of-gray [49]	3.67	2.94	3.03	0.89	7.75
1st-order Gray-Edge [50]	3.35	2.58	2.76	0.79	7.18
2nd-order Gray-Edge [50]	3.36	2.70	2.80	0.89	7.14
Bayesian [48]	3.50	2.36	2.57	0.78	8.02
General gray-world [51]	3.20	2.56	2.68	0.85	6.68
CCC [53]	2.38	1.48	1.69	0.45	5.85
FC4(Squeezenet) [21]	2.23	1.57	1.72	0.47	5.15
FFCC(model J) [45]	1.99	1.31	1.43	**0.35**	4.75
MDLCC [37]	**1.78**	**1.29**	**1.40**	0.42	**3.97**
Proposed model (no noise)	2.16	1.57	1.71	0.54	4.76
Proposed model (with noise)	2.00	1.48	1.59	0.50	4.39

Table 6.4: Comparison of results obtained on the NUS-8 dataset. The best angular error for each metric is bolded.

test gamuts overlap. Tables 6.4 and 6.5 how the usage of the illumination noise pre-processing augmentation can cause a significant improvement in method accuracy when the method is used on an image from a camera that was not used during training. Table 6.7 shows an interesting result. Here, the addition of noise results in a lower accuracy model. This can be explained by looking at Figure 5.19, where gaps in the gamut between the normal filters and the extreme filters can be seen. When noise is added, the model trains on these gap areas to predict illumination that is outside the scope of the dataset.

Finally, Table 6.9 shows how many parameters each of the different CNN methods has. It can be seen that the proposed method has the smallest number of parameters out of all methods. The closest two methods are the Bianco et al.[55] methods with 154k and BoFC with 43k parameters. The Bianco et al. method has fewer convolutional layers than the proposed method, but it performs patch-based illumination estimation. This means that to estimate the illumination of a single image, all the image patches need to be fed into the model.

The closest method in terms of the number of parameters is BoFC, with around double the number of parameters. This method has a more complex setup. In addition to convolutional and max-pooling layers, they use bag-of-features layers [61], attention layers [62], and fully-connected layers. In comparison, the proposed method only uses convolutional, max-pooling, and dropout layers to achieve better results.

Method	Mean	Med.	Trimean	Best 25%	Worst 25%
Gray-world [46]	4.7	3.7	4.0	0.9	10.0
White-Patch [2]	7.0	5.4	6.2	1.1	14.6
Gray-edge [50]	5.3	4.1	4.5	1.0	11.7
2nd order Gray-edge [50]	5.1	3.8	4.2	1.0	11.3
Shades of gray [49]	4.0	2.9	3.2	0.7	9.0
Cheng et al. [28]	4.6	3.4	3.7	0.7	10.3
Weighted grey-edge [54]	6.0	4.2	4.8	0.9	14.2
Bianco et al.[55]	3.4	2.5	2.7	0.8	7.2
C3AE [56]	3.4	2.7	2.8	0.9	7.0
BoCF [57]	2.9	2.4	2.5	0.9	6.1
FC4(VGG16) [21]	2.6	2.0	2.2	0.7	5.5
Proposed model (no noise)	3.3	3.2	3.1	1.1	5.9
Proposed model (with noise)	**2.1**	**1.6**	**1.7**	**0.5**	**4.7**

Table 6.5: Comparison of results obtained on the Intel-TAU dataset camera invariance protocol. The best angular error for each metric is bolded.

6.2 Known number of illuminants estimation

This chapter presents an overview of the method created for multi-illuminant estimation when the number of illuminants in an image is known. For this problem, another convolutional neural network was developed. The neural network is based on the FC4 [37] model. Such a model seemed like the next logical step in the process of creating a general illumination estimation method.

6.2.1 Motivation

The creation of this model started during the development of the Shadows & Lumination dataset. The images in the Shadows & Lumination dataset have the unique feature of having multiple illuminants but having two regions with only a single illuminant. If we use the segmentation mask, we can create two single illuminant images. This is done by setting the pixels under one of the illuminants to 0. There are no non-uniform regions and after segmentation, the two images can be processed by any method that performs single illuminant estimation. The idea was the creation of a method that can estimate the two illuminants without segmenting the image. This is achieved by having a separate output for each of the illuminants present in the image.

Method	Mean	Med.	Trimean	Best 25%	Worst 25%
Gray-world [46]	4.9	3.9	4.1	1.0	10.5
White-Patch [2]	9.4	9.1	9.2	1.4	17.6
Gray-edge [50]	5.9	4.0	4.6	1.0	13.8
2nd order Gray-edge [50]	6.0	3.9	4.8	1.0	14.0
Shades of gray [49]	5.2	3.8	4.3	0.9	11.9
Cheng et al. [28]	4.5	3.2	3.5	0.7	10.6
Weighted grey-edge [54]	6.1	3.7	4.6	0.8	15.1
Yang et al. [48]	3.2	2.2	2.4	0.6	7.6
Color tiger [29]	4.2	2.6	3.2	1.0	9.9
Greyness index [58]	3.9	2.3	2.7	0.5	9.8
PCC_Q2 [59]	3.9	2.4	2.8	0.6	9.6
Bianco et al.[55]	3.5	2.6	2.8	0.9	7.4
C3AE [56]	3.4	2.7	2.8	0.9	7.0
BoCF [57]	2.4	1.9	2.0	0.7	5.1
FFCC [45]	2.4	1.6	1.8	**0.4**	5.6
FC4(VGG16) [21]	2.2	1.7	1.8	0.6	4.7
Proposed model (no noise)	1.92	1.43	1.53	0.45	4.26
Proposed model (with noise)	**1.91**	**1.40**	**1.50**	0.45	**4.25**

Table 6.6: Comparison of results obtained on the Intel-TAU dataset Cross-validation protocol. The best angular error for each metric is bolded. The last two rows show two decimal points so that the effect of noise augmentation can be properly measured.

6.2.2 Model architecture

For our method, the FC4 [21] was used as the baseline. FC4 was chosen because it produces state-of-the-art illumination estimation results and because they introduce an attention mechanism into the neural network. The authors proposed the attention mechanism to combat the problem of irrelevant image regions for illumination estimation.

An example of such a situation is an image of an orange wall. In such a situation, we cannot know whether the wall is white and illuminated by an orange light, the wall is orange and is illuminated by a white light, or any other combination of wall and illumination color. When there are a few surfaces in an image, it is harder to extract the scene illumination.

The authors of FC4 use the attention mechanism to ignore such regions. The attention mechanism can however be expanded to ignore more than just uniformly colored regions. The

Method	Mean	Med.	Trimean	Best 25%	Worst 25%
Gray-world [46]	2.31	1.83	1.95	0.50	4.96
White-Patch [2]	2.60	1.71	1.87	0.52	6.26
Gray-edge [50]	2.19	1.55	1.67	0.48	5.01
1st order Gray-edge [50]	2.78	2.25	2.38	0.73	5.70
2nd order Gray-edge [50]	2.86	2.39	2.50	0.78	5.74
Bianco et al. [55]	1.73	1.46	1.51	0.56	3.41
HypNet-SelNet [60]	2.65	1.83	2.02	0.37	6.32
Proposed model (no noise)	**1.25**	**0.98**	**1.04**	**0.36**	**2.61**
Proposed model (with noise)	1.33	1.04	1.09	0.39	2.78

Table 6.7: Comparison of results obtained on the Full Filters & Lumination dataset. The best angular error for each metric is bolded.

Method	Mean	Med.	Trimean	Best 25%	Worst 25%
Gray-world [46]	2.45	2.01	2.13	0.60	5.04
White-Patch [2]	2.48	1.68	1.83	0.54	5.84
Gray-edge [50]	2.13	1.56	1.66	0.53	4.73
1st order Gray-edge [50]	2.74	2.29	2.41	0.85	5.38
2nd order Gray-edge [50]	2.89	2.48	2.58	0.92	5.53
Bianco et al.[55]	1.59	1.35	1.41	0.55	3.05
HypNet-SelNet [60]	2.74	1.98	2.13	0.40	6.41
Proposed model (no noise)	1.22	0.99	1.04	**0.39**	**2.45**
Proposed model (with noise)	**1.19**	**0.98**	**1.03**	0.49	2.37

Table 6.8: Comparison of results obtained on the Reduced Filters & Lumination dataset. The best angular error for each metric is bolded.

method was adapted to also ignore the regions of the image that do not contain the desired illuminant.

In the original paper, the authors used two different feature extractor models as the backbones and created two varieties of their method. They used AlexNet [63] and SqueezeNet [64]. For the proposed method, the created model uses SqueezeNet because it is the computationally simpler neural network that achieves results comparable to the AlexNet variant.

The main contribution to the low computational complexity of the method is the fire module. The fire module is the main building block of SqueezeNet. It is composed of two layers, the

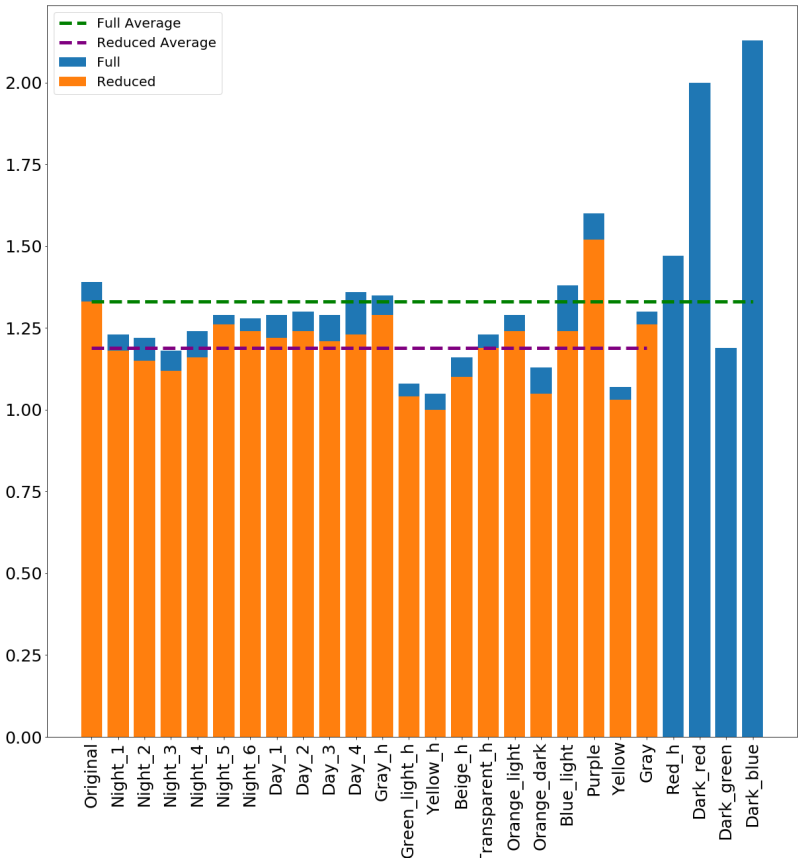


Figure 6.3: A histogram of the average angular error the methods achieves on when different filters are used. Both the Full and Reduced dataset variants are shown.

Squeeze layer, and the Expand layer. The Squeeze layer contains only 1x1 convolutional filters, while the Expand layer contains both 1x1 and 3x3 convolutional filters. This module is the result of the author's desire to create a neural network architecture with few parameters. To achieve this, they used three strategies. The first strategy is to use filters that have a kernel size of 1x1. The reason behind this is that 1x1 filters have the smallest number of parameters, having 9x fewer parameters than 3x3 filters. The second strategy is the reduction of the number of input channels that are fed into the convolutional layer with a kernel size of 3x3. This is also done to reduce the number of parameters a method has. These two strategies are applied in the Squeeze layer. The final strategy is to downsample the inputs of the later convolutional layers so that they have larger activation maps. This strategy was used because of the intuition that larger activation maps will result in better accuracy [64].

The architecture used to create the model can be seen in Figure 6.4. Every convolutional layer is followed by a ReLU activation function. All but the final Max-pooling layer have a kernel size of (3,3) and a stride of (2,2). The first layer is a normal convolutional layer with 64 filters of size (3,3) and a stride of 2. It is followed by the Max-pooling layer. Then comes the

Model	Parameters
Bianco et al.[55]	154k
FC4(Squeezenet) [21]	1.9M
FC4(Alexnet) [21]	3.8M
Koščević et al. [36] (VGG16)	14.7M
BoCF(attention2) [57]	43k
Proposed model	21k

Table 6.9: Number of parameters in different CNN models

first fire module which has 16 filters in the Squeeze layer and 128 filters in the Expand layer. In each Expand layer, half of the filters have a kernel size of (1,1) and half a kernel size of (3,3). After the first fire module comes another fire module with 16 filters in Squeeze and 128 filters in the Expand layers. The output is then fed into a Max-pooling layer. This is followed by two fire modules that both contain 32 filters in the Squeeze layer and 256 filters in the Expand layer. Then comes a Max-pooling layer. Then another two fire blocks with 48 filters in the Squeeze and 384 filters in the Expand layer. They are followed by the final fire module that has 64 filters in the Squeeze layer and 512 filters in the Expand layer.

After the final fire module, n equal branches with separate parameters are added to the model. n is the number of illuminant present in the image. Each branch starts with a Max-pooling layer with a kernel size of (2,2) and a stride of (2,2). It is followed by a convolutional layer with 64 filters of kernel size (6,6). Then we use a Dropout [38] layer with a drop rate of 0.5. That is followed by a convolutional layer with 4 filters with kernels of size (1,1). Three channels of the output represent the RGB values of the illumination in the different regions of the image. The fourth channel is the confidence mask that contains how confident the model is in each illuminant prediction. The confidence mask is then fed into the Softmax activation layer. The RGB mask is multiplied by the Softmax confidence mask. The RGB vectors are then summed and normalized to create the final illumination estimation.

6.2.3 Training setup

This method was implemented in Python [40] using TensorFlow 2.9 [41] for model implementation. For model training, the AdamW [42] optimizer with a weight decay of $5e^{-5}$ was used. The used loss function was Mean Squared Error (MSE). The model was trained for 200 epochs on the Nvidia 2080Ti GPU and AMD Ryzen 7 3700X CPU. A mini-batch of 28 was used. The learning rate was set to $3e^{-4}$.

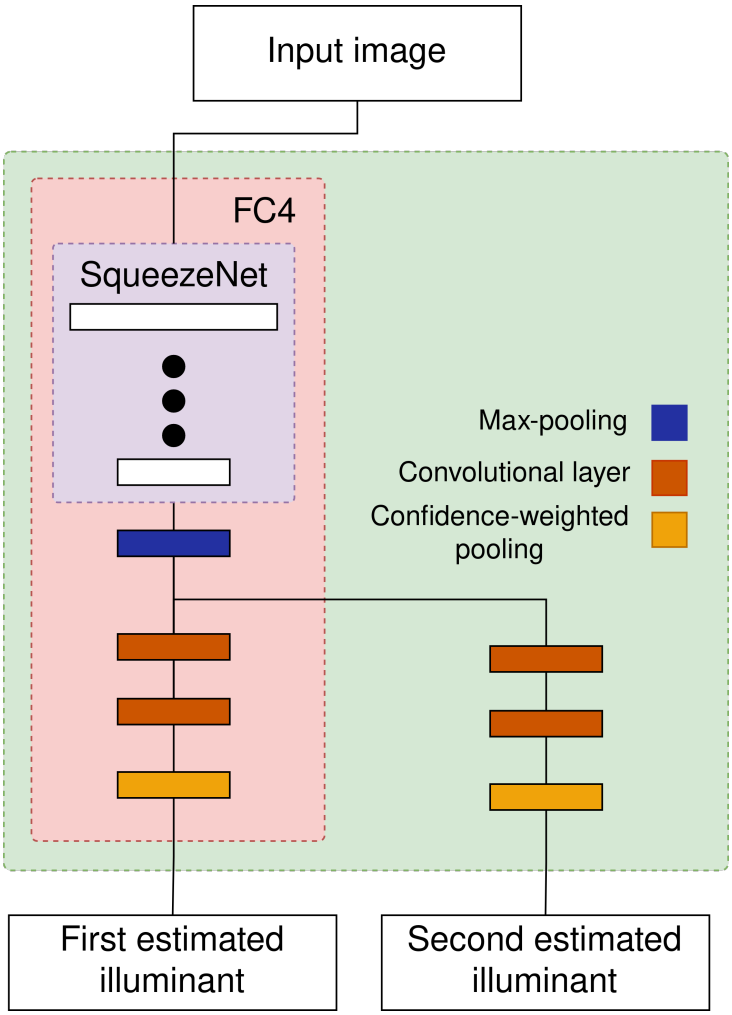


Figure 6.4: Model architecture of the known number of illuminant estimation method

6.2.4 Evaluation

To evaluate the method on images where the number of illuminants is known apriori, we used the Shadows & Lumination dataset. The dataset is perfect to evaluate this method as there are no regions of non-uniform illumination in dataset images. All the protocols introduced in Chapter 5.3.5, which includes Use-All, One-to-Many, Outdoor, Indoor, and Nighttime, were used.

To properly evaluate the method it needs to be compared to existing methods from the literature. To evaluate the method several existing multi-illuminant estimation methods were implemented. Three existing traditional non-learning as well as two learning-based methods were implemented. Two of the traditional methods [32, 65] are based on the Gray-Edge framework [50]. This is a widely used framework as its methods use very simple assumptions and are extremely hardware-friendly. An example from the framework is White-Patch [2], which uses the assumption that an object which perfectly reflects light will possess the color of the illumination. The illuminant is estimated by extracting the highest intensities of each image channel and combining them into a single RGB vector. The other traditional method[66] transforms the

image into La*b* color space and using K-Means calculates centroids that represent image illuminants. All of these methods divide the image into small non-overlapping regions also called patches. [66] divides the image onto patches based on a pixels' Euclidean distance from patch centers, [32] divides the image into square uniformly shaped patches, and [65] segments an image into patches in three different ways, by dividing the image into square uniformly shaped patches, by extracting superpixels [35] from the image or by creating patches around keypoints in the image.

The learning-based methods [55, 60] use convolutional neural networks to predict scene illuminations. They also divide an image into square, uniformly shaped patches. The main difference between the methods is that Bianco et al. method [55] uses one CNN that predicts the illumination for each patch, while [60] consists of two CNNs, one that outputs two illumination predictions for each patch and one that selects which of the predictions is more accurate.

All of these methods use the assumption that each patch contains only one uniform illuminant. After all the patch illuminants have been extracted, they are groped to calculate the two image illuminants.

An important thing to note is that for the traditional methods, Use-All and One-to-Many are equivalent. Traditional methods require no training, and both protocols devolve into testing the method on each image.

To compare the results of the models, the angular error was used. The standard evaluation metrics used in other works were used. This includes mean, median, Best 25% mean, and Worst 25% mean.

6.2.5 Results

The results on the Shadows & Lumination dataset can be seen in Table 6.10, Table 6.11, and Table 6.12.

In Table 6.10, the results of the model on the Use-All protocol are shown. It can be seen that the proposed method produces significantly better results than all other methods. It can also be seen that Direct illuminant estimation is a significantly easier problem than Ambient illuminant estimation, with the average angular error being more than 1° smaller. The Table also shows that unlike in single-illuminant estimation, existing learning-based methods do not outperform existing non-learning-based methods, in some cases having significantly better results.

The results of the One-to-Many protocol can be seen in Table 6.11. Here, the proposed model does not outperform the other methods in all metrics. For this protocol, the model achieved great results when looking at the Worst 25% angular error and the mean angular error for Ambient Illumination. The model does outperform other learning-based methods, but methods from [65] achieve the best results. This can be attributed to the fact that these methods are non-learning and different camera sensors have no effect on them. Also, the difference between

Method	Ambient				Direct				Both			
	mean	med.	best 25%	worst 25%	mean	med.	best 25%	worst 25%	mean	med.	best 25%	worst 25%
[66]	13.19	13.09	6.55	20.09	14.15	13.72	8.72	20.35	13.67	13.46	7.54	20.22
CRF(White-Patch) [32]	8.40	6.84	1.74	17.96	5.98	4.56	1.36	13.03	7.19	5.44	1.52	15.81
Patch-based (White-Patch) [65]	4.89	3.00	0.95	12.22	3.70	2.81	1.09	7.92	4.30	2.89	1.02	10.13
Keypoint-based (White-Patch) [65]	6.90	4.52	1.32	16.62	4.01	2.97	0.96	8.91	5.46	3.59	1.11	13.15
Superpixel-based(2nd Order) [65]	5.00	3.63	1.26	11.25	3.39	2.71	0.97	7.08	4.20	3.10	1.09	9.32
Bianco et al. [55]	9.17	7.36	3.43	18.04	6.85	4.58	1.45	16.91	8.01	5.65	2.15	17.98
HypNet/SelNet [60]	6.20	4.20	0.92	14.94	6.41	3.66	0.78	16.90	6.31	3.95	0.85	15.95
Created method	**2.84**	**2.13**	**0.74**	**6.21**	**1.71**	**1.22**	**0.44**	**3.86**	**2.28**	**1.60**	**0.55**	**5.22**

Table 6.10: The mean, median Best 25%, and Worst 25% angular error scores of different methods tested using the Use-All protocol. Direct represents when only the direct light source illuminant estimation accuracy is examined. Ambient represents when only the ambient light source illumination estimation accuracy is examined. Both represents how well the methods perform in general. The best results are bolded.

the Ambient and Direct mean errors is significantly smaller than in the Use-All protocol.

Finally, Table 6.12 shows us how the model performs with different types of scenes and illumination. Here, the proposed method outperforms all other methods when Indoor and Outdoor images are examined, but for Nighttime the model achieves the second-best results. Here again, a non-learning-based method outperforms all other methods. The difference between the proposed method and the best is much smaller than when looking at One-to-Many. What differentiates Nighttime images from other types is that it only contains artificial illumination and our methods struggle more with this than [65]. Even with this, Nighttime images are not the most difficult type to estimate. Indoor images are the hardest to estimate having almost a 1° higher mean angular error than Nighttime images and unlike Nighttime and Outdoor images the difference between Ambient and Direct angular error is relatively small. This makes sense as Indoor images contain both artificial and natural illumination that often mixes making it harder to separate the Direct illumination and Ambient illumination colors in the Indoor set.

