Master in Computational Colour and Spectral Imaging (COSI)

Deep Learning-Based Illumination SPD Estimation from an RGB Camera

Master Thesis Report

Presented by

Dong Han

and defended at the

Norwegian University of Science and Technology

September 2022

Academic Supervisor(s): Dr. Philippe Colantoni, and Dr. Éric Dinet

Jury Committee:

1. Prof. Ivar Farup, Norwegian University of Science and Technology, Norway
2. Dr. Mika Flinkman, University of Eastern Finland, Finland

Submission of the thesis: 10$^{\text{th}}$ August 2022
Day of the oral defense: 2$^{\text{nd}}$ September 2022

Abstract

Lighting is fundamentally crucial for numerous useful applications. The extended reality system is capable of simulating realistic scenes with precise illumination data. However, it is difficult to consistently obtain the scene's actual lighting SPD. Instead of predicting the entire spectrum of illumination SPD, the majority of known approaches for estimating illumination focus on recovering the illumination's color using spectral images. This work explores the problem of illumination SPD estimation from sRGB images. By discarding global spatial information, we convert the problem into a vector-to-vector regression task. The deep learning model is proposed to predict the illumination SPD using only the pixel values from sRGB images. In order to alleviate the lack of training SPD data, a large sRGB image dataset along with the corresponding lighting SPD is proposed in our work. The various unique illuminations are generated by the advanced 24-channel LED lighting system, which is the first one in Europe. In order to tackle the time-consuming capturing, the virtual camera model is implemented as the data-generation tool to simulate images with a variety of lighting conditions. As a result, the proposed datasets include both real-world and synthetic data for training and evaluating the model. The results of both datasets show consistent performance across a wide range of spectra according to the different numerical evaluation metrics. In the end, the sensor-independent prediction is explored by integrating the transfer learning technique when working on data from the different cameras.

Acknowledgment

I would first like to thank my thesis supervisors Dr Philippe Colantoni and Dr Éric Dinet of the Laboratoire Hubert Curien at Université de Lyon, Université Jean Monnet. They have provided all the equipment needed for this project. Besides, I have been inspired to have more ideas every time after having a discussion with Dr Philippe Colantoni. He also helped develop the data collection tools to make automatic image capture possible.

I wish to show my appreciation to Prof Alain Trémeau and Dr Damien Muselet of the Laboratoire Hubert Curien at Université de Lyon, Université Jean Monnet. They have given valuable suggestions about the deep learning part during the meeting.

I am grateful for the effort of Dr Seyed Ali Amirshahi of the Department of Computer Science at Norwegian University of Science and Technology. As COSI coordinator, he has made significant contributions to COSI administration and dissertation management.

Finally, I must express my gratitude to my parents for their unwavering support and consistent encouragement during my years of education, research, and thesis writing. And thanks to my classmates and friends, they provided me with countless care and encouragement, as well as making my university life full of warmth and happiness. This accomplishment would not have been possible without them.

Acronyms

XR	Extended Reality
VR	Virtual Reality
MR	Mixed Reality
NDT	Neutral digital transformation
DSLR	Digital single-lens reflex
SPD	Spectral power distribution
ISP	Image Signal Processing
DSCs	Digital still color cameras
CFA	Color filter array
CCD	Charge-coupled device
CMOS	Complementary metal–oxide–semiconductor
WB	White Balance
HVS	Human vision system
SVR	Support vector regression
DOCC	Double-opponency color constancy
SVR	Support vector regression
CNN	Convolutional neural network
DNN	Deep neural network
ADU	Analog-to-digital unit
QE	Quantum efficiency
CST	Color space transformation

HDR High dynamic range

RMSE Root Mean Square Error

GFC Goodness-of-fit coefficient

SAM Spectral angle mapper

SID Spectral information divergence

FWHM Full width at half maximum

FC Fully connected

PCA Principal component analysis

MSE Mean square error

ROI Region Of Interest

TL Transfer learning

Contents

1 Introduction

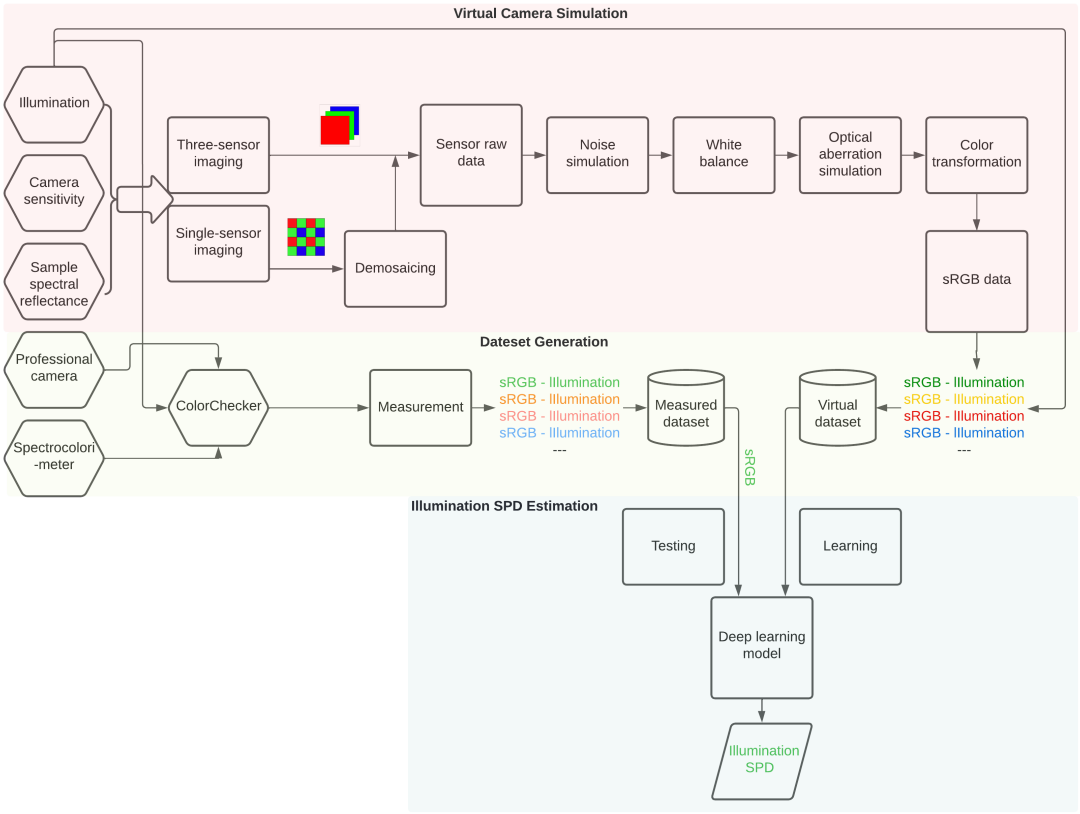


Figure 1.1: *Virtual camera simulation 3, dataset generation 5 and illumination SPD estimation 4 framework.*

Lighting conditions vary spatially and temporally in real life and it significantly influences the real-time perception of people, especially for low-vision patients. The objective of the work is to carry out research to understand how light affects or can change the visual perception of visually impaired people. The main scientific challenge that needs to be addressed is to have a chance to develop adaptive

and smart lighting systems to assist the visually impaired. To overcome such an obstacle in the medium term, extended reality (XR) technologies and the modern multiple-channel lighting system give the possibility for the digital simulation of low vision in order to investigate the impact of different lighting conditions. Such an approach would make it possible to dynamically create visual deformations corresponding to well-identified pathologies likely to evolve over time.

Digitally simulating low vision needs to tackle the question of the observer experience, i.e., how to set up an XR system to transform the perceived scene while preserving its realistic dimension. The selected device must be sufficiently immersive to be able to efficiently deceive the observer's brain. As smartphones are not able to provide such a characteristic, the virtual reality (VR) headset associated with an external stereoscopic camera positioned at eye level is a good option to explore. The main advantage of such a technological choice is that this type of device already exists and is already used to simulate mixed reality (MR) glasses.

Then, instead of using the standard imaging process: Light+objects > Eyes > Brain, the selected device will be set up to have the process: Light+objects > Stereoscopic Camera > VR Headset > Eyes > Brain. It will be possible to digitally modify, in real time, the images provided by the stereoscopic camera before displaying them in the immersive VR headset. The demonstration of those two processes is shown in Figure 1.2. Here, the ColorChecker is used as the object for illustration. Due to distinct imaging processes, there exists a visual distinction in perceived object images.

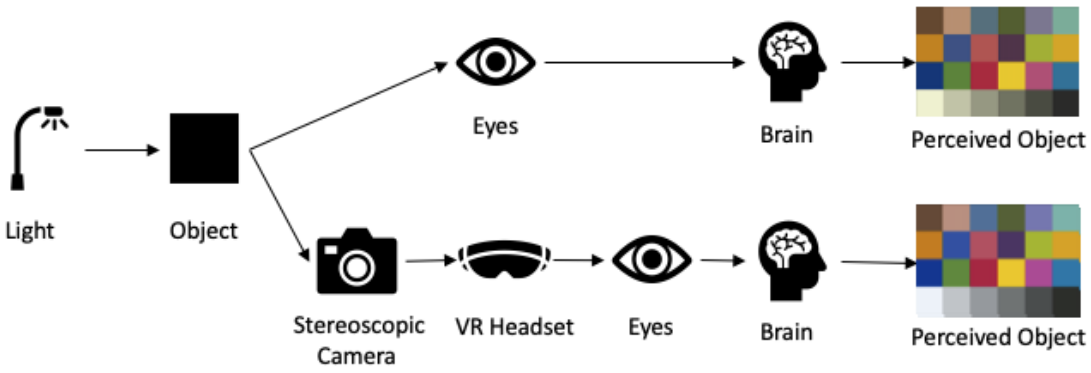


Figure 1.2: *Standard and MR imaging process.*

The first important task that needs to be solved is to ensure the visual similarity between the scene observed by human eyes and the same scene captured and displayed by the stereoscopic camera as well as the VR headset in any lighting condition. This can be achieved by developing a neutral digital transformation (NDT) that processes the image pixels obtained from the stereoscopic camera sensor

and transfers them to the VR headset.

For the implementation of NDT and the processing that will be implemented, the illumination spectrum of each lighting condition is more important to know than just the illumination color. The spectrum of illumination can be measured by the lighting measurement device. However, it is not practical and time-consuming when there are a large number of them. Specifically, we will use the 24-channel LED lighting system. Even though only the 17 channels are in the visible range, it can still produce an infinite variety of lighting by adjusting each channel individually. Therefore, we are searching for a method that recovers the current illumination information only based on pixel value from the captured image to avoid doing the measurement using any extra device. However, the stereoscopic camera dedicated to the VR headset is a "black box", which means that the camera image signal processing (ISP) pipeline is not available. In the meantime, the sensor quality is not that high compared with digital single-lens reflex (DSLR) cameras. Therefore, the simulation of this black box is a good way to understand how the image is computed and how the illumination information is presented in the image.

Based on the above discussion, the entire workflow is proposed in Figure 1.1, the virtual camera model can be implemented in order to approximate the ISP which includes basic image processing modules. The input elements of the virtual camera are illumination, camera sensitivity and sample spectral reflectance, while the output is the sRGB image. The illumination can be simulated as virtual lighting in combination with spectral power distribution (SPD) from each LED channel provided by Telelumen company. The camera sensitivity is acquired from well-known academic open sources. The ColorChecker with pre-measured reflectance is selected as the object sample. By using the virtual camera, we are able to generate synthetic sRGB images in a variety of simulated lighting conditions. One of the advantages of the virtual camera is the ability to produce enormous images within a short period of time. Those images and corresponding illuminations can be organized as a virtual dataset. The deep learning model can learn the relationship between sRGB image and lighting conditions in the form of SPD by training on this virtual dataset. Furthermore, the deep learning model can provide us with a way to predict and estimate the SPD of illumination from the sRGB image captured under unknown lighting conditions. After verifying the estimation model with the virtual camera on the virtual dataset, we can also use the spectrocolorimeter to measure the light directly generated by the lighting system and capture the real image by using the professional camera to acquire the measured dataset. Then, the illumination estimate model can be evaluated in a more practical scenario.

The main objective of this work is to solve the lighting SPD estimation problem only based on sRGB information. The goal is achieved through the three main parts including virtual camera simulation, dataset generation and illumination SPD

estimation as depicted in Figure 1.1. The virtual camera model is implemented to generate the synthetic images. The virtual dataset and measured dataset are generated as SPD database. The 24-channel LED lighting system is utilized to generate diverse natural and artificial illuminations. The deep learning approach is developed for estimating illumination SPD and assessed using several quantitative measures. It is also used to perform some initial tests with transfer learning techniques.

All the ISP pipeline fundamentals, illumination color and SPD estimation are introduced in chapter 2. The implementation details of the virtual camera model based on ISP are explained in chapter 3. In chapter 4, the illumination estimation workflow is proposed after the testing of the different deep learning models on several datasets. Chapter 5 describes the captured dataset configuration and the capturing procedure. Then, several experiments are conducted based on the proposed datasets and the results are presented in chapter 6. The final conclusion part is provided in chapter 7.

2 Background

This chapter discusses the traditional ISP pipeline that is frequently used in the commercial camera and the current illumination estimation approaches for illumination color and SPD. In contrast to traditional machine learning, the requirements for vast amounts of data are particularly crucial in deep learning. Synthetic data becomes an important element of the training process when real data is expensive and time-consuming to generate. The synthetic images under different illumination conditions can be simulated through the ISP pipeline and used for training the illumination estimation model.

2.1 Camera Image Signal Processing (ISP) Pipeline

The original simple camera model implies that the image is a quantitative lighting measurement representing scene radiance, and that the camera itself is a light-measuring instrument. Modern commercial digital cameras can no longer be seen as devices for measuring illumination because they are intended to capture and produce aesthetically pleasing images. With the assistance of the ISP, a number of image processing steps can be performed in real time Fukushima et al. (1983). The ISP pipeline seen in Figure 2.1 is a crucial component of a digital camera that performs image processing from the sensor's raw data to a final rendered image.

2.1.1 Image Formation

For generic imaging aspects in terms of sensor, aperture and lens for digital still color cameras (DSCs), at least three bands need to be measured for each pixel position in order to have color image in the final. One common solution is to places a color filter array (CFA) on top of camera sensor instead of using multiple sensors. There are different CFA patterns such as Bayer CFA Bayer (1976), Yamanaka CFA Yamanaka (1977) and HVS-based CFA Parmar and Reeves (2004). However the

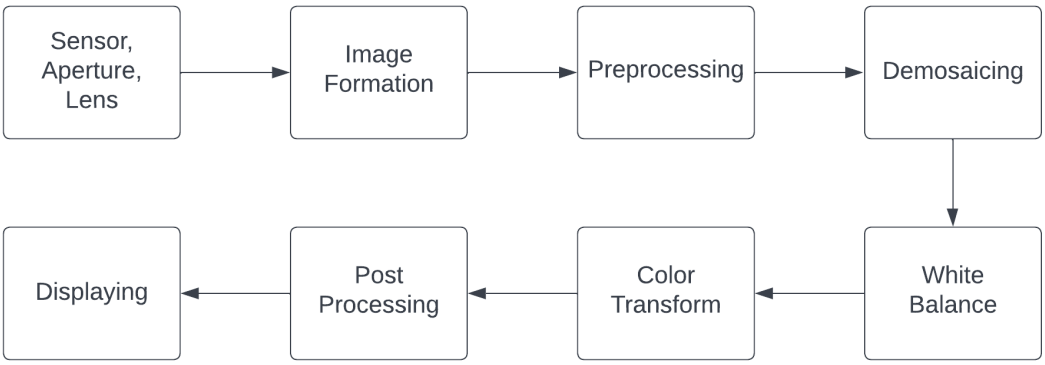


Figure 2.1: *Typical ISP pipeline.*

Bayer CFA and its variants are the most used so far. The CFA affects the control of each pixel's exposure time when changing aperture size, shutter speed and focus of the lens. Besides the CFA, normally an infrared cut filter is also mounted on the surface of sensor to exclude infrared light for avoiding extra heating. The exposure control is based on captured energy of the sensor to adjust the aperture size or shutter speed to make the image well exposed. The focus control needs the information of the object distance which typically measured by infrared light with an extra device or derived from estimation algorithm Krotkov (1988) such as spatial frequency analysis Choi et al. (1999) and phase detection Hamada (2015). The initial recorded image through sensor with CFA is actually the electrical energy converted from lighting falling on the sensor. The Figure 2.2 shows how the sensor with CFA "saw" the image. Theoretically, for each pixel in CFA has one filter which only allows one type of light to pass. Therefore, for each pixel in the CFA raw image, there is only one single value from one of the RGB filters. On the other hand, the sensor receives only a gray-level image before any further steps.

2.1.2 Preprocessing

The raw image captured from the camera sensor includes noise and other artifacts, which may cause the certain image defects that affect the image quality after further color processing. The defective pixel correction is one of the common preprocessing steps. The defective pixels result from defective photo-elements in sensor. Those defective pixels are eliminated by averaging with defective pixels Smith (2005) or interpolating green pixels with known the red and blue values Rashkovskiy and Macy (2001) among the surrounding pixels. The dark current compensation is important step for CCD and CMOS camera sensors since the dynamic range

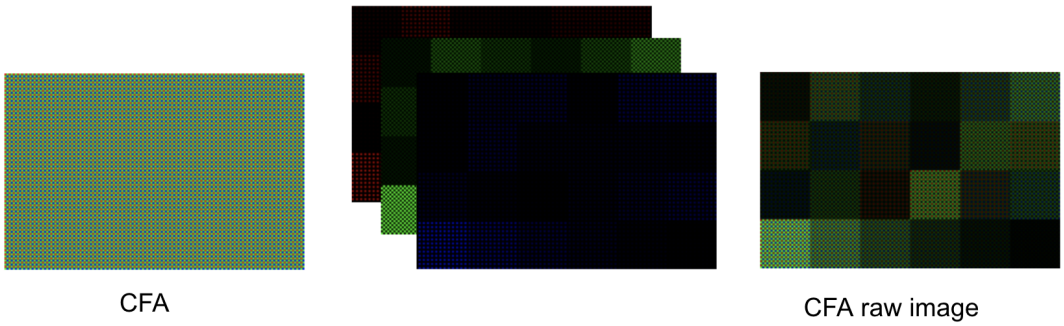


Figure 2.2: *What sensor "saw" ?*

and sensitivity of the sensor are significantly affected by the dark current noise, especially for low-light imaging use cases. The simple way to reduce dark current effect is to substract the dark image value from the captured image or design the specific scheme based on hardware working principle of photodetector Beaudoin et al. (2009). When the light source is too bright, the flare might be observed in the image which is caused by strong scattering and reflecting light. The flare compensation is to reduce the nonuniformity contributed by flare light. The naive approach is to substract the mean signal energy from the captured image. Since the flare also affects the colorfulness and contrast of image, the adaptive approach Kim et al. (2006) is able to use the measured flare to compensate for the flare and chroma at the same time.

