

Hyperspectral Color Constancy in Low Dimensions

Low-Dimensional Spectral Representations for Illuminant Estimation

Gunnar Dofri Viðarsson



Abstract

Illuminant estimation aims to recover the scene illumination to achieve color constancy, i.e., an illuminant-independent representation of surface colors. This problem is inherently ill-posed, since many combinations of surface reflectance and illumination can produce the same measurements. Hyperspectral imaging reduces this ambiguity by capturing richer spectral information, but its high dimensionality makes it difficult to process and motivates compact spectral representations.

In this thesis, we study how low-dimensional projections of hyperspectral data compare to RGB for illuminant estimation. Using the KAUST-MIE hyperspectral reflectance image dataset, we synthetically relight scenes with CIE standard illuminants and simulate RGB observations using camera spectral sensitivities. We evaluate a classical method (Color by Correlation) and a deep model (FC4) on RGB and reduced hyperspectral representations. For Color by Correlation, PCA-, NNMF-, and LDA-based projections consistently outperform RGB, with strong results already at two to three components. In contrast, for FC4, learned three-channel projections do not yield consistent gains over RGB, and RGB pretraining provides only small and unstable improvements. Overall, low-dimensional hyperspectral projections substantially benefit statistical illuminant estimation, while deep models may require different architectures or more data to reliably exploit spectral inputs.

Contents

Abstract	i
1 Introduction	1
1.1 Background and Motivation	1
1.2 Challenges of Illuminant Estimation	1
1.3 Hyperspectral Imaging for Color Constancy	2
1.4 Existing Approaches to Illuminant Estimation	2
1.5 Scope and Assumptions	3
1.6 Goals and Research Questions	3
1.7 Contributions	3
1.8 Thesis Organization	4
2 Related Work	5
2.1 Classical Color Constancy	5
2.2 Learning-Based Illuminant Estimation from RGB Images	6
2.2.1 Histogram-Based Convolutional Methods	6
2.2.2 Spatially Aware Convolutional Methods	7
2.3 Hyperspectral Approaches to Illuminant Estimation	7
2.4 Spectral Representations and Dimensionality Reduction	8
2.5 Summary	9
3 Problem Formulation	11
3.1 Image Formation Model	11
3.2 Color Correction via the von Kries Model	12
3.3 Illuminant Estimation is Ill-Posed	13
3.4 Task Definitions	14
3.4.1 Illuminant Classification	14
3.4.2 Continuous Illuminant Estimation	14
3.4.3 Bridging the two approaches	14
3.5 Output Representations	15
3.5.1 White Point Representations	15
3.5.2 Spectral Representations	15
3.6 Dimensionality Reduction	15
3.7 Evaluation Metrics	15

Contents

3.8 Summary	16
4 Data	17
4.1 Hyperspectral Images	17
4.2 Illuminant Spectral Power Distributions	17
4.3 Camera Sensitivities	18
5 Methodology	21
5.1 Data Generation and Pre-processing	21
5.1.1 Whiteboard masking and inpainting	21
5.1.2 Synthetic Relighting	22
5.1.3 Illuminant Assignment	23
5.1.4 Training, testing and validation splits	25
5.2 Dimensionality Reduction Methods	25
5.2.1 Camera RGB Reduction (RGB)	26
5.2.2 Principal Component Analysis (PCA)	26
5.2.3 Non-Negative Matrix Factorization (NNMF)	26
5.2.4 Linear Discriminant Analysis (LDA)	27
5.3 Illuminant Estimation Method Implementations	28
5.3.1 Grey-World (GW)	28
5.3.2 White-Patch (WP)	28
5.3.3 Color by Correlation	28
5.3.4 FC4	32
5.4 Experiments	32
5.4.1 Dimensionality Reduction Study with CbC	33
5.4.2 Deep Learning Improvement Study with FC4	34
5.4.3 Learned Projection Transfer Study	34
6 Results	35
6.1 Dimensionality Reduction Study with CbC	35
6.1.1 Analysis of Learned Bases	36
6.2 Deep Learning Improvement Study with FC4	37
6.3 Learned Projection Transfer Study	39
7 Conclusions	43
7.1 Answers to Research Questions	43
7.2 Limitations	44
7.3 Implications	45
7.4 Future Work	45
A Numeric Results & Model Architecture	47
Bibliography	53

1 Introduction

1.1 Background and Motivation

In any environment where there are lights, the observed colors of surfaces will be influenced by the color of that illumination. This illumination-induced color cast gives rise to a fundamental problem in computer vision: recovering the intrinsic color of surfaces independent of the scene illuminant.

People are largely insensitive to changes in illumination due to the human visual system's ability to adapt to different lighting conditions, discounting the illuminant information. This process is called chromatic adaptation [5]. Digital imaging systems, however, lack this innate mechanism. This causes an image of a scene captured under a non-neutral illuminant to appear inconsistent with the human perception of the same scene. Correcting for the color cast is therefore an essential step in image processing pipelines, especially for photography where this process is called white balancing. Similar issues arise in industrial and scientific imaging, where stable color appearance is required for inspection and analysis [36], and where hyperspectral sensors are sometimes used to obtain more reliable spectral information [9]. The problem of removing the illuminant color is called *color constancy* [5]. It aims to find an illuminant-agnostic color representation of the surfaces in an image. Many methods first estimate the scene illuminant, a task known as *illuminant estimation*. This thesis focuses on this task.

1.2 Challenges of Illuminant Estimation

Illuminant estimation is inherently ambiguous. The observed image depends on both the surface reflectances and the scene illuminant, with many combinations producing the same results. The problem is made worse by the RGB image sensors that integrate over broad wavelength ranges, collapsing much of the underlying spectral structure. A detailed discussion of ill-posedness is provided in Section 3.3.

1.3 Hyperspectral Imaging for Color Constancy

Hyperspectral imaging captures scene radiance in many narrow spectral bands, producing a per-pixel sampled spectrum rather than three broad RGB measurements. Compared to RGB, this preserves more spectral structure, reducing the information loss from integrating over wide sensor sensitivities. As a result, hyperspectral observations can better distinguish between illuminant spectra and surface reflectances that may be ambiguous in RGB space.

Beyond improving white balancing of RGB images, illuminant estimation in hyperspectral imaging is important in its own right. Many hyperspectral applications require accurate surface reflectance recovery [9], which depends on knowing the illuminant spectrum.

One way in which RGB data is superior is the compactness of representation. Hyperspectral images have many more channels than RGB images, making them larger and often less efficient to process, particularly for methods whose memory or runtime scale poorly with dimensionality. However, hyperspectral bands are highly correlated, making it possible to represent hyperspectral signals in a lower-dimensional space using dimensionality reduction [26]. This motivates studying how low-dimensional representations of hyperspectral data affect illuminant estimation performance, and which reductions best preserve the information needed for this task.

1.4 Existing Approaches to Illuminant Estimation

A wide range of methods has been proposed for illuminant estimation. In controlled settings, the problem can be simplified by placing a known color reference (e.g., a gray card) in the scene, allowing the illuminant to be inferred directly. However, such references are often impractical in practice, motivating automatic approaches that estimate the illuminant from image statistics or learned image features.

Most established illuminant estimation methods are developed for standard RGB images, where the sensor measurements are limited to three broad spectral channels. Fewer works explicitly study illuminant estimation in hyperspectral settings, or compare RGB and hyperspectral performance under controlled conditions. This thesis therefore focuses on evaluating how additional spectral information and low-dimensional hyperspectral representations affect illuminant estimation across both statistical and learning-based methods. Prior work and existing methods are discussed in detail in Chapter 2.

This thesis uses a representative statistical method, *Color by Correlation* (CbC) [13], and a deep learning method, *Fully Convolutional Color Constancy with Confidence-weighted Pooling* (FC4) [16], to study how different input representations affect illuminant estimation performance.

1.5 Scope and Assumptions

This thesis evaluates illuminant estimation under a controlled image formation setting. We assume a single, global scene illuminant and Lambertian surfaces, and we neglect effects such as interreflections between surfaces and non-linear camera responses. Hyperspectral reflectance images are synthetically relit with known illuminant spectra, enabling direct comparisons between RGB inputs (simulated via measured camera spectral sensitivities) and reduced hyperspectral representations. The formal image formation model is given in Chapter 3.