These results show that a convolutional neural network that uses the entire image to perform multi-illuminant estimation can achieve state-of-the-art results with the main obstacle being different camera sensors. In addition, the fact that the number of illuminants needs to be known beforehand and the fact that the model does not explicitly state which light illuminates which region are also downsides of the method. These concerns are discussed in the next chapter.

Method	Ambient				Direct				Both			
	mean	med.	best 25%	worst 25%	mean	med.	best 25%	worst 25%	mean	med.	best 25%	worst 25%
[66]	13.19	13.09	6.55	20.09	14.15	13.72	8.72	20.35	13.67	13.46	7.54	20.22
CRF(White-Patch) [32]	8.40	6.84	1.74	17.96	5.98	4.56	1.36	13.03	7.19	5.44	1.52	15.81
Patch-based (White-Patch) [65]	4.89	**3.00**	**0.95**	12.22	**3.70**	2.81	1.09	7.92	4.30	**2.89**	**1.02**	10.13
Keypoint-based (White-Patch) [65]	6.90	4.52	1.32	16.62	4.01	2.97	**0.96**	8.91	5.46	3.59	1.11	13.15
Superpixel-based(2nd Order) [65]	5.00	3.63	1.26	11.25	3.39	**2.71**	0.97	**7.08**	**4.20**	3.10	1.09	9.32
Bianco et al. [55]	10.09	8.21	3.17	20.40	9.48	7.39	3.26	20.02	9.78	7.67	3.21	20.36
HypNet/SelNet [60]	8.35	6.00	1.31	19.41	7.67	4.98	1.05	19.12	8.01	5.45	1.17	19.31
Created method	**4.83**	4.18	1.83	**9.05**	4.58	4.20	1.89	7.85	4.71	4.19	1.86	**8.45**

Table 6.11: The mean angular error score of different methods tested using the One-to-Many protocol. Direct represents when only the direct light source illuminant estimation accuracy is examined. Ambient represents when only the ambient light source illumination estimation accuracy is examined. Both represents how well the methods perform in general. The best results are bolded.

Method	Outdoor			Indoor			Nighttime		
	Amb.	Direct	Both	Amb.	Direct	Both	Amb.	Direct	Both
[66]	13.44	12.94	13.19	15.42	14.26	14.84	16.91	15.79	16.35
CRF(White-Patch) [32]	9.00	5.62	7.31	8.16	9.40	8.78	7.10	6.81	6.96
Patch-based (2nd Order) [65]	5.21	3.84	4.53	5.76	5.40	5.58	**4.39**	**2.57**	**3.48**
Keypoint-based (White-Patch) [65]	7.39	4.15	5.77	6.99	5.50	6.25	5.06	2.93	3.99
Superpixel-based(White-Patch) [65]	5.88	3.59	4.73	5.78	4.57	5.18	5.98	3.29	4.63
Bianco et al. [55]	3.74	5.98	4.86	7.98	6.85	7.42	7.61	8.82	8.22
HypNet/SelNet [60]	5.99	6.26	6.09	6.49	7.65	7.07	5.28	4.24	4.76
Created method	**2.26**	**1.30**	**1.78**	**4.72**	**4.18**	**4.45**	4.42	2.93	3.68

Table 6.12: The mean, median Best 25%, and Worst 25% angular error scores of different methods tested on the three variants of the By-Type protocol. Direct represents when only the direct light source illuminant estimation accuracy is examined. Amb. represents when only the ambient light source illumination estimation accuracy is examined. Both represent how well the methods perform in general. The best results are bolded.

6.2.6 Discussion

The presented results show us that the model can properly estimate scene illumination when multiple illuminants are present, but this approach has two major disadvantages.

For the method to properly work, the number of illuminants in an image needs to be known beforehand. One could overcome this constraint by creating a neural network ensemble. The number of illuminants an image needs to have is explicitly stated by the number of outputs the model has, and to cover all situations N models could be trained to cover the 1 to N illuminants situations. In addition to the estimation models a classification model that estimates the number of illuminants would also need to be created. Such a method would be significantly more computationally complex than the method presented in this chapter.

Another way to remove this constraint is by combining classification and estimation models into a single neural network. There is one situation in which chromatic adaptation is not needed and that is when the illumination is a perfectly white light, so when a model outputs a white light it means no further action is required to adapt the image. If a model has 5 outputs and the final output produces a white light, we can conclude that a fifth illuminant is not present in the image. This is how the method was modified so that instead of needing to know the number of illuminants in an image we need to know the maximum number of illuminants an image can have, where if an image does not have the maximum number of illuminants the redundant outputs will produce a perfectly white light.

To test this approach, the LSMI [33] dataset was used, as it contains images with 1 to 3 illuminants. A method with three outputs was trained and tested on the three camera subsets of the dataset.

Method/Subset	Num. of images	Mean	1st ill. mean	2nd ill. mean	3rd ill. mean	Num. of two ill. images	2nd ill. F1	Num. of three ill. images	3rd ill. F1
FC42/Sony	475	3.83	2.18	6.97	13.59	190	0.75	19	0.93
FC42/Nikon	308	4.32	2.3	9.29	18.32	111	0.23	5	0.94
FC42/Galaxy	388	4.51	2.36	8.85	18.30	151	0.47	13	0.93
FC42/All	1171	4.18	2.27	8.17	15.88	452	0.54	37	0.94

Table 6.13: The estimation and classification results obtained using the proposed method. Num. means Number and ill. means illuminant

In Table 6.13 the estimation and classification results can be seen. There is no classification metric for the first illuminant, as it is always present in the image. It can be seen that estimation accuracy for the first illuminant is fairly high, close to the 2° threshold of the human eye. This cannot be said for the second and third illuminants, which produce erroneous results, with the best performance obtained on the Sony images. Looking at the F1 score for the second

illuminant it can be seen that the model cannot accurately predict whether the second illuminant is present or not. For the third illuminant, the F1 is very high, but sadly this can be explained by the small number of images with three illuminants, as the model assumes there are no images with three illuminants.

The other problem with the approach of this chapter is the fact that the methods provide no information on what influence the illuminant has on regions of an image. The authors of LMSI [33] used their dataset to evaluate methods that provide spatial illumination information. In Table 6.14 we compare our approach to these methods on the Galaxy subset. For the single illuminant situation, our model outperforms methods from the literature, but it achieves terrible results in multi-illuminant situations, being three to four times worse than the best-performing methods.

Method	Single		Multi		Mixed	
	Mean	Median	Mean	Median	Mean	Median
Pix2Pix [67]	6.53	2.17	4.28	2.63	5.66	2.44
Gijsenij et al.[68]	7.49	6.04	12.38	9.57	10.09	7.43
Bianco et al.[55]	4.15	3.30	5.56	4.33	4.89	3.83
HDRNet [69]	2.85	2.20	3.13	2.70	3.06	2.54
U-Net [70]	2.95	1.86	2.35	2.00	2.63	1.91
Proposed method	2.39	1.83	9.81	7.11	4.51	2.13

Table 6.14: Comparison of the mean and median angular errors of the proposed method with other methods from literature

Unlike Shadows & Lumination, LSMI [33] contain non-uniform illumination and it can occur that the illuminant is not actually present in the image. Instead, the illuminants mix creates illumination that is significantly different from all the illuminants that contributed to the illumination mix. This in conjunction with the fact that the method does not provide any spatial information about the illuminant are the reasons why further research of these method stops here. The results presented here are to showcase what results would be achieved if one pushed illumination estimation with no spatial information to its limit.

6.3 Patch-based multi-illumination estimation

In this chapter, an overview of the first method developed for multi-illuminant estimation is presented. The model is a patch-based illumination estimation model. An image is divided into numerous small patches and the illumination is estimated for each patch. Unlike the method

from Chapter 6.2, this method has no number of illuminants constraints. Also, unlike the previous method it provides spatial information, which region the estimated illumination affects. The method can perform illumination estimation on images with a variable number of illuminants with regions of non-uniform illumination. This method uses the model from Chapter 6.1 as its baseline.

6.3.1 Motivation

During the creation of the multi-illuminant estimation model from Chapter 6.2, several neural networks that can perform multi-illuminant estimation were found in the literature. All of these methods performed illumination estimation on a patch-by-patch basis. The models themselves actually do not perform multi-illuminant estimation. They use the assumption that since the patch is small, it contains only one illuminant. The illumination estimation is single illuminant, but when the patch illuminations are combined, they create a multi-illuminant illumination mask. This approach looked like a good next step as it implicitly includes what region of the image the estimated illumination affects.

Ironically, the problem with the patch-based approach is the fact that the patches are so small. It can happen that a patch does not contain enough information for illumination estimation. A simple example is a patch that contains only one color. Using only the information in the patch we cannot properly estimate the illumination and such patches only introduce errors in the estimation. This is visible in single illuminant datasets, where methods that use the entire image achieve better results [21, 36, 37]. To solve this problem, the method presented in this chapter uses features extracted from the entire image to perform illumination estimation for a patch. This method also uses the assumption that a patch is illuminated by a single uniform illuminant.

6.3.2 Model architecture

The proposed method is a two-part convolutional neural network. The first part is an image feature extractor and the second part is the patch Illuminant Estimator. The output of the feature extractor is fed into the Illuminant Estimator alongside the patch. Both the feature extractor and the illumination estimator are modified versions of the illumination estimation model introduced in Chapter 6.1. The architecture of the model can be seen in Figure 6.5. Like the single illuminant estimation model, all convolutional layers in this model have a kernel size (1,1).

There are several modifications that were done to the model architecture for the feature extractor. The first is the image size which is 256×256 instead of 384×384. The second change is the removal of the final convolutional layer. The final change is the kernel size of the max-pooling layers. It was changed from (8,8) to (4,4). These modifications were the result of

an ablation study presented in Chapter 6.3.5.

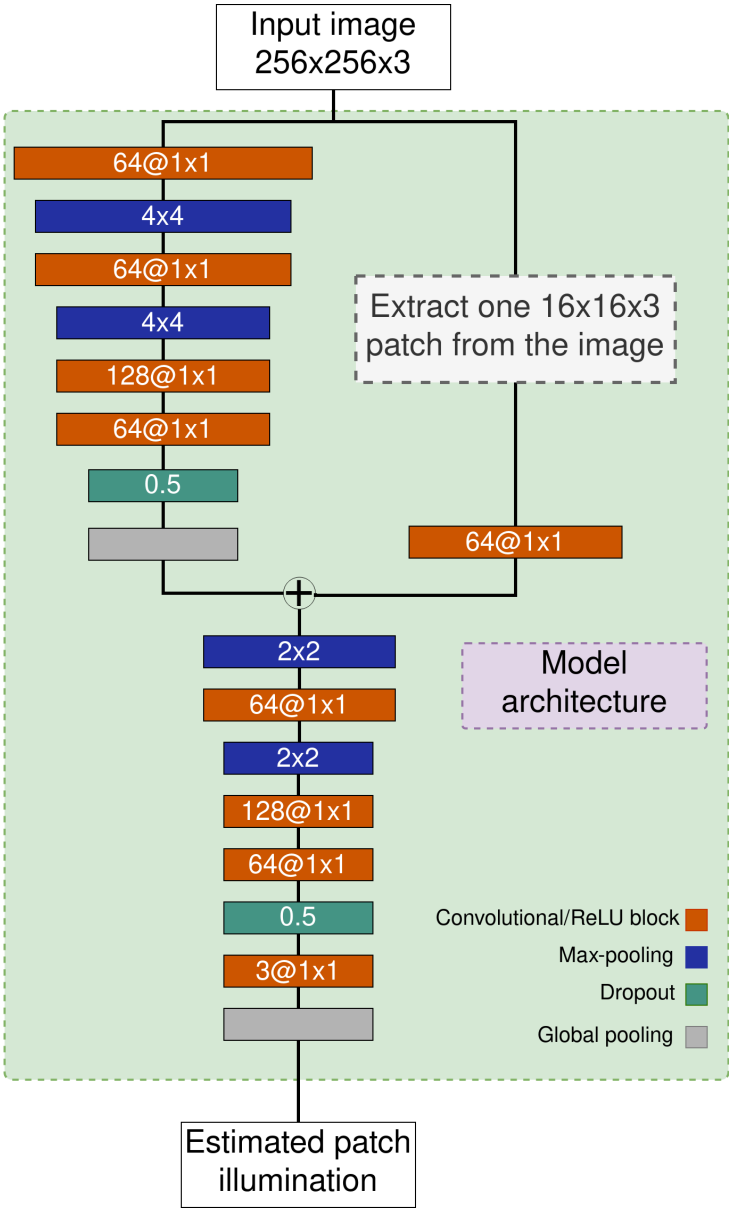


Figure 6.5: Model architecture of the patch-based illuminant estimation method

The architecture of the Illuminant Estimator is very similar to the architecture of the original model. The input shape of the patch is 16×16. The first layer of the Illuminant Estimator can be seen as the patch feature extractor. The output of this layer is combined with the output of the image feature extractor. The combination is performed by adding the two outputs. The rest of the layers in the Illuminant Estimator are identical to the original model, except for the max-pooling layers whose kernel size was reduced from (8,8) to (2,2).

The same philosophy used for the creation of the single illuminant method was used to create this method. This means the created model should be as simple as possible, or in other words have low computational complexity. That is the reason for the usage of the presented single illuminant estimation model as the image feature extractor. Any existing image feature

extractor can be used. Models such as VGG16 [71], EfficientNet [72], or MobileNet [73] could have been used. They were not used as they are significantly more complex than the presented single illuminant estimation model and the methods achieve state-of-the-art results with the simple model as the extractor.

6.3.3 Training setup

To create this method, Python [40] and TensorFlow 2.9 [41] were used. To train the method, the same loss function [43] used to train the single illuminant method from Chapter 6.1 was used. The method was trained for 400 epochs. During training, a learning rate scheduler was used. A cyclical learning rate [44] with a minimum learning rate of $1e^{-7}$, a maximum learning rate of $1e^{-3}$, and a half cycle of 200 epochs was used. For the optimizer, the AdamW [42] optimizer with a weight decay of $5e^{-5}$ was used. To train the model, the Nvidia RTX A6000 GPU and AMD Ryzen 3960X CPU hardware was used.

6.3.4 Data preprocessing

Before being fed to the model, an image is resized to 256×256. The image is also divided into 256 patches of size 16×16. Image patches that contain only black pixels were removed. This occurs in image regions that contain the calibration objects. They were masked out by setting calibration object region pixels to 0. The illumination of a patch is calculated by taking the average value of the patch illumination mask. Random rotation and flipping were applied separately to the entire image and each patch during training. Before an image and a patch are fed into the model, image standardization is applied to both separately. After image standardization, the average value of the image becomes 0 and the standard deviation becomes 1.

6.3.5 Ablation study

To confirm that the applied modifications on the model from Chapter 6.1 produce the best results, an ablation study was performed. To see how the model performs when some layers are removed. More specifically, the final three layers of the feature extractor are used to transform the model output into an RGB vector that represents the predicted scene illumination. The final three layers are a Dropout layer (Drop), a Convolutional layer (Conv), and a Global-Pooling layer (Pool), in that order. All possible combinations of the final three layers were tested and are presented in Table 6.15.

In the variants that contain the Convolutional layer, the output of the feature extractor is added to the patch before the first Convolutional layer for the estimator. This was done because it is the only location where the number of channels of the feature extractor output and the

Illuminant Estimator are the same. Except for the output of the estimation, where it makes no sense to add the global image context to the illumination prediction. Additionally, an ablation study that examines how the change of Max-pooling kernel size affects model accuracy was performed. This was done with the variants that have the Global-pooling layer. In the other variants, only the (4,4) kernel was used because the spatial dimensions of the feature extractor output and the patch would be different.

Model variant				Mean	Std.	Median	Trimean	Best 25%	Worst 25%	95 Percentile
Drop	Conv	Pool	Max*							
-	-	-	(4,4)	1.79	1.91	1.36	1.45	0.39	4.29	5.58
✓	-	-	(4,4)	1.99	1.98	1.43	1.53	0.46	4.50	5.69
-	✓	-	(4,4)	2.74	4.59	1.46	1.64	0.48	7.39	9.10
-	-	✓	(8,8)	2.32	4.34	1.17	1.36	0.37	6.34	7.26
-	-	✓	(4,4)	1.61	1.92	0.95	1.10	0.29	4.03	5.33
✓	✓	-	(4,4)	1.99	2.04	1.29	1.45	0.49	4.70	6.29
-	✓	✓	(8,8)	3.03	4.90	1.69	1.85	0.57	7.95	9.35
-	✓	✓	(4,4)	3.14	4.52	1.77	1.98	0.58	8.20	10.58
✓	✓	✓	(8,8)	2.55	3.40	1.52	1.69	0.50	6.43	8.04
✓	✓	✓	(4,4)	2.09	2.33	1.41	1.52	0.47	4.95	6.32
✓	-	✓	(8,8)	1.94	3.53	1.04	1.16	0.29	5.26	6.08
✓	-	✓	(4,4)	**1.57**	**1.84**	**0.93**	**1.06**	**0.27**	**3.96**	**5.00**

Table 6.15: Comparison of results obtained using the different model variants. *Max refers to the Max-pooling kernel size in the feature extractor.

Table 6.15 shows that the best results are achieved when the Dropout and Global-pooling layers are used. It can also be seen that the usage of the smaller kernel size improves model accuracy in all variants.

To confirm the idea of using features extracted from the entire image for the illumination estimation of a small image patch another ablation study was done. The study was performed on the LSMI dataset [33]. Two variants of the method were tested, one that uses the feature extractor and one that has no image feature extractor. In other words, one method only uses information from the patch for patch illumination estimation, and one model uses information from the patch and from the entire image for patch illumination estimation.

The results of the experiment can be seen in Table 6.16. It shows us that the inclusion of information from the entire image has a drastic effect on the accuracy of the method. This is most evident in the Sony camera images. When using both patch and image information, the

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Patch (Galaxy)	3.68	2.71	2.16	2.41	0.70	9.37	11.07
Image-Patch (Galaxy)	1.57	1.84	0.93	1.06	0.27	3.96	5.00
Patch (Nikon)	3.59	3.34	2.03	2.30	0.60	9.34	10.94
Image-Patch (Nikon)	1.53	2.35	0.85	0.97	0.24	4.02	4.76
Patch (Sony)	4.17	5.98	2.42	2.64	0.64	10.91	14.28
Image-Patch (Sony)	1.76	2.83	0.93	1.10	0.25	4.75	5.69

Table 6.16: Comparison of results obtained only using the information from the patch and results obtained by using information from the patch and the entire image. The presented measures were calculated using the angular error of each patch in the dataset, excluding patches that are completely black.

mean angular distance for the Sony images is less than half of the mean angular distance when only patch information is used.

The significance of the image features is most prominent in the Worst 25% and 95 Percentile. The mean angular distance for the Mean, Median, and trimean angular distance falls under 2°or the threshold under which the Human Visual System cannot distinguish color [34].

During the development of the method, an interesting method property came to light. The memory requirement of this method is larger than other patch-based methods because the method uses both the image and the patch. This means the method uses a smaller batch size. Since each image is divided into 256 patches, the actual number of samples used for training is significantly larger than the actual number of images in the dataset. To get preliminary results faster, only a randomly selected 25% subset of image patches for each epoch was used. In a surprising twist, such a training setup creates a method that is more accurate than a method that is trained on all patches of an image.

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Using all patches	2.03	2.15	1.36	1.36	0.48	4.74	5.93
Using 25% of patches	1.57	1.84	0.93	1.06	0.27	3.96	5.00

Table 6.17: Comparison of results obtained using all data and only a 25% subset on the galaxy subset.

This result can be seen in Table 6.17. To explain the phenomenon of fewer data is better, the method training pipeline needs to be examined. During training, an image is divided into 256 patches and the entire image is fed into the model for each patch. This results in the image feature extractor overfitting on the training images. The removal of part of the patches removes the overfitting problem and also significantly reduces the model training time.

6.3.6 Evaluation

To evaluate the method three datasets were used, the LSMI [33], the Shadows & Lumination, and the multi-illuminant Filters & Lumination datasets. For LSMI the evaluation procedure used by the original authors was used. Experiments are performed on each camera separately. Three instances of the model were created for the LSMI [33] dataset, one for each camera. The train/validation/test split for each camera, provided by the authors was used. The original authors used a 70/20/10 train/validation/test. The test set is also divided into three subsets, single, multi, and mixed number of illuminants.

For the Shadows & Lumination dataset, the evaluation procedure explained in Chapter 5.4.6 was used. Five experiments using three different evaluation protocols were performed. The experiments use all images, only use outdoor images, only use indoor images, only use nighttime images, or train on images from one camera and test on the images from other cameras. For each experiment, a 5-fold split was used, meaning five instances of the model were created and each was trained on 4 folds and tested on 1 fold. The One-to-many protocol is inverted, where 1 fold is used for training and 4 folds were used for testing.

For the Filters & Lumination, two variants of the datasets were evaluated, the full variant created using all filters and the reduced variant created after removing the extreme filters. For each variant, a random 3-fold split was used. No 2 folds contain images from the same scene

To compare different methods, the angular distance equation 5.3.2 was used. The angular distance between the ground-truth illumination vector and the predicted illumination vector was calculated. This is a commonly used metric for method comparison. The mean, median, Best 25% mean, Worst 25% mean, and 95 percentile angular errors are presented.

6.3.7 Results

In this chapter, the results for this method on the three datasets, LSMI, Shadows & Lumination, and Filters & Lumination are presented. Our results are also compared with the results obtained by current state-of-the-art methods from the literature. The results start with the LSMI [33] dataset, after which come the Shadows & Lumination results, and finally, the Filters & Lumination results are presented.

Following the LSMI [33] authors, each camera is evaluated separately using the provided train/validation/test split. The authors evaluated two types of methods on their dataset, patch-based and image-based. Gijsenij et al.[68] and Bianco et al.[55] are patch-based methods. HDRNet [69], U-Net [70] and Pix2Pix[67] are image-based methods.

Since images with non-uniform illumination can be seen as an image-to-image transition problem, where a prediction is needed for each pixel, we also tested several GAN models. Research with GANs [19] has proved their robustness and ability to adapt to many different

problems. These models are computationally complex, having several million parameters, but they have shown potential in color constancy [74]. CycleGAN [75], Pix2Pix[67], Pix2PixHD [76], and CUT [77] were tested. These methods were adapted to input an uncorrected image and output the illumination mask that will correct the image. This was supposed to be a starting point in the development of a new GAN. Surprisingly, these models performed significantly worse than was expected, with most of the models performing worse than the worst methods the authors of LSMI tested. The exception is the Pix2Pix [76], which is the only GAN whose results are present in the result tables. Because of the results and GAN computational complexity, GAN development was abandoned and the focus was shifted to the more traditional Convolutional Neural Networks.