2.1.3 Demosaicing

After obtaining the CFA image, the process of estimating other channel values is needed for computing the RGB color image. The demosaicing operation interpolates the missing channel information based on the neighbor pixel information. The demosaicing performance will affect the image quality in other process steps later. Some image artifacts might be introduced depending on how the demosaicing algorithm performs. The classical demosaicing methods are based on color difference interpolation. The nearest neighbor interpolation Adams Jr (1995), cubic splines interpolation Hou and Andrews (1978). The color difference interpolation method is simple and easy to implement, but it ignores the correlation among channels and the details of edge structure. The reconstruction results often have defects such as color artifacts, zipper effects and blurring. It is suitable for reconstructing relatively smooth image types. Compared with traditional methods, the reconstructed image quality of the learning-based methods is further improved. CNN-based two-stage

demosaicing network Tan et al. (2017), CNN-based three-stage demosaicing network Cui et al. (2018), end-to-end residual network Zhang et al. (2019) and pyramid attention networks (PANet) Mei et al. (2020). The learning-based methods have surpassed most traditional methods in terms of reconstruction accuracy, but the high computational cost and long running time is still a problem. Therefore, the classical interpolation approaches are still mainly used due to their good performance and simplicity to being integrated with sensor systems.

The demosaicing artifacts are observed after demosaicing the CFA image. The most common two are the zipper effect which indicates the color changing along the edge and false color which results in noticeable color errors. Those two artifacts are mainly due to the incorrect interpolation direction. Normally, the interpolation artifacts still exist because of the limitations of interpolation itself even if the interpolation direction is correct. There are several approaches to reduce the artifacts such as spectral correlation Gunturk et al. (2002), spatial correlation Cok (1987), and green-plane-first rule Adams Jr (1997). The trade-off between the complexity of demosaicing algorithm and the quality of the demosaiced image depends on the application.

2.1.4 White Balance (WB)

Color constancy is the ability to accommodate different light spectral content to perceive the same or similar color when the illumination changes. Human vision has the ability to recognize the same object's colors under different illumination conditions due to color constancy. However, the camera sensors do not have this ability. Therefore, the computational color constancy is achieved by the WB. The WB algorithm estimates illumination color and then compensates RGB gain to correct images by discarding the effect of illumination. In general, for traditional methods, the WB correction operation is based on a linear diagonal matrix (with size 3×3). The WB is carried out early in the ISP pipeline. It can be done before or after the demosaicing process. The image after WB is still in linear sensor raw space.

The process of estimating the global illumination has received a lot of attention from researchers. The traditional WB methods are based on statistics like Grey-World Buchsbaum (1980), White-Patch Funt and Shi (2010), Shades of Grey Finlayson and Trezzi (2004) and Grey-Edge Gijsenij et al. (2007). The learning-based methods offer more accurate results by relying on training examples like end-to-end deep learning network Stanikunas et al. (2004), shallow CNN for end-to-end illumination estimation Hernandez-Juarez et al. (2020) and contrastive learning on raw image Lo et al. (2021). The idea of a single global illuminant is an oversimplification, as many settings contain many light sources. There are

some works focus on deal with multi-illuminant scenes such as CNN-based network for outdoor daytime illumination Domislović et al. (2021), pixel-wise DNN-based WB algorithm Kim et al. (2021), DNN-based network learning from different WB setting images Afifi et al. (2022).

2.1.5 Color Transformation

The color transformation includes the transformation from the camera raw image to device-independent space and the rendered image is computed after converting into device-dependent space. The device-independent color spaces are so named because they are not intended for output but rather for storage or conversion medium. The sensor captures data in the camera's color space, which has little to do with colorimetric values. In order to attain colorimetric precision while capturing the image, it may be necessary to convert the image from the sensor's spectral space to the CIEXYZ color space Ohno (2000) which is a device-independent space. This can be done with a linear transformation to approximate the mapping function from camera raw color value to the CIEXYZ value.

From CIEXYZ color space, the visually appealing image that is eventually encoded in a standard RGB color space can be obtained from the colorimetric conversion and further photofinishing techniques. However, it is a lossy process since the rendered image is limited to the 8 bit dynamic range required by most displaying media. sRGB Commission et al. (1999) is one of most popular used rendered space. It is defined as a transformation from device-independent color space (mostly from CIEXYZ color space), the final sRGB is transformed from this perceptual color space. sRGB value obtained after the matrix mapping operation, it is still in linear space. The gamma correction adds the nonlinearity to the final sRGB.

2.1.6 Post Processing

After all of the above-mentioned image processing steps, various camera manufacturers employ a variety of unique methods to improve the image look from an aesthetic perspective or for specific usage. Each of the above stages may introduce undesirable artifacts, making postprocessing required. For example, the demosaicing procedure may generate a zipper artifact along strong intensity edges. Removal of color artifacts, improvement of edges, reduction of noise and enhancement of contrast are frequent postprocessing processes. These strategies are mostly dependent on heuristics and need extensive adjustment.

2.2 Illumination Estimation

Estimating illumination from a color image of an object is a challenging task since there is an ambiguous nonlinear relationship between illumination spectrum and pixel value. And it is an ill-posed problem due to the color constancy effect, the same color might resulting from different illuminations. The illumination information of a image is incorporated in spatial layout and color distribution Lombardi and Nishino (2012). In terms of illumination information, it can be represented as illumination color or SPD.

2.2.1 Illumination Color Estimation

To recover the illumination color, it is usually related to color constancy since the crucial part of the algorithm is to estimate the illumination color. Illuminant color estimation methods can be categorized into two classes: statistics-based and learning-based methods. Statistics-based methods estimate the illuminant based on the statistical properties (e.g., reflectance distributions) of the image. Learning-based methods learn a model from training images and then estimate the illuminant using the model. Statistics-based methods include Grey-World Buchsbaum (1980), White-Patch Land (1977), Shades-of-Grey Finlayson and Trezzi (2004), Grey-Edge Van De Weijer et al. (2007) based on image grayscale and Grey-Pixels Yang et al. (2015) and other methods like local reflectance based Gao et al. (2014), double-opponency color constancy (DOCC) Yang et al. (2015) and Retinal-Mechanism based Zhang et al. (2016). This kind of method estimates the light source through the assumption of statistical invariance of scene reflectance or human vision system (HVS). The light source is estimated through a predetermined parameter configuration in fast computation without training data. Among them, the method of simulating HVS has more sufficient theoretical basis, better accuracy and more practical in other applications. Learning-based methods include support vector regression (SVR) Xiong and Funt (2006), Gamut-Mapping Gijsenij et al. (2010), nearest neighbor method Joze and Drew (2013) and other methods that build the machine learning models mainly based on key features such as the number of colors, brightness and chromaticity distribution, and texture of images to achieve light source color estimation. With the amount of dataset available, the neural networks based model has high accuracy Cardei et al. (2002) such as deep CNN model Lou et al. (2015), Deep Specialized Network (DS-Net) Shi et al. (2016) and bounding illumination solution Koščevic et al. (2019).

2.2.2 Illumination SPD Estimation

However, the color constancy algorithm is not able to recover the SPD of illumination but rendered illumination color. For SPD estimation, the existing proposed methods also have two different types: statistics-based and learning-based. The statistics-based methods include basis functions modeling optimization for projector Hidaka et al. (2020), spectral gray-world, max-spectral, spectral shades of gray and spectral gray edge by extending classical computational color constancy methods for multispectral imaging Khan et al. (2017) and Wiener filter estimators based on sensor-space value from multispectral camera Kitanovski et al. (2021). Most learning-based methods are based on spectral reflectance information and measured SPD of light coming from white reference. The unsupervised learning method proposed in Nieves et al. (2008) directly estimates the SPD of illumination from the camera digital counts that selected from bright areas of natural scenery images by using a three-sensor camera. The CNN-based method for pixelwise illumination estimation Robles-Kelly and Wei (2018) predicts the illumination of the center position of image and obtains the whole spectra by bicubic interpolation. In order to solve the multiple illuminants problem, a DNN-based method is presented in Li et al. (2021), the model is trained on synthesized dataset with single and multiple illumination. For the most current methods mentioned above, the used SPD data is normally measured from the light source of the lamp or the natural daylight. Even though most lighting conditions exist in real life, those acquired SPD have low diversity due to the channel number limitation of the light source device.

3 Virtual Camera Simulation

In this chapter, the implementation of virtual camera simulation as proposed in Figure 1.1 is presented. Based on traditional ISP pipeline, the modified virtual camera is proposed. The implementation of different core components of virtual camera are mainly discussed. From the sensor imaging where the sensor raw image computed, the simulated noise is added before the demosaicing and white balance processing. The optical aberration is then recreated to replicate the defect of a real camera. In the last section, the color correction process is proposed to achieve fast color transformation and the final rendered ColorChecker is also presented in sRGB image.

The virtual camera model is necessary because deep learning approaches typically demand large datasets for training. This is one of the difficulties for our aim since SPDs estimation model needs to be trained on various images with different illuminations. Even though it is possible to produce illuminations by using the 24-channel LED lighting system, the capturing of images and the measurement of SPDs under a number of different lighting scenarios are tedious and time-consuming work. Meanwhile, by employing virtual illumination, the spectrocolorimeter can be omitted because the SPD data is modeled rather than measured. According to the ISP pipeline, the virtual camera model can be implemented to take advantage of virtual lighting to simulate the images with the object reflectance. In this case, the real camera is also can be avoided. The virtual lights and simulated corresponding sRGB images can be used as a synthetic dataset to train the deep learning model for estimating the SPD of lighting. At this simulation stage, we are able to verify the estimation model without using any measurement or imaging devices.

3.1 Virtual Camera Implementation

3.1.1 Sensor Imaging

For general image formation model:

$$P(i,j) = \int I(i,j,\lambda)R(i,j,\lambda)S(i,j,\lambda)d(\lambda) \tag{3.1}$$

Each sensor raw pixel value P for an image is resulting from scene illumination I, object spectral reflectance R and camera sensitivity S within whole spectrum λ. For capturing the three-channel RGB, there are two types of sensor imaging techniques: three-sensor imaging and single-sensor imaging. For three-sensor imaging, three sensors are dedicated for red, green and blue channels, respectively, which has high expense and a limitation in system size. Therefore, the most color cameras usually use single-sensor imaging scheme. Normally, there is a color filter on top of the sensor surface to select the desired wavelength of light spatially in order to generate the color image later on. The main difference between single-sensor and three-sensor imaging is that the demosaicing algorithm is mandatory for the former one.

According to the matrix camera image modeling from Karaimer and Brown (2018), the three-sensor imaging model can be represented by formula:

$$Cam = S(i,j,\lambda)diag(I(i,j,\lambda))R(i,j,\lambda) \tag{3.2}$$

The camera sensitivity can be formed as matrix $S = [Cr, Cg, Cb]^T$ which denotes the R,G,B channel spectral response. The illumination information can be organised into a N×N (N bands) diagonal matrix where the illumination spectral is in diagonal position and zero anywhere else. Since the ColorChecker is used as an object, the color information is more interesting than spatial information. The spectral reflectance of the object can be reshaped to N×P (P patches).

However, considering singe sensor, the extra CFA selection matrix should be integrated into the general image formation model. Inspired from multispectral image, the optimized image formation equation is proposed in Sadeghipoor et al. (2012). The camera sensitivity matrix S can be seen as CFA filter spectral sensitvity. In order to incorporate the CFA selection matrix, the Kronecker product operation is done between CFA filter spectral sensitvity and identity matrix. However, the computational cost is high for the large image since the size of identity matrix depends on the total number of pixels within image. Besides, there is another limitation is that they did not include illumination spectrum in final computing but assuming the lighting power is uniform for every pixel.

Therefore, in order to reduce computational cost and include the illumination spectrum inside the single-sensor imaging in the formation model, the CFA selection matrix is proposed according to the bayer pattern. The ColorChecker will be mainly used to contain the illumination information in following experiment. Assuming each color patch has number of pixels m, firstly the bayer pattern is organized as $BP^{1\times m} = [rg...gb...]$, by using the bayer elements rg and gb. Then the CFA

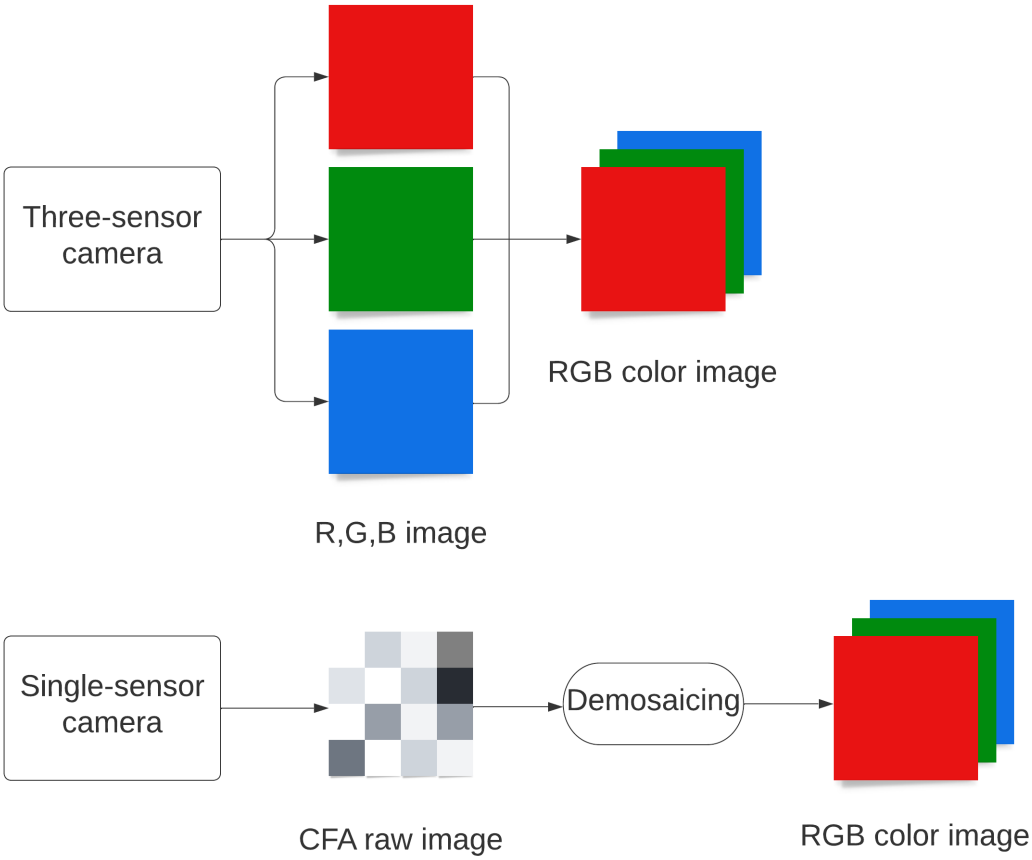


Figure 3.1: *Sensor imaging pipline.*

selection matrix $A^{m\times N}$ where each row represent one CFA filter, is computed according to the bayer pattern. In the end, the single-sensor imaging model is proposed as the following:

$$CamCFA^{m\times P} = A^{m\times N} diag(I)^{N\times N} R^{N\times P} \tag{3.3}$$

In the output matrix $CamCFA$, each column stores the CFA value for each color patch. Therefore, each column vector can be reshaped into size of $\sqrt{m} \times \sqrt{m}$ and P patches in total. The final CFA image can be concatenated with those patches. The test CFA raw image is shown in Figure 3.2. For the visual comparison of two different sensor imaging methods, the three-sensor imaging process addressed in the paper Karaimer and Brown (2018) is also implemented, and the result image is shown in Figure 3.3.

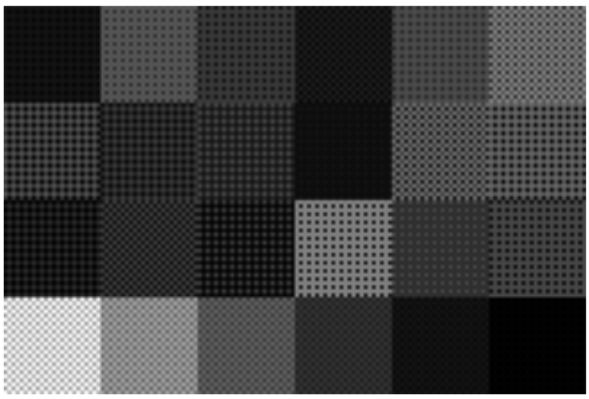


Figure 3.2: *CFA raw image computed with proposed method.*

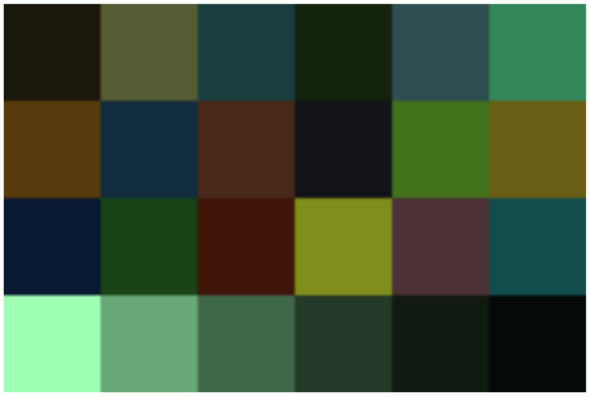


Figure 3.3: *Sensor raw image computed with the three-sensor imaging method following equation 3.2. The image shows in RGB for visulization.*

As previously stated, the single-sensor imaging technique is more commonly employed; hence, the subsequent steps will all be based on single-sensor imaging.