1.6 Goals and Research Questions

The main goal of this thesis is to evaluate how hyperspectral observations, and in particular low-dimensional representations, affect the performance of illuminant estimation methods. Under this setup, we can directly compare RGB and hyperspectral inputs and systematically vary spectral dimensionality.

The research questions addressed in this thesis are:

- **RQ1:** How much does illuminant estimation performance improve when moving from RGB representations to hyperspectral representations?
- **RQ2:** How does illuminant estimation performance change as the dimensionality of hyperspectral data is varied?
- **RQ3:** How do different dimensionality reduction methods compare in preserving the information needed for illuminant estimation?
- **RQ4:** Under pretrained and randomly initialized RGB backbones, do learned three-channel projections of hyperspectral data improve deep illuminant estimation?
- **RQ5:** Can a learned linear projection from a deep learning setting be transferred to improve statistical illuminant estimation?

1.7 Contributions

This thesis makes the following contributions:

- An evaluation of illuminant estimation performance using low-dimensional hyperspectral representations compared to RGB data under controlled conditions.
- A quantitative study of how illuminant estimation performance changes as hyperspectral data is reduced to different numbers of dimensions.

Chapter 1. Introduction

- A quantitative comparison of different dimensionality reduction methods for illuminant estimation on reduced hyperspectral representations.
- An evaluation of a learning-based illuminant estimation model on reduced hyperspectral inputs, including adaptation of an RGB-trained network backbone.
- An analysis of how learned linear reductions from deep learning models can be transferred to improve statistical methods.

1.8 Thesis Organization

This thesis is organized as follows. Chapter 2 reviews related work on illuminant estimation, covering both classical statistical approaches and modern learning-based methods, along with spectral representations. Chapter 3 presents the problem formulation, including the image formation model, assumptions, and evaluation metrics used throughout the thesis. Chapter 4 describes the data used in the experiments, including hyperspectral reflectance images, illuminant spectra, and camera sensitivity functions. Chapter 5 details the methodology, outlining the illuminant estimation approaches used and the dimensionality reduction techniques applied to hyperspectral data, and describing the experimental setup. Chapter 6 presents the experimental results and analysis, comparing performance across input representations and methods. Finally, Chapter 7 discusses the findings, limitations, and directions for future work.

2 Related Work

This chapter summarizes prior works on illuminant estimation and color constancy that are most relevant to this thesis. We first review classical statistical methods and learning-based RGB approaches, and then discuss hyperspectral methods and the low-dimensional spectral models that motivate the dimensionality reduction experiments in later chapters.

2.1 Classical Color Constancy

This section describes early, basic color constancy methods. They are based on very simple image statistics and are quick to compute but have some limitations.

Gray World. The Gray World method is based on the assumption that the average surface reflectance in a scene is achromatic [8], i.e., that the world is gray on average. Under this assumption, any deviation of the image's average color from gray is attributed to the illuminant, and can be used to estimate its chromaticity. The main limitation is that many scenes violate the assumption (e.g., scenes dominated by a single color, such as nature scenes with a lot of green), which can lead to incorrect color correction.

White Patch. The White Patch method [23] assumes that the scene contains a white surface. Such a surface is expected to produce the highest responses in the red, green, and blue channels, and the chromaticity of these maxima is taken as an estimate of the illuminant. A common modification to reduce sensitivity to noise is to use the top n -th percentile of each channel rather than the single maximum. A limitation of this method is the strong assumption that a white surface (or sufficiently strong highlight) is present in the scene.

Gamut Mapping. Gamut mapping [15] is based on the assumption that, while the 3-D RGB space of possible color signals is theoretically infinite, the range of colors observed in real scenes is not. Moreover, the set (or gamut) of observed colors changes with the scene illuminant. By assembling a diverse set of surface reflectances and imaging them under different illuminants, one can estimate a gamut for each illuminant, typically represented as a convex hull in RGB space. Given a new image, the illuminant is then estimated by finding which illu-

Chapter 2. Related Work

minant gamut best contains the image's pixel colors. This method requires both a varied set of reflectances and a predefined set of illuminants to construct the gamuts, and as a classification approach it can only predict illuminants for which a gamut has been defined.

Color by Correlation. Similar to gamut mapping, Color by Correlation [13] replaces hard gamut constraints with a correlation scoring of color statistics. For each illuminant, it models the distribution of observed colors as a histogram in a 2-D chromaticity space. Given an input image, each pixel chromaticity is scored under each illuminant histogram, and the per-pixel log-likelihoods are summed to obtain an overall score. The illuminant with the highest score is selected. This can be more robust than gamut methods when the image colors are consistent with multiple illuminant gamuts, since it exploits a statistical prior rather than only containment. However, it can be sensitive to biases in the data used to estimate the histograms, and like other classification approaches it can only predict among the illuminants represented in its model. In this thesis, we use Color by Correlation as a representative statistical baseline. Implementation details are described in Section 5.3.3.

Overall, these classical methods provide simple and efficient baselines, but rely on strong assumptions about scene content. Since they do not exploit higher-level spatial structure or semantic cues, they tend to break down in scenes that violate these assumptions.

2.2 Learning-Based Illuminant Estimation from RGB Images

This section reviews learning-based illuminant estimation methods for RGB images. In contrast to classical approaches based on simple image statistics, these methods learn features from data and, in many cases, leverage local spatial structure to improve robustness.

2.2.1 Histogram-Based Convolutional Methods

Convolutional Color Constancy (CCC). CCC [3] formulates illuminant estimation as a localization problem on a two-dimensional log-chromaticity histogram. In this formulation, changes in illuminant become translations of the histogram. Convolutional filters are learned to score locations in this space. The illuminant chromaticity is taken to be the highest scoring location. Color correction is then achieved by shifting the illuminant to a desired white point.

Fast Fourier Color Constancy (FFCC). FFCC [4] extends CCC with an implementation based on the fast Fourier transform, improving efficiency while following the same basic principle.

Cross-Camera Convolutional Color Constancy (C5). C5 [1] further improves CCC-style methods to support cross-camera generalization, aiming to increase robustness to camera variability while working under the same basic principle.

The key limitation of these methods is that, like classical approaches, they discard spatial and semantic structure and rely only on chromaticity distributions. Limited spatial information

2.3 Hyperspectral Approaches to Illuminant Estimation

can be introduced through simple prefiltering of images (e.g., blurring or edge enhancement), but they do not explicitly model semantic cues.

2.2.2 Spatially Aware Convolutional Methods

This section covers methods that explicitly model spatial structures.

Color Constancy using CNNs. Color Constancy using CNNs [6] was one of the first methods to use convolutional neural networks to regress the illuminant color directly from an input image. The method uses a single convolutional layer followed by pooling and a regression head to predict the global illuminant. It demonstrated the usefulness of learning-based methods for color constancy.

Fully Convolutional Color Constancy with Confidence (FC4). More complex models followed, such as FC4 [16]. A fully convolutional network, initialized from a pretrained backbone such as AlexNet [21] or SqueezeNet [17], produces per-area illuminant predictions along with confidence values to represent reliability of each prediction. Confidence-weighted pooling is used to calculate the global illuminant estimate. By explicitly modeling spatial and semantic features, FC4 improves robustness compared to earlier CNN-based methods and is a widely used baseline for learning-based color constancy methods. In this thesis, we use FC4 as a representative deep-learning baseline. Implementation details are described in Section 5.3.4.

Cascading Convolutional Color Constancy (C4). C4 [37] builds on FC4 using an FC4-like regressor in several stages, where each stage predicts an illuminant, corrects the image, and passes the corrected image to the next stage. The model is trained with a weighted loss over all stages and improves over FC4 particularly on more difficult cases.

Other Learning-Based Approaches. Beyond the methods above, many other deep-learning approaches to illuminant estimation have been proposed. Some aggregate local CNN features into a single image-level representation [22], others treat illuminant estimation as predicting among a set of representative illuminants and then combining them to obtain a final estimate [27]. Additional works explore regressing the illuminant from pretrained CNN features [29], learning feature spaces that group images by illuminant [35], using multi-stage networks that generate and select illuminant hypotheses [31], and patch-based methods that also address multiple illuminants [7].

2.3 Hyperspectral Approaches to Illuminant Estimation

This section reviews approaches that leverage hyperspectral image data for illuminant estimation. By observing scenes at more than three spectral bands, these methods aim to reduce the ambiguity inherent in RGB-based estimation and, in some cases, recover the full illuminant spectrum rather than only three-channel white points.