Method	Single		Multi		Mixed	
	Mean	Median	Mean	Median	Mean	Median
Pix2Pix [67]	6.53	2.17	4.28	2.63	5.66	2.44
Gijsenij et al.[68]	7.49	6.04	12.38	9.57	10.09	7.43
Bianco et al. [55]	4.15	3.30	5.56	4.33	4.89	3.83
HDRNet [69]	2.85	2.20	3.13	2.70	3.06	2.54
U-Net [70]	2.95	1.86	2.35	2.00	2.63	1.91
Proposed method	**1.19**	**0.75**	**2.16**	**1.53**	**1.57**	**0.93**

Table 6.18: Comparison of results obtained on the Galaxy phone camera. The proposed method results are bolded.

Table 6.18 shows the results achieved on the Galaxy phone camera data subset. It can be seen that the proposed method vastly outperforms other patch-based methods, with less than half the angular error. When looking at single-illuminant images, the method is significantly better than any other method, whilst the multi-illuminant results are better but comparable to the image-based methods. A statistical analysis was performed to compare the presented model to the best-performing model from the literature. Using the one-tailed Z-test, a p-value of less than 0.001 was observed, concluding that the results of the presented method are significantly different from the results of U-Net[70].

In Table 6.19 the results obtained on the Nikon camera subset are presented. Again, the presented method outperforms both the patch-based Bianco et al. method [55] and the image-based HDRNet [69], U-Net [70], and Pix2Pix [67] methods. For Nikon images, there is a smaller difference between our model and the best-performing model from literature U-Net [70], with the presented model being around 0.5° better in all metrics. Also, the result achieved by the presented method has a smaller accuracy deviation between cameras than U-Net.

Table 6.20 presents the results achieved on the Sony images data subset. The presented

Method	Single		Multi		Mixed	
	Mean	Median	Mean	Median	Mean	Median
Pix2Pix [67]	6.10	2.27	4.18	2.76	5.41	2.49
Bianco et al. [55]	3.18	2.61	4.65	4.19	3.93	3.48
HDRNet [69]	2.76	2.43	3.20	3.01	2.99	2.61
U-Net [70]	1.51	1.14	2.36	1.84	1.95	1.45
Proposed method	**1.27**	**0.67**	**1.99**	**1.43**	**1.53**	**0.85**

Table 6.19: Comparison of results obtained on the Nikon camera. The proposed method results are bolded.

Method	Single		Multi		Mixed	
	Mean	Median	Mean	Median	Mean	Median
Pix2Pix [67]	4.08	1.72	4.37	3.26	4.20	2.20
Bianco et al. [55]	3.25	2.62	4.38	3.93	3.86	3.19
HDRNet [69]	2.70	2.37	3.65	3.33	3.21	2.89
U-Net [70]	2.83	2.44	3.04	2.78	2.94	2.66
Proposed method	**1.45**	**0.60**	**2.23**	**1.65**	**1.76**	**0.93**

Table 6.20: Comparison of results obtained on the Sony camera. The proposed method results are bolded.

method outperforms all other methods in all metrics as well, with the difference being the largest out of all the subsets. In the comparison with U-Net, it can also be seen that the proposed method has a significantly smaller accuracy deviation between the test subset, with the proposed method having a mean angular error variation of around 0.3° and U-Net having a variation as high as 1°.

Dataset subset	Mean	Std.	Median	Trimean	Best 25%	75 Percentile	95 Percentile
Galaxy	3.44	3.79	1.85	2.39	0.54	4.99	10.27
Nikon	3.28	3.87	1.71	2.19	0.51	4.55	10.80
Sony	4.06	4.07	2.34	2.97	0.47	6.42	12.00

Table 6.21: Comparison of the worst patch illuminant estimation error between the dataset subsets.

Since the method performs patch-based illumination estimation, an analysis of how the method performs in the worst-case scenario was observed. In Table 6.21 the compiled angular

error measures when only the worst patch from each image is examined. It shows that the most difficult patches to estimate are present in the Sony subset, with the highest values in most measures and a mean angular error of over 4°. The results of this table can also be compared with Table 6.16 where the results obtained using only patch information are present, which has similar results. Another interesting thing in the table is the massive error jump between the 75 Percentile and 95 Percentile, which tells us there are outlier samples for which our model performs poorly. This is further explored in Chapter 6.3.8.

The next dataset is the Shadows & Lumination dataset from Chapter 5.3. In addition to the estimation results, the segmentation results are presented. The results are compared with the results of segmentation methods from the literature. To get the segmentation mask the predicted illumination mask is resized to the size of the image, after which KMeans [78] was used to calculate the two centroids that represent the two image illuminants, and finally, each pixel in the illumination mask is assigned to one of the centroids which result in a segmentation mask.

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Patch (Gray World [65]	5.08	4.36	3.76	4.10	1.21	11.14	13.74
Keypoint (Gray World) [65]	5.45	4.32	4.31	4.60	1.38	11.43	13.80
Superpixel (Gray World) [65]	5.09	4.53	3.65	4.03	1.15	11.42	14.10
Hyp-Sel [60]	6.80	7.43	4.60	5.06	0.97	16.47	20.55
Bianco et al. [55]	3.05	2.69	2.32	2.47	0.85	6.52	7.92
Proposed method	**2.46**	**2.45**	**1.73**	**1.90**	**0.53**	**5.62**	**6.80**

Table 6.22: The mean, standard deviation, median Best 25%, Worst 25%, and 95 Percentile angular error scores on the Use-All protocol. The best results are bolded.

Table 6.22 presents the results on the Use-All protocol. The methods from Chapter 6.3.7 were used to compare our results, but predictions from all patches have not been grouped to create two illuminant predictions. Instead, each patch is a sample used to calculate the angular error metrics. Another major difference is the keypoint method, where in the previous chapter the keypoint region was used as a patch, whilst here the keypoint patch contains the pixels that are closest to the particular keypoint. This change was made because certain image regions contain no keypoints and they were not considered as spatial information was ignored.

It can be seen that the learning-based methods are significantly better than non-learning-based methods and that the proposed method outperforms all other methods from the literature in all angular error measures. Another interesting thing to note is that the Bianco et al. method [55] achieves better results when patch illuminants are not grouped with local to global aggregation described by the authors.

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Patch (Gray World) [65]	5.08	4.36	3.76	4.10	1.21	11.14	13.74
Keypoint (Gray World) [65]	5.45	4.32	4.31	4.60	1.38	11.43	13.80
Superpixel (Gray World) [65]	5.09	4.53	**3.65**	**4.03**	1.15	11.42	14.10
Hyp-Sel [60]	5.62	5.85	3.93	4.29	**0.83**	13.35	16.5
Bianco et al. [55]	4.68	**3.06**	4.18	4.25	1.77	**8.48**	9.96
Proposed method	**4.65**	3.30	4.12	4.18	1.62	8.63	**9.78**

Table 6.23: The mean, standard deviation, median Best 25%, Worst 25%, and 95 Percentile angular error scores on the One-to-Many protocol. The best results are bolded.

Table 6.23 shows the results on the One-to-Many protocol. Here the results of the non-learning-based methods are the same as in Table 6.22 because without training both protocols devolve into testing on each image. Unlike the situation in chapter 6.2.5, where the non-learning-based methods achieved the best results, learning-based methods achieve the best results. Hyp-Sel [60] achieves the best median and Best 25% results, but has significantly worse results in Worst 25% and 95 Percentile than the other two learning methods. The proposed model and Bianco et al. method [55] achieve similar results, with both methods being the best in some metrics. This tells us that both have a similar ability to ignore a camera sensor when predicting illumination.

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Patch (Gray World) [65]	5.14	4.24	3.89	4.22	1.31	11.04	13.53
Keypoint (Gray World) [65]	5.69	4.28	4.60	4.88	1.57	11.59	13.88
Superpixel (Gray World) [65]	5.15	4.44	3.75	4.14	1.24	11.35	13.91
Hyp-Sel [60]	6.68	6.97	4.66	5.09	1.01	15.83	19.64
Bianco et al. [55]	2.47	**1.83**	2.01	2.12	0.73	5.00	6.03
Proposed method	**2.23**	1.86	**1.69**	**1.82**	**0.54**	**4.82**	**5.95**

Table 6.24: The mean, standard deviation, median Best 25%, Worst 25%, and 95 Percentile angular error scores on the Outdoor By-Type protocol. The best results are bolded.

Table 6.24 Table 6.25, and Table 6.26 show the results achieved By-Type protocol subsets. The proposed method achieves state-of-the-art results in all metrics for all three types of images. The method achieves the best results on the Outdoor images and the worst on the Indoor images. This tells us that natural illumination is the easiest to estimate since Outdoor images only contain such illumination. Indoor images are the hardest because they contain both nat-

ural and artificial illumination. This is also proven by the Nighttime images that only contain artificial illumination. Here, the method achieves accuracy in between the Outdoor and Indoor accuracies.

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Patch (White Patch) [65]	6.37	5.43	4.79	5.22	1.22	14.15	17.20
Keypoint (White Patch) [65]	6.13	5.46	4.62	4.94	1.12	13.78	17.16
Superpixel (White Patch) [65]	6.57	5.53	5.02	5.44	1.24	14.51	17.59
Hyp-Sel [60]	8.73	8.11	6.53	7.01	1.47	19.77	24.37
Bianco et al. [55]	4.43	3.03	3.73	3.88	1.42	8.60	10.26
Proposed method	**3.84**	**2.76**	**3.25**	**3.36**	**1.08**	**7.66**	**9.48**

Table 6.25: The mean, standard deviation, median Best 25%, Worst 25%, and 95 Percentile angular error scores on the Indoor By-Type protocol. The best results are bolded.

Table 6.26 with Nighttime results is also interesting, as all methods achieve very similar results, with both non-learning and learning methods achieving competitive results. Nighttime images have the lowest illumination intensity and in such situations, non-learning-based methods are comparable with learning-based methods. The Keypoint segmentation method achieves the most similar results to the proposed model, having the same standard deviation and 0.02° better Worst 25%.

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Patch (Gray World) [65]	3.53	3.66	2.45	2.68	0.82	8.14	9.87
Keypoint (Gray World) [65]	3.38	**3.30**	2.45	2.65	0.81	7.57	9.22
Superpixel (Gray World) [65]	3.50	3.57	2.45	2.66	0.81	8.01	9.63
Hyp-Sel [60]	4.22	6.66	**2.12**	2.51	**0.43**	11.75	14.82
Bianco et al. [55]	3.41	3.92	2.53	2.68	0.91	7.48	**9.01**
Proposed method	**3.25**	**3.30**	2.31	**2.47**	0.76	**7.40**	9.22

Table 6.26: The mean, standard deviation, median Best 25%, Worst 25%, and 95 Percentile angular error scores on the Nighttime By-Type protocol. The best results are bolded.

Table 6.27 presents the Dice mean and standard deviation results obtained by the proposed method. Several methods from the literature were implemented for comparison.

The implemented methods are simple thresholding, Max Entropy [79], and OTSU [80] as they are some of the most popular non-learning-based methods from literature. In simple thresholding, the image is transformed into HLS color space and 0.05 was selected as the threshold

Method	Use-All	One-to-Many	Outdoor	Indoor	Nighttime
Simple Thresholding	80.9±12.6	80.9±12.6	80.8±12.8	73.6±14.9	80.2±15.3
Max Entropy [79]	71.5±22.4	71.5±22.4	73.0±20.2	65.6±24.2	68.4±28.9
OTSU Thresholding [80]	80.1±10.7	80.1±10.7	80.9±10.6	74.8±09.6	77.9±11.8
UNet (VGG16) [70]	88.2±12.0	85.1±13.3	88.8±12.0	79.1±13.0	86.4±12.8
UNet (SEResNet18) [70]	89.2±12.5	**86.8±12.8**	**90.1±12.2**	80.1±11.4	87.4±12.3
LinkNet (VGG16) [81]	87.9±12.0	84.9±13.2	89.1±01.5	78.3±12.3	85.5±12.8
LinkNet (SEResNet18) [81]	88.9±12.4	85.5±12.9	89.3±12.0	78.0±11.7	86.9±12.7
FPN (VGG16) [82]	88.7±12.2	85.5±13.6	89.4±11.9	**80.3±12.3**	86.1±12.7
FPN (SEResNet18) [82]	**89.3±11.9**	86.5±12.9	90.0±12.4	79.6±12.1	**87.7±12.4**
PSPNet (VGG16) [83]	88.0±12.0	85.3±13.8	79.8±11.5	75.2±16.8	85.5±13.0
PSPNet (SEResNet18) [83]	86.8±12.5	84.1±13.3	79.1±12.1	72.3±15.9	86.8±13.0
Patch estimation method	77.6±15.5	72.5±16.3	79.2±14.6	64.9±16.1	77.2±17.7

Table 6.27: The mean and standard deviation dice scores of different methods tested using different protocols. The results with the best mean are bolded.

to create a binary mask from the L channel. In OTSU the image is transformed into grayscale color space, after which a threshold is selected so that the variance in each class is minimized. In Max Entropy, the image is transformed into grayscale color space. Here, the threshold is selected so that the entropy of each class is maximized.

For the learning-based methods several widely used segmentation models, U-Net[70], PSP-Net[83], Link-Net[81], and FPN-Net [82] were implemented. A defining feature of these models is that they are made out of two parts, an encoder, and a decoder. The encoder is used to extract useful image information, progressively downsampling the image using Convolutional and Max-pooling layers to create an important image features vector. The decoder is used to upsample the useful extracted information to the spatial dimensions of the original image using Deconvolutional layers. Each downsampling stage is combined with its corresponding upsampling stage using skip connections. U-Net combines stages by concatenating the downsampling and upsampling stages. Link-Net combines the stages by adding the downsampling and upsampling stages. FPN-Net combines the upsampling and downsampling stages by addition as well, with each Deconvolutional layer in the decoder having the same number of filters. FPN-Net performs predictions for each upsampling stage. PSPNet uses a pyramid pooling module which is placed between the encoder and decoder. This module uses convolutional layers of different kernel sizes to extract global and local image information.

These models are used in combination with two backbones, VGG16 [71] and SEResNet18 [84]. These backbones are trained on ImageNet [85] and they are used as encoders in the segmentation models.

Table 6.27 shows that the proposed method with K-means does not compare to the models that were explicitly trained to segment the image into regions based on illumination, which is not surprising as the model was not designed to segment the image based on illumination. No pure segmentation method was created, as the presented model corrects images quite well without an explicit segmentation mask. Table 6.27 was created for baseline results to see how accurately existing segmentation methods perform on the dataset.

The results on the full Filters & Lumination dataset are presented in Table 6.28 for the Full Filters variant and Table 6.29 for the Reduced Filters variant. The proposed method achieves the best results for both variants in all measures except for the Best 25% measure, with all methods except for [55] achieving better results. This tells us that the other methods do not generalize well on the data, which can be seen in the Worst 25% and 95 Percentile where the proposed models achieve around 2° better results. Unlike the other datasets, for this dataset, the non-learning methods outperform the learning methods from the literature. This shows us that our neural network can not properly generalize on a diverse set of illumination colors and the number of illuminants in a scene.

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Patch (Gray World) [65]	5.03	5.06	3.41	3.77	1.00	11.89	15.24
Keypoint (Gray World) [65]	4.84	4.82	3.31	3.66	**0.91**	11.41	14.38
Superpixel (Gray World) [65]	5.02	5.00	3.38	3.76	0.99	11.86	15.19
Hyp-Sel [60]	8.98	13.88	4.96	5.58	0.94	24.32	30.08
Bianco et al. [55]	6.02	4.55	4.92	5.13	1.98	12.00	14.66
Proposed method	**4.32**	**4.24**	**3.09**	**3.30**	1.17	**9.67**	**12.23**

Table 6.28: Comparison of results obtained on the Full Filters dataset. The best results are bolded. std. means standard deviation, Med. means median, and Tri. means trimean.

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Patch (Gray World) [65]	3.75	3.23	2.80	3.03	0.89	8.18	10.03
Keypoint (Gray World) [65]	3.67	3.14	2.74	2.98	0.82	8.09	9.93
Superpixel (Gray World) [65]	3.80	3.35	2.80	3.04	0.89	8.41	10.36
Hyp-Sel [60]	6.09	6.86	4.11	4.51	**0.78**	14.88	18.37
Bianco et al. [55]	5.05	3.11	4.55	4.63	1.81	9.17	10.61
Patch estimation method	**2.90**	**1.99**	**2.49**	**2.57**	0.96	**5.58**	**6.62**

Table 6.29: Comparison of results obtained on the Reduced Filters dataset. The best results are bolded. std. means standard deviation, Med. means median, and Tri. means trimean.

6.3.8 Qualitative results

In this chapter, a couple of visual examples of how an image looks after it has been corrected using our method are presented. An analysis of situations when the model does not give satisfactory results, how the corrected image looks in those situations, and why the method fails in such situations is also given.

To start, Figure 6.6 shows a couple of random examples from the LSMI [33] dataset. There are a couple of one-illuminant, two-illuminant, and three-illuminant images. It can be seen that the angular error differs between the patches in the same image. Some patches are easier to predict than others. Looking at the colorbar it can be seen that this does not have a significant effect on how the image looks because the maximum error in each image is less than 2°. The difference between the ground truth and model correction is not noticeable, which is consistent with research from [34]. Figure 6.6 also shows us that the method can also perform accurate illumination estimation on patches that contain only one color. The most common such situation is a patch of a single color wall.

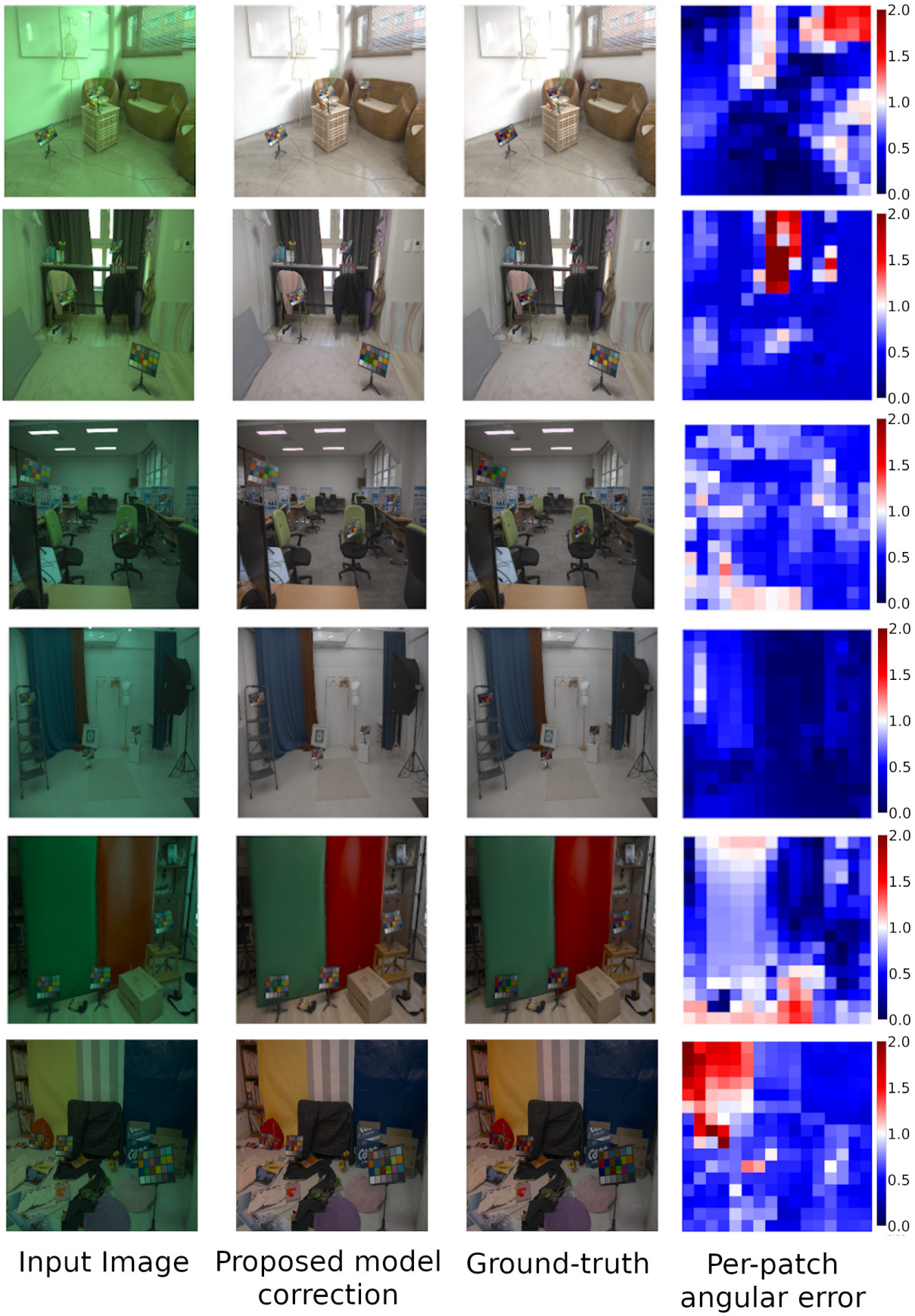


Figure 6.6: Visual comparison of model-corrected image and ground-truth corrected image on the LSMI dataset. The angular error heatmap for each image is also shown. The max angular error in the heatmaps is 2°. All images have been tone-mapped for better visualization.

Figure 6.7 shows some random examples of method performance on the Shadows & Lumination dataset. The images show that some patches are harder to predict, but when comparing the method-corrected image to the ground-truth corrected image there are no noticeable differences.