3.1.2 Noise Simulation

During the imaging procedure of the camera, noise is almost unavoidable. It is also not feasible to create a noise model for the camera. The input signal is obviously already noisy. The quantity of photons striking a pixel throughout the exposure duration varies statistically. The probability is Poisson distributed according to quantum mechanics rules Hu (2008). As a result, the variance of the fluctuations equals the mean number of photons. This noise, which is commonly referred to as shot noise (or photon noise Hasinoff (2014)), is determined by the fundamental

principles of physics and is the same for all different kinds of cameras. According to Figure 3.4, the camera sensor receives a number of photons and converts them into a number of analog-to-digital unit (ADU) as output. Unlike the shot noise (s), the dark noise (k) and quantization noise happen inside of the camera sensor.

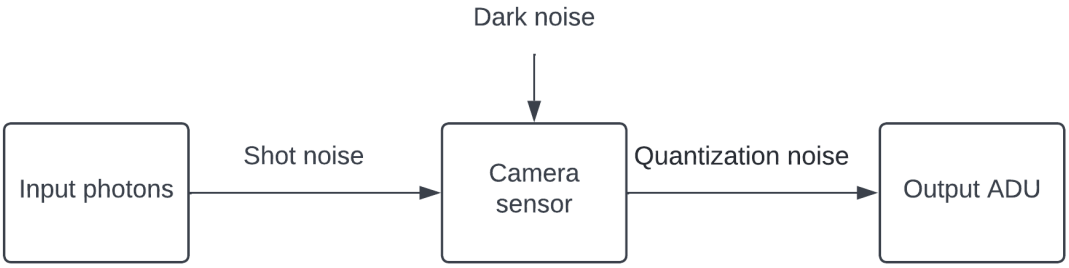


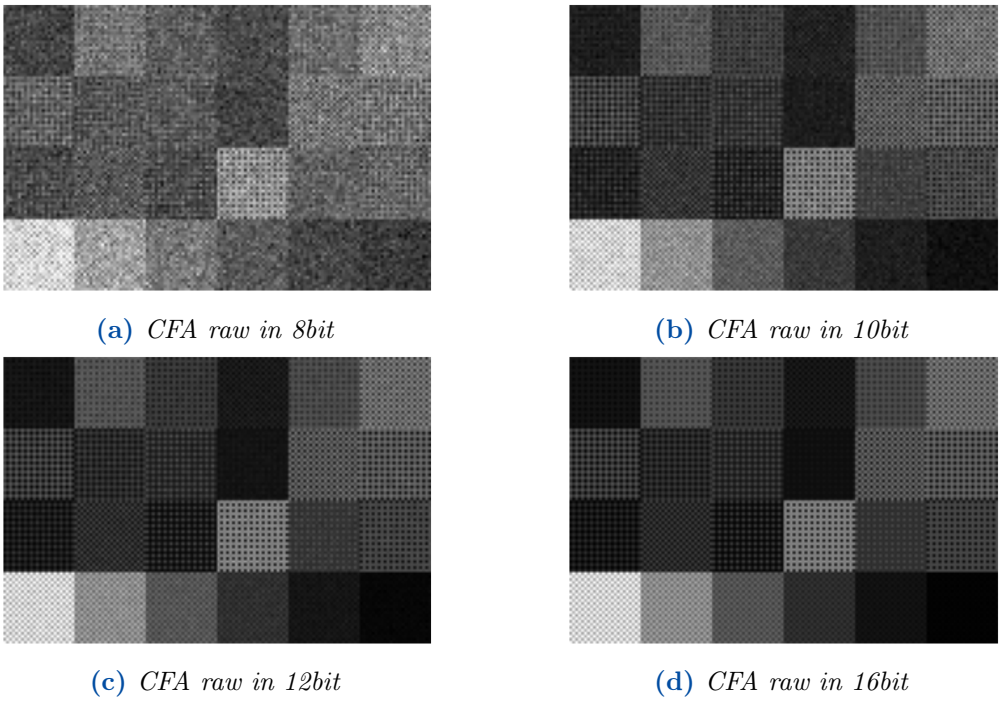
Figure 3.4: *Noise transformation*

Dark noise refers to the noise in the pixels when there is no light on the camera. There are mainly two sources of dark noise: readout noise and dark current. The dark current is not very significant in microscopy or machine vision. However, it is significant in astronomy due to the extremely long exposure lengths of several seconds or more, which allow large dark current electron accumulation during exposure. Therefore, when dark noise is primarily read noise, it can be modeled as a Gaussian distribution Boncelet (2009). Quantization noise indicates the raw value rounded to a nearby integer value when the analog voltage signal is digitalized. This rounding error is known as quantization noise or quantization error. In practice, this form of noise has a relatively minor impact on total noise.

In order to simulate the shot noise and dark noise, the number of input photons of each pixels are necessary Standard (2010). The CFA raw image that is computed by using a single-sensor imaging process can be seen as ADU. The electrons can be obtained by dividing each pixel in the CFA raw image by the conversion gain of the camera sensor (which is associated to camera sensitivity function). The next step is to convert the electrons to photons by dividing the electrons of each pixel by the quantum efficiency (QE) of the associated camera sensor. The photon to ADU formula can be summarized as:

$$ADU = Gaussian(k) + Poisson(s) \tag{3.4}$$

The terms $Gaussian(k)$, $Poisson(s)$ denote the dark noise that followed Gaussian distribution and shot noise in form of Poisson distribution. The total noise added to the ADU is in the form of a linear combination. The output ADU might need clipping since some value might be out of the final bit range.



(a) *CFA raw in 8bit* (b) *CFA raw in 10bit*

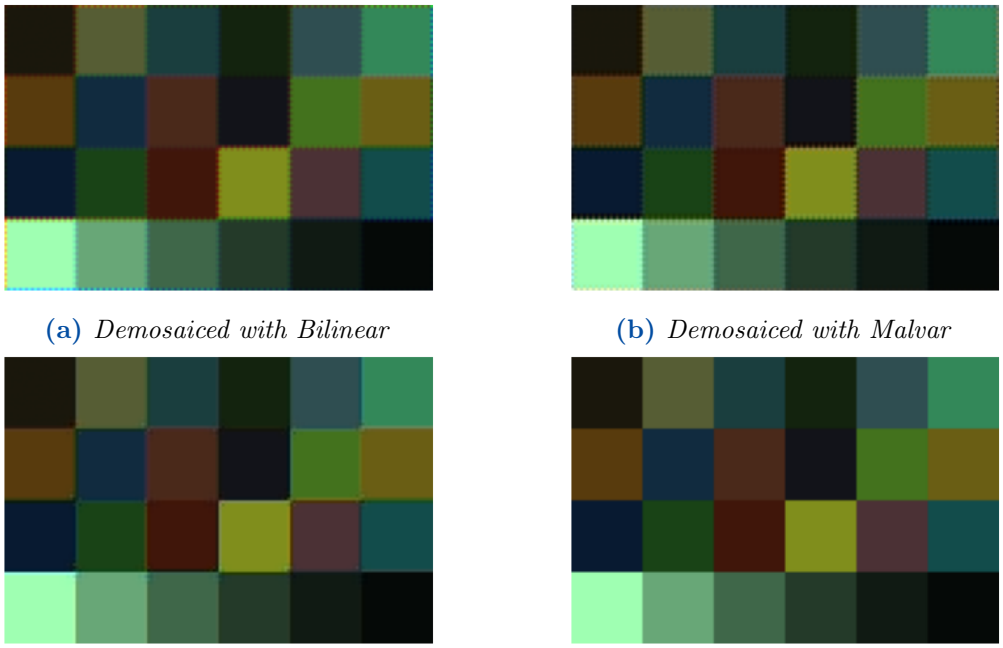
(c) *CFA raw in 12bit* (d) *CFA raw in 16bit*

Figure 3.5: *CFA raw image after adding the noise.*

The Figure 3.5 shows four different CFA raw noise images, each with a different bit range. The noise effect is strongly dependent on the precision (dynamic range) of the camera sensor. When applied to a range with 8 bits, the same quantity of noise has a relatively greater influence than when applied to a range with 16 bits. The quality of the image is adversely damaged by noise, particularly in situations in which the noise values are relatively significant in comparison with the pixel values.

3.1.3 Demosaicing Implementation

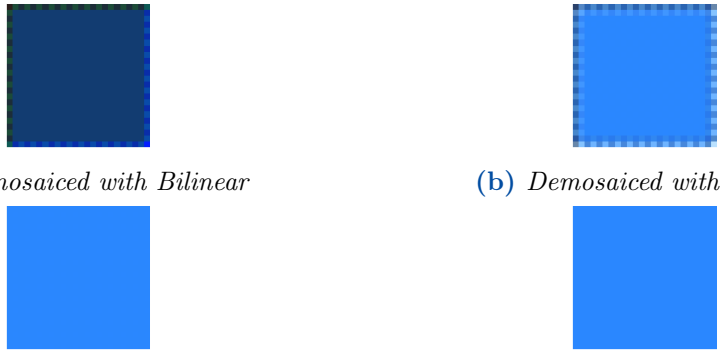
Since our main goal is not intended to propose a new demosaicing approach, the existing public demosaicing algorithm is enough for our implementation of tentative virtual camera simulation. Three different public well-used methods Losson et al. (2010), Malvar et al. (2004), Menon et al. (2006) are chosen. They are all linear demosaicing methods, Menon approach is selected as one part of the virtual camera model since it can reduce the visible artifacts to some extent, specifically for edge region. In Figure 3.6, the sensor raw image computed by three-sensor imaging can be seen as a ground truth demosaiced image 3.6d. The quite obvious zipper effect along with the edge of color patches is observed in 3.6a and 3.6b. The test illumination is fixed to D65 for the comparison.



(a) *Demosaiced with Bilinear* (b) *Demosaiced with Malvar*

(c) *Demosaiced with Menon* (d) *Ground truth*

Figure 3.6: *Demosaiced image with different methods.*



(a) *Demosaiced with Bilinear* (b) *Demosaiced with Malvar*

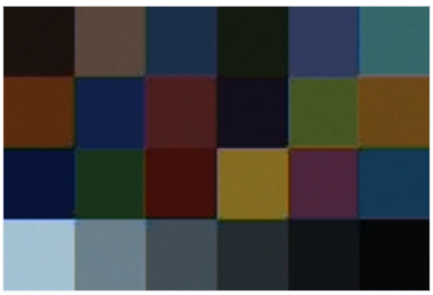
(c) *Demosaiced with Menon* (d) *Ground truth*

Figure 3.7: *Demosaiced single color patch with different methods.*

When running the demosaicing algorithms on a single color patch from the ColorChecker, the demosaicing artifact is readily apparent and easy to spot. In Figure 3.7, the certain edge effect are appeared in 3.6a and also in 3.7b.

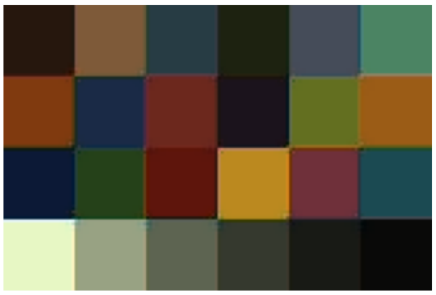
3.1.4 White Balance Implementation

For simplicity and robustness, statistical methods like Grey World, Shades of Grey and White Patch are used for our implementation. Since there is no strict order of demosaicing and WB, we conducted WB operation on demosaiced image. The appearance of WB raw images computed from different WB approaches shown in Figure 3.8 are quite different due to the properties of the used algorithm. The most noticeable effect of WB is shown in the hue of the white patch in ColorChecker. The bluish white and yellowish white is noticeable in white patches from Figure 3.8a and Figure 3.8c, respectively.



(a) *Grey world WB*

(b) *Shades of gray WB*



(c) *White patch WB*

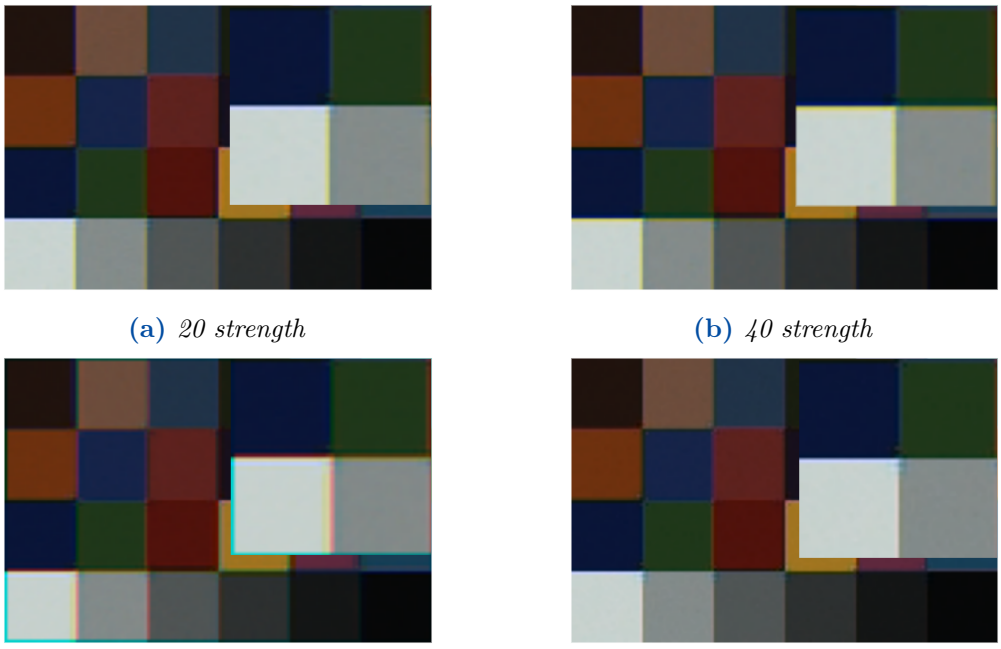
Figure 3.8: *White balanced image with different methods.*

The shades of gray algorithm (Minkowski norm $p = 6$) is selected as the main WB step for the virtual camera model since the competitive performance among other advanced colour constancy algorithm. Notably, the shades of gray algorithm uses the Minkowski norm p and the best results are obtained with $p = 6$ in paper Finlayson and Trezzi (2004). Therefore, in the virtual camera model implementation, the same value of p is maintained as the paper specified.

3.1.5 Optical Aberration Simulation

Besides the noise, camera sensors and lens systems have various optical distortions and aberrations which degrade the final rendered image. There are different types of optical aberrations such as spherical aberration Koomen et al. (1949), astigmatism Harris (2000), coma, vignetting Zheng et al. (2008), chromatic aberration Johnson and Farid (2006), distortion, and etc. However, we only focus on chromatic aberration by considering the feasibility of implementation without specifying the structure of the lens system of virtual camera.

Chromatic aberration is due to unequal magnifying of all image points across the whole spectrum. It is related to dispersion and the light near to blue spectral range is refracted more than the ones near to red spectral range, therefore the shift between them is noticeable. It has two kinds, longitudinal and lateral. The simulating longitudinal chromatic aberration of RGB image needs specific point spread function of lens to be applied to each color channel which is complex and has high computational cost while the simulating lateral chromatic aberration can be simulated by scaling each color channel to produce an increasing offset along the distance from the image center. Therefore, mainly the lateral chromatic aberration is simulated for our virtual camera model.



(a) *20 strength* **(b)** *40 strength*

(c) *80 strength* **(d)** *Without aberration applied*

Figure 3.9: *Chromatic aberration images with different strength.*

The figure 3.9 shows the effect which is brought by different levels of lateral chromatic aberration. Zooming in on the left bottom corner of the image, observing from the center, the white color patch shows red fringes at the side pointing to the image center and violet or blue fringes at the opposite side. Similar visual changes also happened at the edges of other color patches.

3.1.6 Color Transformation Implementation

For white-balanced raw RGB, the CIEXYZ can be computed as $CIEXYZ = T * RGB$, T is 3×3 matrix. This matrix is not fixed since the illumination condition is not always the same and not always in standard illumination. Therefore, from an academic viewpoint, each illumination condition has one unique transfer matrix. There are some optimization methods to solve this matrix. For example, researcher in paper Rowlands (2020) uses least square optimization to get preliminary solution then further minimizes the color difference by using nonlinear optimization technique to compute refined T. And they declare that the demosaiced raw RGB is needed instead of using white-balanced raw RGB. Therefore, the data used is the illumination SPD, demosaiced raw RGB and corresponding CIEXYZ value. The CIEXYZ value can be measured by a colorimeter or computed by using color matching function. However this is not practical, since the illumination condition always changes, the algorithm needs to be executed on each illumination. And CIEXYZ value is not always available. For most commercial cameras and smartphone cameras, the two characterized matrices are predefined by calibration in two certain standard CIE illuminants (A and D65, respectively). Once computing those matrices, any matrix T can be interpolated by them according to different illumination conditions. After getting the CIEXYZ from the sensor raw RGB, sRGB can be easily acquired.

In order to avoid the complex computing color space transformation (CST) matrix for each single illumination condition as we mentioned above, color correction is proposed and it has two phases shown in Figure 3.10. Firstly, the CST matrix is predetermined under standard D65 illumination by using the white balanced image and CIEXYZ values computed from the spectral reflectance of ColorChecker in the CST matrix computation process. Afterwards, the CST matrix is used for arbitrary illumination in the CST matrix applying process in order to generate the final output in sRGB space. The advantage of this method is that the CST matrix only needs to be calculated once for each sample and it is convenient to compute the image when there are a large number of different SPDs.