Chapter 2. Related Work

Illuminant and Reflectance Spectra Separation via Low-Rank Factorization. [38] formulates illuminant estimation as a low-rank matrix factorization to separate the illuminant and reflectance spectra from a hyperspectral image. The method works on the assumption that surface reflectances lie in a low-dimensional subspace and that the image scene is lit by a single spatially uniform illuminant.

Illuminant Separation in Non-Lambertian Scenes. Another non-deep-learning method is proposed in [33]. It targets hyperspectral scenes with non-Lambertian surfaces by exploiting specular reflections. It separates diffuse and specular components, and the illuminant is estimated using weighted principal component analysis. The approach is effective when specular highlights are present, but assumes spatially uniform illumination and relies on noticeable specularity in the scene.

Illuminant Estimation in Multispectral Imaging. [2] analyzes how different numbers of sensor channels, sensor sensitivity shapes and bandwidths affect illuminant estimation by classical color constancy methods such as Gray World, Max-RGB, Shades of Gray, and Gray Edge. While multispectral extensions can improve performance over RGB in controlled settings, results are highly dependent on the sensitivity shape and noise, with diminishing returns beyond a moderate number of spectral bands.

Pixelwise Illuminant Recovery for Color and Spectral Images. [30] proposes a CNN architecture that can be applied to either RGB or hyperspectral inputs, and predicts a per-pixel illuminant (a white point for RGB, or a spectrum for hyperspectral data). The model is trained using an angular-error loss, and the authors evaluate its RGB and hyperspectral performance on separate datasets.

Hyperspectral Illuminant Estimation Using a Deep Unrolling Network. [25] views the problem as a constrained matrix factorization problem, and employs a loop-unrolled ADMM network to solve it. They also introduce the KAUST-MIE [24] dataset of hyperspectral reflectance images used in this thesis.

2.4 Spectral Representations and Dimensionality Reduction

As noted in the introduction, a core assumption underlying several illuminant estimation approaches and central to this thesis is that reflectance and illuminant spectra lie in low-dimensional subspaces.

Low-Dimensional Modeling of Surface Reflectance. Early work showed that surface reflectance spectra can be accurately approximated using a small number of basis functions. [26] demonstrated that linear models with five to seven components can capture the majority of variation in measured surface reflectances.

Low-Dimensional Modeling of Illuminant Spectra. Similarly, natural illuminants exhibit strong low-dimensional structure. In particular, [32] demonstrated that daylight spectra can

be modeled with seven components, and the CIE D-series illuminants [10], that are commonly used as reference daylight illuminants, are constructed from three components.

Von Kries Hypothesis and Diagonal Models. A classical motivation for low-parameter illumination change models is the von Kries hypothesis [34], which models chromatic adaptation as independent per-channel scaling. [14] showed that, under low-dimensional linear modeling of illuminants and reflectances, illuminant change can be represented by a diagonal transform after a fixed change of sensor basis.

Overall, these results support the use of low-dimensional spectral representations for both reflectance and illumination. They motivate dimensionality reduction techniques that aim to preserve illumination-discriminative information while reducing computational complexity, an idea that is central to the experiments presented in this thesis.

2.5 Summary

The literature on illuminant estimation spans simple statistical baselines, deep-learning-based RGB methods, and hyperspectral approaches. Classical methods such as Gray World, White Patch, and gamut-based techniques are efficient and interpretable, but rely on strong scene assumptions and limited image statistics. Learning-based RGB approaches improve robustness by learning features from data, ranging from histogram-based formulations (CCC/FFCC/C5) to spatially aware CNN models such as FC4 and its cascading extension C4.

Hyperspectral methods reduce the ambiguity of RGB representations by providing richer spectral measurements. However, few works systematically study how much of the spectral information is actually needed for accurate illuminant estimation, or how different low-dimensional spectral representations affect performance across method classes. Motivated by prior evidence that both reflectance and illuminant spectra lie in low-dimensional subspaces, this thesis addresses this gap by evaluating illuminant estimation under controlled conditions. We vary the dimensionality and type of hyperspectral projection and use Color by Correlation and FC4 as representative statistical and deep-learning methods.

3 Problem Formulation

This chapter specifies the problem setting used throughout this thesis. We consider illuminant estimation under a controlled image formation setting, as outlined in Chapter 1, and we evaluate methods on both RGB images and hyperspectral image representations derived from the same underlying reflectance data.

Two related task formulations are considered. In *illuminant classification*, the goal is to identify the illuminant from a predefined set of known illuminant spectra. In *continuous illuminant estimation*, the goal is to estimate the scene illuminant directly, either as a spectral power distribution or via a corresponding device-specific white point. The remainder of this chapter introduces the image formation model, defines the output representations, and specifies the evaluation metrics used in the experiments.

3.1 Image Formation Model

The image formation model used in this thesis is a Lambertian one. That is, it assumes that all surfaces are perfectly diffuse. This model is a common assumption for color constancy algorithms. Equation 3.1 describes the color signal $C(\lambda, \mathbf{x}) \in \mathbb{R}$ at a given wavelength $\lambda \in \Lambda$ and location \mathbf{x} . $E(\lambda) \in \mathbb{R}$ is the illuminant spectral power distribution and $R(\lambda, \mathbf{x}) \in [0, 1]$ is the surface reflectance function at \mathbf{x} . The response of a camera sensor $p \in P$ depends on its sensitivity function $S_p(\lambda)$ and is defined by Equation 3.2, where Λ denotes the wavelength range considered and P is the set of camera sensors.

$$C(\lambda, \mathbf{x}) = E(\lambda) R(\lambda, \mathbf{x}), \quad (3.1)$$

$$I_p(\mathbf{x}) = \int_{\Lambda} E(\lambda) R(\lambda, \mathbf{x}) S_p(\lambda) d\lambda, \quad p \in P, \quad (3.2)$$

Given a hyperspectral image that samples the spectrum of a scene at d wavelength bands, a

Chapter 3. Problem Formulation

simulated RGB image for a camera with known sensor responses can be constructed using Equation 3.3. Here λ_i is the i -th sampling point and $\Delta\lambda$ is the sampling interval. In this model we assume uniform sampling across the spectrum.

$$I_p(\mathbf{x}) \approx \sum_{i=1}^d S_p(\lambda_i) E(\lambda_i) R(\lambda_i, \mathbf{x}) \Delta\lambda, \quad p \in \{R, G, B\}, \quad (3.3)$$

This is formulated with matrices in Equation 3.4. Here, $\mathbf{E} \in \mathbb{R}^d$ and $\mathbf{R}(\mathbf{x}) \in \mathbb{R}^d$ denote the discretely sampled illuminant spectrum and surface reflectance. \odot is the element-wise (Hadamard) product. The matrix $\mathbf{S} \in \mathbb{R}^{3 \times d}$ contains the sampled spectral sensitivity functions of the RGB camera. The resulting vector $\mathbf{I}_{\text{RGB}}(\mathbf{x}) \in \mathbb{R}^3$ represents the RGB color signal at pixel \mathbf{x} .

$$\mathbf{I}_{\text{RGB}}(\mathbf{x}) = \mathbf{S} (\mathbf{E} \odot \mathbf{R}(\mathbf{x})) \Delta\lambda, \quad (3.4)$$

Note that the factor $\Delta\lambda$ can be omitted from these equations, since illuminant estimation recovers the illuminant only up to an unknown scale factor.

3.2 Color Correction via the von Kries Model

As mentioned in Chapter 1, illuminant estimation is the first step in many color constancy algorithms. After estimating the illuminant, the image is color corrected. A widely used model for color correction is the von Kries model [34, 14]. The von Kries model assumes that chromatic adaptation can be approximated by scaling each sensor channel independently.

Let $\mathbf{I}(\mathbf{x}) \in \mathbb{R}^3$ denote the RGB color signal at pixel \mathbf{x} captured under an unknown illuminant with white point $\mathbf{e} \in \mathbb{R}^3$. Given an estimate $\hat{\mathbf{e}}$ of the illuminant white point, color correction is performed by

$$\mathbf{I}_{\text{corr}}(\mathbf{x}) = \mathbf{D}(\hat{\mathbf{e}}) \mathbf{I}(\mathbf{x}), \quad (3.5)$$

where $\mathbf{D}(\hat{\mathbf{e}})$ is a diagonal matrix that scales each channel inversely proportional to the estimated illuminant components.