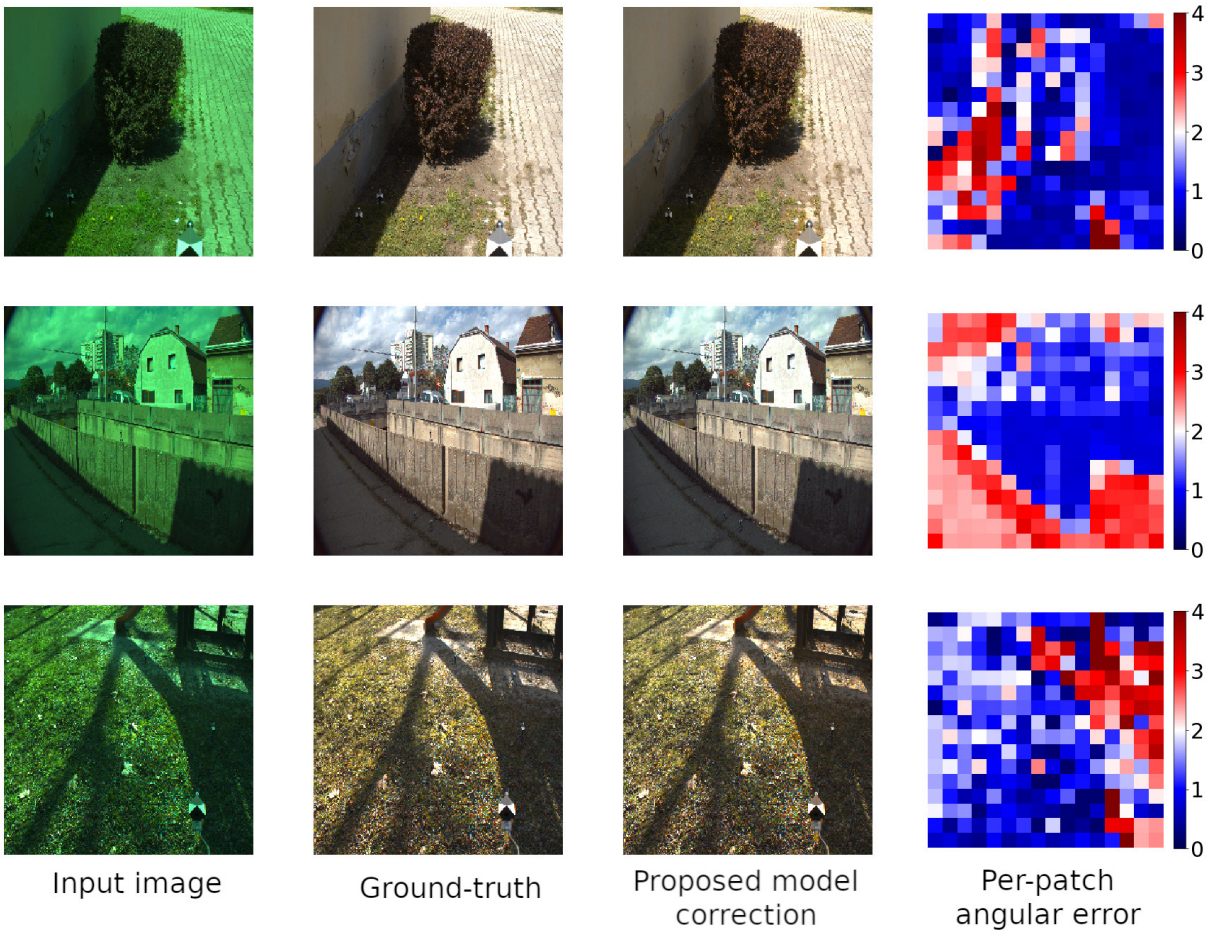


Figure 6.7: Visual comparison of model-corrected image and ground-truth corrected image on the Shadows & Lumination dataset. The angular error heatmap for each image is also shown. The max angular error in the heatmaps is 4°. All images have been tone-mapped for better visualization.

Figure 6.8 shows some examples of how the method performs on the Reduced Filters & Lumination dataset. The first image shows a situation where the method generally performs good with a few outlier regions that were incorrectly estimated, but these outliers do not have a noticeable influence on the look of the image.

The next Figure 6.9 shows four situations in which the proposed method fails to properly predict the image illumination. In the first image, it can be seen that the method has problems with estimating artificial illumination. Concretely artificial lighting with strong intensity. The region with the biggest error is the upper right region that contains the artificial illuminants. The model also has a problem with the region of the image that contains a glossy table. The artificial illumination is reflected by the table, and the model struggles to properly predict the illumination.

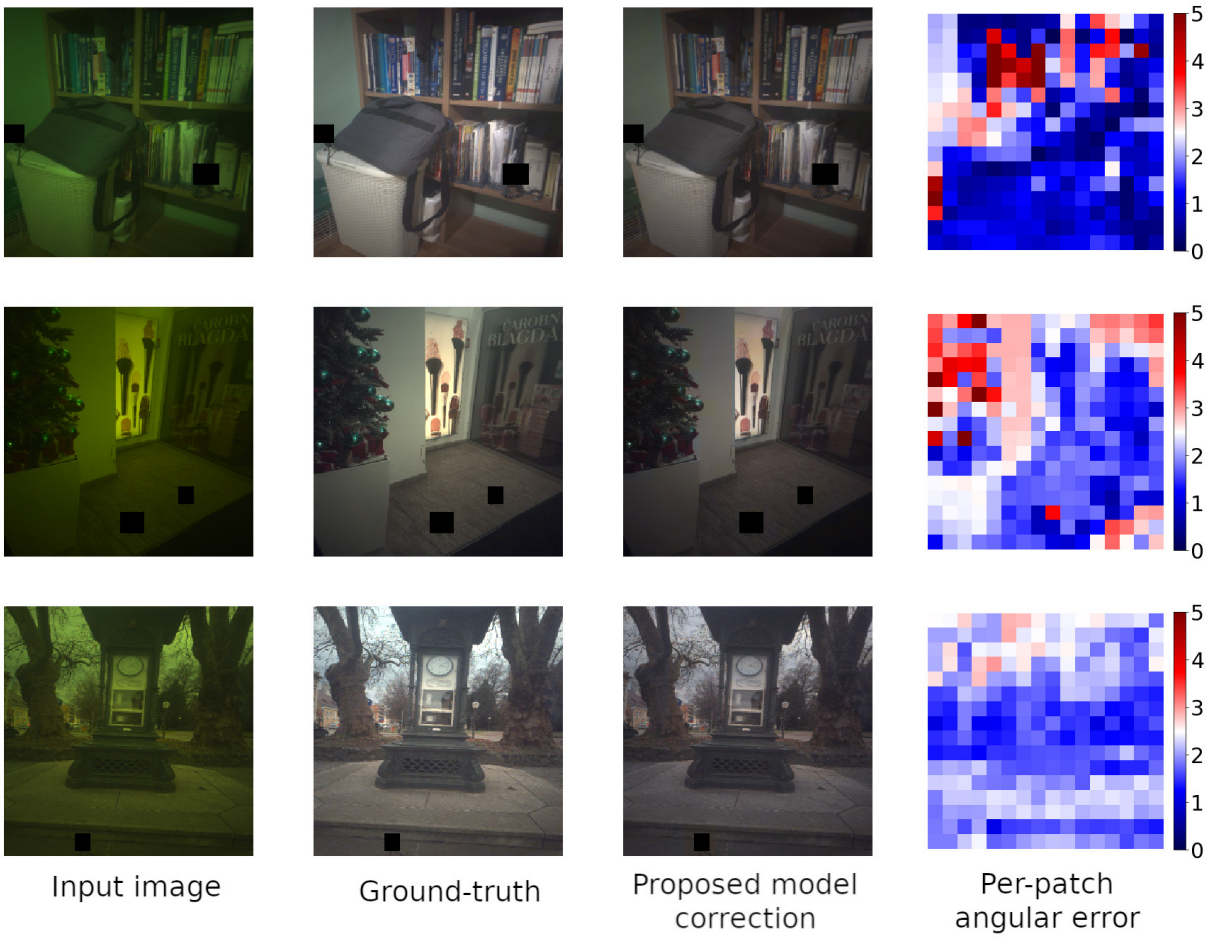


Figure 6.8: Visual comparison of model-corrected image and ground-truth corrected image on the Reduced Filters & Lumination dataset. The angular error heatmap for each image is also shown. The max angular error in the heatmaps is 5°. All images have been tone-mapped for better visualization.

The second situation where the model does not accurately estimate the illumination is in regions that contain highly saturated pixels. Here it can be seen the blue color of the sky gets lost. This is confirmed by the angular error map. Parts of the image illumination are properly estimated while some parts are not. The area with erroneous results contains highly saturated pixels created by the strong natural outdoor lighting.

The third image showcases a really problematic image for illumination estimation. In this image, no region of the image is properly predicted. The scene in the image is illuminated by a low-intensity light. The top right of the image is completely black and the entire image is full of noisy data. These factors are the causes of the model's poor performance on the image.

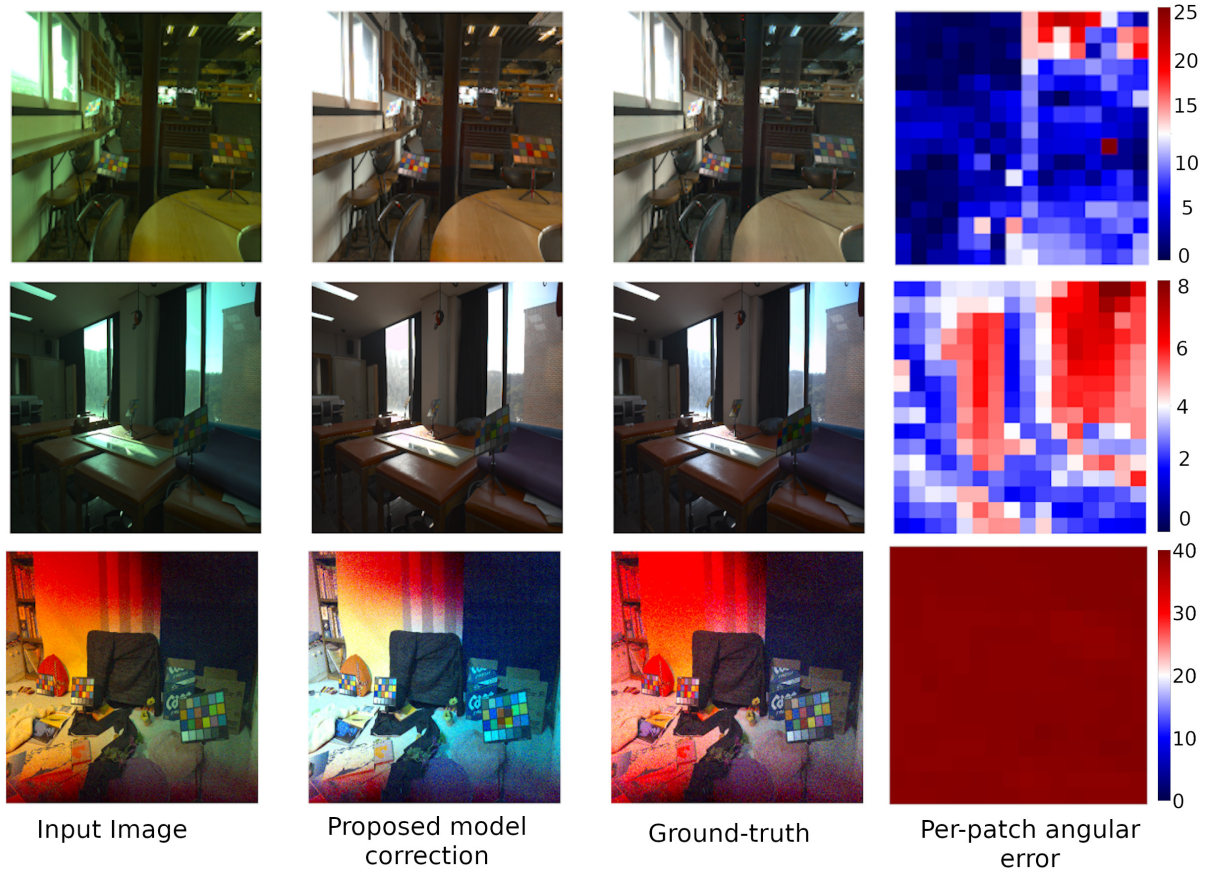


Figure 6.9: Showcase of situations where the model gives sub-par results. The angular error for each patch is also shown. The max angular error is different for each image. All images have been tone-mapped for better visualization.

The final Figure 6.10 highlights the problem the model faces with the Filters & Lumination dataset. The first image shows the problem with extreme filters, where both the ground truth and method correction result in unnatural images in the regions with extreme filters. It can also be seen that the regions without extreme filters are relatively ok. The extreme filters cause another problem which can be seen in the second image. Since the extreme filters change the image color drastically, the method has learned to assume that any highly saturated region is just the color of the illuminant, resulting in a desaturated image of a blue pipe.

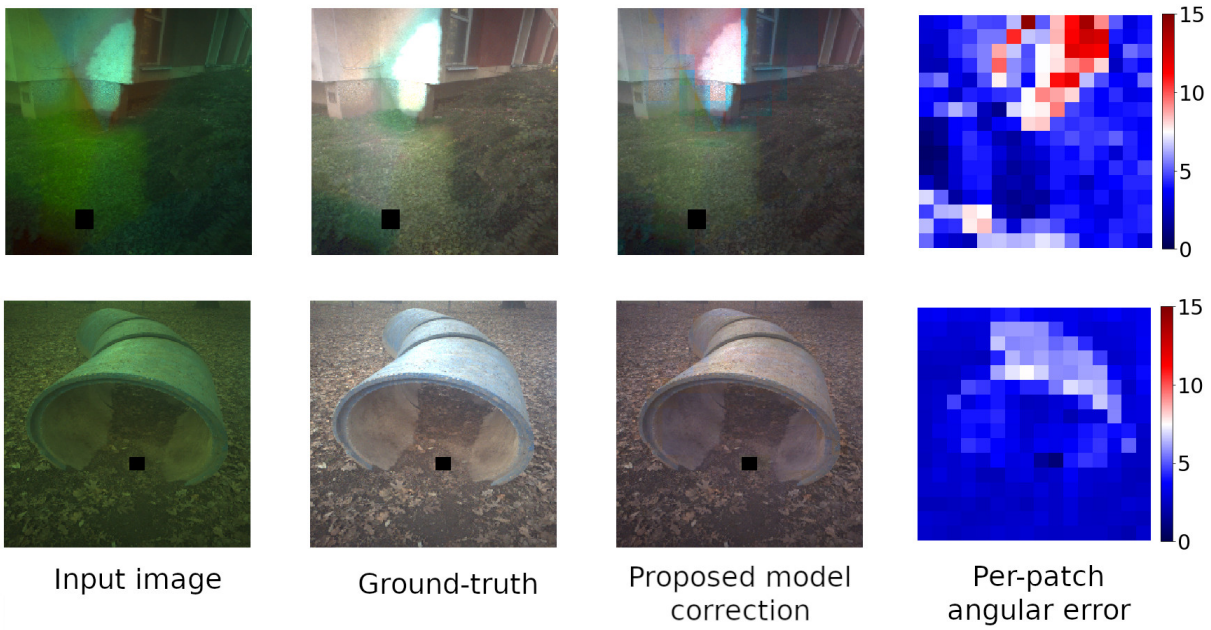


Figure 6.10: Showcase of situations where the model fails to give satisfactory results on images with extreme filters. The maximum angular error in the heatmaps is 15°. All images have been tone-mapped for better visualization.

6.4 Pixel-based multi-illumination estimation

In this chapter, an overview of experiments performed and methods created for multi-illuminant estimation is given. The main difference between these models from the model presented in Chapter 6.3 is that they perform illumination estimation on a pixel-per-pixel basis. These models use the model from Chapter 6.1 to find the limitation of the created model architecture. Several modifications were tested and their results are presented in Chapter 6.4.5. This is the final phase of this research's development of a convolutional neural network for illumination estimation.

6.4.1 Motivation

Even though the method from Chapter 6.3 can be used to perform multi-illuminant estimation, at its core the model performs single-illuminant estimation. That model also uses the assumption that since the patch is small, it contains only one illuminant. The idea here is to see if this architecture can be used without this assumption. A true multi-illuminant estimation model. This is possible if the patch-based method is adapted to perform per-pixel illumination estimation. Patch-based methods have a more complex image processing pipeline as to get an image illumination mask, each patch needs to be fed into the model. For example, if we have an image of 256 $\times$ 256 pixels with a patch size of 16 $\times$ 16 pixels, we would need to feed 256 patches into the model to get the illumination mask. The use of image patches allows us to create a less

computationally complex model as the input is smaller. Since our model from Chapter 6.3 uses the entire image in conjunction with the patch, it would be interesting to see how the model would fare if the image is not divided into patches.

6.4.2 Model architecture

There are two ways in which we can create a per-pixel illumination estimation model based on the methods from Chapter 6.1.

One way is to expand the method so that instead of estimating a single illuminant for each patch, it estimates the illumination of each pixel in the patch. With this approach, the scope of the model is limited, as it would still only need to estimate the illumination of a single image patch. It would still use the global image features from the feature extractor, but only to assist in patch illumination estimation.

The other way would be to modify the method so that the input of the model is the entire image and the output of the model is the entire image illumination mask. The main motivation is the creation of a method with a simplified process of image chromatic adaptation, where the image is fed into the model only once.

Several variations of both of these methods were tested and their results are presented in Chapter 6.4.5. The architectures of best-performing variations for both approaches are presented here.

Again, both approaches are based on the model presented in 6.1 and they take inspiration from U-Net [70], more specifically, the "U" shaped architecture that was proposed in the paper. The U-Net architecture is made out of two parts, the Downsampler which creates an important image feature vector by downscaling the image with Convolutional and Max-pooling layers, and the Upsampler which upscales the important image feature vector to the spatial dimensions of the original image using Transposed Convolutional layers.

Both of the approaches have three components based on the model from Chapter 6.1, the Global Feature Extractor, the Downsampler, and the Upsampler.

The Global Feature Extractors are the same as the Image Feature Extractor described in Chapter 6.3, it takes an image of 256×256 pixels, has 5 Convolutional layers with a kernel size of (1,1), 2 Max-pooling layers with a kernel size of (4,4) and it outputs a 64 element feature vector. There is no difference in the Global Feature Extractors between the approaches.

The Downsamplers are modified versions of the Illuminant Estimator described in Chapter 6.1. Just like in the Illuminant estimator, the outputs of the Global feature Extractors are combined with the outputs of the first layers of the Downsamplers. Unlike the Illuminant Estimator neither the Patch-approach nor the Image-approach have a Global-pooling layer, also the final Convolutional layer of the Illuminant Estimator was removed from both Downsamplers. They were removed because they were used to transform the output of the Illuminant Estimator into a

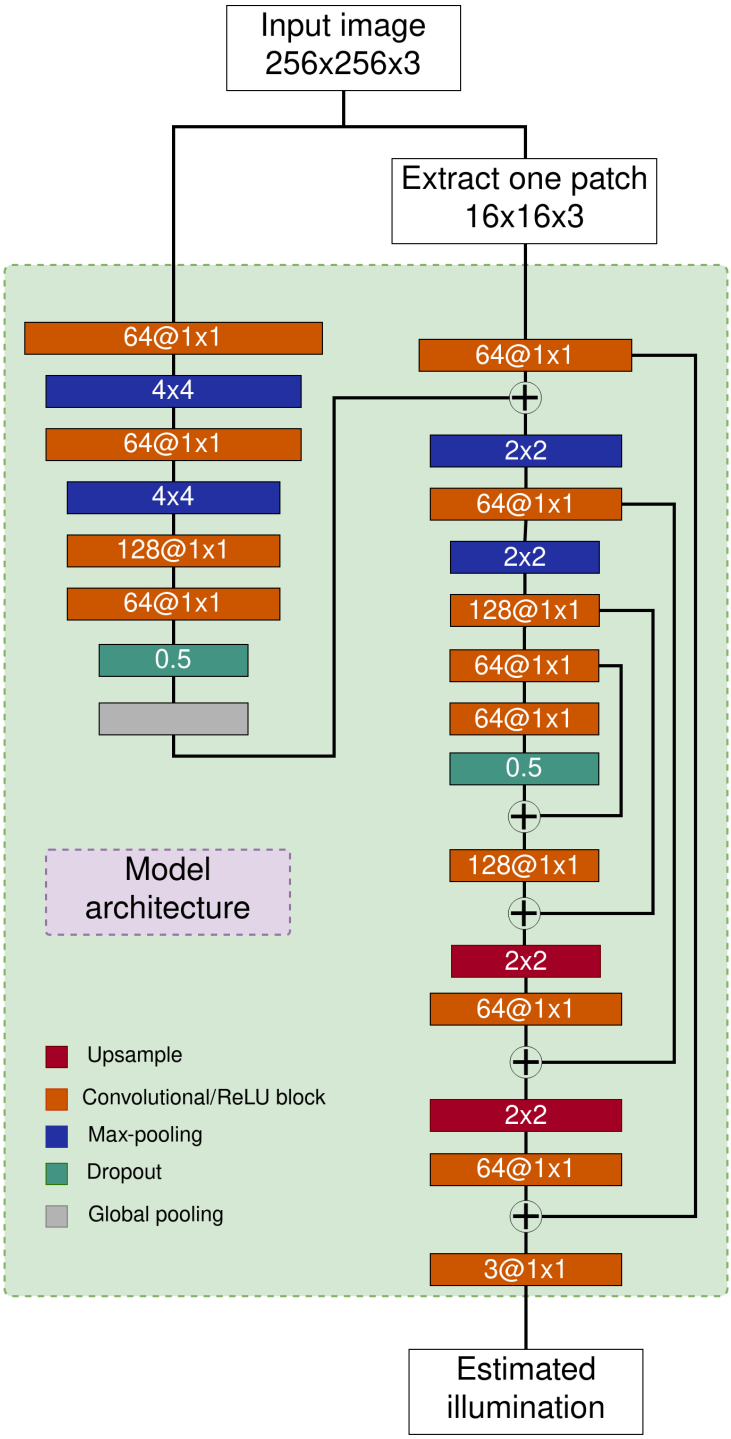


Figure 6.11: Model architecture of the patch-based per-pixel illuminant estimation method

3-value vector that represents the RGB values of the predicted illumination. The Dropout layer has been moved to the Upsampler thanks to the ablation study. The difference between the two Downsamplers in the Patch-approach the Dropout layer was moved from the Downsampler to the Upsampler, while the Image-approach does not move the Dropout layers. Another difference between the Downsamplers is that the Patch-approach takes an input of size 16×16 pixels and the Image-approach takes an input of size 256×256 pixels.