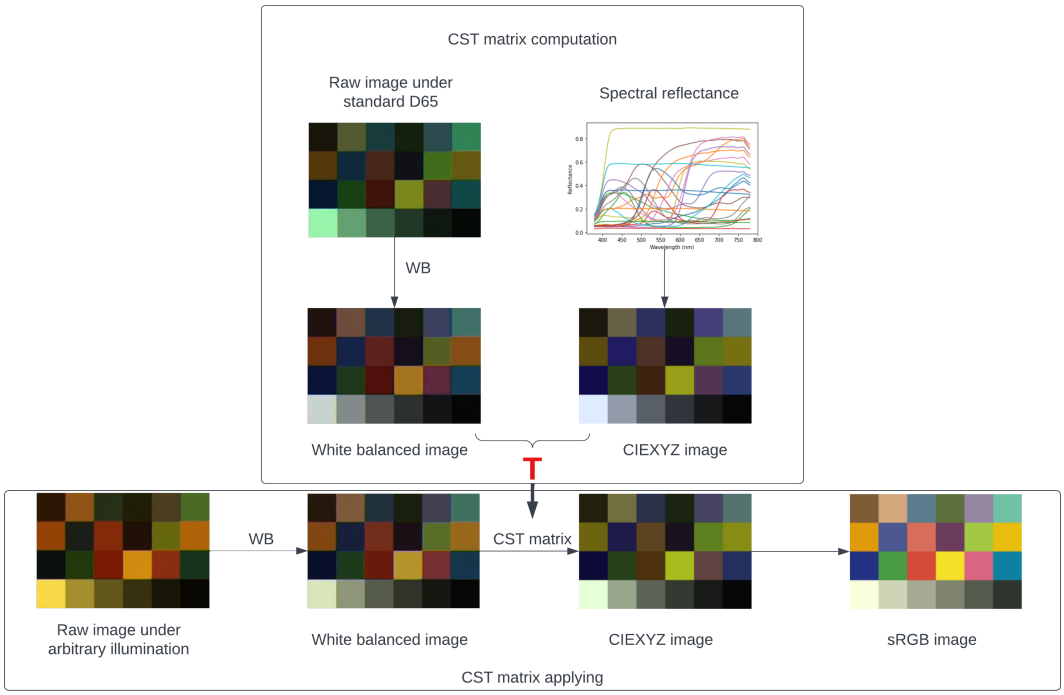


Figure 3.10: *Color correction process.*

4 Illumination Estimation in Deep Learning

In this chapter, the illumination SPD estimation part as stated in Figure 1.1 is mainly discussed. In specific, starting from the problem formulation section 4.1, two different potential deep learning models are presented in methodology section 4.2. As the final output result is the spectrum of the SPD, the different numerical metrics to evaluate the similarity of spectrum are introduced in the evaluation metrics section 4.3. The way to generate different lighting conditions and the content of synthetic images are in sections virtual lighting 4.4 and ColorChecker 4.5, respectively. Then, different deep learning models are tested and evaluated by conducting experiments based on synthesized image datasets in the model selection section 4.6. In the last section 4.7, the proposed framework for illumination estimation with a comprehensive deep learning network topology is presented.

4.1 Problem Formulation

As our goal is to reconstruct the whole illumination spectrum with only the sRGB image of ColorChecker, the spatial information of ColorChecker is not very relevant to illumination information due to the same spatial layout of all the images (the position and the order of each color patch remain the same all the time). For the purpose of the lighting estimation, the global spatial information is irrelevant, even for the training images with various contents Laakom et al. (2020). In addition, excluding global spatial information allows the network to extract illumination without global spatial inference, hence enhancing model resilience and ensuring illumination assumption adherence. Therefore, the color values of the ColoChecker have more contribution to containing the illumination information. The average pixel values of each color patch can be arranged as one single vector instead of all pixels because there is only a tiny variance within each color patch. It is obvious that the SPD is also in the form of a 1D vector with a relatively high dimension by considering the whole range of the spectrum and the small interval. In the end,

the problem can be regarded as a nonlinear vector-to-vector regression task.

4.2 Methodology

The vector-to-vector regression problem is becoming more and more popular in the machine learning domain. The classical regression methods such as linear regression Pearson (1896), support vector regression (SVR) Smola and Schölkopf (2004) and random forest have lower performance or fail when dealing with multi-output and nonlinear regression problems. The illumination estimation task can be considered as a regression problem since the output is the color values or the SPD of lighting. With the advance of neural networks, it is possible to represent an efficient mapping function in relation to a regression problem. Inspired by traditional universal approximation theory, the DNN-based vector-to-vector regression is proposed inside a machine learning framework, where DNN parameters may be modified using machine learning optimization methods. It is an efficient method for determining the underlying connections between vector inputs and their corresponding outputs and outperforms other approaches. There are illumination related works such as illumination color vector estimation from raw-RGB images based on DNN Afifi and Brown (2019); outdoor high dynamic range (HDR) illumination estimation from CNN Hold-Geoffroy et al. (2017) and lighting estimation with CNN for mixed reality Marques et al. (2022).

Moreover, the 1D CNNs have been proposed and achieved significant performance in many applications such as data classification tasks and detection tasks Kiranyaz et al. (2021). 1D CNNs can also be used for solving nonlinear regression problems and 1D signals are inherently compatible with 1D CNNs without any reshaping operation. On the contrary, when the data is in 2D format as for images, a 2D-to-1D conversion can be achieved in order to be compatible with 1D CNNs if the spatial information is not relevant for the task.

In the following experiments, the DNN and 1D CNNs deep learning models are mainly explored and compared for solving high-dimensional SPD estimation problem.

4.3 Evaluation Metrics

To quantitatively evaluate the performance of a model, four distinct types of metrics are utilized: root mean square error (RMSE) in Eq.4.1, the goodness-of-fit coefficient (GFC), spectral angle mapper (SAM), and spectral information divergence (SID). The ground-truth illumination SPD is normalized by using the maximum to divide each value so that the range is kept within [0-1]. The predicted illumination SPD

also has the same range in order to evaluate the relative errors. Parameters n, $f_i(\lambda)$ and $f_p(\lambda)$ denote the number of data points, the measured SPD and the predicted SPD for equations in this section.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_\lambda \Big(f_p(\lambda) - f_i(\lambda)\Big)^2} \tag{4.1}$$

GFC is defined as the cosine of the angle between the ground truth spectrum and predicted spectrum in Eq.4.2.

According to the interpretation of GFC in Romero et al. (1997): GFC $>$ 0.995 indicates the colorimetrically accurate illumination estimation; GFC $>$ 0.999 represents good spectral fit; GFC $>$ 0.9999 almost exact matching.

$$GFC = \frac{\sum_\lambda f_i(\lambda) f_p(\lambda)}{(\sum_\lambda f_i(\lambda)^2)^{1/2} (\sum_\lambda f_p(\lambda)^2)^{1/2}} \tag{4.2}$$

In some applications, researchers are more concerned with spectral shape differences than spectral amplitude differences. The spectral similarity measurement method based on projection can reflect the spectral shape difference to a certain extent, which is very important for the analysis of spectral data. SAM as in Eq.4.3 is one of the most typical projection-based spectral similarity measurement methods Kruse et al. (1993). The SAM value goes from 0 (very similar) to 1 (highly dissimilar).

$$SAM = \arccos\left(\frac{\sum_\lambda f_i(\lambda) f_p(\lambda)}{\sqrt{\sum_\lambda f_i(\lambda)^2} \sqrt{\sum_\lambda f_p(\lambda)^2}}\right) \tag{4.3}$$

Another typical spectral similarity measurement method is based on information measures and the representative approach is the SID as in Eq.4.4. The SID method was proposed in Chang (1999). This method transforms the spectral similarity evaluation problem into the redundancy evaluation problem between the probabilities of two spectral vectors. The higher SID value indicates the high dissimilarity of the spectrum. The $D(f_i(\lambda)||f_p(\lambda))$ is the relative entropy Lindley (1959) of $f_i(\lambda)$ with respect to $f_p(\lambda)$.

$$SID = D(f_i(\lambda)||f_p(\lambda)) + D(f_p(\lambda)||f_i(\lambda)) \tag{4.4}$$

4.4 Virtual Lighting

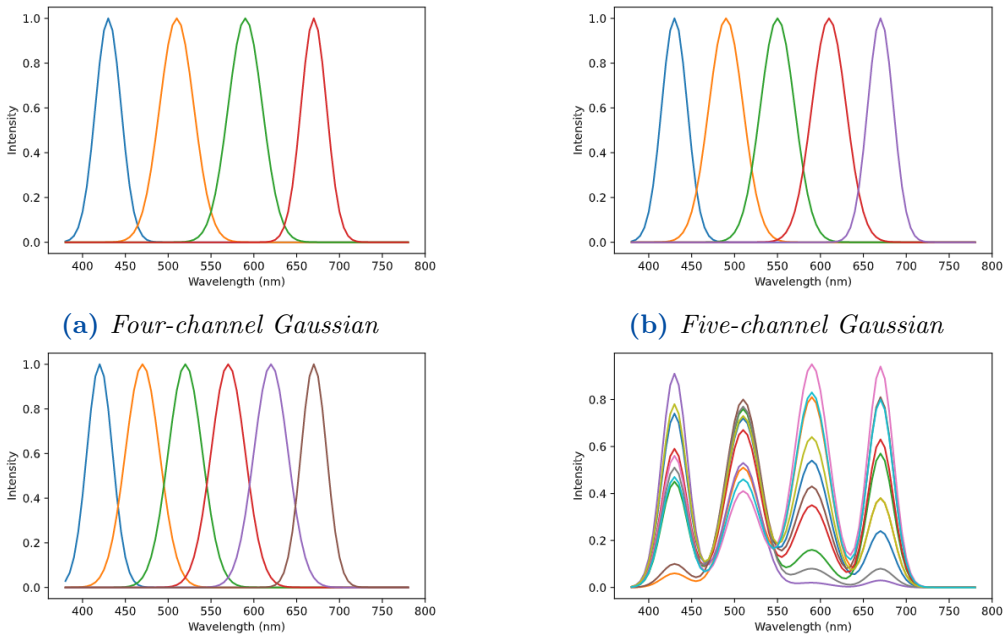
In order to generate the different lighting conditions without any real measurement, the SPD data of each LED channel provided along with the lighting system is used instead of producing the illumination by the lighting device itself. By assuming that

the output SPD from the lighting system can be approximated as an accumulation of the SPD from all LED channels, in this way, we can manipulate the weight for each LED and compute the simulated SPD by using a linear combination according to Eq.4.5.

$$SPDs = \sum_{i=1}^{n} w_i * I(\lambda)_i \tag{4.5}$$

where w_i (range from 0 to 1) and $I(\lambda)_i$ are the weight and SPD of each individual light channel, and n is the number of channels used to compute SPDs. The SPDs can be simulated by using multiple Gaussian distributions or by modifying the power distribution of each light channel from the existing lighting system to obtain various SPDs.

In Figure 4.1, the 4∼6 dimension Gaussian curves show here for demonstration. The wavelength range corresponds to the camera sensitivity which does not response after 700 nm and peak of each channel is set to one. The center wavelength of each channel and its full width at half maximum (FWHM) are adjusted to spread equally throughout the validated wavelength range. It is also possible to generate SPD with an even higher dimension Gaussian curve. The generated SPDs in Figure 4.1d are based on a 4-channel Gaussian distribution for illustration.



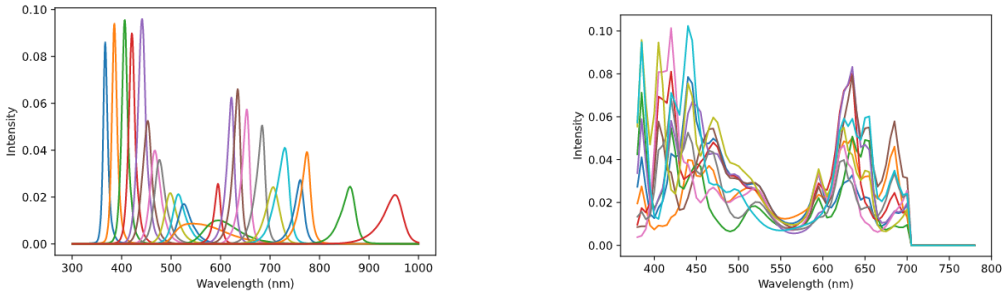
(a) *Four-channel Gaussian*

(b) *Five-channel Gaussian*

(c) *Six-channel Gaussian*

(d) *Generated SPDs*

Figure 4.1: *Simulated illumination from Gaussian modeling.*



(a) *LED SPD of lighting system* **(b)** *Generated SPDs*

Figure 4.2: *Simulated illumination from Telelumen LED light system.*

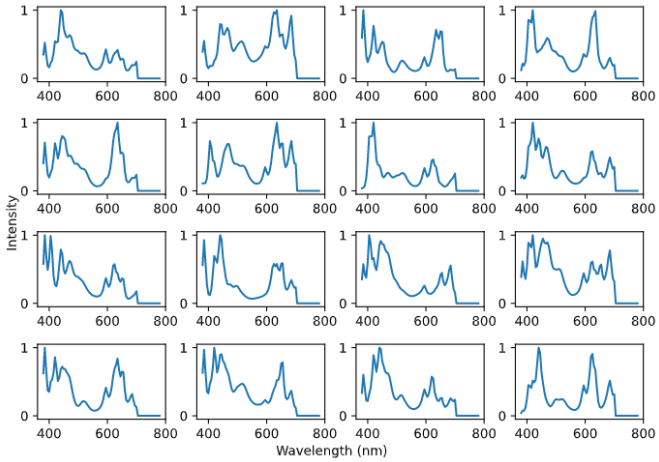
Although the generated SPDs by Gaussian can be a good approximation of simple lighting conditions, the SPDs that are generated from the SPD data of the 24-channel LED lighting system are more closed to the lighting SPD in real cases. It means that not all of the channels have the same shapes or peaks (show in Figure 4.2a) and the generated SPDs (show in Figure 4.2b) is not very smooth but with sharp peaks. The generated SPDs by Gaussian and by lighting system LED SPD data are all computed by assigning random weight (from zero to one) to each channel following the equation 4.5 as mentioned before.

4.5 ColorChecker

For the experiment samples, the Macbeth ColorChecker and the X-Rite ColorChecker Digital SG are chosen in order to make comparison as in Figure 4.3. For the latter, only the 96 out of 140 patches in the center are selected since the color patches contribute more information instead of gray-scale patches for illumination estimation. Generally, the more different color patches in the ColorChecker, the more illumination information can be stored. The Macbeth ColorChecker is mostly employed for generating simulated datasets for evaluating the tentative illumination estimating model. The ColorChecker Digital SG is mainly used as the object for real lighting capturing later.

The sRGB images of ColorChecker computed by the virtual camera under the different illumination conditions show in Figure 4.4. In this case, all the 24 channels are included.

Figure 4.3: *Rendered ColorCheckers by virtual camera under D65 illumination.*



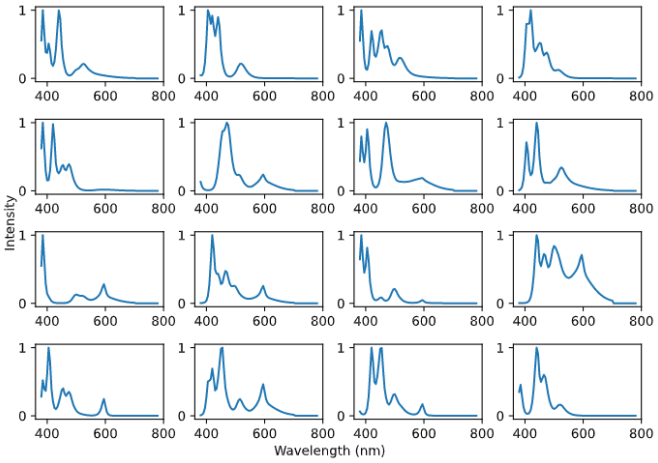
(a) *Simulated illumination*

(b) *Simulated sRGB*

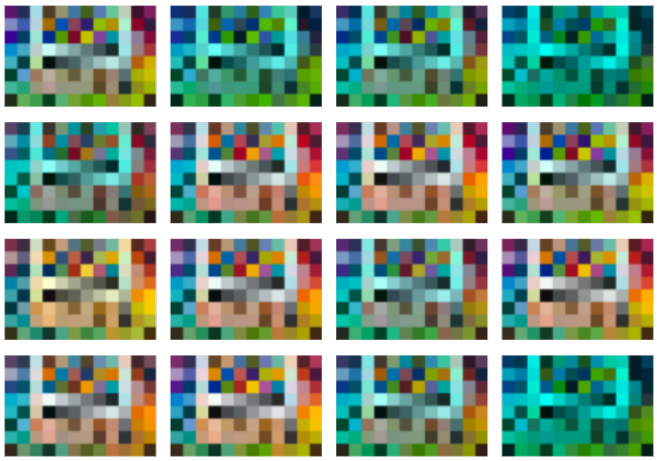
 Figure 4.4: *Simulated sRGB images with various random SPDs based on 24 channels.*

The difference among illuminations reflects quite small visual differences in sRGB images since the SPD gets contributions from all the channels. The white balance algorithm is able to tackle those types of illumination properly.

However, it is important to notice that not all the center wavelength of 24 LED channels are in visible wavelength range. We can disable the UV and IR LED channels by setting corresponding weights equal to zero when those channels are not interested for the purpose. The Figure 4.5 shows the SPDs which only consist of 7 activated LED channels within visible wavelength range to make a comparison with Figure 4.4.



(a) *Simulated illumination*



(b) *Simulated sRGB*

Figure 4.5: *Simulated sRGB images with various random SPDs based on 7 channels.*

The images are quite visually different to each other due to the characteristics of lighting SPD, especially when the high intensity of SPD only appears within a short wavelength range. E.g, the simulated illuminations in the first row of Figure 4.5a all have a relative high intensity from 400 nm to 500 nm but nearly zero response after 600 nm. This might be the reason why the corresponding simulated sRGB in 4.5b shows bluish and greenish colors. In those cases, the white balance is not working very well on those monochromatic-like illuminations.

4.6 Model Selection

In this part, several different experiments are carried out to test DNN and 1D CNNs methods on generated SPDs datasets from Gaussian and Telelumen LED data. The datasets are generated by the method introduced in section 4.4. In following experiments, the weight vector $[w_i]$ of each generated SPD is chosen as the model target variable instead of SPD itself due to the low dimension. And it is straightforward to recover the whole lighting spectrum with the predicted weight vector by Eq.4.5. Therefore, simulated sRGB images and the weight vectors can be used as the input-output pairs for training the deep learning model.

The width of the hidden layer of models are tuned on various simulated datasets in Table 4.1. The size of the input and output layers depend on the used dataset. The fully connected (FC) layer with 512 neurons are stacked at the last hidden layer in 1D CNNs model. The tentative models are mainly for the purpose of investigation about the performance of different models and the impact of the output dimension. The detailed information is presented in the following subsections.