The von Kries model connects illuminant estimation and color correction and is widely used in computational color constancy. Even though it ignores more complex adaptation effects, it is an effective approximation. In this thesis, we therefore evaluate some illuminant estimates in terms of their white point, under the assumption that accurate estimates translate to effective correction via channel-wise scaling.

The same idea extends naturally to hyperspectral data as an element-wise normalization across wavelength channels. Because hyperspectral bands are narrow and involve far less spectral mixing than RGB sensor responses, this channel-wise correction is closer to the

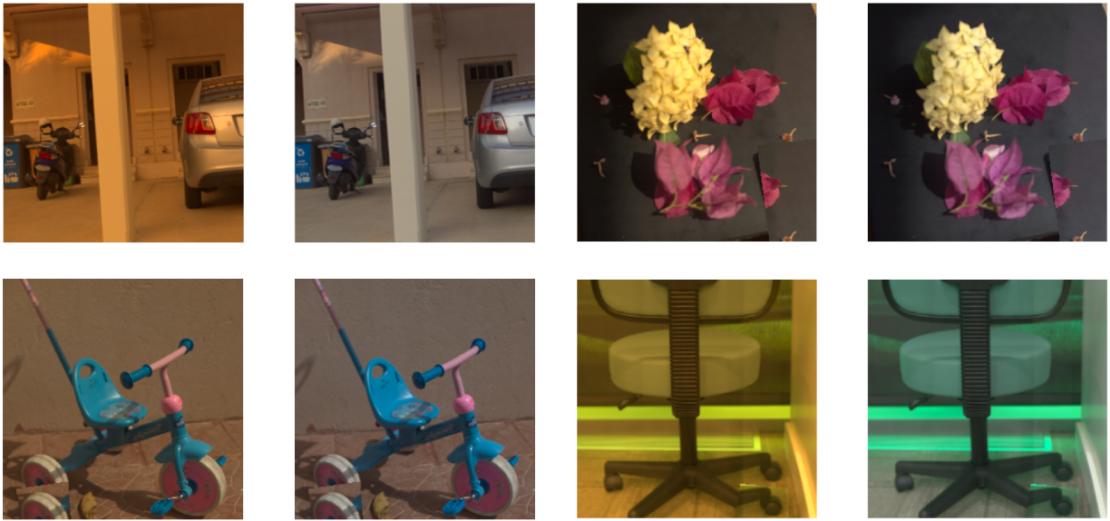


Figure 3.1: Illustrative examples of von Kries color correction using illuminant estimates from CbC on images from the KAUST-MIE dataset. For each pair, the left image is the uncorrected input and the right image is corrected to appear as if it were illuminated by D65. Both images are rendered in sRGB.

Lambertian image formation model (Eq. 3.2) and therefore a more accurate approximation.

Figure 3.1 shows illustrative examples of color correction using estimated illuminants, demonstrating the effect of the von Kries model in practice.

3.3 Illuminant Estimation is Ill-Posed

Even under the simple Lambertian model, illuminant estimation is ill-posed because the image formation process entangles surface reflectance and illumination. For an RGB image with n pixels, there are $3 \cdot n$ measured values, but the unknowns include the $3 \cdot n$ illuminant-independent surface reflectance values and, under a single global illuminant assumption, the 3 color values of the illuminant. Therefore, we have $3 \cdot n$ knowns and $3 \cdot n + 3$ unknowns, making the problem underdetermined. Many combinations of reflectances and illuminants can produce the same observed colors.

With hyperspectral images, the same argument remains. For a d -channel hyperspectral image, there are $d \cdot n$ measured values and $d \cdot n + d$ unknowns. However, hyperspectral images have two advantages over RGB images. First, RGB sensors aggregate the signal over wide portions of the spectrum, irreversibly mixing spectral information into three measurements. Hyperspectral images maintain spectral structure and reduce ambiguity introduced by sensor integration. Secondly, empirical studies have shown that real-world surface reflectance spectra can be approximated accurately using a small number of basis functions [26]. Hyperspectral images maintain enough information to project onto these basis functions, further constraining the

space of feasible illuminant and reflectance combinations.

3.4 Task Definitions

This section defines the task formulations used in this thesis. Illuminant estimation is commonly formulated either as a classification task or as a continuous estimation task.

3.4.1 Illuminant Classification

As a classification task, the goal is to predict which illuminant is lighting a scene from a predetermined illuminant set. This formulation incorporates prior knowledge of true illuminants and restricts the output space to these true illuminants. That way, arbitrarily incorrect illuminant predictions are avoided. The classification setting is therefore a robust setting to evaluate the ability of methods to discriminate between illuminants.

3.4.2 Continuous Illuminant Estimation

In the continuous setting, the goal is to estimate the illuminant in a continuous space, either as chromaticity values (white points) or as a full spectrum. Unlike in the classification setting, the output is not restricted to a predetermined illuminant set, so methods must produce a physically plausible illuminant estimate without relying on an explicit list of allowed illuminants. An advantage of this setting is that it naturally allows generalization to unseen illuminants, which is the case in most practical scenarios.

3.4.3 Bridging the two approaches

Results from these two tasks are not directly comparable. Therefore, results in this thesis will primarily be reported separately for methods within each task. However, we can take the output of a continuous method and trivially *snap* it to the closest allowed output in the classification setting, thereby giving the continuous method the same prior knowledge of the illuminants.

Let $\hat{\mathbf{e}} \in \mathbb{R}^d$ denote the illuminant predicted by a continuous method, and let $\mathcal{E} = \{\mathbf{e}_1, \dots, \mathbf{e}_K\}$ denote the set of allowed illuminants in the classification setting. We snap $\hat{\mathbf{e}}$ to the closest illuminant in \mathcal{E} by selecting the illuminant that maximizes cosine similarity:

$$k^* = \arg \max_{k \in \{1, \dots, K\}} \frac{\hat{\mathbf{e}} \cdot \mathbf{e}_k}{\|\hat{\mathbf{e}}\| \|\mathbf{e}_k\|}, \quad \mathbf{e}^* = \mathbf{e}_{k^*}. \quad (3.6)$$

where $\|\cdot\|$ denotes the ℓ_2 norm.

3.5 Output Representations

There are two main ways to represent the output of illuminant estimation methods: an illuminant white point or the illuminant spectral power distribution. The choice of representation influences what properties of the illuminant are modeled and how estimation accuracy is interpreted.

3.5.1 White Point Representations

White point representations describe the chromaticity of the illuminant. In this thesis, white points are represented as normalized RGB values measured using a specific camera's sensor sensitivities. This representation is sufficient to color-correct RGB images, but it discards spectral information and is inherently device-dependent.

3.5.2 Spectral Representations

An alternative output representation is the full illuminant spectrum. This representation is device-independent and preserves full spectral information. It is therefore sufficient to remove the illuminant effect from hyperspectral images, which is useful when converting hyperspectral radiance images to reflectance images.

3.6 Dimensionality Reduction

Hyperspectral images contain many more dimensions than RGB images. This can make hyperspectral images impractical to work with directly as they take up more space and processing resources. This makes dimensionality reduction desirable. Ideally the reductions would retain information that is relevant to the illuminant estimation problem. Reducing the dimensions of hyperspectral images introduces an inherent trade-off between compactness and information retention. Lower dimensions might improve efficiency but might lose information necessary for discrimination of illuminants. Understanding this trade-off is therefore beneficial and motivates some of the experimental design in later chapters.

3.7 Evaluation Metrics

For evaluation of illuminant estimation methods we use an angular error metric. It measures the angle between the predicted and ground-truth illuminant vectors. Both illuminant white points and illuminant spectra are represented as vectors that we only care about up to a scale factor so the angular error is an appropriate metric for both. We do note that angular errors from the two are not directly comparable.

Given a predicted illuminant vector $\hat{\mathbf{e}} \in \mathbb{R}^d$ and a ground-truth illuminant vector $\mathbf{e} \in \mathbb{R}^d$, where

Chapter 3. Problem Formulation

$d = 3$ for white point representations, the angular error is defined as

$$\theta = \arccos\left(\frac{\hat{\mathbf{e}} \cdot \mathbf{e}}{\|\hat{\mathbf{e}}\| \|\mathbf{e}\|}\right) \quad (3.7)$$

where $\|\cdot\|$ denotes the ℓ_2 norm.