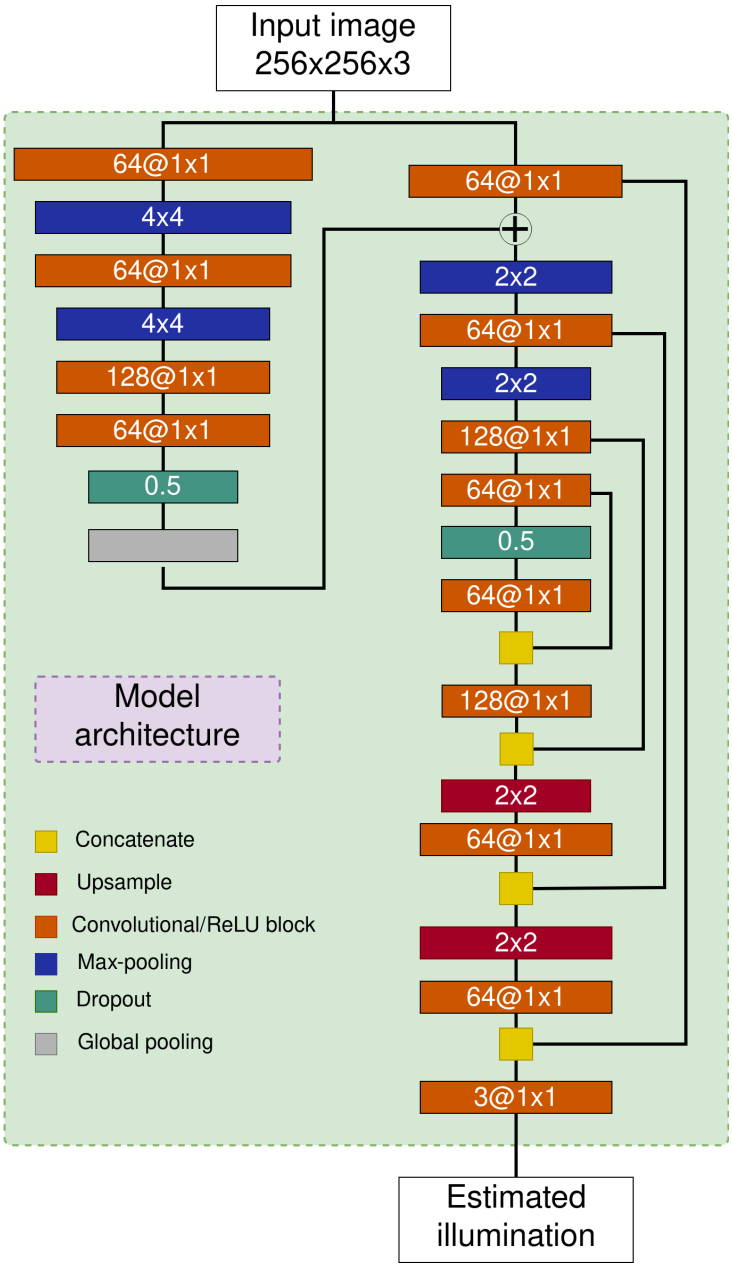


Figure 6.12: Model architecture of the image-based per-pixel illuminant estimation method

The two Upsamplers are in essence the inverse of their respective Downsamplers. Their major difference is the size of the created output image, with the Patch-approach having an output of 16$\times$ 16 pixels and the Image-approach having an output of 256 $\times$ 256. The Upsampler is an inverted version of the original model from Chapter 6.1. The first layer of the Upsampler is a convolutional layer with 64 filters with a kernel size of (1,1). The next layer is also a difference between the Upsamplers, where the Patch-approach unlike the Image-approach has a Dropout layer with a drop-out rate of 0.5 before the Convolutional layer output is combined with the output of the last convolutional layer of the Downsampler. The upsampled tensor is fed into a convolutional layer with 128 filters with a kernel size of (1,1). Its output is combined with the output of the second to last convolutional layer. That output is fed into an Upsampler layer with

a stride of 2, which doubles the spacial dimensions of the tensor. After that, a convolutional layer with 64 filters with a kernel size of (1,1) is used. The output is combined with the second convolutional layer of the Downsampler. The tensor is then upsampled with another Upsample layer with a stride of 2. A convolutional layer with 64 filters of kernel size (1,1) follows. Its output is combined with the output of the first convolutional layer of the Downsampler. The output is fed into the final convolutional layer of the Upsampler which has 3 filters with a kernel size of (1,1). This layer is used to transform the output of the model so that it has the same dimensions as the input of the Downsampler. The combination of tensors differs between the approaches, with the Image-approach Upsampler concatenating the two tensors while the Patch-approach adds the two tensors. All Upsample layers use standard nearest neighbor interpolation resizing. The created architectures are the result of several ablation studies presented in Chapter 6.4.5

The exact architecture of each component can be seen in Figure 6.11 and Figure 6.12.

6.4.3 Training setup

To create this method, Python [40] and TensorFlow 2.9 [41] were used. To train the method, the loss function [43] used to train the single illuminant method from Chapter 6.1 was combined with Mean Absolute Error (MAE). The loss is applied to each pixel of the output image, and the final loss is calculated by averaging the losses for each pixel. The method was trained for 400 epochs. During training, a learning rate scheduler, with cyclical learning rate [44] was used. The minimum learning rate of $1e^{-7}$, maximum learning rate of $2e^{-3}$, and half cycle of 200 epochs were used. The used optimizer was AdamW [42] with a weight decay of $5e^{-5}$. To train the model, the Nvidia RTX A6000 GPU and AMD Ryzen 3960X CPU hardware was used.

6.4.4 Data preprocessing

For the Patch-approach, the preprocessing steps from Chapter 6.3 were used. Before being fed to the model, an image is resized to 256x256, the image is divided into non-overlapping patches of size 16x16 pixels. Patches with only black pixels are not used. Completely black patches are patches that contain the calibration objects. These regions were set to 0 to mask out the calibration object regions as they contain the grout truth being predicted. For the Image-approach, the image is resized to 256x256 pixels and after data augmentation, the image is fed into both the Global Feature Extractor and the Downsampler.

For data augmentation, image flipping and rotation were used. In the Patch-approach, the augmentations were applied independently on the image and the patches. Following the results of the ablation study in Chapter 6.3.5 only 25% of patches were used during training of the Patch-approach. Before being fed into the model image standardization was used on the input

image and input patch. Image standardization makes the average value of the image 0 and the standard deviation of the image 1.

6.4.5 Ablation study

In this chapter, the ablation studies performed to prove that the proposed architectures achieve the best results are presented. All the ablation experiments were performed on the LSMI galaxy subset using the same train/validation/test split of 70/20/10 used by the original authors.

In the first ablation study, several variations of the Patch-approach are compared to prove the proposed architecture achieves the best results. The first variant is where the Upsample layers are replaced with transposed convolution layers. In the second variant, the combination of Downsampler and Upsampler convolutional outputs is removed. The third variant replaces layer combination by addition with combination by concatenation. Since the proposed architecture moves the Dropout layer from the Downsampler to the Upsampler each variant with the Dropout in the Downsampler and the Dropout in the Upsampler was also tested. All these variants were tested on both the Image-approach and the Patch-approach.

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
No skip/Dropout Upsampler	2.70	2.77	1.96	2.05	0.77	6.01	7.68
No skip/Dropout Downsampler	2.74	3.00	1.83	1.97	0.64	6.51	8.85
Concatenation/Dropout Upsampler	1.76	2.47	0.98	1.12	0.30	4.58	5.60
Concatenation/Dropout Downsampler	1.74	2.56	0.96	1.09	0.25	4.63	5.61
Transpose/Dropout Upsampler	4.67	3.22	3.77	3.15	1.79	8.92	14.85
Transpose/Dropout Downsampler	4.47	5.73	2.61	3.04	0.73	11.42	14.28
Addition/Dropout Upsampler	1.79	2.48	0.95	1.14	0.32	4.70	5.53
Addition/Dropout Downsampler	**1.65**	**2.40**	**0.89**	**1.04**	**0.25**	**4.35**	**5.28**

Table 6.30: Results of the ablation study of the Patch-approach. The combination of different skip connections and dropout layer placements are examined. The best angular error result for each measure is bolded.

Table 6.30 shows the results of the ablation study for the Patch-approach. It can be seen that the architecture with tensor combination by addition with the Dropout in the Downsampler achieves the best results. An interesting thing to note is that the usage of Transpose Convolutions significantly reduces the accuracy of the model, even more than the variant where the Downsampler and Upsampler are not combined.

Table 6.31 shows the results of the ablation study for the Image-approach. Here the best results are obtained with the architecture where the Downsampler and Upsampler are combined

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
No skip/Dropout Upsampler	2.75	2.84	2.02	2.13	0.82	6.03	7.13
No skip/Dropout Downsampler	2.57	2.97	1.72	1.84	0.75	5.97	7.27
Concatenation/Dropout Upsampler	**1.78**	**2.79**	**0.94**	**1.08**	**0.27**	**4.80**	**5.65**
Concatenation/Dropout Downsampler	1.84	2.86	0.99	1.10	0.31	4.93	6.22
Transpose/Dropout Upsampler	4.56	5.98	2.54	2.89	0.76	11.98	15.73
Transpose/Dropout Downsampler	4.61	6.03	2.56	2.94	0.76	12.11	15.93
Addition/Dropout Upsampler	1.95	2.88	1.08	1.21	0.31	5.21	6.80
Addition/Dropout Downsampler	1.88	2.90	1.00	1.12	0.30	5.08	6.41

Table 6.31: Results of the ablation study of the Image-approach. The combination of different skip connections and dropout layer placements are examined. The best angular error result for each measure is bolded.

by concatenation with the Dropout in the Downsampler. Again, the Transpose Convolutions significantly reduced the model accuracy.

The next ablation study is how the use of the Global Feature Extractor affects the accuracy of the Image-approach. The results can be seen in Table 6.32, and it shows us that the usage of Global Feature Extractor significantly improves the model accuracy.

Method	Mean	std.	Med.	Tri.	best 25%	worst 25%	95 Percentile
Without Global Feature Extractor	4.26	5.92	2.39	2.74	0.72	11.10	13.57
With Global Feature Extractor	1.78	2.79	0.94	1.08	0.27	4.80	5.65

Table 6.32: Comparison of Image-approach method accuracy with the Global Feature Extractor and without the Global Feature Extractor.

The final ablation study compares the loss used for the method in chapter 6.33 and Chapter 6.34 with an edited version that combines the [43] loss with the Mean Absolute Error (MAE) loss.

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Li et al.[43] loss	1.70	2.56	0.88	1.03	0.25	4.61	5.56
Li et al. [43] loss & MAE loss	1.65	2.40	0.89	1.04	0.25	4.35	5.28

Table 6.33: Comparison of Patch-approach method accuracy when the method uses [43] loss combined with MSE loss and when the method only uses [43] loss.

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Li et al. [43] loss	1.85	2.92	1.00	1.11	0.30	5.02	6.15
Li et al. [43] loss & MAE loss	1.78	2.79	0.94	1.08	0.27	4.80	5.65

Table 6.34: Comparison of Image-approach method accuracy when the method uses [43] loss combined with MAE loss and when the method only uses [43] loss.

Table 6.33 shows how the two losses affect the accuracy of the Patch-approach. It can be seen that the addition of MAE improves the average angular error, Worst 25% average angular error, 95 percentile error, and the standard deviation, while the Trimean angular error and Best 25% angular error are similar with the loss without MAE being 0.01° better. Loss with MAE was chosen since a smaller Worst 25% is more important.

Table 6.34 shows us how the two losses affect the Image-approach. It can be seen that the addition of MAE improves the results of all measures.

6.4.6 Evaluation

To evaluate the method LSMI [33], Shadows & Lumination, and Filters & Lumination datasets were used. The train/validation/test procedure proposed in the LSMI paper [33]. Three different models were created, one for each camera in the dataset, with a 0.7/0.2/0.1 train/validation/test split. The test set results are presented as three separate test sets. The single, multi, and mixed number of illuminants test sets.

For the Shadows & Lumination dataset, the evaluation procedure explained in Chapter 5.4.6 was used. Five experiments using three different evaluation protocols were performed. The experiments use all images, only use outdoor images, only use indoor images, only use night-time images, or train on images from one camera and test on the images from other cameras. For each experiment, a 5-fold split was used, meaning five instances of the model were created and each was trained on 4 folds and tested on 1 fold. The One-to-Many protocol experiment is inverted, 1 fold is used for training and 4 folds were used for testing. The segmentation capabilities of the models were also tested on this dataset. For comparison, Mean Dice error and Dice Standard Deviation were used.

For the Filters & Lumination dataset, two variants of the dataset, the full variant and the reduced variant were used. For each variant, a random 3-fold split, where each scene was present in only one fold was used.

To compare different methods, the angular distance between the ground-truth illumination vector and the predicted illumination vector 5.3.2 was used. This is a commonly used metric for computational color constancy method comparison. The angular distance for each pixel in the predicted illumination mask was calculated. The mean, median, Best 25% mean, Worst 25%

mean, and 95 percentile angular error metrics are presented.

6.4.7 Results

This chapter presents the results the model has achieved on the three datasets and those results are compared with results achieved by methods from the literature.

The results start with the LSMI dataset and the Galaxy subset in Table 6.35. It shows us that both the Image-approach and Patch-approach outperform all other methods from the literature, but they do not outperform the method from Chapter 6.3. When comparing the two methods, it can be seen that generally the Patch-approach performs better, but achieves similar results when images with multiple illuminants are tested. The Image-approach achieves better median results, while the Patch-approach achieves better mean results. This shows that the Patch-approach performs better with outliers.

Method	Single		Multi		Mixed	
	Mean	Median	Mean	Median	Mean	Median
Pix2Pix [67]	6.53	2.17	4.28	2.63	5.66	2.44
Gijsenij et al.[68]	7.49	6.04	12.38	9.57	10.09	7.43
Bianco et al. [55]	4.15	3.30	5.56	4.33	4.89	3.83
HDRNet [69]	2.85	2.20	3.13	2.70	3.06	2.54
U-Net [70]	2.95	1.86	2.35	2.00	2.63	1.91
Patch estimation method	**1.19**	0.75	**2.16**	1.53	**1.57**	0.93
Per-Patch per-pixel approach	1.23	**0.67**	2.30	1.55	1.65	**0.90**
Per-Image per-pixel approach	1.45	0.73	2.33	**1.51**	1.78	1.32

Table 6.35: Comparison of results obtained on the Galaxy phone camera. The best angular error results are bolded.

Next is the Nikon LSMI subset in Table 6.36. Here again, both methods outperform methods from the literature and both methods have lower accuracy than the method from Chapter 6.3. The results of the two approaches are similar, but the Patch-approach achieves better results for all measures.

The final LSMI subset for the Sony camera is presented in Table 6.37. The methods again do not outperform the method from Chapter 6.3 and both outperform other methods from the literature. The interesting thing in this table is that the Image-approach achieves around 0.2° degrees better results while having similar or better median angular error results when compared to the method from Chapter 6.3. This shows us that the Image-approach has more trouble with outliers than the method from Chapter 6.3.

Method	Single		Multi		Mixed	
	Mean	Median	Mean	Median	Mean	Median
Pix2Pix [67]	6.10	2.27	4.18	2.76	5.41	2.49
Bianco et al. [55]	3.18	2.61	4.65	4.19	3.93	3.48
HDRNet [69]	2.76	2.43	3.20	3.01	2.99	2.61
U-Net [70]	1.51	1.14	2.36	1.84	1.95	1.45
Patch estimation method	**1.27**	0.67	**1.99**	1.43	**1.53**	0.85
Per-Patch per-pixel approach	1.47	**0.58**	2.09	**1.36**	1.68	**0.74**
Per-Image per-pixel approach	1.49	0.71	2.18	1.45	1.73	0.93

Table 6.36: Comparison of results obtained on the Nikon camera. The best angular error results are bolded.

Method	Single		Multi		Mixed	
	Mean	Median	Mean	Median	Mean	Median
Pix2Pix [67]	4.08	1.72	4.37	3.26	4.20	2.20
Bianco et al. [55]	3.25	2.62	4.38	3.93	3.86	3.19
HDRNet [69]	2.70	2.37	3.65	3.33	3.21	2.89
U-Net [70]	2.83	2.44	3.04	2.78	2.94	2.66
Patch estimation method	**1.45**	0.60	**2.23**	**1.65**	**1.76**	0.93
Per-Patch per-pixel approach	1.73	0.64	2.68	1.86	2.10	**0.90**
Per-Image per-pixel approach	1.77	**0.59**	2.54	**1.65**	2.08	0.91

Table 6.37: Comparison of results obtained on the Sony camera. The best angular error results are bolded.

The second dataset presented is the Shadows & Lumination dataset. For this dataset, both the estimation results and the segmentation results when the model outputs are combined with KMeans clustering are presented.

The results start with the Use-All protocol in Table 6.38. Both methods outperform all other methods from the literature, but only the Image-approach has results similar to the method from Chapter 6.3. While the Image-approach has a better mean, median, and Best 25% it achieves slightly worse Worst 25% and 95 Percentile. This is expected, since here the error is calculated on a per-pixel basis.

Table 6.39 shows how the approaches perform on the One-to-Many protocol. Again, the Image-approach outperforms other methods from the literature, except for Hyp-Sel which achieves the best median and Best 25% results. The same situation as in previous tables occurs, where

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Patch (Gray World) [65]	5.08	4.36	3.76	4.10	1.21	11.14	13.74
Keypoint (Gray World) [65]	5.45	4.32	4.31	4.60	1.38	11.43	13.80
Superpixel (Gray World) [65]	5.09	4.53	3.65	4.03	1.15	11.42	14.10
Hyp-Sel [60]	6.80	7.43	4.60	5.06	0.97	16.47	20.55
Bianco et al. [55]	3.05	2.69	2.32	2.47	0.85	6.52	7.92
Patch estimation method	2.46	**2.45**	1.73	1.90	0.53	**5.62**	**6.80**
Per-Patch per-pixel approach	2.72	2.61	1.96	2.15	0.56	6.15	7.41
Per-Image per-pixel approach	**2.41**	2.54	**1.62**	**1.80**	**0.48**	5.72	7.06

Table 6.38: The mean, standard deviation, median Best 25%, Worst 25%, and 95 Percentile angular error scores on the Use-All protocol. The best results are bolded.

Hyp-Sel achieves good Best 25% but achieves significantly worse Worst 25%. The Image-approach even outperforms the Bianco et al. method [55] in the Worst 25%. The Patch-approach does not achieve stellar results and is worse than the Bianco et al. method [55]. This leads us to conclude that the Image-approach has good camera invariance capabilities, while the Patch-approach does not.

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Patch (Gray World) [65]	5.08	4.36	3.76	4.10	1.21	11.14	13.74
Keypoint (Gray World) [65]	5.45	4.32	4.31	4.60	1.38	11.43	13.80
Superpixel (Gray World) [65]	5.09	4.53	3.65	4.03	1.15	11.42	14.10
Hyp-Sel [60]	5.62	5.85	3.93	4.29	**0.83**	13.35	16.50
Bianco et al. [55]	4.68	3.06	4.18	4.25	1.77	8.48	9.96
Patch estimation method	4.65	3.30	4.12	4.18	1.62	8.63	9.78
Per-Patch per-pixel approach	4.94	3.38	4.33	4.43	1.74	9.20	10.69
Per-Image per-pixel approach	**4.55**	**3.02**	**4.06**	**4.15**	1.71	**8.24**	**9.40**

Table 6.39: The mean, standard deviation, median Best 25%, Worst 25%, and 95 Percentile angular error scores on the One-to-Many protocol. The best results are bolded.

Table 6.40 shows the results of the approaches on the Outdoor By-Type protocol. Both approaches achieve good results, with the Patch-approach achieving similar results to the Bianco et al. method [55] and the Image-approach achieving better results than the method from Chapter 6.3.

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Patch (Gray World) [65]	5.14	4.24	3.89	4.22	1.31	11.04	13.53
Keypoint (Gray World) [65]	5.69	4.28	4.60	4.88	1.57	11.59	13.88
Superpixel (Gray World) [65]	5.15	4.44	3.75	4.14	1.24	11.35	13.91
Hyp-Sel [60]	6.68	6.97	4.66	5.09	1.01	15.83	19.64
Bianco et al. [55]	2.47	**1.83**	2.01	2.12	0.73	5.00	6.03
Patch estimation method	2.23	1.86	1.69	1.82	0.54	4.82	5.95
Per-Patch per-pixel approach	2.45	2.07	1.83	1.99	0.55	5.37	6.57
Per-Image per-pixel approach	**2.03**	1.90	**1.43**	**1.58**	**0.44**	**4.65**	**5.84**

Table 6.40: The mean, standard deviation, median Best 25%, Worst 25%, and 95 Percentile angular error scores on the Outdoor By-Type protocol. The best results are bolded.

Table 6.41 shows the results of the approaches on the Nighttime By-Type protocol. The interesting thing to note in this table is that all methods, except for Hyp-Sel [60], achieve comparable results with a mean angular error difference of around 0.2°. We can see that non-learning-based methods outperform learning-based methods with Keypoint segmentation achieving the best results for a method from the literature. Both of the approaches achieve results that are worse than the results of Chapter 6.3 method. The Image-approach achieves better results and is similar to Bianco et al. method [55] and Keypoint segmentation assisted method [65] with the most noticeable difference being the 95 Percentile metric.

Method	Mean	std.	Med.	Trimean	Best 25%	Worst 25%	95 Percentile
Patch (Gray World) [65]	3.53	3.66	2.45	2.68	0.82	8.14	9.87
Keypoint (Gray World) [65]	3.38	**3.30**	2.45	2.65	0.81	7.57	9.22
Superpixel (Gray World) [65]	3.50	3.57	2.45	2.66	0.81	8.01	9.63
Hyp-Sel [60]	4.22	6.66	2.12	2.51	**0.43**	11.75	14.82
Bianco et al. [55]	3.41	3.92	2.53	2.68	0.91	7.48	**9.01**
Patch estimation method	**3.25**	**3.30**	**2.31**	**2.47**	0.76	**7.40**	9.22
Per-Patch per-pixel approach	3.54	3.60	2.44	2.65	0.83	8.21	10.25
Per-Image per-pixel approach	3.39	3.41	2.38	2.58	0.72	7.84	9.96

Table 6.41: The mean, standard deviation, median Best 25%, Worst 25%, and 95 Percentile angular error scores on the Nighttime By-Type protocol. The best results are bolded.