Table 4.1: *Tentative model structures.*

Model	**Structure (Input – hidden layers – Output)**
DNN	Input-32-32-64-64-128-128-256-256-512-512-Output
1D CNNs	Input-32-32-64-64-FC-Output

4.6.1 The choice between DNN and 1D CNNs

For the generated SPD by Gaussian, the different number of channels can be selected. The used channels determine the dimension of the weight vector. E.g, the weight vector of the four-channel Gaussian has four dimensions. The dimension of the weight vector can affect the accuracy of the model prediction since it is difficult to predict high-dimensional variables than the lower ones.

In order to investigate the performance of DNN model and 1D CNNs model in terms of illumination SPD estimation, models are tested on several datasets which are generated by different numbers of channels. In this case, the Macbeth ColorChecker is chosen as the sample for this experiment. G1, G2, G3 and G4 datasets are generated from 4, 5, 6 and 7-channel Gaussian distributions, respectively, and each dataset has 10000 input-output pairs.

Table 4.2: *RMSE, GFC, SAM, SID values of prediction results with DNN model on Gaussian datasets with different channel numbers.*

Dataset	RMSE		GFC		SAM		SID	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
G1	**0.0205**	**0.0089**	**0.9989**	**0.0009**	**0.0436**	**0.0174**	**0.0049**	**0.0097**
G2	0.0235	0.0097	0.9988	0.001	0.0455	0.0175	0.005	0.0061
G3	0.0314	0.0151	0.9982	0.0049	0.0537	0.0266	0.007	0.0224
G4	0.0431	0.0203	0.9972	0.0027	0.0688	0.029	0.0087	0.0094

Table 4.3: *RMSE, GFC, SAM, SID values of prediction results with 1D CNNs model on Gaussian datasets with different channel numbers.*

Dataset	RMSE		GFC		SAM		SID	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
G1	**0.0094**	**0.0054**	**0.9997**	**0.0003**	**0.0203**	**0.0105**	**0.0021**	**0.0071**
G2	0.0155	0.0069	0.9994	0.0011	0.0313	0.016	0.0033	0.008
G3	0.0195	0.0099	0.9992	0.0028	0.0344	0.0216	0.0036	0.0166
G4	0.0243	0.0108	0.9990	0.0012	0.037	0.0175	0.0039	0.0048

The prediction performance is shown in Table 4.2 and Table 4.3. It is noticed that the RMSE increases when more channels are used for simulating the illumination. It is reasonable since the output dimension adds the complexity to the regression problem. In terms of GFC performance, the results from the 1D CNNs model all exceed 0.999 (indicates good spectral fit), which means the prediction spectrum from 1D CNNs model has better spectral approximation than the DNN model. The 1D CNNs model has overall better regression performance across different kinds of datasets according to the numerical metrics.

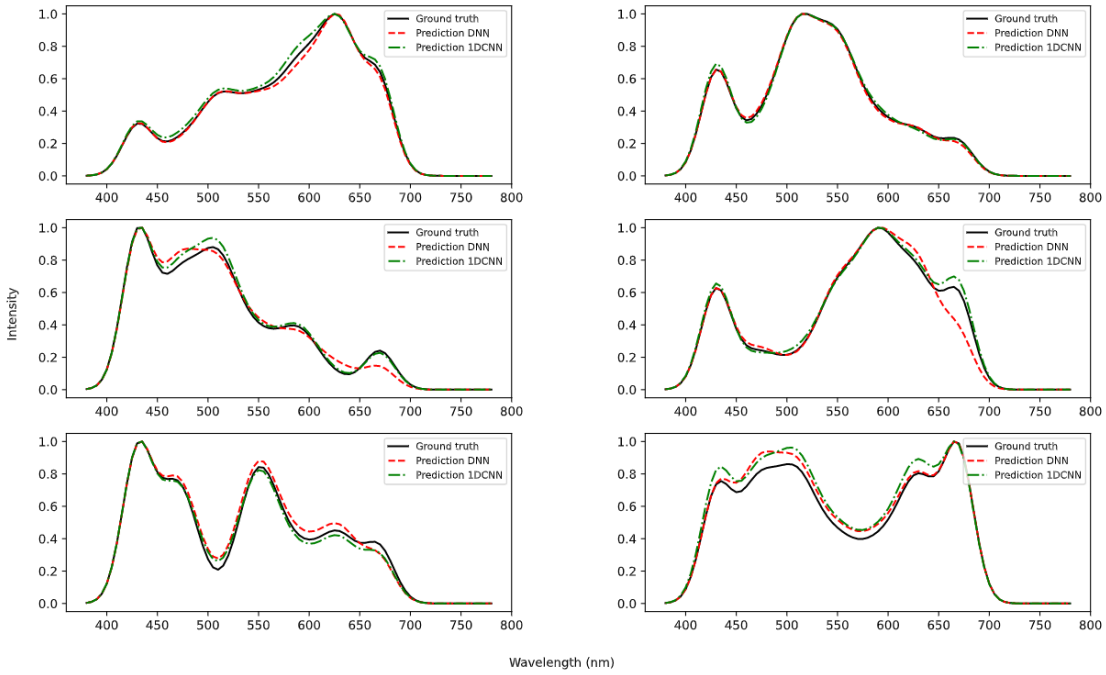


Figure 4.6: *Illumination SPDs prediction results on 7 channel dataset.*

The predicted illumination SPDs are shown together with ground truth SPDs in Figure 4.6. Those results are based on model trained from 7-channel dataset. The 1D CNNs model can predict better in terms of the shape of SPD.

4.6.2 The effect of regression dimension

After experiments with the DNN and 1D CNNs on Gaussian datasets, the 1D CNNs model is selected to test on generated SPDs from Telelumen LED data Figure 4.2a. The experimental ColorChecker is X-Rite ColorChecker Digital SG. The generated SPDs data are more similar to the scenario that an actual lighting system can produce. In order to test the effect of the regression target dimension on model performance, datasets H1 and H2 are generated from the 7 channel (randomly select 7 channels) and 24 channel of Telelumen SPD. In this case, the dimension of weight vectors for datasets H1 and H2 are 7 and 24, respectively. And the H1 and H2 both have 10000 input-output pairs. The images and SPDs in those two datasets are like Figure 4.4 and Figure 4.5.

Table 4.4: *RMSE, GFC, SAM, SID values of prediction results with 1D CNNs model on simulated Telelumen datasets.*

Dataset	RMSE		GFC		SAM		SID	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
H1	0.064	0.0224	0.9858	0.0103	0.1595	0.0561	0.1039	0.111
H2	0.1143	0.0532	0.9318	0.0586	0.3428	0.1472	0.2533	0.216

In Table 4.4, according to the model performance on RMSE, GFC, SAM, and SID, the model that trained on high dimension output variable has higher error regarding to the ones that trained on low dimension output variable.

4.6.3 Principal component analysis (PCA) on SPD

Considering the model needs to be compatible with real lighting SPD produced directly from the lighting system, the relationship between the real lighting SPD and each LED SPD is unknown. The real lighting can not be computed as a linear combination by using the weighted LED SPD of each channel. Therefore, the lighting SPD instead of the weight vector should be chosen as the output target variable of the model for the real lighting scenario. However, from the previous experiments, the high-dimensional target variable brings more errors for the model prediction. Due to the high dimension of the original SPD data, if it is directly used as the output of the machine learning model, it will cause slow convergence and a long training time. At the same time, redundant spectral data of SPD also reduces the prediction accuracy of the model. PCA is commonly used as a pattern classification-oriented data dimensionality reduction method. In the study Howley et al. (2005), it is utilized for the processing of the high-dimensional spectral data in order to increase the accuracy prior to the application of the machine learning algorithms. On the basis of reflecting more original information as much as possible, PCA replaces the original data with fewer principal components, so as to achieve the purpose of simplifying the data. Therefore, the PCA can be as the preprocessing step of illumination SPD data.

4.7 Illumination Estimation Framework

After choosing 1D CNNs model as the based regression model and integrating the PCA as one of necessary step to process the SPD data. The proposed workflow is shown in Fig 4.7, the virtual lighting SPDs generated by assigning the random weights and the simulated sRGB image based on those SPDs are the synthetic dataset for the neural network 1D CNNs model. Moreover, the average pixel values

of each patch from the ColorChecker are computed as the representation of the image to discard global spatial information. To consider the high dimensionality property of SPD spectral data, several PCA features are chosen as SPD's approximation. In the model training stage, the average pixel value and PCA features are constructed into 1D vectors, respectively, as the input and target dataset. After training the model with the synthetic dataset, for a given sRGB image illuminated under unknown lighting, the model output is the predicted PCA features and the whole spectrum of SPD can be recovered directly from those features with linear PCA reverse operation. It is noticed that the proposed scheme is not only compatible with virtual lighting but also with real lighting. For consistency, all the experiments later will be based on PCA processed SPD data.

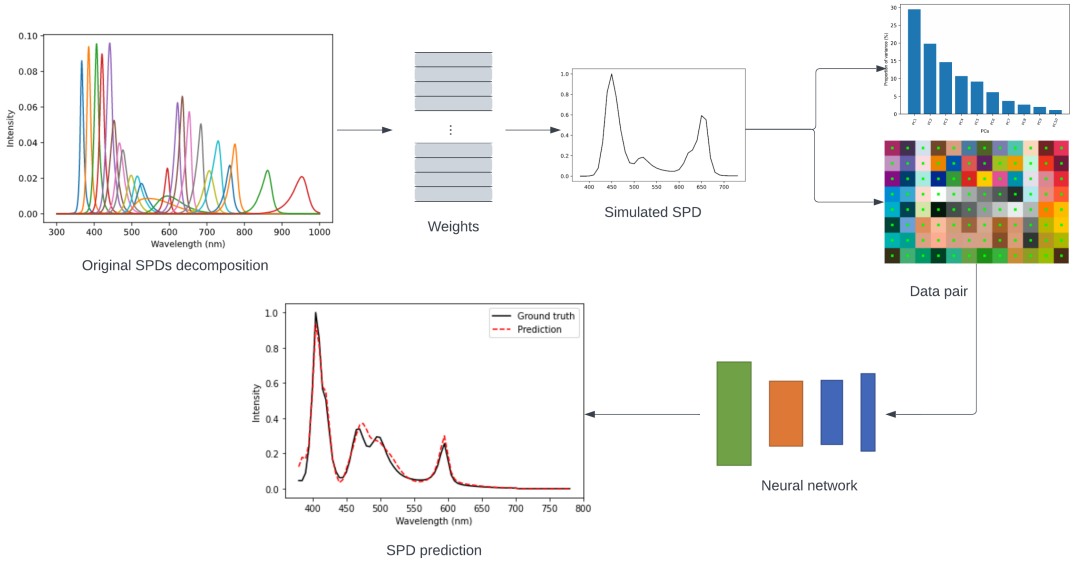


Figure 4.7: *SPD estimation scheme.*

The proposed 1D CNNs model detailed information are shown in Table 4.5. The input layer is 1D vector with a size 288×1 since 96 RGB pixels when using X-Rite ColorChecker Digital SG. The size of the output layer depends on how many PCA components are computed from SPD data. In general, the number of PCA components is determined based on the explained variance ratio that exceeds 99.0%. The Adam optimizer Kingma and Ba (2014) with a learning rate 0.001 and the mean square error (MSE) loss function are chosen as the model configuration. The bath size and the total epoch number are set to 64 and 200, respectively. The hyperparameters and structures are further tuned by exhaustive testing on different simulated datasets. The network is implemented based on Tensorflow and Keras using the NVIDIA GeForce RTX 3090 GPU. The model with the proposed

Table 4.5: *Neural network structure.*

Layer	The number of units	Activation function
Input	288×1	
Conv1D	282×32	ReLU
Conv1D	278×64	ReLU
MaxPooling1D	139×64	
Dropout	139×64	
Flatten	8896	
Dense	64	ReLU
Output	14	Sigmoid

configuration, the training time on dataset with 10000 samples is around 2 minutes.

Any database and sensor can be utilized as long as the testing is performed on data that was obtained from the same surface and illuminant databases and by utilizing the same sensor sensitivities. Nevertheless, our ultimate objective is to evaluate the deep learning model using real image data captured with a digital camera of a real ColorChecker object. If a deep learning model that was trained on synthetic data is going to be evaluated on actual images, the sensor sensitivity functions that were used to train it need to be as similar as the real sensors that were being utilized in the tests. Any departure of the actual camera from the trained model will result in discrepancies in the RGB values observed by the camera and, as a consequence, the accuracy of illumination estimation provided by the model is reduced.

In this context, the camera sensitivity used for the virtual camera model should be similar to the camera that used for capturing the real image. Since the Canon 5D Mark II camera sensitivity function is used for the virtual camera model implementation, the Canon 5D Mark IV (which has the similar image sensor to the Canon 5D Mark II) is selected as the camera for the real image capturing. The detailed discussion about how to generate real image datasets is discussed in the next chapter 5.

5 Dataset Generation

In this chapter, the datasets for training the proposed illumination SPD estimation framework 4.7 are generated. As mentioned in the data generation part in Figure 1.1, the measured dataset is captured in real scenarios with a camera and spectrocolorimeter while virtual dataset is simulated by the virtual camera model. In order to maintain data consistency for the measured dataset, the data collection tools are developed to ensure the automatic acquisition. The captured image preprocessing and the time cost for the generation of both datasets are compared in the last section.

5.1 Real Lighting Dataset

Compared with the virtual lighting generated by LED SPD data, the lighting produced by the lighting system is measured directly using a spectrocolorimeter. The illumination estimation model is necessary to be validated under different real lighting conditions since they are directly associated with the lighting system itself. For generating various lightings, the software equipped with the lighting system enables us to control the intensity of each LED channel. After excluding the UV and IR channels, there are 17 channels left. We can assign the weights for all other channels to generate arbitrary lighting. Unlike the simulated SPD, the way that the lighting system combining LED channels is not clear. However, based on the previous system testing, the lighting system has power limitations such that it is impossible to turn all the channels to the maximum intensity at the same time. The software of lighting system allows the user to adjust the intensity of each LED channel from 0 to 1. Since the each LED consumes different amount of power, it is difficult to ensure the valid weight before sending to the lighting system. Due to the invalid weight, all channels are turned off after being received by the lighting system. Based on previous tests, the lighting system hardly emits any light when the number of selected channels is increased to 10 (with random intensity of each channel).

Taking into account the technical constraints of the lighting system and previous

testing, several experiments were conducted to investigate the optimal configuration:

- Experiment 1: randomly selecting 7, 8 or 9 channels (from 17) and assigning the random weight (0∼1) to each channel by considering that the turning 10 channels on will shut down the system as mentioned before and in the meanwhile we want to keep turn on as many channels as possible. The weight of other non-selected channels is set to zero. By repeating this process, a number of sample weights can be generated. If there is no lighting after sending the weight to system, the weight is referred to as an invalid weight; otherwise, it is referred to as a valid weight. The valid ratio is derived by dividing the number of valid weights by the total number of sample weights. The optimal number of channels can be determined based on the valid ratio.

- Experiment 2: randomly selecting 9 channels (from 17) and limiting the maximum intensity from 0 to 1 with an interval of 0.15 since even 9 channels can sometimes exceed the power limitation without limiting the maximum intensity. As in experiment 1, the sample weights are generated and tested in the lighting system. The highest optimal maximum intensity can be selected by checking the valid ratio of different cases.

- Experiment 3: Turning on all 17 channels of the visible range instead of only choosing a few of them in order to take advantage of the full ability of the lighting system in terms of generating complex high-dimension lighting. The optimal maximum intensity can be determined with the same procedure as in experiment 2.

After exhaustive testing, the optimal configuration according to each experiment mentioned above is determined as following:

- Configuration 1: randomly selecting 7 channels (from 17) without limiting the maximum intensity.

- Configuration 2: randomly selecting 9 channels (from 17) with maximum set to 0.75.

- Configuration 3: turning on all 17 channels with maximum set to 0.4.

The three different datasets can be generated according to each configuration. The proposed dataset information is shown in Table 5.1. The dataset includes the sRGB images and the corresponding illumination SPDs. The obtaining of the sRGB image will be discussed later in last section of this chapter. The total number of sample weights for each configuration is set to 3170 by considering the long time of capturing and potential overheating problems of the devices. The time for

Table 5.1: *Proposed datasets.*

Dataset	Total number	Valid number	Channel	Size
D1	3170	2167	7	
D2	3170	2395	9	1709×1212
D3	3170	2631	17	

capturing one image and measuring the corresponding SPD is around 7.5 seconds. Therefore, D1, D2, D3 three datasets cost more than 15 hours. After feeding the sample weights to the lighting system, the valid number of lighting SPD in each dataset are 2167, 2395 and 2631 respectively. The each corresponding sRGB image is cropped to the same size with 1709×1212.

5.2 Virtual Lighting Dataset

After generating the measured dataset according to the configuration in Table 5.1. The valid weights from the sample weights that produce the real lighting can be used for generating corresponding virtual lighting. Afterwards, the synthetic sRGB images are easily computed by our virtual camera model. In this way, the certain similarity between the measured dataset and virtual dataset is ensured by the similar distribution of real lighting and virtual lighting.

5.3 Captured Dataset Generation

There are several existing datasets for the usage of illumination color estimation purposes like Gelher-Shi Gehler et al. (2008), NUS Cheng et al. (2014) and Cube++ Ershov et al. (2020). Even though the image content is various and the ground truth illumination color is provided, the illumination type is still limited to indoor, outdoor natural lighting or artificial lighting from low-end light source device. The lack of illumination types (e.g., non-existing lighting) and the SPD information is still the problem for the illumination SPD estimation task. We decided to do long capture using a professional camera (Canon 5D Mark IV) by taking the advantage of the powerful 24-channel LED Telelumen lighting system which is able to produce infinite unique lighting. In order to achieve the automatic image capturing to obtain the datasets as mentioned in Table 5.1, the implementation of two different data collection tools in charge of driving the three devices is required:

- The 24-band spectral lighting from Telelumen company, allowing to produce our lights (with a Python SDK).