For each method, we report the mean angular error aggregated over the test set described in Section 5.1.4.

We use the angular error for both classification and continuous evaluations. Even though classification does make predictions from a finite set of illuminants, we can still use the spectra or white points for these illuminants to calculate the error. Traditional classification error metrics do not have the nuance to see if a wrongly predicted illuminant is close to being the correct one or a very different illuminant. This is why those methods are not used.

3.8 Summary

This chapter introduced the main problem addressed in this thesis: illuminant estimation under a Lambertian image formation model with a single global illuminant. The problem is ill-posed, but hyperspectral images provide additional constraints on possible solutions. Two task formulations were defined, illuminant classification and continuous illuminant estimation. Methods can output illuminant white points or full spectra, and both are evaluated using an angular error metric. Having defined the problem, the next chapter describes the datasets used in this work.

4 Data

Three types of data were used for the experiments presented in this thesis. First, a dataset of hyperspectral reflectance images serves as input for all of our methods. Because these are reflectance images, we can relight them with arbitrary illuminants using the second data source: illuminant spectral power distributions (SPDs). A diverse set of illuminant SPDs is used as ground truths and for relighting hyperspectral images as input data. The third type of data we use is a set of camera sensitivities measured from a consumer RGB camera. They are used to simulate the image acquisition of an RGB camera and define device-specific white points.

4.1 Hyperspectral Images

Hyperspectral images are crucial to our experiments. We use them both as inputs for hyperspectral methods and to simulate RGB images using the camera sensitivity functions. For our experiments, we use the KAUST Multispectral Illumination Estimation (KAUST-MIE) dataset [24]. It contains 409 hyperspectral images with spatial resolution 512×512 and 31 spectral bands. The spectra are sampled uniformly at 10 nm intervals from 400–700 nm, spanning the visible spectrum. The dataset includes a diverse selection of indoor and outdoor scenes. The images include a whiteboard reference, and the effect of the scene illuminant has been removed. As a result, pixel values approximate surface reflectance functions, corresponding to $\mathbf{R}(\mathbf{x})$ in Equation 3.4. Examples from the KAUST-MIE dataset are shown in Figure 4.1.

4.2 Illuminant Spectral Power Distributions

To serve as illuminant spectra for our experiments, we chose a selection of CIE Standard Illuminant spectral power distributions (SPDs) [10]. Care was taken to pick a diverse set of illuminants that spans a variety of color temperatures and represents common lights encountered in typical modern scenes. We use a total of 28 illuminants from four families presented in Table 4.1. The CIE 1931 xy chromaticities [10] of each illuminant are shown in

Chapter 4. Data

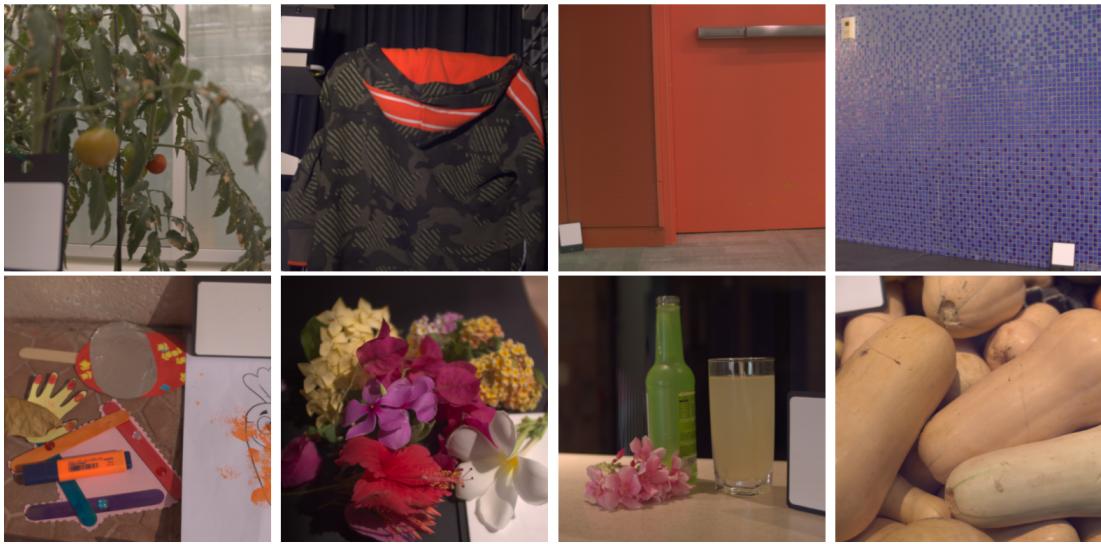


Figure 4.1: Examples of images from the KAUST-MIE hyperspectral reflectance image dataset. All images are rendered in sRGB under the D65 illuminant for visualization purposes.

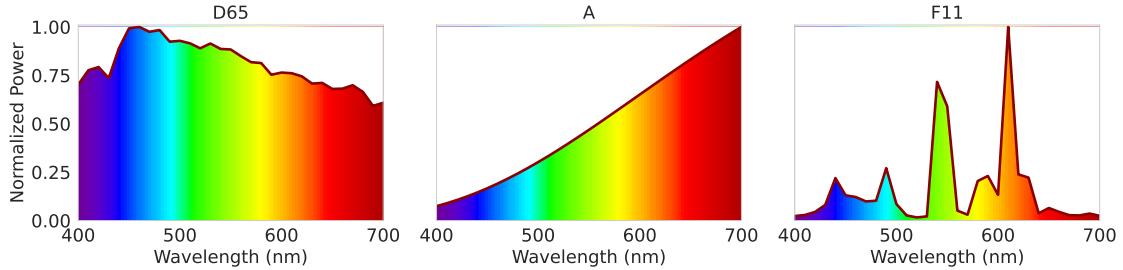


Figure 4.2: Example CIE standard illuminant spectral power distributions (SPDs) from the illuminant set used in this thesis. Each SPD is normalized to have a maximum value of 1.

Figure 4.3. The chromaticities are calculated using the CIE 1931 2° standard observer [11]. We sample the CIE-defined SPDs at the same wavelengths as our hyperspectral images. Each illuminant SPD is therefore represented by a 31-dimensional vector. In addition to the discrete illuminants listed here, we also sample continuous daylight SPDs from the CIE daylight locus for more diverse training data. Figure 4.2 shows three example illuminant SPDs from our set, illustrating the variation in spectral shape across illuminants.

4.3 Camera Sensitivities

For simulating images taken by an RGB camera we use the spectral sensitivity functions of a Canon 300D camera from the Camera Spectral Sensitivity Database [18], originally introduced by Jiang et al. [19]. The spectral sensitivity functions are plotted in Figure 4.4. The database contains measured camera sensitivities from 28 cameras. The sensitivity functions are provided as three 33-dimensional vectors, uniformly sampled at 10 nm intervals from 400–720 nm.

4.3 Camera Sensitivities

Table 4.1: CIE standard illuminants used in this thesis.