Table 6.42 shows the results of the approaches on the Indoor By-Type protocol. Here the

best results are achieved by the Image-approach with results slightly better than the results of the method from Chapter 6.3. The Patch-approach outperforms other methods from the literature but has the worst results out of our three proposed methods.

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Patch (White Patch) [65]	6.37	5.43	4.79	5.22	1.22	14.15	17.20
Keypoint (White Patch) [65]	6.13	5.46	4.62	4.94	1.12	13.78	17.16
Superpixel (White Patch) [65]	6.57	5.53	5.02	5.44	1.24	14.51	17.59
Hyp-Sel [60]	8.73	8.11	6.53	7.01	1.47	19.77	24.37
Bianco et al. [55]	4.43	3.03	3.73	3.88	1.42	8.60	10.26
Patch estimation method	3.84	2.76	3.25	3.36	1.08	7.66	9.48
Per-Patch per-pixel approach	4.04	2.91	3.40	3.53	1.14	8.09	9.99
Per-Image per-pixel approach	**3.70**	**2.64**	**3.08**	**3.23**	**1.06**	**7.39**	**9.15**

Table 6.42: The mean, standard deviation, median Best 25%, Worst 25%, and 95 Percentile angular error scores on the Indoor By-Type protocol. The best results are bolded.

Table 6.43 shows how the methods perform when they are adapted to perform image segmentation. This is achieved by taking the output of the methods and using K-means to cluster the pixels into two groups. Just like the method from Chapter 6.3, these methods do not achieve better results than methods created for image segmentation. The Patch-approach achieves similar results to the method from Chapter 6.3. The interesting thing to note is that the Image-approach achieves far better results than the Patch-approach or the method from Chapter 6.3. This leads us to conclude that segmenting an image on a patch-by-patch basis is not a good approach.

The final dataset tested is the Filters & Lumination dataset. The Full Filters variant results are in Table 6.44. Again, the method from Chapter 5.4.6 achieves the best results. The next best method is neither of the approaches presented in this chapter. Instead, it is the Keypoint method that achieves better results in all measures except for the Worst 25% where the Image-approach achieves the best results.

The interesting thing with this variant is that the use of extreme filters corrupts the images so that even when an image is corrected with the ground truth it will look unnatural. This benefits the non-learning-based methods since the information from the corrupted images does not affect how the method will estimate the illumination in an image that has no extreme filters. Examples of how the proposed methods perform on such images can be seen in Chapter 6.4.8.

Table 6.45 presents the results achieved on the Reduced Filters variant. Here both approaches achieve better results than all other methods from literature while still having a lower

Method	Use-All	One-to-Many	Outdoor	Indoor	Nighttime
Simple Thresholding	80.9±12.6	80.9±12.6	80.8±12.8	73.6±14.9	80.2±15.3
Max Entropy [79]	71.5±22.4	71.5±22.4	73.0±20.2	65.6±24.2	68.4±28.9
OTSU Thresholding [80]	80.1±10.7	80.1±10.7	80.9±10.6	74.8±09.6	77.9±11.8
UNet (VGG16) [70]	88.2±12.0	85.1±13.3	88.8±12.0	79.1±13.0	86.4±12.8
UNet (SEResNet18) [70]	89.2±12.5	**86.8±12.8**	**90.1±12.2**	80.1±11.4	87.4±12.3
LinkNet (VGG16) [81]	87.9±12.0	84.9±13.2	89.1±01.5	78.3±12.3	85.5±12.8
LinkNet (SEResNet18) [81]	88.9±12.4	85.5±12.9	89.3±12.0	78.0±11.7	86.9±12.7
FPN (VGG16) [82]	88.7±12.2	85.5±13.6	89.4±11.9	**80.3±12.3**	86.1±12.7
FPN (SEResNet18) [82]	**89.3±11.9**	86.5±12.9	90.0±12.4	79.6±12.1	**87.7±12.4**
PSPNet (VGG16) [83]	88.0±12.0	85.3±13.8	79.8±11.5	75.2±16.8	85.5±13.0
PSPNet (SEResNet18) [83]	86.8±12.5	84.1±13.3	79.1±12.1	72.3±15.9	86.8±13.0
Patch estimation method	77.6±15.5	72.5±16.3	79.2±14.6	64.9±16.1	77.2±17.7
Per-Patch per-pixel approach	75.6±15.2	70.7±15.9	75.6±14.7	67.3±17.4	76.3±16.3
Per-Image per-pixel approach	87.3±12.3	85.2±12.6	89.2±09.9	76.3±16.3	85.3±14.5

Table 6.43: The mean and standard deviation dice scores of different methods tested using different protocols. The results with the best mean are bolded.

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Patch (Gray World) [65]	5.03	5.06	3.41	3.77	1.00	11.89	15.24
Keypoint (Gray World) [65]	4.84	4.82	3.31	3.66	**0.91**	11.41	14.38
Superpixel (Gray World) [65]	5.02	5.00	3.38	3.76	0.99	11.86	15.19
Hyp-Sel [60]	8.98	13.88	4.96	5.58	0.94	24.32	30.08
Bianco et al. [55]	6.02	4.55	4.92	5.13	1.98	12.00	14.66
Patch estimation method	**4.32**	**4.24**	**3.09**	**3.30**	1.17	**9.67**	**12.23**
Per-Patch per-pixel approach	5.18	5.16	3.61	3.88	1.33	11.87	15.63
Per-Image per-pixel approach	4.92	4.89	3.42	3.69	1.24	11.27	14.66

Table 6.44: Comparison of results obtained on the Full Filters dataset. The best results are bolded. std. means standard deviation, Med. means median, and Tri. means trimean.

accuracy than the model from Chapter 6.3. Other methods achieve better results in the Best 25% measure, but they achieve significantly worse results in the Worst 25% and 95 percentile measures.

Method	Mean	std.	Med.	Tri.	Best 25%	Worst 25%	95 Percentile
Patch (Gray World) [65]	3.75	3.23	2.80	3.03	0.89	8.18	10.03
Keypoint (Gray World) [65]	3.67	3.14	2.74	2.98	0.82	8.09	9.93
Superpixel (Gray World) [65]	3.80	3.35	2.80	3.04	0.89	8.41	10.36
Hyp-Sel [60]	6.09	6.86	4.11	4.51	**0.78**	14.88	18.37
Bianco et al. [55]	5.05	3.11	4.55	4.63	1.81	9.17	10.61
Patch estimation method	**2.90**	**1.99**	**2.49**	**2.57**	0.96	**5.58**	**6.62**
Per-Patch per-pixel approach	3.31	2.26	2.83	2.92	1.09	6.35	7.52
Per-Image per-pixel approach	3.18	2.38	2.64	2.74	1.00	6.30	7.45

Table 6.45: Comparison of results obtained on the Reduced Filters dataset. The best results are bolded. std. means standard deviation, Med. means median, and Tri. means trimean.

6.4.8 Qualitative results

In this chapter, a couple of example images of how the model performs are shown to see how the images look after it has been color corrected. The results are compared to the ground truth correction. Examples for all three used datasets are presented and some examples where the models fail to estimate illumination are analyzed.

The Image-approach results on the LSMI [33] dataset are shown in Figure 6.13. It contains images from all three cameras and it can be seen that there are not any noticeable differences between the method-corrected and ground-truth corrected images. Some image regions are easier to predict than others, as seen in the error heatmap, but they fall under the acceptable threshold for the Human Visual System.

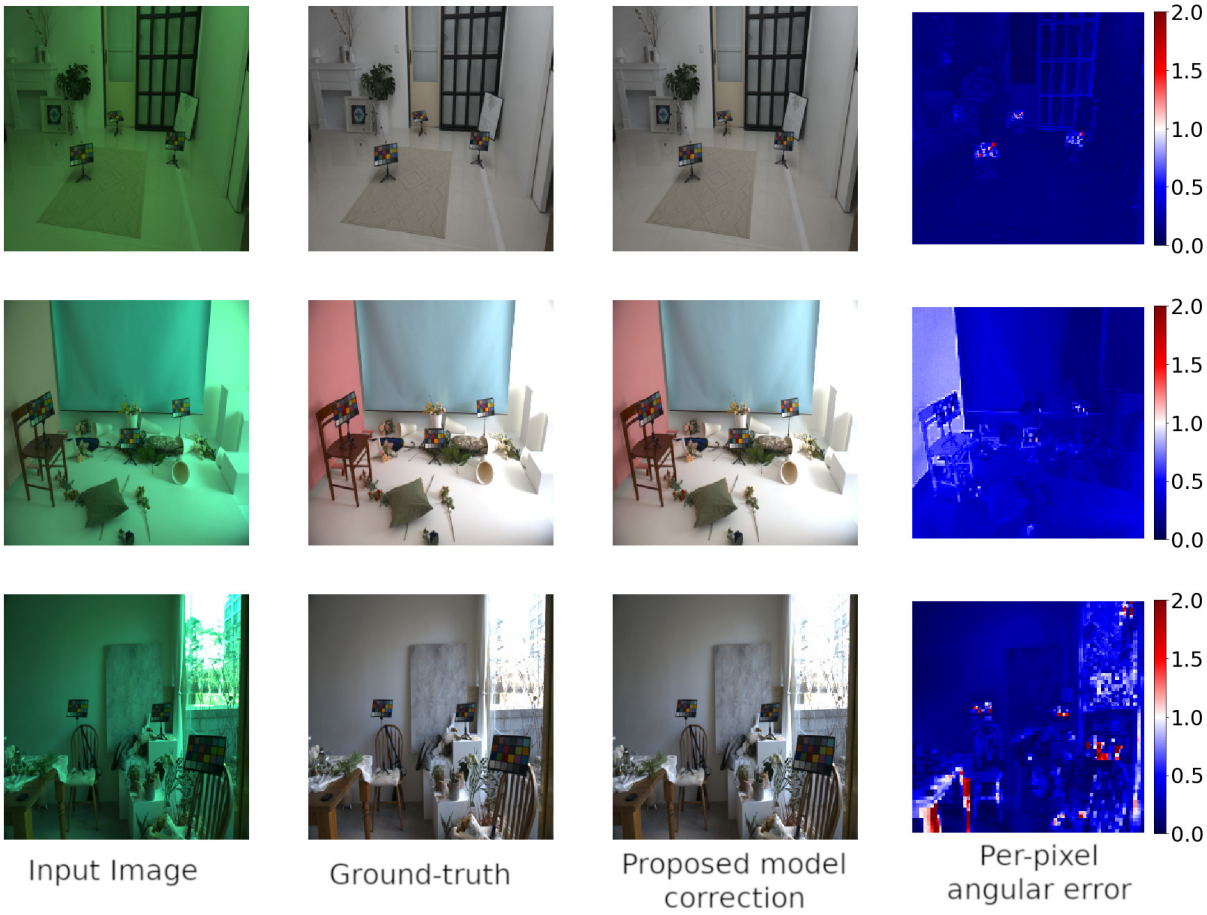


Figure 6.13: Visual comparison of Image-approach model-corrected image and ground-truth corrected image on the LSMI dataset. The angular error heatmap for each image is also shown. The max angular error in the heatmaps is 2°. All images have been tone-mapped for better visualization.

Figure 6.14 shows a couple of random examples of the Image-approach on the Shadows & Lumination dataset. Again some regions are easier to predict than others with regions in the shadow being harder to predict in most situations. Even with this, there is no noticeable difference between the ground truth and method corrected images.

Figure 6.15 showcases some random examples where the Image-approach performs illumination estimation on the Filters & Lumination dataset. The results show that this dataset is the hardest for illumination estimation, but without extreme filters, the method can produce accurate results. There are some regions in the images where the error is around 10°, but these regions do not significantly alter how the corrected image looks.

Figure 6.16 shows a couple of examples where the Image-approach fails to predict the illu-

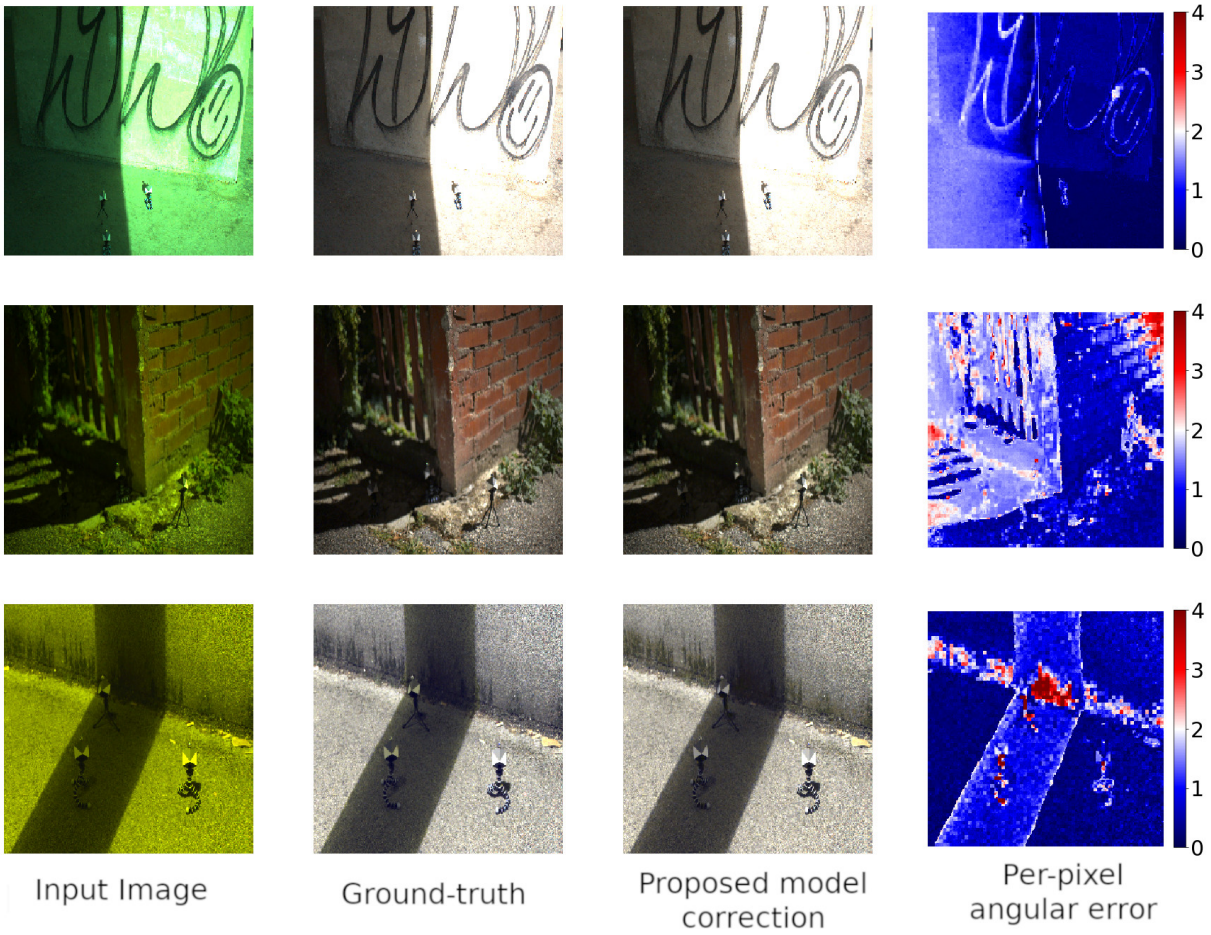


Figure 6.14: Visual comparison of Image-approach model-corrected image and ground-truth corrected image on the Shadows & Lumination dataset. The angular error heatmap for each image is also shown. The max angular error in the heatmaps is 4°. All images have been tone-mapped for better visualization.

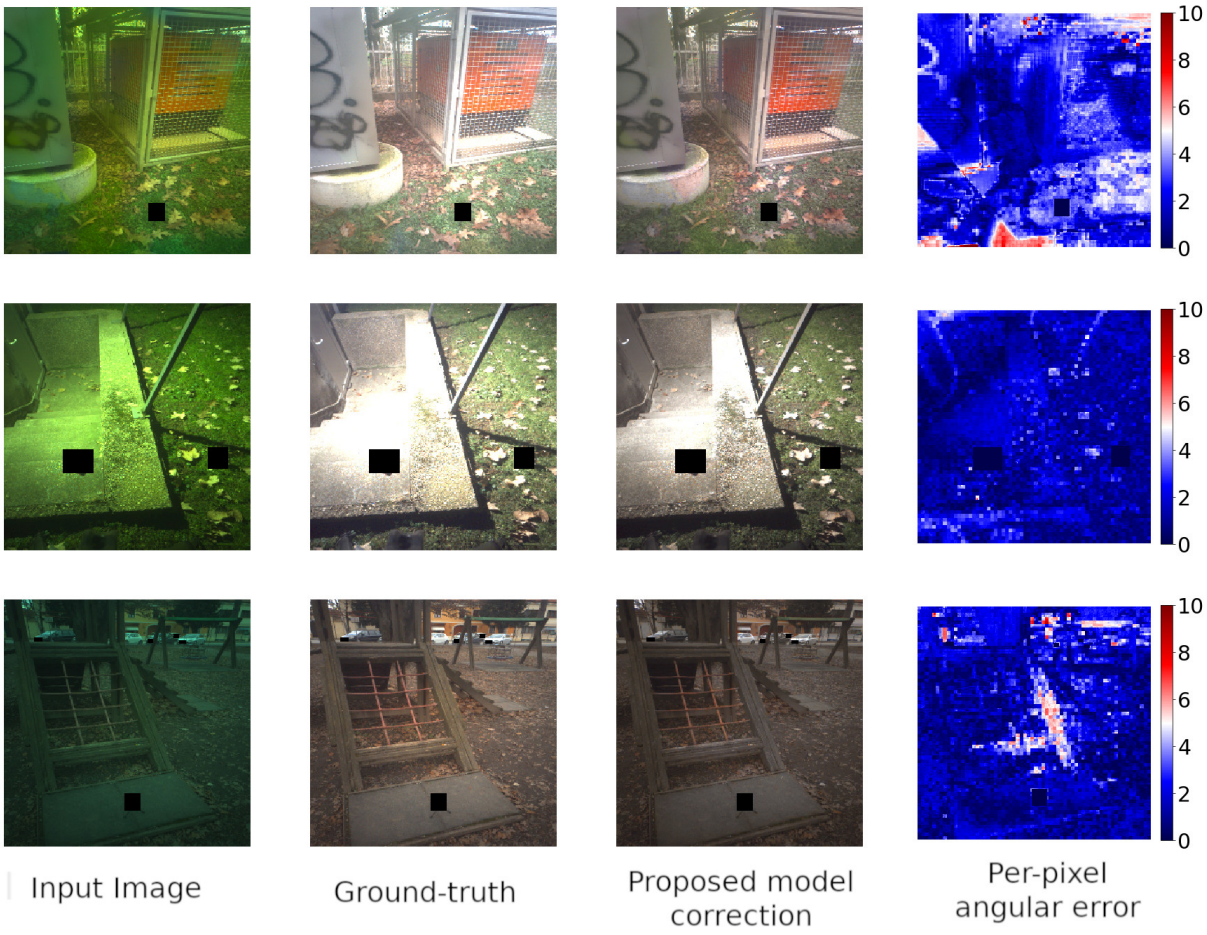


Figure 6.15: Visual comparison of Image-approach model-corrected image and ground-truth corrected image on the Filters & Lumination dataset. The angular error heatmap for each image is also shown. The max angular error in the heatmaps is 10°. All images have been tone-mapped for better visualization.

mination. It can be seen that the model fails when an image contains oversaturated pixels, as seen in the error heat map where the window is. The second region where the method fails is the top of the image where artificial illumination is present, confirming theories that artificial light is harder to predict than natural illumination. The image correction is generally good for this image, but these two regions decrease the overall illuminant prediction accuracy. The second image showcases when the model cannot predict any illumination. This image is the same as the one presented in Figure 6.9, where Chapter 6.3 method fails. This image has an extremely low illumination intensity, so much that noise is visible in the image. The final image showcase does not look improperly corrected, but the heatmap shows the model has problems predicting the illumination of pixels on the border of two illumination regions.

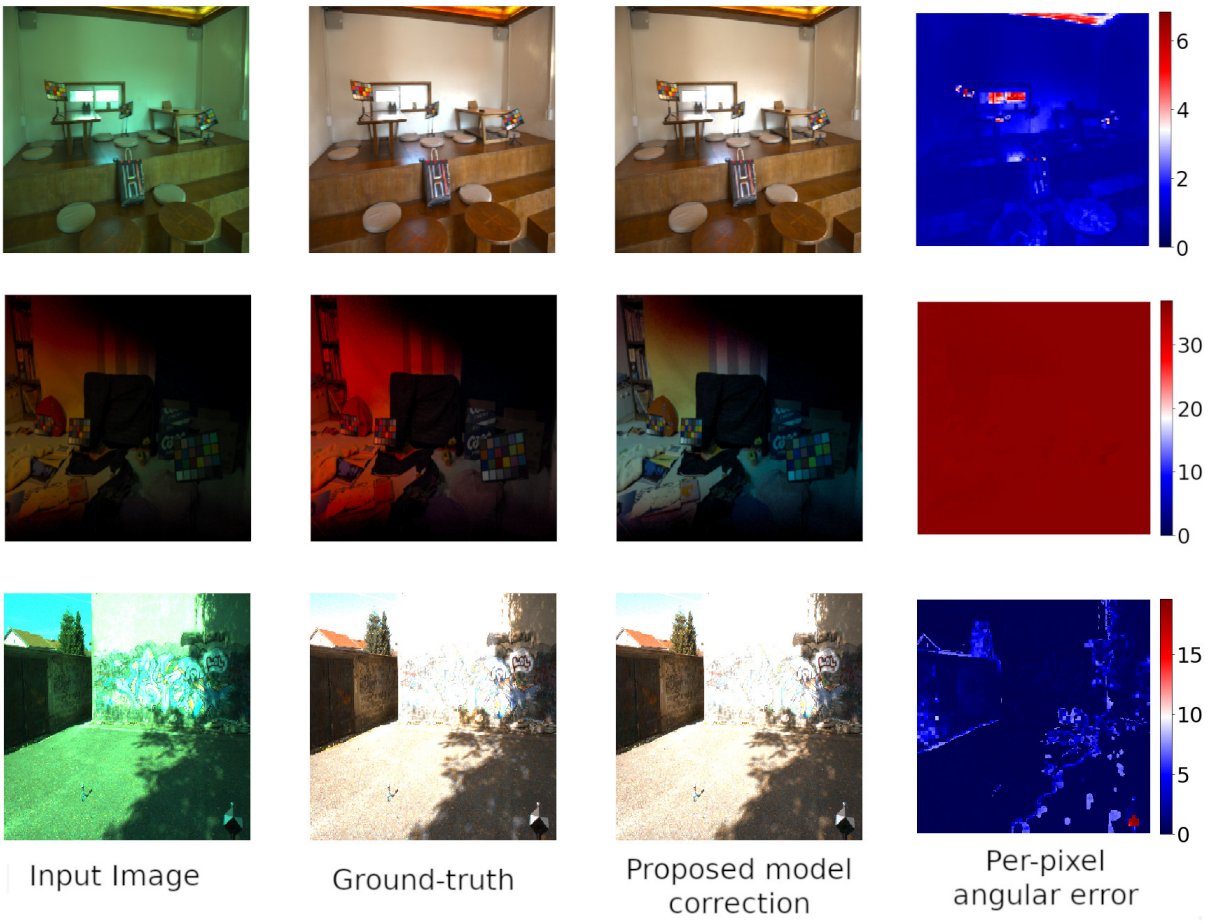


Figure 6.16: Three examples situations where the Image-approach fails to properly estimate the illumination in an image. Max angular error is different for each image. All images have been tone-mapped for better visualization.