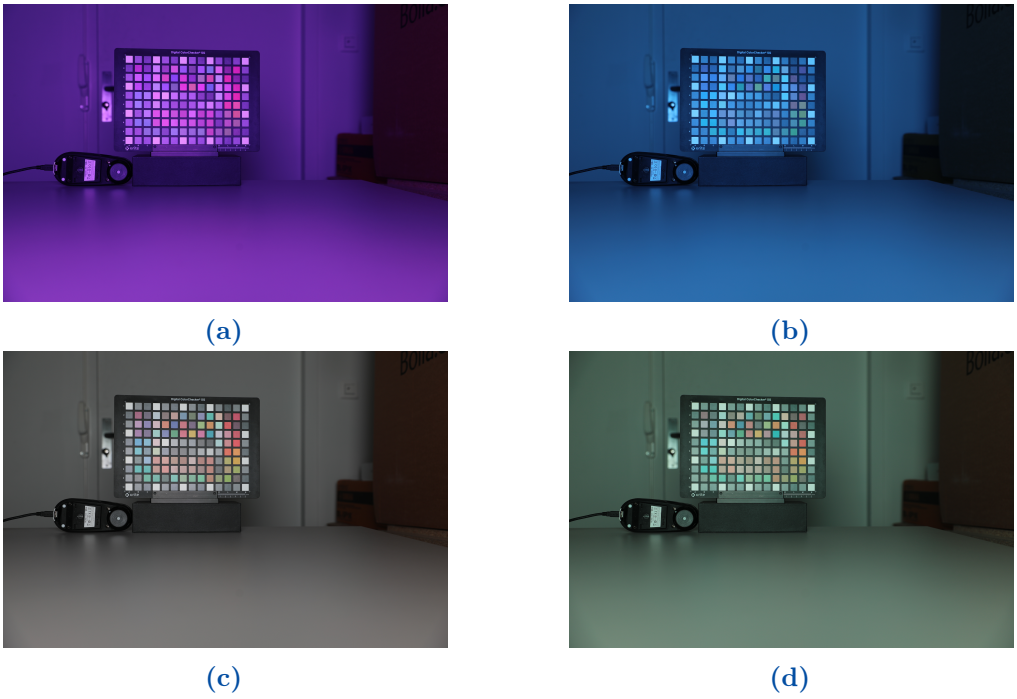
- An I1Photo Pro 2 spectrocolorimeter from X-Rite allowing to measure the SPD of the light sources as well as the reflectance of the patches of our color chart (with a C++ SDK).

- A Canon EOS 5D Mark IV camera used to capture the RGB images of the color chart (with a C++ SDK).

The first tool, named ReflectenceMeasure, allows to measure the reflectances of the patches with the help of the x-rite I1Photo Pro2 spectrocolorimeter. Each of the reflectances obtained is the result of the average of 4 measurements and takes the form of a 36-dimensional vector with a wavelength from 380 nm to 740 nm and 10 nm steps.

The second tool, named MultiCapture, was much more complex to implement because it allows a list of 24-dimensional vectors (corresponding to the activation values of our spectral lighting system) to activate and measure the light produced and then shoot and download the chart from our camera.

For all the captures, our camera was set in Automatic Shooting (Scene Intelligent Auto) mode where the camera analyzes the scene and sets the optimum settings automatically. It makes images in the JPEG format with a high quality setting so that data loss from compression is kept to a minimum.



(a) **(b)** **(c)** **(d)**

Figure 5.1: *Captured sRGB images under Telelumen lighting.*

The whole capturing process takes place in a room without windows and with all the other lights off. Several captured images are shown in Figure 5.1.

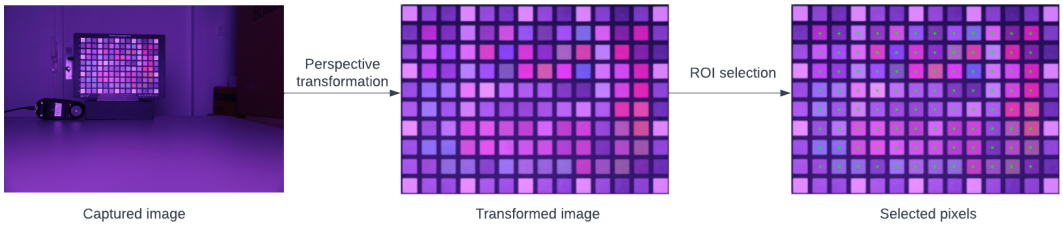


Figure 5.2: *Captured image preprocessing.*

The camera angle is not perfectly perpendicular to the ColorChecker the image is distorted as shown in Figure 5.2, the further image processing steps are needed to correct and acquire the useful information. Since the only ColorChecker is our interesting object in image, the perspective transformation Szeliski (2010) is applied to exclude outliers and get the perspective view image. Perspective transformation requires that the perspective center, the image point, and the target point be collinear. According to the law of perspective rotation, the perspective surface is rotated around the perspective axis by a certain angle, destroying the original image. It projects an object to a new imaging plane based on the law of object imaging projection. After that, the average pixel values from each center of the color patch are computed. The ROI (Region Of Interest) is shown as green square in the image and the size of ROI can be adjusted depending on the original captured image resolution. It is noticed that the neutral color patches where locate in outermost layer of ColorChecker are excluded during ROI selection since the colorful area is brighter and has various degree of saturation.

All the collected data used for this work are available from this address: `https://www.couleur.org/articles/SITIS2022-WAI`.

The time consumed by generating the measured datasets and the virtual datasets is shown in Table 5.2. The advantage of the virtual camera model is reflected in the dataset acquisition time.

Table 5.2: *Dataset acquisition time comparison.*

Dataset	Number of images	Total time spend
Measured dataset	7193	15 hours
Virtual dataset	7193	55 minutes

6 Experiments, Results Analysis and Transfer Learning Tests

6.1 Model Verifying

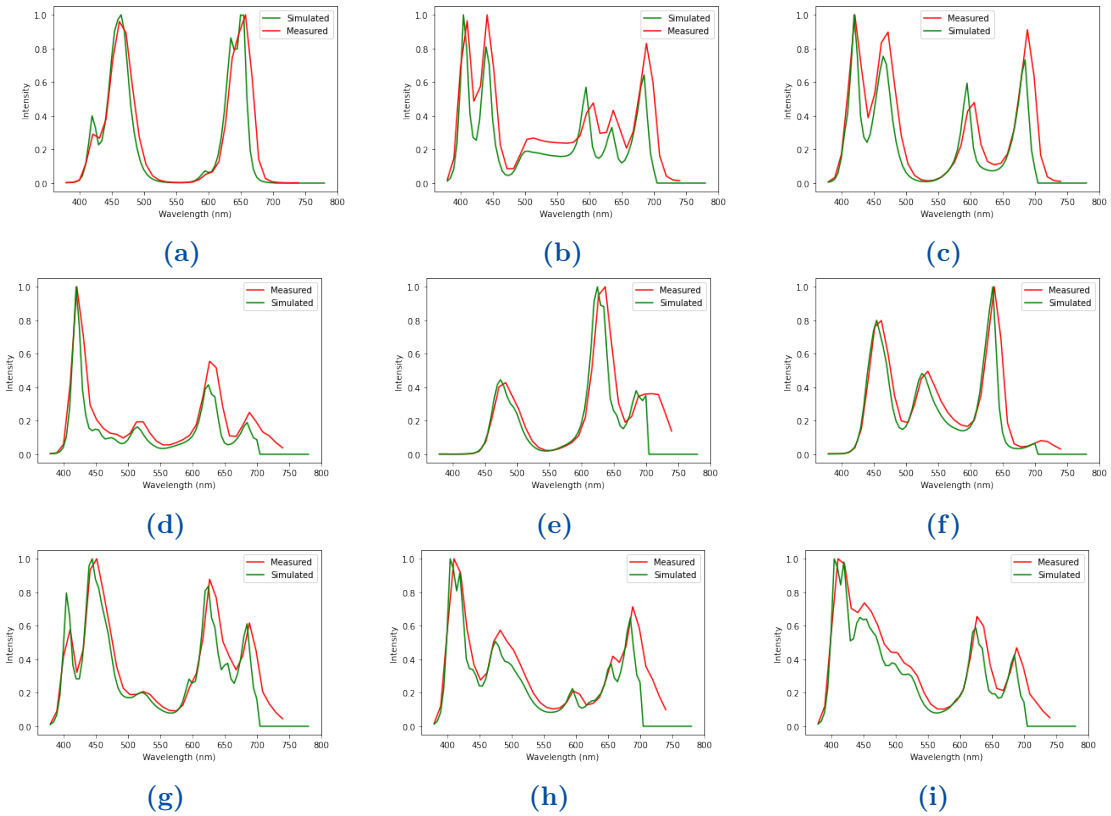


Figure 6.1: *Simulated virtual lighting through the equation 4.5 by using the same valid weights according to configuration in Table 5.1 for generating the real lighting, first row is 7 channel, middle row is 9 channel and last row is 17 channel SPD.*

According to the configurations mentioned in section 5.1, the measured datasets are generated by using the data collection tools introduced in the previous Section 5.3 according to the Table 5.1. The corresponding virtual datasets can also be easily computed by virtual camera model using the same weights that were sent to the lighting system when generating measured datasets. Several simulation examples are shown in Figure 6.1, the simulated virtual lighting and measured real lighting are not identical to each other since the light measurement device accuracy is restricted to a 10 nm interval and the power limitation of the light system as mentioned before. This indicates that the lighting system produces the lighting from LED channels not in a purely linear-like combination mechanism.

In order to test the illumination estimation model on proposed datasets. Several different dataset combinations are designed to verify the performance.

Table 6.1: *Model verifying scenarios.*

Scenario	**Training dataset**		**Prediction dataset**	
	Measured	Virtual	Measured	Virtual
V1	D1	S1	D1	S1
V2	D1	S1	D2	S2
V3	D1	S1	D3	S3
V4	D2	S2	D1	S1
V5	D2	S2	D2	S2
V6	D2	S2	D3	S3
V7	D3	S3	D1	S1
V8	D3	S3	D2	S2
V9	D3	S3	D3	S3
V10	D1+D2+D3	S1+S2+S3	D1	S1
V11	D1+D2+D3	S1+S2+S3	D2	S2
V12	D1+D2+D3	S1+S2+S3	D3	S3

Table 6.1 shows all the experiments conducted on proposed datasets. The model is trained and validated on different dataset settings. The same verifying scenarios are also applied to the virtual datasets. The virtual datasets are denoted as S1, S2 and S3 to make a comparison with measured datasets.

From Table 6.2 and Table 6.3, for the scenarios with the same training dataset GFC, SAM, and SID indicate better results when on a high-channel prediction dataset. E.g., the V1, V2 and V3 are all trained on D1 dataset, the better performance observed on prediction dataset D3 which represents 17 channel case. Based on the model performance on RMSE, the mixed D1+D2+D3 is the best training dataset among all the scenarios. The overall better performance is observed in Table 6.3 than Table 6.2 since the synthetic image can contain more information from the illumination than captured ones.

Table 6.2: *RMSE, GFC, SAM and SID of prediction results with 1D CNNs model on proposed measured datasets.*

Scenario	RMSE		GFC		SAM		SID	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
V1	0.1106	0.0367	0.9561	0.028	0.2833	0.0918	0.3498	0.2581
V2	0.1112	0.0381	0.9625	0.0234	0.2628	0.0812	0.3192	0.3486
V3	0.1514	0.0454	**0.9772**	0.0116	**0.2072**	0.0529	**0.1757**	0.3094
V4	0.1202	0.035	0.9505	0.0265	0.3056	0.0824	0.3642	0.2558
V5	0.1081	0.0327	0.963	0.0194	0.264	0.0704	0.2839	0.3377
V6	0.1162	0.0403	**0.9828**	0.0093	**0.1793**	0.0489	**0.0625**	0.066
V7	0.1833	0.0486	0.9065	0.049	0.4223	0.1122	0.5286	0.3438
V8	0.1588	0.0452	0.9365	0.035	0.3466	0.0933	0.3316	0.2239
V9	0.889	0.0279	**0.9825**	0.0101	**0.1802**	0.0513	**0.0602**	0.038
V10	0.1036	0.0312	0.9624	0.0208	0.265	0.0748	0.3348	0.2898
V11	0.0998	0.0325	0.97	0.016	0.2363	0.068	0.2236	0.205
V12	**0.0808**	0.028	**0.9868**	0.0076	**0.1561**	0.0449	**0.048**	0.0301

Table 6.3: *RMSE, GFC, SAM and SID of prediction results with 1D CNNs model on virtual datasets.*

Scenario	RMSE		GFC		SAM		SID	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
V1	0.0822	0.0258	0.9643	0.022	0.2573	0.076	0.4314	0.4091
V2	0.0859	0.0314	0.9688	0.0182	0.2416	0.0669	0.3035	0.2689
V3	0.1428	0.0428	**0.982**	0.0107	**0.183**	0.052	**0.0857**	0.126
V4	0.0889	0.026	0.9584	0.0348	0.2751	0.0919	0.3665	0.2579
V5	0.0814	0.0254	0.9692	0.0173	0.2392	0.0689	0.226	0.1884
V6	0.1009	0.0365	**0.9871**	0.0083	**0.1531**	0.0492	**0.039**	0.0311
V7	0.169	0.0525	0.9029	0.0741	0.4216	0.1476	0.6109	0.9404
V8	0.143	0.044	0.939	0.0445	0.3357	0.1063	0.3581	0.5429
V9	0.0679	0.0228	**0.9871**	0.0088	**0.1526**	0.0499	**0.0405**	0.0463
V10	0.0837	0.024	0.9656	0.0189	0.2545	70.0675	0.3874	0.3063
V11	0.0827	0.028	0.9716	0.0149	0.2304	0.0635	0.2529	0.2154
V12	**0.0698**	0.0243	**0.988**	0.0074	**0.1483**	0.0453	**0.035**	0.0229

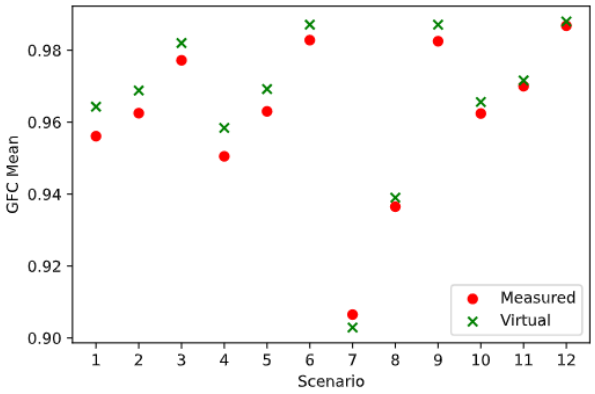
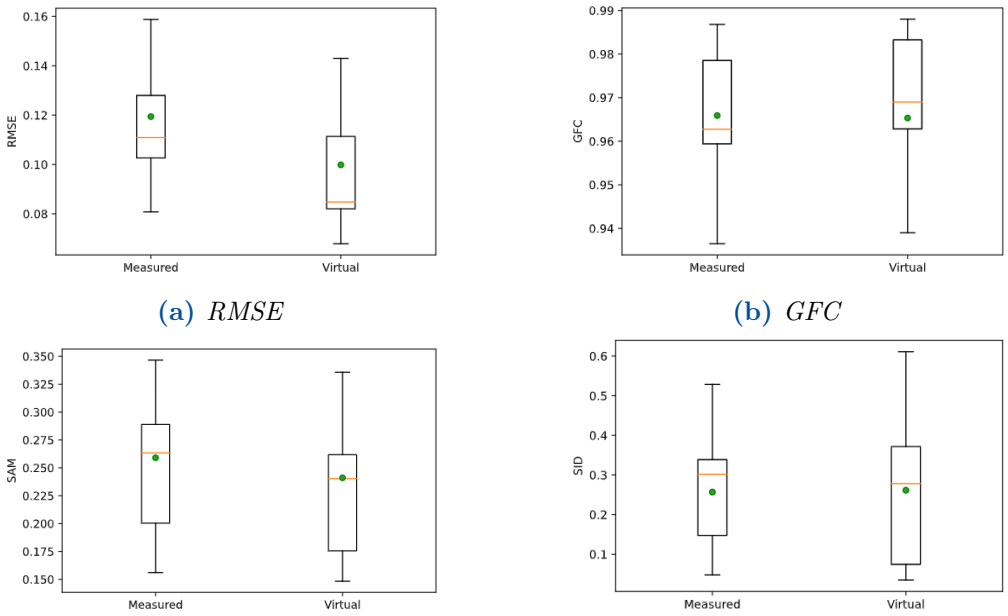


Figure 6.2: *GFC mean values between measured and virtual dataset.*

The Figure 6.2 shows the GFC mean values throughout all scenarios for the model that trained on measured and datasets. The difference in each scenario is quite small, which means the simulated virtual dataset can approximate the measured dataset to some extent since the model configuration is the same for both cases.



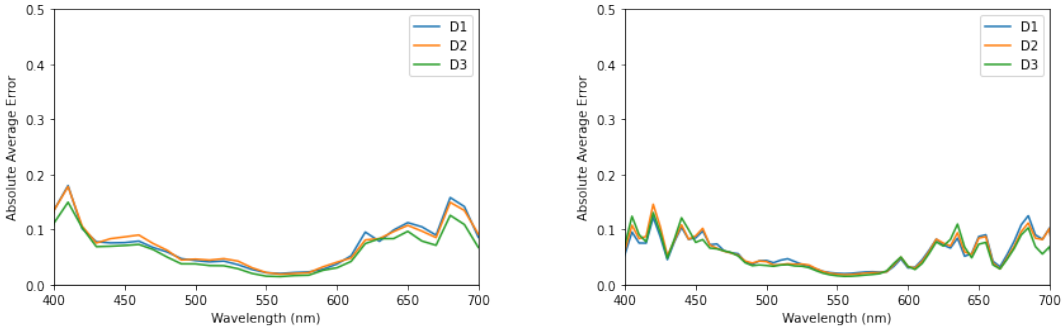
(a) *RMSE* **(b)** *GFC*

(c) *SAM* **(d)** *SID*

Figure 6.3: *Box plots of the SPD estimation for the four evaluated cases under measured and virtual datasets. The median values are shown with orange lines, the mean values with green dots, and the 25%-75% percentile range with black rectangles.*

The metric values in 12 scenarios (V1 to V12) of measured and virtual datasets are presented in Fig .6.3. The model has overall better performance on virtual datasets than measured datasets according to those four metrics. However, the performance of each metric between measured and virtual datasets is similar in terms of median and mean values. This is an important observation because we might have insight about the model performance on measured dataset ahead (without having measured dataset). After determining the lights that need to be produced by using the light system, the corresponding virtual lights can be simulated by virtual camera model quickly. The model performance based on virtual lights can indicate how much performance the model can achieve when using the measured dataset that we are going to capture. In this way, the measured dataset can be optimized before conducting the real capturing process.