Family	Illuminants	Count
Incandescent	A	1
Daylight (D-series)	D50, D55, D60, D65, D75, D93	6
Fluorescent (F-series)	F1–F12	12
LED (LED-series)	LED-B1–B5, LED-BH1, LED-RGB1, LED-V1–V2	9
Total		28

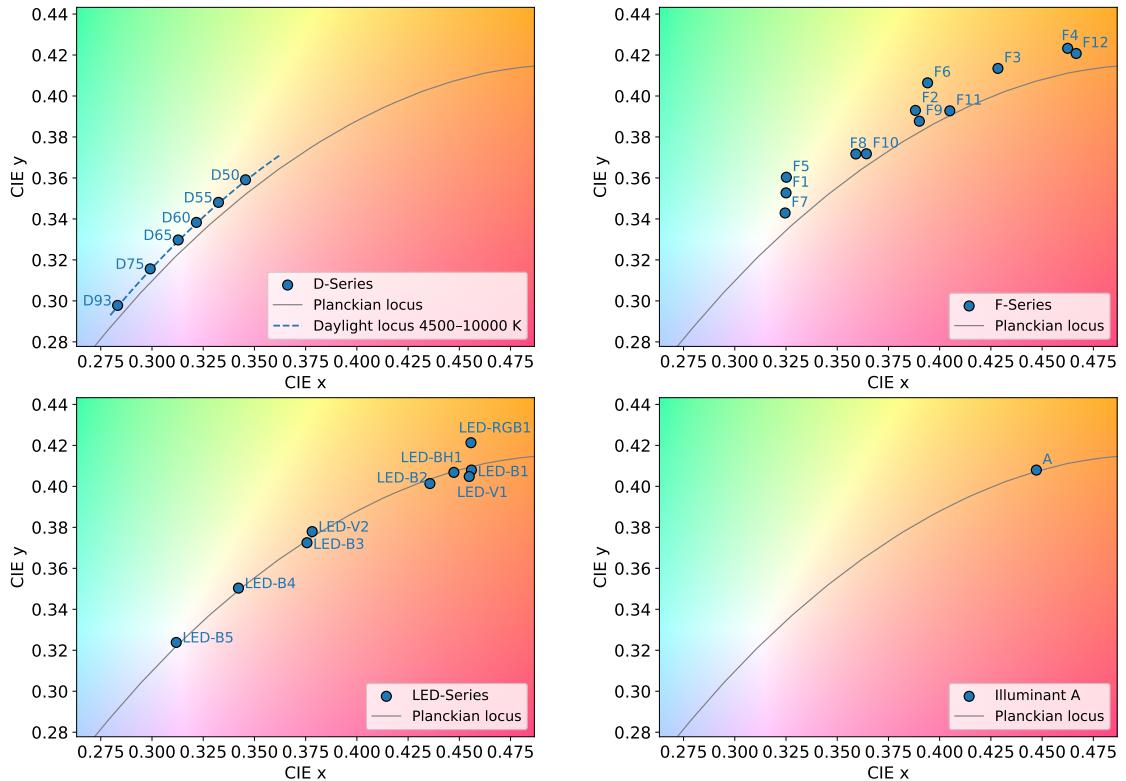


Figure 4.3: CIE 1931 xy chromaticities of the illuminants used in this thesis, grouped by family, including the daylight locus. The Planckian locus is overlaid for reference.

For our experiments, we omit the values corresponding to 710 nm and 720 nm, making the sampling aligned with our hyperspectral images and illuminant spectra.

Chapter 4. Data

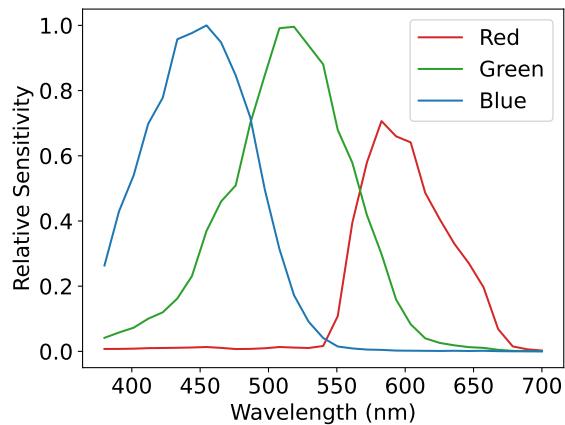


Figure 4.4: Spectral sensitivity functions of a Canon 300D camera. Normalized so the highest value across the three functions is one.

5 Methodology

This chapter describes the implementation details of the experimental setup used in this thesis. We first describe the data processing, including ground truth masking and inpainting protocols, target specifications, the synthetic relighting method and illuminant assignment strategies. We then introduce the various dimensionality reduction methods used to investigate the effect of different representations of spectral information. The illuminant estimation methods used for our experiments are introduced: Gray World [8] and White Patch [23] are simple classical methods used as baselines. Color by Correlation (CbC) [13] is a statistical method used for analyzing performance on spectral representations, and then we use a deep learning method FC4 [16] to study how popular deep-learning methods perform on spectral data. The last section of this chapter provides the specific sets of experiments conducted.

5.1 Data Generation and Pre-processing

In this section we describe the pre-processing we do to transform our data sources to create the input data and the targets for our methods.

5.1.1 Whiteboard masking and inpainting

As mentioned in Section 4.1, the hyperspectral images from the KAUST-MIE [24] dataset include a whiteboard color reference. We remove this reference to prevent information leakage. This is especially important for deep learning methods that could easily learn to identify the reference, but also for simpler statistical methods where the presence of the reference will skew the image statistics. We remove this reference by manually creating masks for each image and following the inpainting method described here.

Let the hyperspectral reflectance image be $R : \{0, \dots, H - 1\} \times \{0, \dots, W - 1\} \rightarrow \mathbb{R}^{31}$ and the binary mask be $M : \{0, \dots, H - 1\} \times \{0, \dots, W - 1\} \rightarrow \{0, 1\}$, where $M(y, x) = 1$ indicates masked pixels. We define the masked set $\Omega = \{(y, x) : M(y, x) = 1\}$. The height of the mask is thus $h = (\max_{(y, x) \in \Omega} y) - (\min_{(y, x) \in \Omega} y) + 1$. We use neighboring pixels to paint in the masked area.



Figure 5.1: Example of the masking and inpainting procedure. Shows the base image with the whiteboard reference (a). The image with the manually added mask in red (b) and the inpainted image (c).

Specifically, we choose a vertical shift of $s = h$ if the vertical center of the mask is below the center of the image, otherwise $s = -h$. To prevent out-of-bounds values, we define a reflection operator

$$r(t) = \begin{cases} t, & 0 \leq t \leq H-1, \\ 2H-t-1, & t \geq H, \\ -t-1, & t < 0. \end{cases} \quad (5.1)$$

Then, the inpainted reflectance image R' is

$$R'(y, x) = \begin{cases} R(r(y+s), x), & (y, x) \in \Omega, \\ R(y, x), & (y, x) \notin \Omega. \end{cases} \quad (5.2)$$

There are some edge-cases with very large masks where this inpainting method may fill the mask with pixels from the masked area, but that is not the case for our data. An example of this method is shown in Figure 5.1.

5.1.2 Synthetic Relighting

Another important part of creating the inputs for our methods is applying an illuminant spectrum to the hyperspectral reflectance images. We refer to this process as synthetic relighting. This is achieved with an element-wise multiplication between the illuminant SPD and the image reflectance spectrum at each pixel. Using the same notation as in Equation 3.4, the illuminated spectral image \mathbf{C} is defined by its pixel spectra, $\mathbf{C}(\mathbf{x}) \in \mathbb{R}^d$.

5.1 Data Generation and Pre-processing

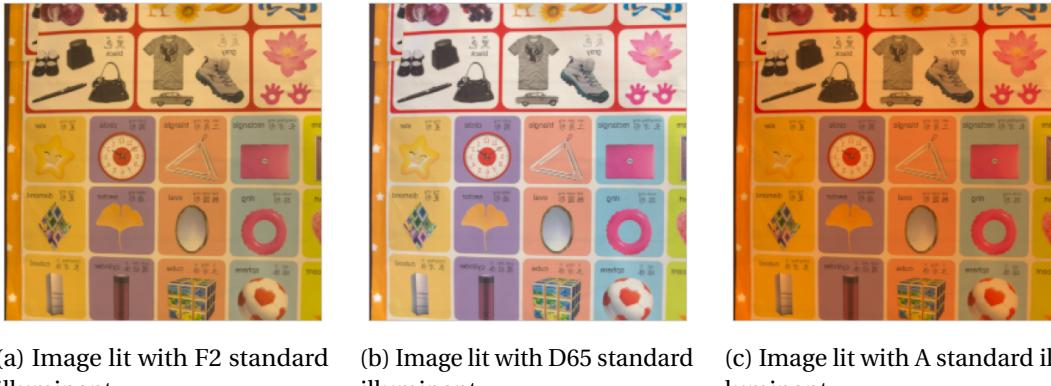


Figure 5.2: The same image lit with 3 different illuminants, rendered in sRGB.

Table 5.1: Illuminant usage and assignment strategies for the two experimental setups.