Figure 6.17, Figure 6.18, and Figure 6.19 show how the Patch-approach performs on the three datasets.

Figure 6.17 shows some random examples from the three cameras of the LSMI [33] dataset. It can be seen that the method creates an image indistinguishable from the ground truth correction. These results are expected as out of the three datasets, the method achieves the best results

on the LSMI [33] dataset.

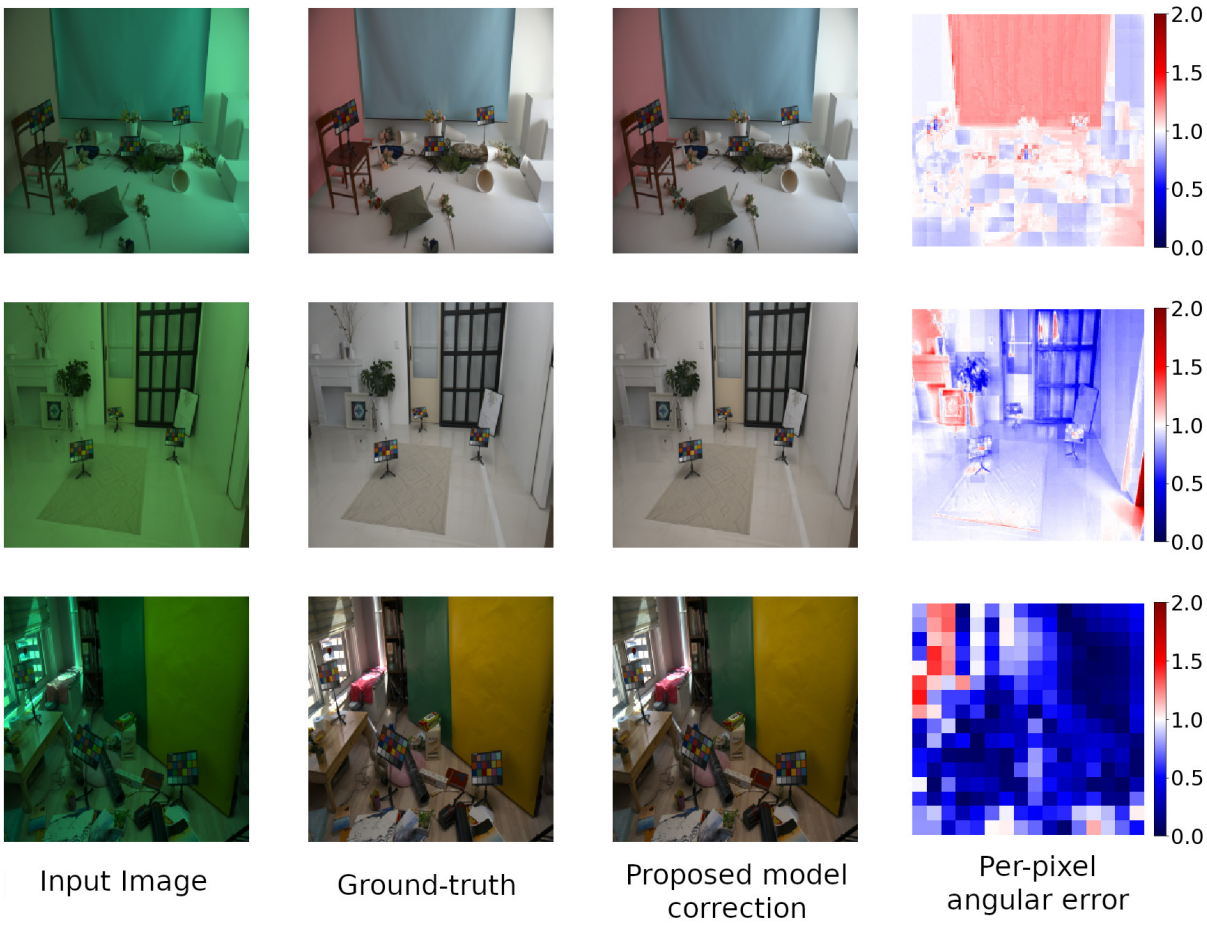


Figure 6.17: Visual comparison of Patch-approach model-corrected image and ground-truth corrected image on the LSMI dataset. The angular error heatmap for each image is also shown. The max angular error in the heatmaps is 2°. All images have been tone-mapped for better visualization.

Similar results can be seen in Figure 6.18 where some example images from the Shadows & Lumination dataset are shown and Figure 6.19 where some samples from the Filters & Lumination dataset are shown.

Figure 6.20 shows a couple of examples where the Patch-approach fails to properly estimate illumination. The first image shows that the Patch-approach also has a problem with artificial illumination. Here the image contains a strong red illuminant, and it can be seen that the model struggles with this, resulting in a tablecloth with an unnatural red bar on the table corner. The second image is again the extremely dark image where the model cannot predict any illumination correctly. All our methods have trouble predicting illumination in low-light images. This can be attributed to the fact that in such a situation, image noise becomes a significant factor in a pixel's value, resulting in improperly corrected images. The final image shows that this method has a problem with estimating illumination border regions, with the error being less noticeable than the method from Chapter 6.3 but more noticeable than the Image-approach as seen in the heatmap.

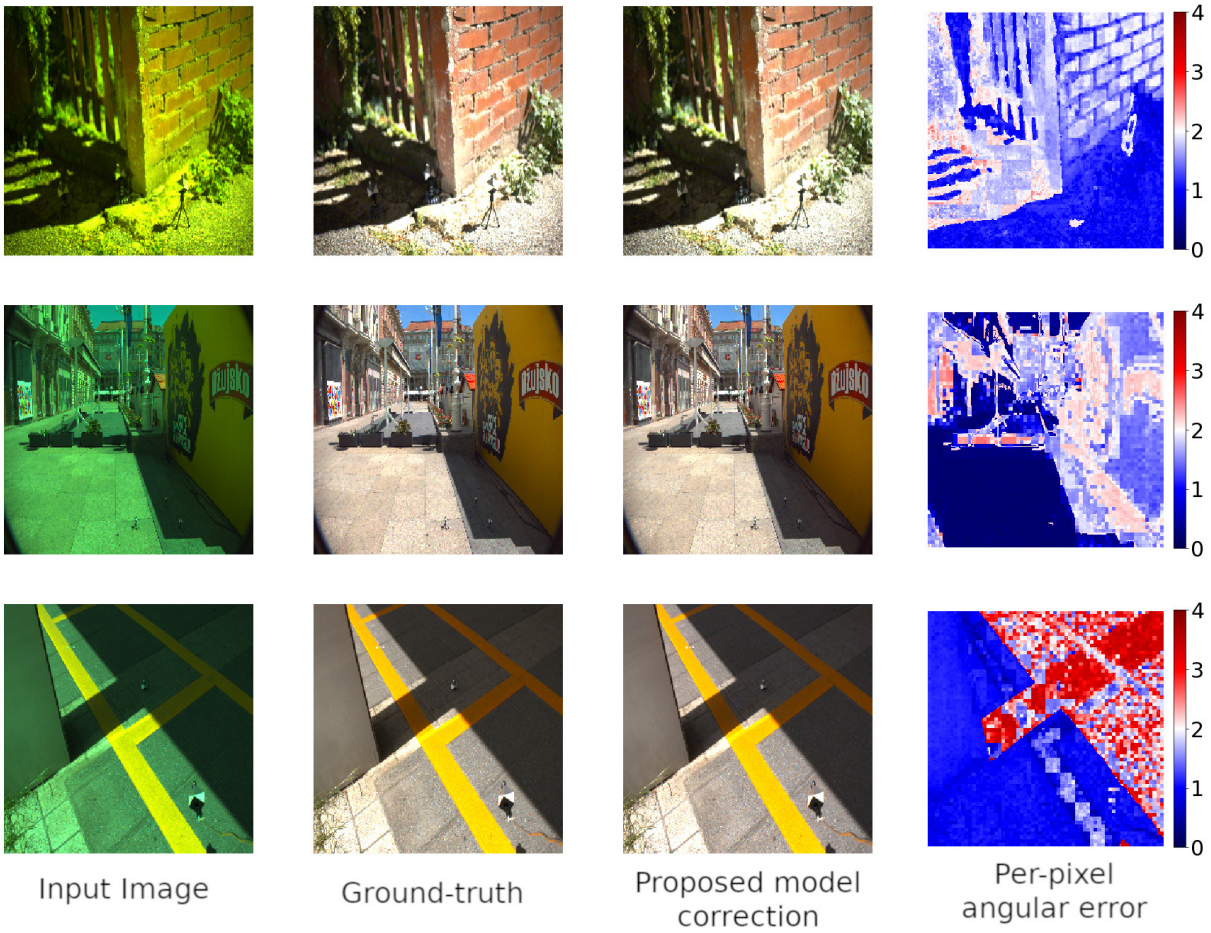


Figure 6.18: Visual comparison of Patch-approach model-corrected image and ground-truth corrected image on the Shadows & Lumination dataset. The angular error heatmap for each image is also shown. The max angular error in the heatmaps is 4°. All images have been tone-mapped for better visualization.

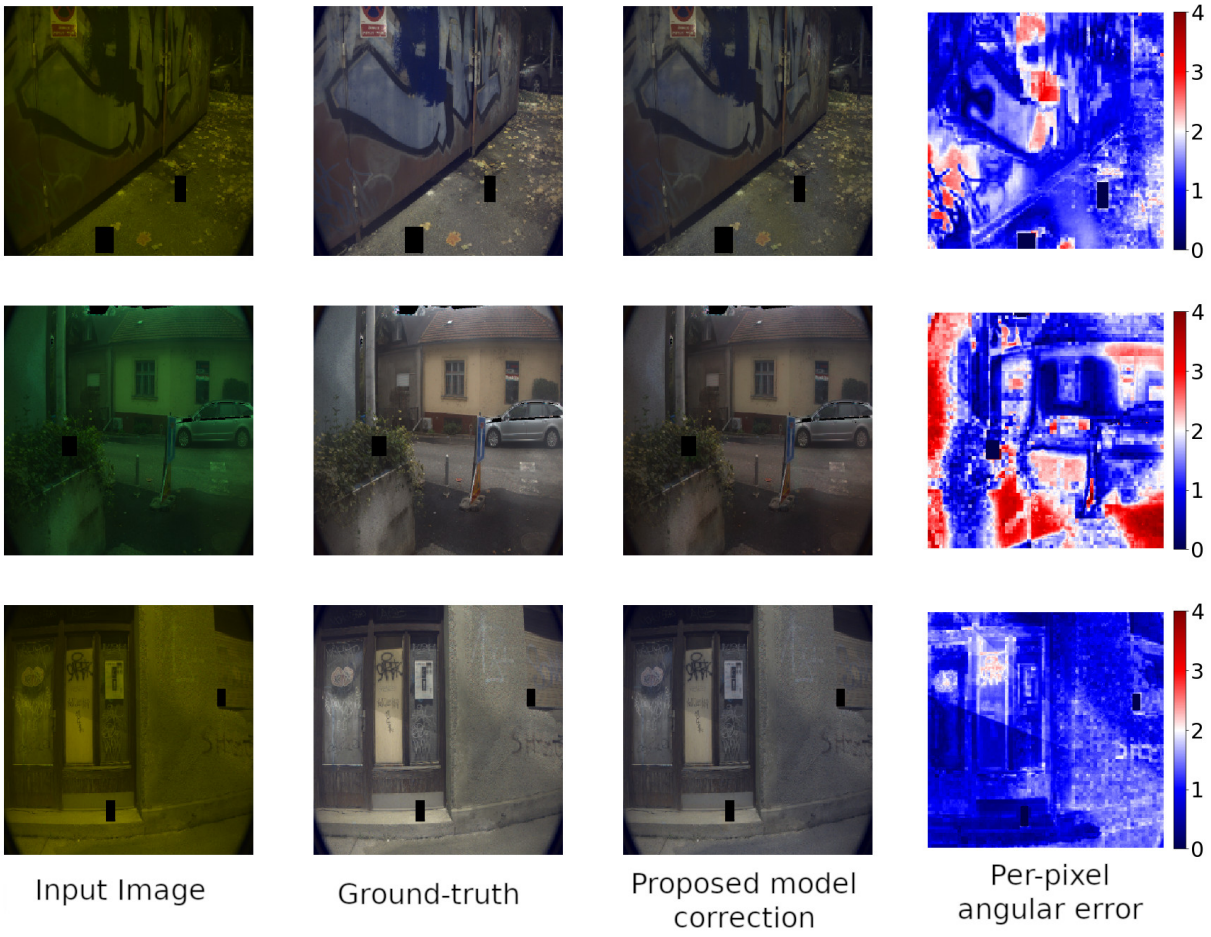


Figure 6.19: Visual comparison of Patch-approach model-corrected image and ground-truth corrected image on the Filters & Lumination dataset. The angular error heatmap for each image is also shown. The max angular error in the heatmaps is 4°. All images have been tone-mapped for better visualization.

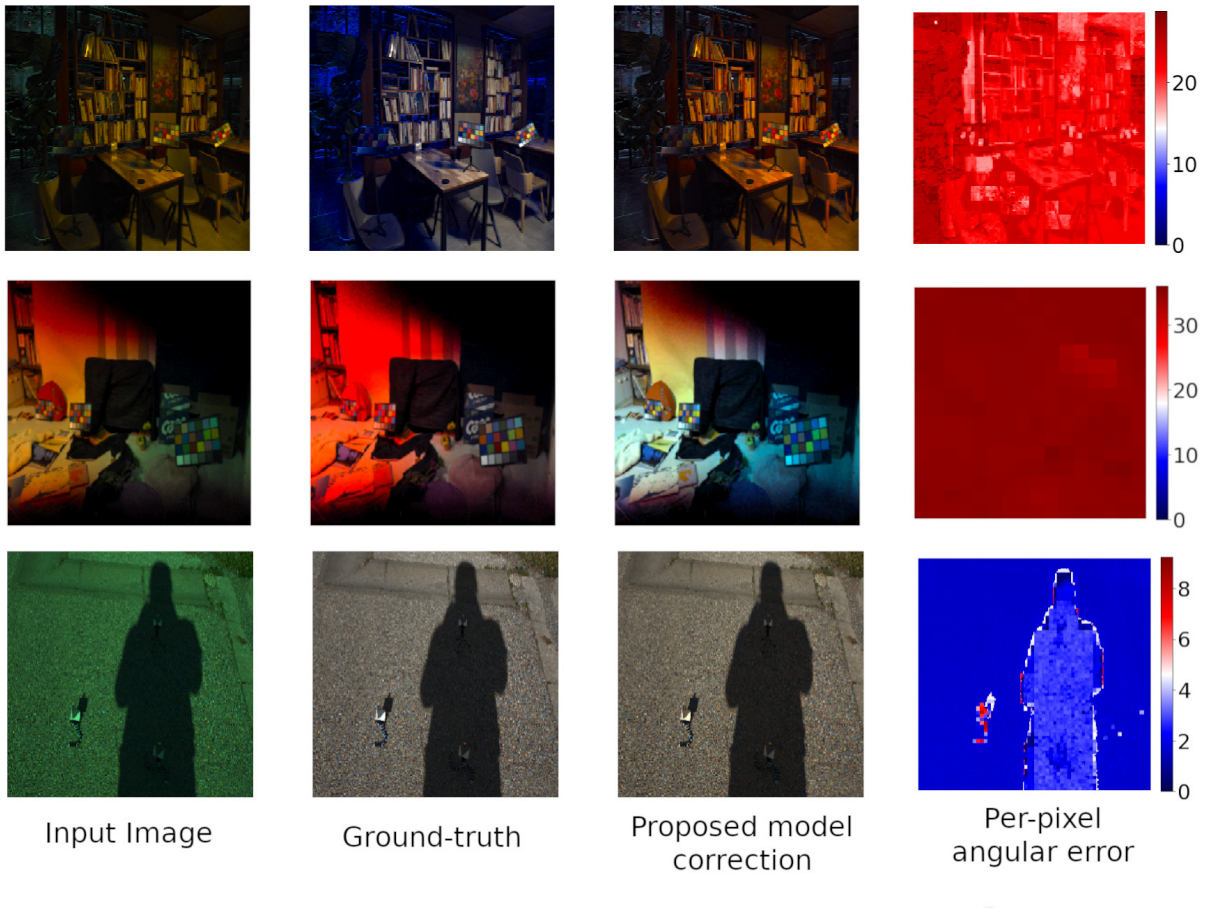


Figure 6.20: Three examples situations where the Patch-approach fails to properly estimate the illumination in an image. Max angular error is different for each image. All images have been tone-mapped for better visualization.

Figure 6.21 showcase the difficulties with estimating the illumination in the Filters & Lumination dataset with extreme filters. The first image is from the Patch-approach and the second image is from the Image-approach. We can see that even with the ground truth, the correction looks unnatural. The extreme filters cause the method to fail since the ground truth is corrupted.

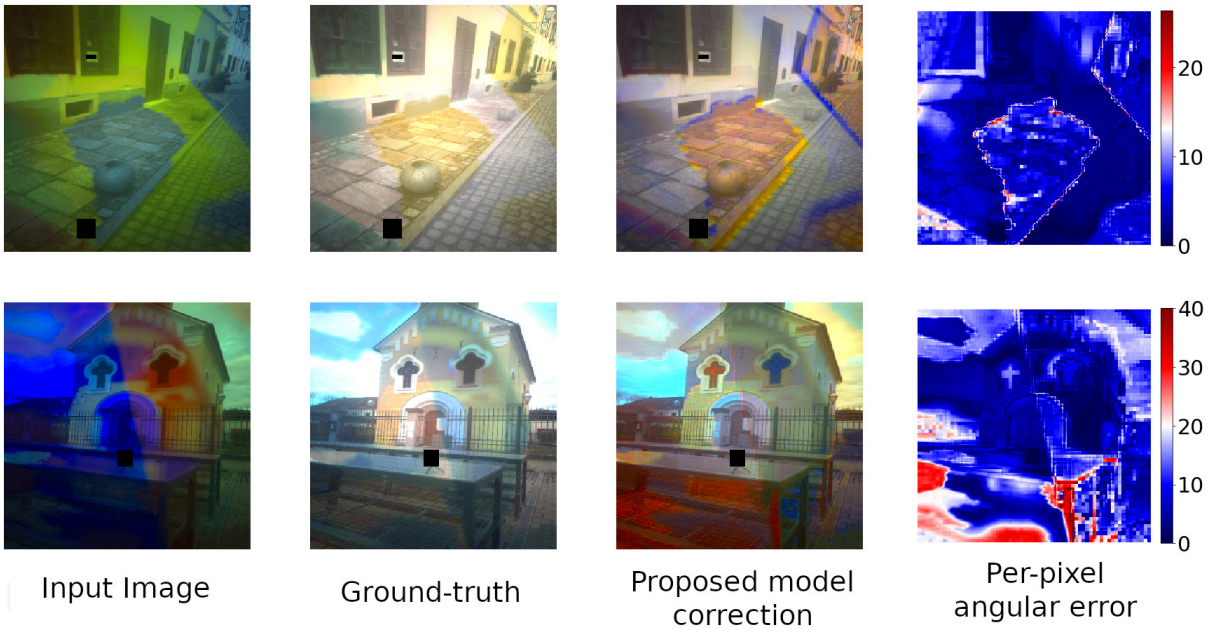


Figure 6.21: A couple of examples showcasing the problem with illumination estimation on images that use extreme filters. The extreme filters degrade images so that natural images cannot be obtained. All images have been tone-mapped for better visualization.

6.4.9 Comparison

In this chapter, two different approaches for true multi-illuminant estimation were introduced. They achieve similar results on the three datasets, with the Patch-approach achieving better results on the LSMI [33] dataset and the Image-approach achieving better results on the Shadows & Lumination and the Filters & Lumination datasets. The Patch-approach has fewer parameters (67K), but the Image-approach with more parameters (88K) has a simpler image processing pipeline. To process a single image the image needs to be fed to the Image-approach only once while for the Patch-approach, the image needs to be divided into small patches and each patch needs to be fed into the method. Both of these methods are more complex that the method from Chapter 6.3 that only has 42K parameters which has the same processing pipeline as the Patch-approach. Out of the three methods, Chapter 6.3 method achieves the best results on Filters & Lumination and LSMI datasets, with the Shadows & Lumination dataset being a toss-up between it and the Image-approach. A downside of this method is that it is not a truly multi-illuminant estimation method, as it uses the assumption that a patch contains only one single illuminant. All three methods have difficulty estimating illumination in low-light conditions,

difficulty estimating oversaturated image regions, and difficulty estimating artificial illumination. All of these methods have their benefits, and in many situations, the Human Visual System would not be able to distinguish between the method-corrected image and the ground-truth corrected image.