6.2 Camera Sensitivity Restriction



(a) *Performance on measured dataset* **(b)** *Performance on virtual dataset*

Figure 6.4: *Absolute average error.*

The absolute average error is computed between ground truth and predicted SPD at each band for different datasets, as shown in Figure 6.4. The high error is present at around 400 nm and 700 nm both in the measured and virtual datasets. The possible reason for that could be the valid range of camera sensitivity function. For the virtual camera, the spectral response is from the camera Canon EOS 5D Mark II, the data can be found in EPFL (2022). It is noticed that the response is quite low near 400 nm and 700 nm in Figure 6.5. The sensor is relatively limited to acquiring the SPD information at the edge of the response range. The lack of information around and beyond this wavelength range can deteriorate the prediction accuracy of the illumination estimation model. Due to the similarity of the real camera (EOS 5D Mark IV) and the virtual camera, we can assume

that their camera sensitivity functions are also related, which can explain the relative identical performance on both the measured and virtual dataset in terms of absolute average error.

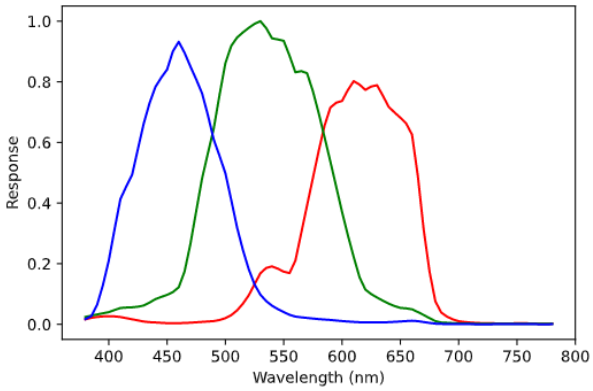


Figure 6.5: *Camera sensitivity function.*

By considering the model performance in RMSE, GFC, SAM, SID and absolute average error on measured as well as virtual datasets, it seems that the proposed illumination estimation model has nearly the same error pattern on those datasets. This kind of tendency gives the possibility to approximately predetermine the error results of measured datasets that are to be acquired by using corresponding virtual datasets.

6.3 Worst-Case Prediction

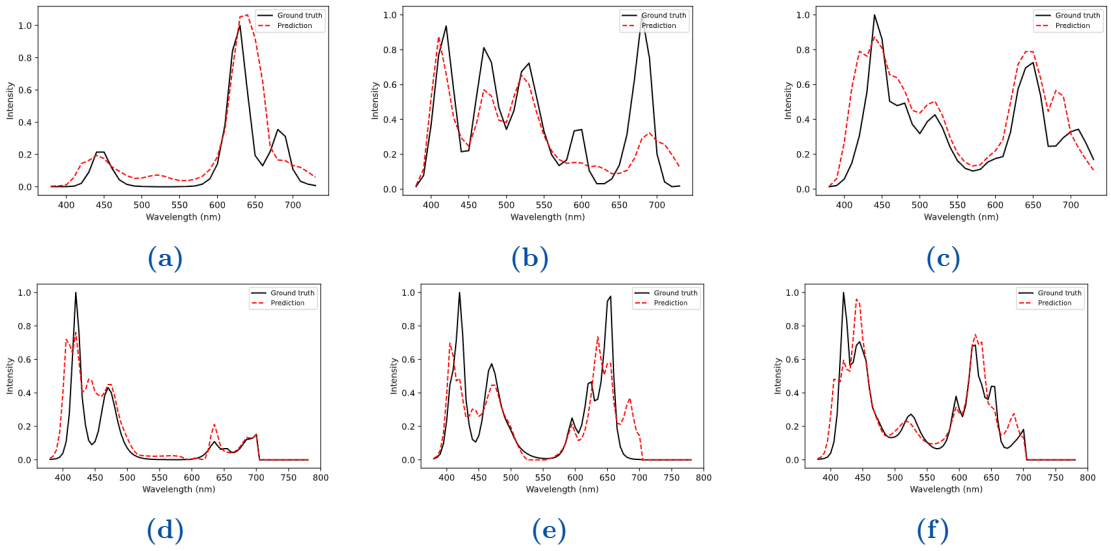


Figure 6.6: *The prediction SPD with maximum prediction error with respect to GFC, first row is based on measured dataset and second row is based on virtual dataset.*

The prediction results in Figure 6.6 are based on measured and virtual datasets. From left to right, the predicted SPD and ground-truth SPD with maximum GFC error for 7, 9 and 17 channel cases are presented. The relatively high prediction error can be visually observed at the beginning and end of the wavelength range, which corresponds to the main finding from the Figure 6.4.

With predicted SPD from the illumination estimation model, it is possible to simulate the corresponding predicted sRGB images by applying the virtual camera model. The influence of the SPD difference can be represented by the images of ColorChecker. In Figure 6.7, the predicted SPD with the same GFC value (0.9598) compared with ground truth for 7, 9, 17 channel cases are selected. In each case, the predicted sRGB image is generated with the virtual camera model by applying predicted SPD. The visual difference can be observed from the color patches between the ground-truth and predicted ColorChecker images. The pixel-wise differences are also computed by subtraction of the ground-truth images from the predicted ones. It is noticeable that the pixel difference in 7-channel case is more than the other two cases even though the GFC value is the same. The potential reason can be the low uniformity of SPD across the wavelength range. There are three main peaks in predicted SPD in 7-channel case and the other parts are all nearly zero intensity.

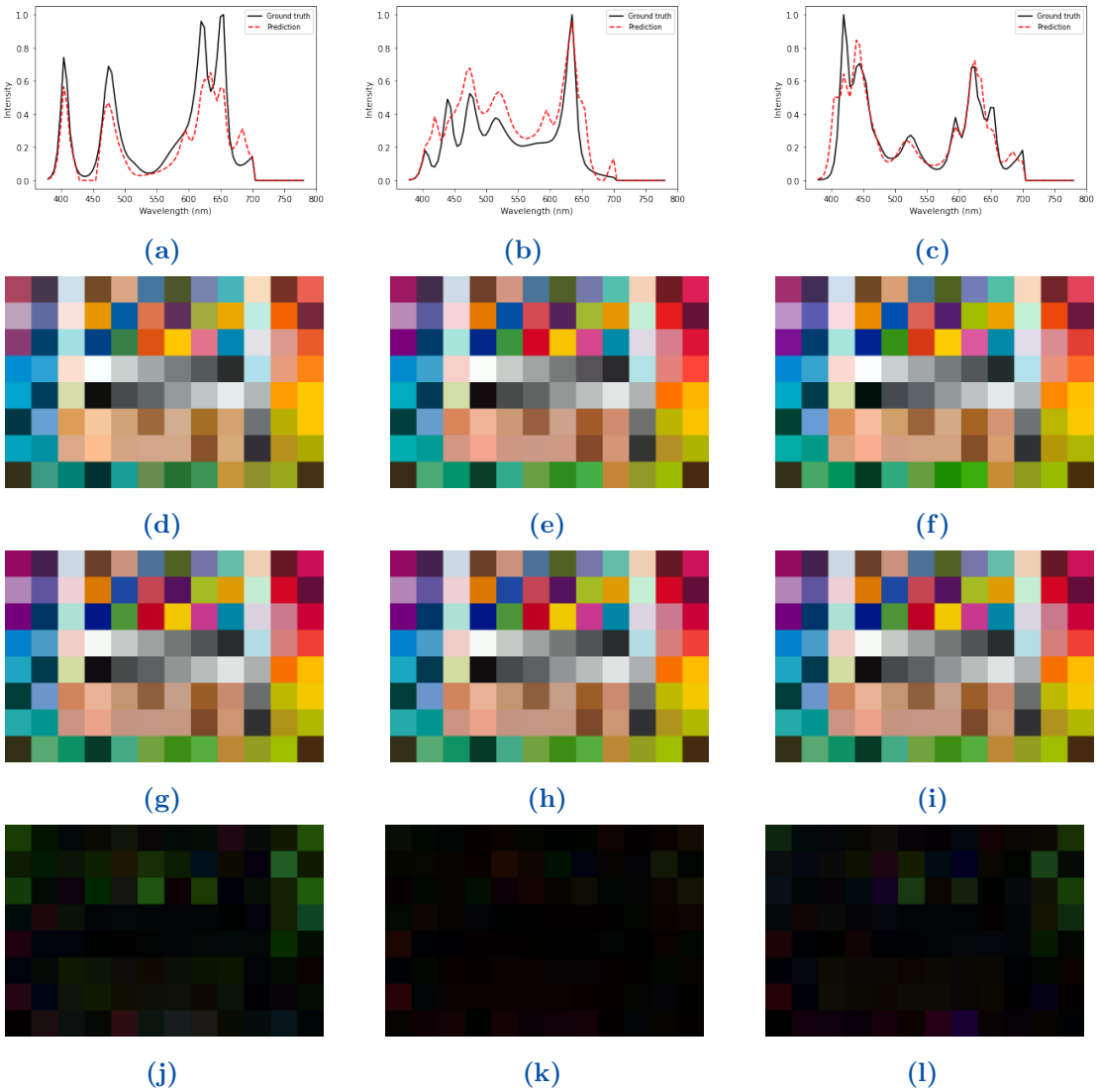


Figure 6.7: *Simulated images by virtual camera with predicted SPDs and the visual differences versus the ground-truth images. From top to bottom are the predicted and ground-truth SPDs, ground-truth images, simulated images and pixel-wise difference images.*

6.4 Transfer Learning Experimentation: Sensor-independent Performance

The previous experiments are all based on datasets that are associated with the specific camera sensor. The illumination estimation model needs to be retrained

when working on the new dataset that is produced by the new camera sensor. Thus, the model has sensor-dependent performance. In general, it is hard to find or capture the large amount of datasets with desired camera devices. The transfer learning (TL) Torrey and Shavlik (2010) can assist the model training when working on similar task or similar datasets but there is only a small amount of available data captured by an unknown camera sensor.

In order to verify the basic transfer learning technique, the two new testing datasets are captured with different lighting sources and imaging devices. The capturing procedure follows the steps mentioned in the previous chapter 5. Datasets T1 and T2 in Table 6.4 contain sRGB images and also the corresponding SPDs. The image content is the same ColorChecker Digital SG as used in proposed datasets D1∼D3. The light source in T1 is the RGB lamp which can change the lighting by adjusting the R,G,B values by the software. Since the lightness of the lamp is quite weak, the two lamps positioned in two directions are used during the capturing. The same camera Canon 5D Mark IV is used as an imaging device. Besides, the SPDs of the lighting from the lamps are not very diverse since it is only a three-channel light source. The imaging device in T2 is the tablet Galaxy Tab S7 and the associated camera sensor is low quality compared with the Canon 5D Mark IV.

Table 6.4: *Testing datasets for transfer learning.*

Dataset	Total number	Light source	Imaging device
T1	1000	Philips hue E27 color	Canon 5D Mark IV
T2	70	Telelumen lighting system	Galaxy Tab S7

The pre-trained 1D CNNs model (with same configuration as described in 4.5) is based on a virtual dataset (with 24 channels in order to include different illumination SPD as many as possible to avoid an unbalanced dataset) that has 10000 samples. T1 dataset is used for testing the transfer learning performance when the lighting source device is changed. T2 dataset is desired to test the situation when only a few data samples are available and the images are captured by a low-quality and high-dissimilar camera sensor as the ones that used in pre-trained model. The model performance results on different metrics with and without transfer learning applied are shown in Table 6.5.

The transfer learning improves the performance on both testing datasets. Specifically, the GFC mean value exceeds 0.999 on dataset T1. There is also around 6.7% improvement according to GFC mean value on dataset T2 when applying transfer learning.

The ground-truth SPD and predicted SPD with and without transfer learning are in Figure 6.8 and Figure 6.9. The black, red, and green color curves indicate

Table 6.5: *RMSE, GFC, SAM and SID of prediction results of 1D CNNs model with and without transfer learning on testing datasets.*

Dataset	RMSE		GFC		SAM		SID	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
T1	0.2065	0.0953	0.9479	0.0423	0.2998	0.1258	0.1002	0.0817
T1(TL)	0.0535	0.033	**0.9991**	0.0009	0.0385	0.0185	0.0032	0.0036
T2	2.4884	0.5853	0.8518	0.0686	0.539	0.1237	0.5225	0.24
T2(TL)	1.9888	0.7153	**0.9091**	0.0682	0.404	0.1523	0.4535	0.6042

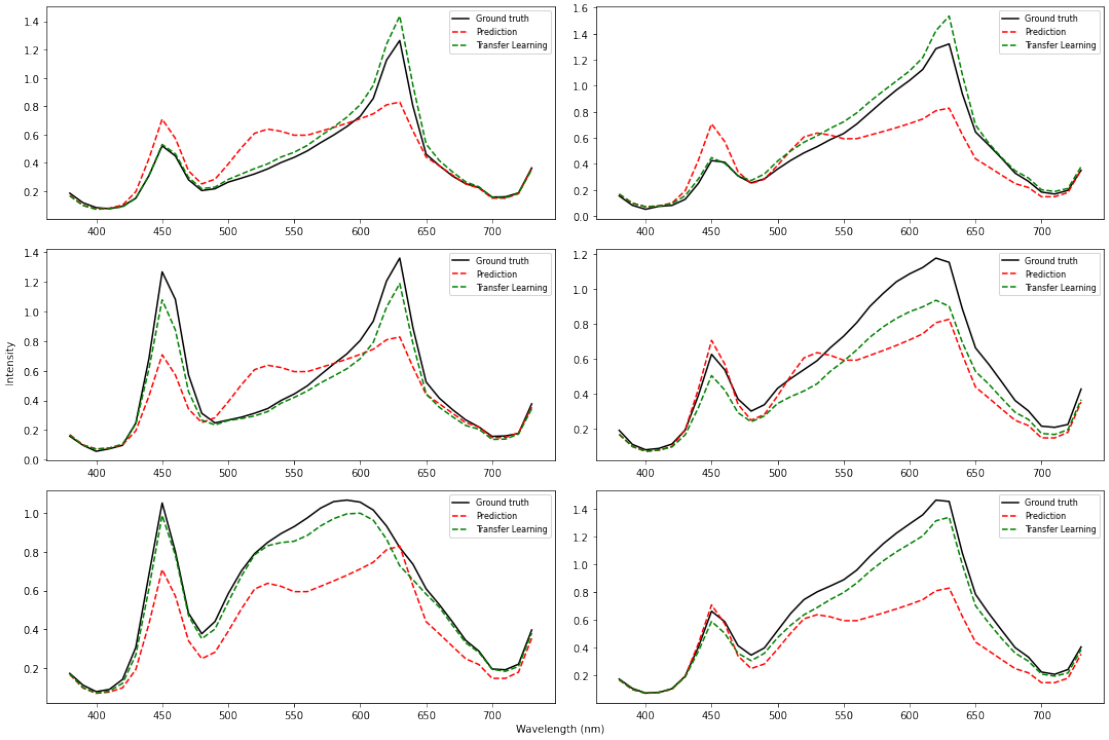


Figure 6.8: *Illumination SPDs prediction results with transfer learning on dataset T1.*

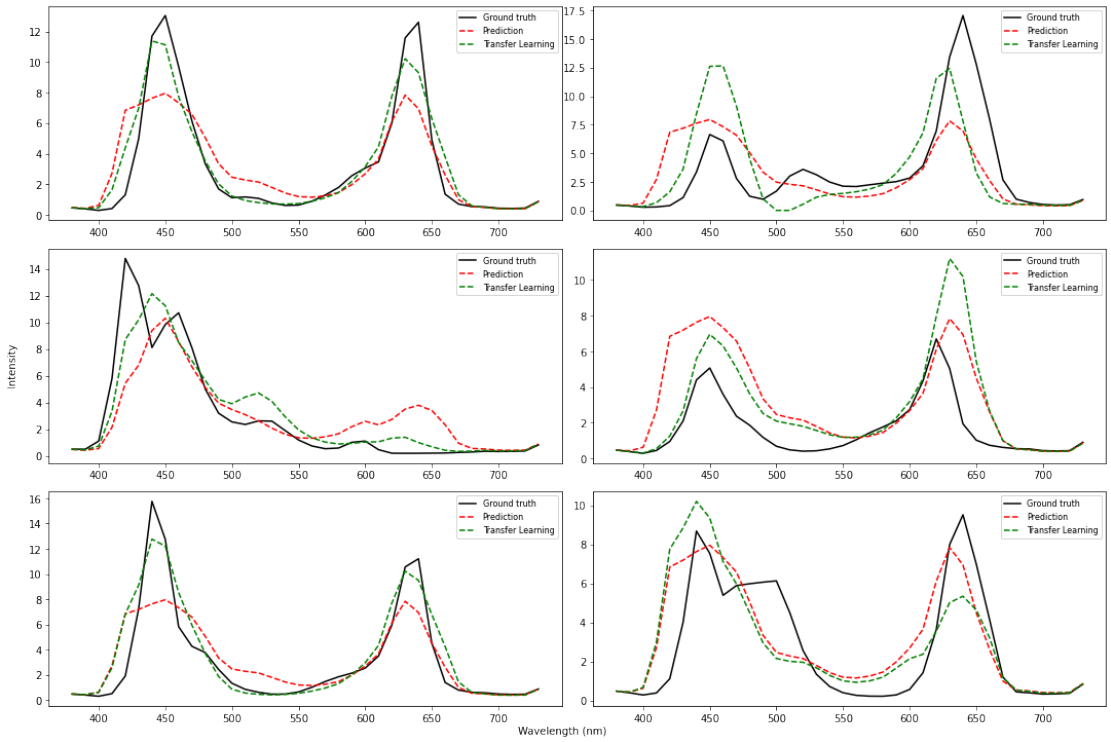


Figure 6.9: *Illumination SPDs prediction results with transfer learning on dataset T2.*

ground-truth SPD, prediction without transfer learning and prediction with transfer learning. The predicted SPD with transfer learning on both T1 and T2 datasets is closer to the shape of ground-truth SPD. The potential reason that transfer learning improves the performance on both datasets might be due to the complexity and similarity of the lighting SPDs. For dataset T1, the lighting source device is a low-end lamp which can only produce three-channel lighting. Thus, the complexity and diversity of this dataset is much lower than the simulated 24-channel dataset. For dataset T2, although only few data and the low-end imaging device, the lighting SPD is similar to the ones that pre-trained model learned.