	Color by Correlation	FC4
Illuminant families	D- and F-series, A	D-, F- and LED-series, A
Number of illuminants	19	28 total (14 training, 14 evaluation)
<i>Illuminant Assignment:</i>		
Training	Exhaustive pairing	$p = 0.3$: daylight locus sampling $p = 0.7$: stratified sampling (training)
Validation	Not used	Stratified sampling (evaluation)
Test	Exhaustive pairing	Exhaustive pairing (evaluation)

$$\mathbf{C}(\mathbf{x}) = \mathbf{E} \odot \mathbf{R}(\mathbf{x}) \quad (5.3)$$

An example of an image lit with three different illuminants can be seen in Figure 5.2.

5.1.3 Illuminant Assignment

Another important step in preparing our data is choosing which illuminants are applied to which images. Since the Color by Correlation (CbC) and FC4 experiments serve different purposes, we use different illuminant sets and assignment strategies. A high-level summary is given in Table 5.1.

Illuminant sets

For the CbC experiments, we use a subset of the full illuminant pool (D- and F-series, and A). Since these experiments require a large number of runs across different parameter settings, a

Chapter 5. Methodology

Table 5.2: Dataset splitting strategies for the two experimental setups.

	Color by Correlation	FC4
Splitting strategy	80/20 holdout	4-fold cross-validation
Training images	327 (80%)	307 (75%)
Validation images	Not used	51 (12.5%)
Test images	82 (20%)	51 (12.5%)
Number of trainings	1	4

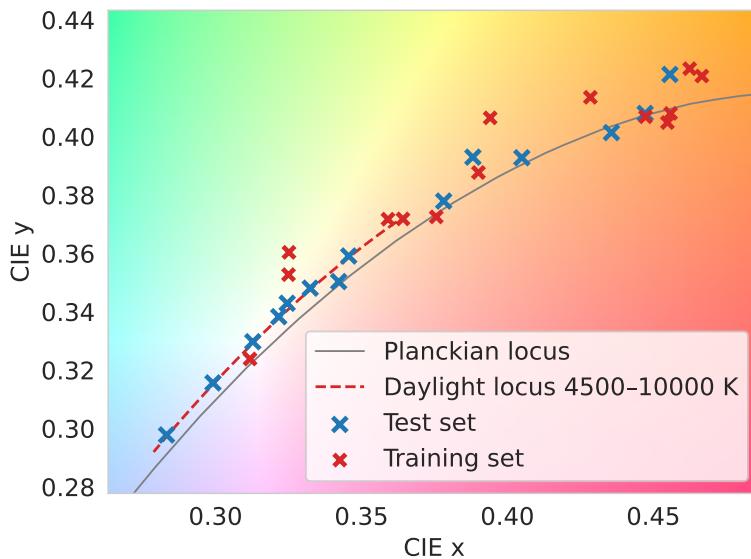


Figure 5.3: CIE 1931 xy chromaticities [10] of the illuminants used in the FC4 experiments, split into training and evaluation (validation/test) sets. The split was chosen to provide broad coverage of chromaticity space.

reduced illuminant set is used to limit computational overhead.

For the FC4 experiments, we use the full illuminant pool (D-, F-, and LED-series, and A), but we split illuminants into a *training illuminant set* and an *evaluation illuminant set* (validation/test). This enables evaluation on illuminants that are not seen during training. The two sets are listed in Table 5.3 and were selected to provide a broad spread of CIE 1931 xy chromaticities. Figure 5.3 visualizes the coverage of the training and evaluation illuminant sets.

Pairing strategy

For testing, we use exhaustive illuminant–image pairing. For CbC, we evaluate all combinations of test images and illuminants. For FC4, we evaluate all combinations of test images with all illuminants in the evaluation set.

5.2 Dimensionality Reduction Methods

Table 5.3: Deep learning illuminant split into training and evaluation sets.

Training illuminants	Evaluation illuminants (val/test)
(Continuous D-series)	D50, D55, D60, D65, D75, D93
F1, F3, F4, F5, F6	F2, F7, F11
F8, F9, F10, F12	A
LED-B1, LED-B3, LED-B5	LED-B2, LED-B4
LED-BH1, LED-V1	LED-RGB1, LED-V2
Total: 14 discrete + (continuous D-series)	Total: 14

For training, CbC also uses exhaustive pairing. For FC4 training, we use a stratified assignment scheme. With probability 0.3, a daylight illuminant is sampled uniformly along the daylight locus for correlated color temperatures in the range 4500 K–10000 K. Otherwise, an illuminant is chosen from the discrete FC4 training illuminant set using a shuffled-cycle scheme: we iterate through a random permutation of the training illuminants and reshuffle when exhausted. This ensures an even usage of discrete illuminants across training while allowing different image–illuminant pairings across epochs.

For FC4 validation, we apply the same shuffled-cycle scheme but restricted to the evaluation illuminant set (no continuous daylight sampling), using a single pass over the validation images per epoch.

5.1.4 Training, testing and validation splits

We employ two different dataset splitting schemes for our main experimental setups, as shown in Table 5.2.

For the CbC method, which is a statistical classification method, we use random 80/20 holdout split of our 409 image KAUST-MIE dataset, resulting in 327 images for training and 82 for testing. No validation set is needed for this method.

For evaluating FC4, we use 4-fold cross-validation to reduce the influence of specific train–test splits on the results. Each model is trained 4 times, each time with a different 75% of images for training, and the remaining 25% split evenly for validation and test sets. (307 training, 51 validation, and 51 testing images).

5.2 Dimensionality Reduction Methods

Hyperspectral data is high-dimensional, so projecting it to lower dimensions can improve computational efficiency and act as a controlled way to study how much even low-dimensional spectral information can improve illuminant estimation methods. In this thesis, we use four dimensionality reduction methods: RGB Sensor Reduction (RGB), Principal Component

Chapter 5. Methodology

Analysis (PCA), Non-negative Matrix Factorization (NNMF), and Linear Discriminant Analysis (LDA).

The RGB Sensor Reduction uses camera sensitivity functions (Section 4.3) as the new basis to project onto. The three other methods are all fitted to the statistics of the pixel-level spectra in the training split. That is, we discard spatial information and treat each pixel spectrum from the synthetically illuminated images (Section 5.1.2) as an independent sample. We then apply the learned transformation to reduce the data dimension from d to d' components.

5.2.1 Camera RGB Reduction (RGB)

RGB is the simplest reduction method. It uses the Canon 300D camera spectral sensitivity functions introduced in Section 4.3 to reduce the dimensionality of the hyperspectral data down to three (RGB) channels. Given a matrix of the camera sensitivity functions $S \in \mathbb{R}^{3 \times d}$ we transform the pixel spectra $x \in \mathbb{R}^d$ with

$$z = Sx, \quad z \in \mathbb{R}^3. \quad (5.4)$$

This method serves as a baseline representing raw RGB camera values.

5.2.2 Principal Component Analysis (PCA)

PCA is an unsupervised method to find the orthogonal linear transform that aligns the data in such a way that the variance is maximal around the axes. Components are ordered by their explained variance. By choosing to keep only the top d' components, it is often possible to reduce the number of dimensions considerably without losing too much information.

Let $X \in \mathbb{R}^{N \times d}$ be the matrix of training pixel spectra ($d = 31$) where N is the number of pixels in the training set. We fit PCA on that data, after centering it by the per-band mean μ , yielding the top- d' principal components $U_{d'} \in \mathbb{R}^{d \times d'}$. Each spectrum $x \in \mathbb{R}^d$ is then represented by

$$z = U_{d'}^\top (x - \mu), \quad z \in \mathbb{R}^{d'}. \quad (5.5)$$

A variant we also tried was performing PCA on the illuminant set itself. In this case, PCA is fitted to the collection of illuminant spectra used in the experiments, rather than to pixel spectra from images. We refer to this variant as *Illuminant Principal Component Analysis* (Ill. PCA).

5.2.3 Non-Negative Matrix Factorization (NNMF)

NNMF is an unsupervised method to decompose a matrix into two, usually lower-rank matrices with non-negative values. It does so by minimizing a loss function. The non-negative

nature has benefits when it comes to interpreting the results, as many physical systems, such as image sensor responses, are inherently non-negative.