Chapter 7

Conclusion

In this thesis, the phenomenon of color constancy and its implications for digital photography are analyzed. Color constancy is the adaptability of the Human Visual System to environmental illumination. This adaptability allows us to perceive an object's color as relatively constant regardless of how much the illumination modifies the object's color. Digital cameras are not able to do this automatically and a method that emulates the Human visual system is needed. The process of removing the illumination chromaticity from a digital image is called white-balancing. White-balancing is a very important step in a camera's image-processing pipeline. Without a white-balancing image would look unnatural. The removal of the illumination chromaticity is also important for many computer vision tasks as color is a very important feature of an object. Computer vision tasks such as object detection, object recognition, and object tracking rely on object color not changing under different lighting conditions. The research done with White-balancing can be divided into two groups, white-balancing when the photographed scene contains only one illuminant and white-balancing when a photographed scene contains an arbitrary number of illuminants.

Both situations were tackled in this thesis, with a method for single illuminant estimation and five different approaches for multi-illuminant estimation. All the multi-illuminant estimation methods were analyzed and the benefits and the limitations of each method were presented. In addition to the estimation method, this thesis presented two different methods for the creation of labeled high-quality multi-illuminant digital images.

Much research has been done in the area of single-illuminant estimation and the created methods can even outperform the Human Visual System. Research has shown that learning-based methods significantly outperform the more classical approaches. A problem with learning-based methods is their computational complexity. These methods require powerful hardware and cannot be easily used on less powerful devices. One of the contributions of this research is the development of learning-based illumination estimation methods. These methods are based on a convolutional neural network based on a novel architecture. The created method is a

lightweight model that has little computational complexity. The method was tested on a variety of publicly available single illuminant color constancy datasets. The results obtained from the experiments show that the proposed methods have equal or better results than other far more complex learning-based methods.

The first multi-illuminant estimation method presented in this thesis addresses the situation when the number of illuminants in an image is known beforehand. The proposed neural network architecture is a modified version of a single-illuminant estimation method from the literature. The authors of the original architecture used an attention mechanism to filter out useless information from the image. The useless information in the image are regions of the image that contain a few surfaces and little variety in color. An example would be a region containing an orange wall. Here we cannot say whether the wall is orange and the illumination is white, the wall is white and the illumination is orange, or any other combination of wall and illumination color. In addition to useless region filtering, the attention mechanism can be used to filter out regions that are not illuminated by the illuminant we are trying to predict. To evaluate and compare this neural network architecture, several methods from the literature were implemented and the performed tests showed this method achieves comparable or better results than methods from the literature.

The limitation of the method is the fact that the number of illuminants needs to be known beforehand. To remove this requirement, another method was developed. Here the illumination is estimated on a patch-by-patch basis. Each patch is a uniform square. The idea is to divide the image into a large number of small patches and estimate the illumination for each patch. Since a patch is small the assumption that it contains only one uniform illuminant is used. There are a couple of methods that used this approach, but they only use information from the patch for illumination estimation. Here the problem of useless information becomes significant, as many patches will not contain enough information for accurate estimation. In this thesis, a method is proposed that solves this problem. The proposed method uses local information from the patch as well as global illumination information from the entire image. Performed experiments show that the addition of global image information significantly improves the accuracy of the method. The method is based on the single-illuminant method presented in this thesis and is also a lightweight convolutional neural network. This method was also compared with existing methods from the literature, and the results show the method achieves comparable or better results while being less complex than existing learning-based approaches.

The limitation of this method is that it performs illumination estimation on a patch-by-patch basis. It achieves great results on multi-illuminant images, but in essence, the method estimates a single illuminant for each patch. The thesis presents two methods that remove this restriction and perform illumination estimation on a pixel-by-pixel basis. Both methods have a similar neural network architecture that is based on the architecture of the presented single-illuminant

estimation method. One method uses the entire image and estimates the illumination for each pixel in the image, while the other divides the image into many small patches and estimates the illumination for each pixel separately. Both methods use global and local image information to estimate the illumination for each pixel. The two methods achieve great results, but neither method is better than the other in all aspects. Both methods were evaluated and compared to existing methods from the literature and both achieve comparable or better results than methods from the literature.

To properly evaluate and compare estimation methods, two methods for the creation of multi-illuminant images were developed. The collection, labeling, and analysis processes for both methods are presented in this thesis. The motivation for the creation of new multi-illuminant datasets was the lack of publicly available large-scale multi-illuminant datasets. Many of the existing datasets are limited in the number of images and not suited for learning-based methods. The first created dataset is a publicly available two-illuminant dataset. It contains 2500 images from a variety of indoor, outdoor, daytime, and nighttime images captured using five different cameras, Canon 5D, Canon 550D, Sony α300, Panasonic FZ1000, and the Motorola one fusion+ mobile camera. For each image, the present illuminants and the segmentation mask are provided. The segmentation mask shows which illuminant illuminates which region of the image. Shadows were used to create the multi-illuminant situation. One of the illuminants is the more intense direct light source. The other illuminant is the weaker ambient scene illumination that illuminates the regions in the shadows. This setup creates a clear border between regions under one illuminant and the regions under the other illuminant. This significantly simplifies the creation of the segmentation mask, as the regions are uniformly illuminated by one of the illuminants. An extensive study was performed to test the quality of the extracted information. The created dataset was used to evaluate the proposed methods.

The limitation of the two-illuminant dataset is the fact that there can only be two illuminants in the image and the fact there is uniform illumination in the two regions of the image. Therefore, another multi-illuminant image creation method was developed and described in this thesis. Using this method a dataset containing 7800 images from 300 scenes was created. The dataset contains a wide variety of indoor, outdoor, daytime, and nighttime images captured using three cameras, Canon 550D, Panasonic FZ1000, and the Motorola one fusion+ mobile camera. The novelty of the proposed method is that with it one can create an infinite number of different multi-illuminant illumination masks for each image. This is achieved by taking several images of the same scene. The difference between the images of the same scene is that each image was created using a different lighting filter placed over the camera lens, effectively changing the scene illumination without editing the scene. Using such images, one can create a multi-illuminant image by combining multiple images of the same scene because the color of an object illuminated by multiple illuminants is a linear combination of the object's color when

it is illuminated by the different illuminants separately. The thesis presents a method for the automatic creation of an image with a variable number of non-uniform overlapping illuminants. A deep analysis of the filters and their effect on images is presented in this thesis. Using this method two color constancy datasets containing 6000 images were created and were used to evaluate the proposed multi-illuminant estimation methods.

Bibliography

[1]Chromatic Adaptation. John Wiley & Sons, Ltd, 2013, ch. 8, str. 156-180, dostupno na: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118653128.ch8

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

[3]Chromatic Adaptation Models. John Wiley & Sons, Ltd, 2013, ch. 9, str. 181-198, dostupno na: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118653128.ch9

[4]Tominaga, S., Dichromatic Reflection Model. Boston, MA: Springer US, 2014, str. 191–193, dostupno na: https://doi.org/10.1007/978-0-387-31439-6_532

[5]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/

[6]Finlayson, G. D., Hordley, S. D., "Color constancy at a pixel", J. Opt. Soc. Am. A, Vol. 18, No. 2, Feb 2001, str. 253–264, dostupno na: http://opg.optica.org/josaa/abstract.cfm?URI=josaa-18-2-253

[7]Tooms, M. S., (ur.), The Bradford Colour Adaptation Transform. John Wiley & Sons, Ltd, 2015, str. 645-647, dostupno na: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119021780.app3

[8]Moroney, N., CIECAM02. New York, NY: Springer New York, 2016, str. 195–202, dostupno na: https://doi.org/10.1007/978-1-4419-8071-7_6

[9]Rizzi, A., Bonanomi, C., "Milano Retinex family", Journal of Electronic Imaging, Vol. 26, No. 3, 2017, str. 1 – 7, dostupno na: https://doi.org/10.1117/1.JEI.26.3.031207

[10]Bani ć, N., Lončarić, S., "Light random sprays retinex: Exploiting the noisy illumination estimation", IEEE Signal Processing Letters, Vol. 20, No. 12, 2013, str. 1240-1243.

[11]McCulloch, W. S., Pitts, W., “A logical calculus of the ideas immanent in nervous activity”, The bulletin of mathematical biophysics, Vol. 5, No. 4, Dec 1943, str. 115-133, dostupno na: https://doi.org/10.1007/BF02478259

[12]Hebb, D. O., The Organization of Behavior: A Neuropsychological Theory. New York: Wiley, 1949.

[13]Rosenblatt, F., “The perceptron: a probabilistic model for information storage and organization in the brain.”, Psychological review, Vol. 65 6, 1958, str. 386-408.

[14]Linnainmaa, S., “Taylor expansion of the accumulated rounding error”, BIT Numerical Mathematics, Vol. 16, No. 2, Jun 1976, str. 146-160, dostupno na: https://doi.org/10.1007/BF01931367

[15]Robbins, H. E., “A stochastic approximation method”, Annals of Mathematical Statistics, Vol. 22, 1951, str. 400-407.

[16]Kingma, D. P., Ba, J., “Adam: A method for stochastic optimization”, 2017.

[17]Schmidhuber, J., “Deep learning in neural networks: An overview”, Neural Networks, Vol. 61, 2015, str. 85-117, dostupno na: https://www.sciencedirect.com/science/article/pii/S0893608014002135

[18]Jordan, M. I., “Chapter 25 - serial order: A parallel distributed processing approach”, in Neural-Network Models of Cognition, ser. Advances in Psychology, Donahoe, J. W., Packard Dorsel, V., (ur.). North-Holland, 1997, Vol. 121, str. 471-495, dostupno na: https://www.sciencedirect.com/science/article/pii/S0166411597801112

[19]Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., “Generative adversarial nets”, in Advances in Neural Information Processing Systems, Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K., (ur.), Vol. 27. Curran Associates, Inc., 2014, dostupno na: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[20]Fukushima, K., “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”, Biological Cybernetics, Vol. 36, No. 4, Apr 1980, str. 193-202, dostupno na: https://doi.org/10.1007/BF00344251

[21]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.

[22]Redmon, J., Divvala, S., Girshick, R., Farhadi, A., "You only look once: Unified, real-time object detection", in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, str. 779-788.

[23]Liu, S., Deng, W., "Very deep convolutional neural network based image classification using small training sample size", in 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 2015, str. 730-734.

[24]Hinton, G., "Neural networks for machine learning, lecture 6a, overview of mini-batch gradient descent", http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf, accessed: 2023–03-23.

[25]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.

[26]Finlayson, G., Hemrit, G., Gijsenij, A., Gehler, P., "A curious problem with using the colour checker dataset for illuminant estimation", 2017.

[27]Bani ć, 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), 2019, str. 366-371.

[28]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://opg.optica.org/josaa/abstract.cfm?URI=josaa-31-5-1049

[29]Bani ć, N., Koščević, K., Lončarić, S., "Unsupervised learning for color constancy", arXiv preprint arXiv:1712.00436, 2017.

[30]Laakom, F., Raitoharju, J., Nikkanen, J., Iosifidis, A., Gabbouj, M., "Intel-tau: A color constancy dataset", IEEE Access, Vol. 9, 2021, str. 39 560–39 567.

[31]Bleier, M., Riess, C., Beigpour, S., Eibenberger, E., Angelopoulou, E., Tröger, T., Kaup, A., "Color constancy and non-uniform illumination: Can existing algorithms work?", in 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). IEEE, 2011, str. 774–781.

[32]Beigpour, S., Riess, C., Van De Weijer, J., Angelopoulou, E., "Multi-illuminant estimation with conditional random fields", IEEE Transactions on Image Processing, Vol. 23, No. 1, 2013, str. 83–96.

[33]Kim, D., Kim, J., Nam, S., Lee, D., Lee, Y., Kang, N., Lee, H.-E., Yoo, B., Han, J.-J., Kim, S. J., "Large scale multi-illuminant (lsmi) dataset for developing white balance algorithm under mixed illumination", in 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, str. 2390-2399.

[34]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.

[35]Li, Z., Chen, J., "Superpixel segmentation using linear spectral clustering", in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, str. 1356-1363.

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

[37]Xiao, J., Gu, S., Zhang, L., "Multi-domain learning for accurate and few-shot color constancy", in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, str. 3258–3267.

[38]Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., "Dropout: A simple way to prevent neural networks from overfitting", J. Mach. Learn. Res., Vol. 15, No. 1, jan 2014, str. 1929–1958.

[39]Afifi, M., Brown, M. S., "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, str. 243–252.

[40]Van Rossum, G., Drake, F. L., Python 3 Reference Manual. Scotts Valley, CA: CreateSpace, 2009.

[41]Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., "TensorFlow: Large-scale machine learning on heterogeneous systems", dostupno na: https://www.tensorflow.org/ Software available from tensorflow.org. 2015.

[42]Loshchilov, I., Hutter, F., “Fixing weight decay regularization in adam”, CoRR, Vol. abs/1711.05101, 2017, dostupno na: http://arxiv.org/abs/1711.05101

[43]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.

[44]Smith, L. N., “Cyclical learning rates for training neural networks”, in 2017 IEEE winter conference on applications of computer vision (WACV). IEEE, 2017, str. 464–472.

[45]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.

[46]Buchsbaum, G., “A spatial processor model for object colour perception”, Journal of the Franklin institute, Vol. 310, No. 1, 1980, str. 1–26.

[47]Gao, S.-B., Yang, K.-F., Li, C.-Y., Li, Y.-J., “Color constancy using double-opponency”, IEEE transactions on pattern analysis and machine intelligence, Vol. 37, No. 10, 2015, str. 1973–1985.

[48]Yang, K.-F., Gao, S.-B., Li, Y.-J., “Efficient illuminant estimation for color constancy using grey pixels”, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, str. 2254–2263.

[49]Finlayson, G. D., Trezzi, E., “Shades of gray and colour constancy”, in Color and Imaging Conference, Vol. 2004, No. 1. Society for Imaging Science and Technology, 2004, str. 37–41.

[50]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.

[51]Barnard, K., Martin, L., Coath, A., Funt, B., “A comparison of computational color constancy algorithms. ii. experiments with image data”, IEEE transactions on Image Processing, Vol. 11, No. 9, 2002, str. 985–996.

[52]Koš čević, K., Subašić, M., Lončarić, S., “Guiding the illumination estimation using the attention mechanism”, in Proceedings of the 2020 2nd Asia Pacific Information Technology Conference, 2020, str. 143–149.

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

[54]Gijsenij, A., Gevers, T., Van De Weijer, J., “Physics-based edge evaluation for improved color constancy”, in 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009, str. 581–588.

[55]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.

[56]Laakom, F., Raitoharju, J., Iosifidis, A., Nikkanen, J., Gabbouj, M., “Color constancy convolutional autoencoder”, in 2019 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2019, str. 1085–1090.

[57]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.

[58]Qian, Y., Kamarainen, J.-K., Nikkanen, J., Matas, J., “On finding gray pixels”, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, str. 8062–8070.

[59]Laakom, F., Raitoharju, J., Iosifidis, A., Tuna, U., Nikkanen, J., Gabbouj, M., “Probabilistic color constancy”, in 2020 IEEE International Conference on Image Processing (ICIP). IEEE, 2020, str. 978–982.

[60]Shi, W., Loy, C. C., Tang, X., “Deep specialized network for illuminant estimation”, in Computer Vision – ECCV 2016, Leibe, B., Matas, J., Sebe, N., Welling, M., (ur.). Cham: Springer International Publishing, 2016, str. 371–387.

[61]Passalis, N., Tefas, A., “Neural bag-of-features learning”, Pattern Recognition, Vol. 64, 2017, str. 277–294.

[62]Bahdanau, D., Cho, K., Bengio, Y., “Neural machine translation by jointly learning to align and translate”, arXiv preprint arXiv:1409.0473, 2014.

[63]Krizhevsky, A., Sutskever, I., Hinton, G. E., “Imagenet classification with deep convolutional neural networks”, Advances in neural information processing systems, Vol. 25, 2012, str. 1097–1105.

[64]Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., Keutzer, K., “Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size”, arXiv preprint arXiv:1602.07360, 2016.

[65]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.

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

[67]Isola, P., Zhu, J.-Y., Zhou, T., Efros, A. A., “Image-to-image translation with conditional adversarial networks”, in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, str. 5967-5976.

[68]Gijsenij, A., Gevers, T., van de Weijer, J., “Improving color constancy by photometric edge weighting”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, No. 5, 2012, str. 918-929.

[69]Li, J., Fang, P., “Hdrnet: Single-image-based hdr reconstruction using channel attention cnn”, in Proceedings of the 2019 4th International Conference on Multimedia Systems and Signal Processing, ser. ICMSSP 2019. New York, NY, USA: Association for Computing Machinery, 2019, str. 119–124, dostupno na: https://doi.org/10.1145/3330393.3330426

[70]Ronneberger, O., Fischer, P., Brox, T., “U-net: Convolutional networks for biomedical image segmentation”, in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Navab, N., Hornegger, J., Wells, W. M., Frangi, A. F., (ur.). Cham: Springer International Publishing, 2015, str. 234–241.

[71]Simonyan, K., Zisserman, A., “Very Deep Convolutional Networks for Large-Scale Image Recognition”, arXiv:1409.1556 [cs], Sep. 2014, arXiv: 1409.1556, dostupno na: http://arxiv.org/abs/1409.1556

[72]Tan, M., Le, Q., “EfficientNet: Rethinking model scaling for convolutional neural networks”, in Proceedings of the 36th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, Chaudhuri, K., Salakhutdinov, R., (ur.), Vol. 97. PMLR, 09–15 Jun 2019, str. 6105–6114, dostupno na: https://proceedings.mlr.press/v97/tan19a.html

[73]Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., “Mobilenets: Efficient convolutional neural networks for mobile vision applications”, CoRR, Vol. abs/1704.04861, 2017, dostupno na: http://arxiv.org/abs/1704.04861

[74]Das, P., Baslamisli, A. S., Liu, Y., Karaoglu, S., Gevers, T., “Color constancy by gans: an experimental survey”, arXiv preprint arXiv:1812.03085, 2018.

[75]Zhu, J.-Y., Park, T., Isola, P., Efros, A. A., “Unpaired image-to-image translation using cycle-consistent adversarial networks”, in 2017 IEEE International Conference on Computer Vision (ICCV), 2017, str. 2242-2251.

[76]Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., Catanzaro, B., “High-resolution image synthesis and semantic manipulation with conditional gans”, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[77]Park, T., Efros, A. A., Zhang, R., Zhu, J.-Y., “Contrastive learning for conditional image synthesis”, in ECCV, 2020.

[78]Lloyd, S., “Least squares quantization in pcm”, IEEE Transactions on Information Theory, Vol. 28, No. 2, 1982, str. 129-137.

[79]Leung, C.-K., Lam, F.-K., “Image segmentation using maximum entropy method”, in Proceedings of ICSIPNN ’94. International Conference on Speech, Image Processing and Neural Networks, Apr. 1994, str. 29–32 vol.1.

[80]Otsu, N., “A threshold selection method from gray-level histograms”, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, str. 62-66.

[81]Chaurasia, A., Culurciello, E., “Linknet: Exploiting encoder representations for efficient semantic segmentation”, in 2017 IEEE Visual Communications and Image Processing (VCIP), 2017, str. 1-4.

[82]Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S., “Feature pyramid networks for object detection”, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017, str. 2117-2125.

[83]Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., “Pyramid scene parsing network”, in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, str. 6230-6239.

[84]Hu, J., Shen, L., Sun, G., “Squeeze-and-excitation networks”, in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, str. 7132-7141.

[85]Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., “Imagenet: A large-scale hierarchical image database”, in 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, str. 248–255.

List of Figures

List of Tables

Biography

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.

List of publications

Publications in scientific journals

1.Domislovi ć, I., Vršnak, D., Subašić, M., Lončarić, S., "One-net: Convolutional color constancy simplified", Pattern Recognition Letters, Vol. 159, Srpanj 2022, str. 31-37

2.Domislovi ć, I., Vršnak, D., Subašić, M., Lončarić, S., "Shadows & Lumination: Two-illuminant multiple cameras color constancy dataset", Expert Systems with Applications, Vol. 224, Kolovoz 2023, str. 9

3.Domislovi ć, I., Vršnak, D., Subašić, M., Lončarić, S., "Color constancy for non-uniform illumination estimation with variable number of illuminants", Neural Computing and Applications, Vol. 224, Ožujak 2023, str. 31-37

4.Vršnak, D., Domislovi ć, I., Subašić M., Lončarić S., "Autoencoder-based training for multi-illuminant color constancy", Journal of the Optical Society of America A, Vol. 39, 2022, str. 1076-1084

5.Vršnak, D., Domislovi ć, I., Subašić M., Lončarić S., "Illuminant segmentation for multi-illuminant scenes using latent illumination encoding", Signal Processing: Image Communication, Vol. 108, Listopad 2022, str. 116822

6.Vršnak, D., Domislovi ć, I., Subašić M., Lončarić S., "Framework for Illumination Estimation and Segmentation in Multi-Illuminant Scenes", IEEE Access, Vol. 11, Siječanj 2023, str. 2128-2137

Works in scientific conferences

1.Domislovi ć, I., Vršnak, D., Subašić, M., Lončarić, S., "Outdoor daytime multi-illuminant color constancy", 12th International Symposium on Image and Signal Processing and Analysis, 2021, str. 270-275

2.Domislovi ć, I., Vršnak, D., Subašić, M., Lončarić, S., "Filters & Lumination: Creating multi-illuminant images for computational color constancy", 8th International Conference on Machine Learning Technologies, 2021, str. 5

3.Vršnak, D., Domislovi ć, I., Subašić M., Lončarić S., "Illuminant estimation error detection for outdoor scenes using transformers", 2th International Symposium on Image and Signal Processing and Analysis, 2021, str. 276-281

Životopis

Ilija Domislović (Graduate Student Member, IEEE) stekao je zvanje prvostupnika računarstva (bacc. comp.) 2018. godine i zvanje magistra (mag. comp.) 2020. na Fakultetu elektrotehnike i računarstva, Sveučilište u Zagrebu. Trenutno je na doktorskom studiju u znanstvenom polje računarstva (tehničke znanosti) na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu, Hrvatska. Njegovi istraživački interesi uključuju obradu slike, analizu slike, neuronske mreže i postojanost boja, s fokusom na procjenu osvjetljenja.