According to the foregoing trials, the illumination estimate model can learn sensor-independent characteristics via data generation and transfer learning techniques.

7 | Conclusion and Perspectives

In this work, the virtual camera model is implemented according to the ISP pipeline including sensor imaging, demosaicing, white balance, color transformation parts with additional noise and optical aberration simulation adding certain non-linearity to approximate defects of the real cameras. The virtual camera model can be viewed as a data-generation tool that aids the training process for deep learning. The illumination SPD comprises both the virtual lighting simulated by the linear combination equation and the real lighting measured by the lighting measurement device. The 24-channel LED lighting system is utilized as the primary light source for the real lighting case. There are two proposed datasets, including virtual datasets and measured datasets. The virtual datasets are created using a virtual camera model and virtual lighting, whereas the measured datasets are captured using a commercial camera in real lighting. With those proposed datasets, the deep learning model 1D CNNs is trained on input-output pairs of PCA components that are derived from illumination SPD data and average pixel values that are computed from each color patch of the ColorChecker sRGB images. In the end, the proposed illumination estimation framework is assessed on synthetic images and also on real captured images by using different numerical metrics. The consistent prediction performance between the measured dataset and the virtual dataset is observed, which indicates the data validity of the synthetic image that is simulated by the virtual camera model. This gives the opportunity to optimize the real lighting dataset setting based on the performance from the virtual dataset, hence preventing random long-duration captures. Furthermore, simulated datasets and transfer learning enable sensor-independent prediction, i.e., predicting lighting SPD when images are acquired by a different camera.

Unquestionably, the quality of the images obtained by a high-end camera contributed to the performance of the proposed lighting SPD estimate model. To make the model more practical, instead of a professional camera, a phone camera might be utilized as an image capturing device. Meanwhile, a tiny and portable ColorChecker could be adopted to cut computing costs and improve usability. Even though the precision of the SPD estimation may be lowered to some extent, the simpler configuration allows for future real-time applications such as cosmetic

rendering, which renders the human face with the approximate SPD of the real scene illumination. For XR applications where illumination is critical for the display of virtual objects, several ColorCheckers could be placed and automatically identified using ArUco markers. This would enable real-time prediction of the SPD of multiple lighting sources from any desired location, allowing realistic representations to be entirely coherent with the real lighting in the area.

Bibliography

Adams Jr, J. E. (1995). Interactions between color plane interpolation and other image processing functions in electronic photography. In *Cameras and Systems for Electronic Photography and Scientific Imaging*, volume 2416, pages 144–151. SPIE. (cited on page 7)

Adams Jr, J. E. (1997). Design of practical color filter array interpolation algorithms for digital cameras. In *Real-Time Imaging II*, volume 3028, pages 117–125. SPIE. (cited on page 8)

Afifi, M. and Brown, M. S. (2019). Sensor-independent illumination estimation for dnn models. *arXiv preprint arXiv:1912.06888*. (cited on page 26)

Afifi, M., Brubaker, M. A., and Brown, M. S. (2022). Auto white-balance correction for mixed-illuminant scenes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1210–1219. (cited on page 9)

Bayer, B. E. (1976). Color imaging array. *United States Patent 3,971,065*. (cited on page 5)

Beaudoin, P. M., Audet, Y., and Ponce-Ponce, V. H. (2009). Dark current compensation in cmos image sensors using a differential pixel architecture. In *2009 Joint IEEE North-East Workshop on Circuits and Systems and TAISA Conference*, pages 1–4. IEEE. (cited on page 7)

Boncelet, C. (2009). Image noise models. In *The essential guide to image processing*, pages 143–167. Elsevier. (cited on page 17)

Buchsbaum, G. (1980). A spatial processor model for object colour perception. *Journal of the Franklin institute*, 310(1):1–26. (cited on pages 8 and 10)

Cardei, V. C., Funt, B., and Barnard, K. (2002). Estimating the scene illumination chromaticity by using a neural network. *JOSA a*, 19(12):2374–2386. (cited on page 10)

Chang, C.-I. (1999). Spectral information divergence for hyperspectral image analysis. In *IEEE 1999 International Geoscience and Remote Sensing Symposium. IGARSS'99 (Cat. No. 99CH36293)*, volume 1, pages 509–511. IEEE. (cited on page 27)

Cheng, D., Prasad, D. K., and Brown, M. S. (2014). Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. *JOSA A*, 31(5):1049–1058. (cited on page 41)

Choi, K.-S., Lee, J.-S., and Ko, S.-J. (1999). New autofocusing technique using the frequency selective weighted median filter for video cameras. *IEEE transactions on Consumer Electronics*, 45(3):820–827. (cited on page 6)

Cok, D. R. (1987). Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal. US Patent 4,642,678. (cited on page 8)

Commission, I. E. et al. (1999). Iec 61966-2-1: Multimedia systems and equipment-colour measurement and management-part 2-1: Colour management-default rgb colour space-srgb. *Geneva, Switzerland: International Electrotechnical Commission*. (cited on page 9)

Cui, K., Jin, Z., and Steinbach, E. (2018). Color image demosaicking using a 3-stage convolutional neural network structure. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2177–2181. IEEE. (cited on page 8)

Domislović, I., Vrsnak, D., Subašić, M., and Lončarić, S. (2021). Outdoor daytime multi-illuminant color constancy. In *2021 12th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 270–275. IEEE. (cited on page 9)

EPFL (2022). Camera sensitivities. `https://www.epfl.ch/labs/ivrl/intranet/page-74944-en-html/page-74946-en-html/`, Last accessed on 2022-6-12. (cited on page 49)

Ershov, E., Savchik, A., Semenkov, I., Banić, N., Belokopytov, A., Senshina, D., Koščević, K., Subašić, M., and Lončarić, S. (2020). The cube++ illumination estimation dataset. *IEEE Access*, 8:227511–227527. (cited on page 41)

Finlayson, G. D. and Trezzi, E. (2004). Shades of gray and colour constancy. In *Color and Imaging Conference*, volume 2004, pages 37–41. Society for Imaging Science and Technology. (cited on pages 8, 10, and 20)

Fukushima, T., Kobayashi, Y., Hirasawa, K., Bandoh, T., Ejiri, M., and Kuwahara, H. (1983). An image signal processor. In *1983 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, volume 26, pages 258–259. IEEE. (cited on page 5)

Funt, B. and Shi, L. (2010). The rehabilitation of maxrgb. In *Color and imaging conference*, volume 2010, pages 256–259. Society for Imaging Science and Technology. (cited on page 8)

Gao, S., Han, W., Yang, K., Li, C., and Li, Y. (2014). Efficient color constancy with local surface reflectance statistics. In *European Conference on Computer Vision*, pages 158–173. Springer. (cited on page 10)

Gehler, P. V., Rother, C., Blake, A., Minka, T., and Sharp, T. (2008). Bayesian color constancy revisited. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE. (cited on page 41)

Gijsenij, A., Gevers, T., and Van de Weijer, J. (2007). Color constancy by derivative-based gamut mapping. In *PACV-1st International Workshop on Photometric Analysis For Computer Vision*, pages 8–p. INRIA. (cited on page 8)

Gijsenij, A., Gevers, T., and Van De Weijer, J. (2010). Generalized gamut mapping using image derivative structures for color constancy. *International Journal of Computer Vision*, 86(2):127–139. (cited on page 10)

Gunturk, B. K., Altunbasak, Y., and Mersereau, R. M. (2002). Color plane interpolation using alternating projections. *IEEE transactions on image processing*, 11(9):997–1013. (cited on page 8)

Hamada, M. (2015). Imaging device including phase detection pixels arranged to perform capturing and to detect phase difference. US Patent 9,197,807. (cited on page 6)

Harris, W. (2000). Astigmatism. *Ophthalmic and physiological Optics*, 20(1):11–30. (cited on page 21)

Hasinoff, S. W. (2014). Photon, poisson noise. *Computer Vision, A Reference Guide*, 4. (cited on page 16)

Hernandez-Juarez, D., Parisot, S., Busam, B., Leonardis, A., Slabaugh, G., and McDonagh, S. (2020). A multi-hypothesis approach to color constancy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2270–2280. (cited on page 8)

Hidaka, H., Monno, Y., and Okutomi, M. (2020). Spectral reflectance estimation using projector with unknown spectral power distribution. In *Color and Imaging Conference*, volume 2020, pages 205–209. Society for Imaging Science and Technology. (cited on page 11)

Hold-Geoffroy, Y., Sunkavalli, K., Hadap, S., Gambaretto, E., and Lalonde, J.-F. (2017). Deep outdoor illumination estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7312–7321. (cited on page 26)

Hou, H. and Andrews, H. (1978). Cubic splines for image interpolation and digital filtering. *IEEE Transactions on acoustics, speech, and signal processing*, 26(6):508–517. (cited on page 7)

Howley, T., Madden, M. G., O'Connell, M.-L., and Ryder, A. G. (2005). The effect of principal component analysis on machine learning accuracy with high dimensional spectral data. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 209–222. Springer. (cited on page 35)

Hu, H. (2008). Poisson distribution and application. *A Course in Department of Physics and Astronomy; University of Tennessee at Knoxville: Knoxville, TN, USA*. (cited on page 16)

Johnson, M. K. and Farid, H. (2006). Exposing digital forgeries through chromatic aberration. In *Proceedings of the 8th workshop on Multimedia and security*, pages 48–55. (cited on page 21)

Joze, H. R. V. and Drew, M. S. (2013). Exemplar-based color constancy and multiple illumination. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):860–873. (cited on page 10)

Karaimer, H. C. and Brown, M. S. (2018). Improving color reproduction accuracy on cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6440–6449. (cited on pages 14 and 15)

Khan, H. A., Thomas, J.-B., Hardeberg, J. Y., and Laligant, O. (2017). Illuminant estimation in multispectral imaging. *JOSA A*, 34(7):1085–1098. (cited on page 11)

Kim, D., Kim, J., Nam, S., Lee, D., Lee, Y., Kang, N., Lee, H.-E., Yoo, B., Han, J.-J., and Kim, S. J. (2021). Large scale multi-illuminant (lsmi) dataset for developing white balance algorithm under mixed illumination. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2410–2419. (cited on page 9)

Kim, J.-M., Son, C.-H., Lee, C.-H., and Ha, Y.-H. (2006). Illuminant adaptive color reproduction based on lightness adaptation and flare for mobile phone. In *2006 International Conference on Image Processing*, pages 1513–1516. IEEE. (cited on page 7)

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. (cited on page 36)

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., and Inman, D. J. (2021). 1d convolutional neural networks and applications: A survey. *Mechanical systems and signal processing*, 151:107398. (cited on page 26)

Kitanovski, V., Thomas, J.-B., and Yngve Hardeberg, J. (2021). Reflectance estimation from snapshot multispectral images captured under unknown illumination. In *Color and Imaging Conference*, volume 2021, pages 264–269. Society for Imaging Science and Technology. (cited on page 11)

Koomen, M., Tousey, R., and Scolnik, R. (1949). The spherical aberration of the eye. *JOSA*, 39(5):370–376. (cited on page 21)

Košcevic, K., Banic, N., and Loncaric, S. (2019). Color beaver: Bounding illumination estimations for higher accuracy. In *Proc. 14th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, pages 183–190. (cited on page 10)

Krotkov, E. (1988). Focusing. *International Journal of Computer Vision*, 1(3):223–237. (cited on page 6)

Kruse, F. A., Lefkoff, A., Boardman, J., Heidebrecht, K., Shapiro, A., Barloon, P., and Goetz, A. (1993). The spectral image processing system (sips)—interactive visualization and analysis of imaging spectrometer data. *Remote sensing of environment*, 44(2-3):145–163. (cited on page 27)

Laakom, F., Passalis, N., Raitoharju, J., Nikkanen, J., Tefas, A., Iosifidis, A., and Gabbouj, M. (2020). Bag of color features for color constancy. *IEEE Transactions on Image Processing*, 29:7722–7734. (cited on page 25)

Land, E. H. (1977). The retinex theory of color vision. *Scientific american*, 237(6):108–129. (cited on page 10)

Li, Y., Fu, Q., and Heidrich, W. (2021). Multispectral illumination estimation using deep unrolling network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2672–2681. (cited on page 11)

Lindley, D. (1959). Information theory and statistics. solomon kullback. new york: John wiley and sons, inc.; london: Chapman and hall, ltd.; 1959. pp. xvii, 395. 12.50. (cited on page 27)

Lo, Y.-C., Chang, C.-C., Chiu, H.-C., Huang, Y.-H., Chen, C.-P., Chang, Y.-L., and Jou, K. (2021). Clcc: Contrastive learning for color constancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8053–8063. (cited on page 8)

Lombardi, S. and Nishino, K. (2012). Reflectance and natural illumination from a single image. In *European Conference on Computer Vision*, pages 582–595. Springer. (cited on page 10)

Losson, O., Macaire, L., and Yang, Y. (2010). Comparison of color demosaicing methods. In *Advances in Imaging and electron Physics*, volume 162, pages 173–265. Elsevier. (cited on page 18)

Lou, Z., Gevers, T., Hu, N., Lucassen, M. P., et al. (2015). Color constancy by deep learning. In *BMVC*, pages 76–1. (cited on page 10)

Malvar, H. S., He, L.-w., and Cutler, R. (2004). High-quality linear interpolation for demosaicing of bayer-patterned color images. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages iii–485. IEEE. (cited on page 18)

Marques, B. A. D., Clua, E. W. G., Montenegro, A. A., and Vasconcelos, C. N. (2022). Spatially and color consistent environment lighting estimation using deep neural networks for mixed reality. *Computers & Graphics*, 102:257–268. (cited on page 26)

Mei, Y., Fan, Y., Zhang, Y., Yu, J., Zhou, Y., Liu, D., Fu, Y., Huang, T. S., and Shi, H. (2020). Pyramid attention networks for image restoration. *arXiv preprint arXiv:2004.13824*. (cited on page 8)

Menon, D., Andriani, S., and Calvagno, G. (2006). Demosaicing with directional filtering and a posteriori decision. *IEEE Transactions on Image Processing*, 16(1):132–141. (cited on page 18)

Nieves, J. L., Plata, C., Valero, E. M., and Romero, J. (2008). Unsupervised illuminant estimation from natural scenes: an rgb digital camera suffices. *Applied Optics*, 47(20):3574–3584. (cited on page 11)

Ohno, Y. (2000). Cie fundamentals for color measurements. In *NIP & Digital Fabrication Conference*, volume 2000, pages 540–545. Society for Imaging Science and Technology. (cited on page 9)

Parmar, M. and Reeves, S. J. (2004). A perceptually based design methodology for color filter arrays [image reconstruction]. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages iii–473. IEEE. (cited on page 5)

Pearson, K. (1896). Vii. mathematical contributions to the theory of evolution.—iii. regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, (187):253–318. (cited on page 26)

Rashkovskiy, O. and Macy, W. (2001). Method of determining missing color values for pixels in a color filter array. US Patent 6,181,376. (cited on page 6)

Robles-Kelly, A. and Wei, R. (2018). A convolutional neural network for pixelwise illuminant recovery in colour and spectral images. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 109–114. IEEE. (cited on page 11)

Romero, J., Garcıa-Beltrán, A., and Hernández-Andrés, J. (1997). Linear bases for representation of natural and artificial illuminants. *JOSA A*, 14(5):1007–1014. (cited on page 27)

Rowlands, D. A. (2020). Color conversion matrices in digital cameras: a tutorial. *Optical Engineering*, 59(11):110801. (cited on page 22)

Sadeghipoor, Z., Lu, Y. M., and Süsstrunk, S. (2012). Optimum spectral sensitivity functions for single sensor color imaging. In *Digital photography VIII*, volume 8299, pages 26–39. SPIE. (cited on page 14)

Shi, W., Loy, C. C., and Tang, X. (2016). Deep specialized network for illuminant estimation. In *European conference on computer vision*, pages 371–387. Springer. (cited on page 10)

Smith, S. G. (2005). Defect correction in electronic imaging systems. US Patent 6,970,194. (cited on page 6)

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222. (cited on page 26)

Standard, E. (2010). 1288, standard for characterization of image sensors and cameras. *European Machine Vision Association*, 3. (cited on page 17)

Stanikunas, R., Vaitkevicius, H., and Kulikowski, J. J. (2004). Investigation of color constancy with a neural network. *Neural Networks*, 17(3):327–337. (cited on page 8)

Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media. (cited on page 43)

Tan, R., Zhang, K., Zuo, W., and Zhang, L. (2017). Color image demosaicking via deep residual learning. In *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, pages 793–798. (cited on page 8)

Torrey, L. and Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global. (cited on page 53)

Van De Weijer, J., Gevers, T., and Gijsenij, A. (2007). Edge-based color constancy. *IEEE Transactions on image processing*, 16(9):2207–2214. (cited on page 10)

Xiong, W. and Funt, B. (2006). Estimating illumination chromaticity via support vector regression. *Journal of Imaging Science and Technology*, 50(4):341–348. (cited on page 10)

Yamanaka, S. (1977). Solid state camera. *United States Patent 4,054,906*. (cited on page 5)

Yang, K.-F., Gao, S.-B., and Li, Y.-J. (2015). Efficient illuminant estimation for color constancy using grey pixels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2254–2263. (cited on page 10)

Zhang, X.-S., Gao, S.-B., Li, R.-X., Du, X.-Y., Li, C.-Y., and Li, Y.-J. (2016). A retinal mechanism inspired color constancy model. *IEEE Transactions on Image Processing*, 25(3):1219–1232. (cited on page 10)

Zhang, Y., Li, K., Li, K., Zhong, B., and Fu, Y. (2019). Residual non-local attention networks for image restoration. *arXiv preprint arXiv:1903.10082*. (cited on page 8)

Zheng, Y., Lin, S., Kambhamettu, C., Yu, J., and Kang, S. B. (2008). Single-image vignetting correction. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2243–2256. (cited on page 21)

List of Figures

List of Tables