Let $X \in \mathbb{R}_{\geq 0}^{N \times d}$ be the matrix of training pixel spectra ($d = 31$). NMF finds two non-negative matrices $U \in \mathbb{R}_{\geq 0}^{N \times d'}$ and $V \in \mathbb{R}_{\geq 0}^{d' \times d}$ such that

$$X \approx UV. \quad (5.6)$$

In our experiments, we used scikit-learn's MiniBatchNMF with default parameters, using Frobenius loss:

$$\min_{U, V \geq 0} \|X - UV\|_F^2. \quad (5.7)$$

After fitting, we can transform each pixel spectrum $x \in \mathbb{R}^d$, with the basis V fixed, into a reduced representation $z \in \mathbb{R}^{d'}$ by solving

$$z = \arg \min_{z \geq 0} \|x - zV\|_2^2. \quad (5.8)$$

5.2.4 Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis is a supervised dimensionality reduction method that uses class labels to find a linear projection that maximizes inter-class scatter and minimizes intra-class scatter.

In our setting, each training pixel spectrum $x \in \mathbb{R}^d$ is associated with an illuminant label $y \in \{1, \dots, C\}$ where C is the number of illuminants.

LDA finds a projection matrix $W \in \mathbb{R}^{d \times d'}$ that maximizes the ratio of inter-class scatter to intra-class scatter. Let μ be the global mean, μ_c be the mean of class c , and n_c be the number of samples in class c . The inter-class and intra-class scatter matrices are

$$S_W = \sum_{c=1}^C \sum_{i \in c} (x_i - \mu_c)(x_i - \mu_c)^\top, \quad S_B = \sum_{c=1}^C n_c (\mu_c - \mu)(\mu_c - \mu)^\top. \quad (5.9)$$

The projection is obtained by maximizing Fisher's criterion,

$$\max_W \frac{|W^\top S_B W|}{|W^\top S_W W|}, \quad (5.10)$$

and each spectrum is mapped to $z \in \mathbb{R}^{d'}$ by

$$z = W^\top x. \quad (5.11)$$

PCA, NMF, and LDA were implemented using the following functions from scikit-learn (ver-

Chapter 5. Methodology

sion 1.7.2) [28] with default parameters:

- PCA: `sklearn.decomposition.IncrementalPCA`
- NMF: `sklearn.decomposition.MiniBatchNMF`
- LDA: `sklearn.discriminant_analysis.LinearDiscriminantAnalysis`

5.3 Illuminant Estimation Method Implementations

5.3.1 Grey-World (GW)

The classic Grey-World (GW) method works on the assumption that the average reflectance in a scene is achromatic. It is typically employed on RGB images but the implementation can trivially be extended to hyperspectral images. With the Grey-World method, the illuminant chromaticity is simply the mean of each individual channel.

So given an illuminated reflectance image $\mathbf{C} : \{0, \dots, H-1\} \times \{0, \dots, W-1\} \rightarrow \mathbb{R}^d$, we define the GW illuminant SPD estimate $\hat{\mathbf{e}}_{GW}$ with

$$\hat{e}_{GW}(k) = \frac{1}{HW} \sum_{y=0}^{H-1} \sum_{x=0}^{W-1} C_k(y, x), \quad k = 1, \dots, d. \quad (5.12)$$

5.3.2 White-Patch (WP)

The classic White-Patch method works on the assumption that there exists an achromatic, bright patch somewhere in the image. That patch should reflect the illuminant SPD and be the brightest patch in the image. It simply searches for the highest value per channel in the image. Given an illuminated reflectance image $\mathbf{C}(\mathbf{x})$, we define the WP illuminant SPD estimate $\hat{\mathbf{e}}_{WP}$ with

$$\hat{e}_{WP}(k) = \max_{y,x} C_k(y, x), \quad k = 1, \dots, d. \quad (5.13)$$

A common variation for stability uses the top n -percent brightest values.

5.3.3 Color by Correlation

The Color by Correlation method models the probability distributions of chromaticities that appear under each illuminant as histograms. It is a classification method that assumes a predefined set of illuminants. In this thesis, we use CbC to investigate how different spectral representations affect the ability to discriminate between illuminants.

A practical limitation of CbC is that the histogram representations suffer from the curse of

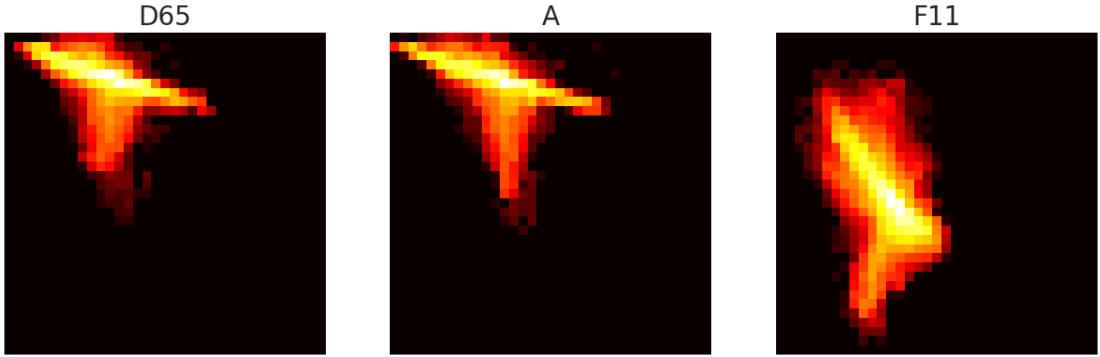


Figure 5.4: Example CbC histograms for $d' = 2$ and $B = 35$. Each panel shows the learned chromaticity log-likelihood over the two-dimensional reduced space for a single illuminant.

dimensionality. As the number of spectral dimensions increases, the number of bins in the histograms grows exponentially. With d dimensions and B bins per dimension, the size of the histogram is B^d bins. This makes it impractical to apply CbC directly to high-dimensional spectral representations. The histograms quickly take up a lot of memory and become difficult to process. We therefore apply dimensionality reductions to the spectral data and investigate how different reduction methods and number of components affect performance.

Here we describe our implementation. It follows the original closely but has slight deviations that we discuss at the end of this section. For the original formulation we refer to the paper [13].

Figure 5.4 visualizes example CbC histogram log-likelihoods for $d' = 2$ and $B = 35$, illustrating how chromaticity distributions differ across illuminants.

Implementation

We construct the histograms from the training data as described in Sections 5.1.3 and 5.1.4. To construct the histograms efficiently, we subsample the training images by a factor of 4 in each spatial dimension. This reduces the total number of pixels for training while preserving the overall chromaticity distribution.

Given a certain illuminant \mathbf{e}_i , we define the set of all images rendered under said illuminant \mathbf{C}_i . For this method all spatial information is discarded, and the images are flattened into a single set of pixels

$$\mathbf{P}_i = \left\{ \mathbf{p}_j \in \mathbb{R}^d \right\}_{j=1}^N \quad (5.14)$$

where d is the spectral dimensionality, and N is the number of pixels under each illuminant.

Chapter 5. Methodology

Each pixel is converted into a chromaticity representation $\mathbf{c} \in \mathbb{R}^d$ by applying L_1 normalization.

$$\mathbf{c} = \frac{\mathbf{p}}{\|\mathbf{p}\|_1} \quad \text{with} \quad \|\mathbf{p}\|_1 = \sum_{k=1}^d |p_k| \quad (5.15)$$

After normalization, a dimensionality reduction transform

$$\mathbf{z} = T(\mathbf{c}) \quad (5.16)$$

is applied, where $T : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}, d' < d$ is the dimensionality reduction method used in a given experiment.

Before constructing the histograms, common bounds need to be defined to ensure that all the histograms correspond to the same chromaticity space. Let

$$\mathbf{Z}_i = \left\{ \mathbf{z}_j \in \mathbb{R}^{d'} \right\}_{j=1}^N \quad (5.17)$$

be the set of all transformed chromaticities under illuminant \mathbf{e}_i and

$$\mathbf{Z} = \bigcup_i \mathbf{Z}_i \quad (5.18)$$

denote the set of all transformed chromaticities across the whole training set. The histogram bounds are defined per dimension as

$$z_k^{\min} = \min_{\mathbf{z} \in \mathbf{Z}} z_k, \quad z_k^{\max} = \max_{\mathbf{z} \in \mathbf{Z}} z_k \quad (5.19)$$

for $k = 1, \dots, d'$.

Using these bounds we use a uniform binning scheme, with B bins per dimension. Histogram H_i is constructed for each illuminant \mathbf{e}_i by counting the number of chromaticities that fall into each bin:

$$H_i(b) = |\{\mathbf{z} \in \mathbf{Z}_i \mid h(\mathbf{z}) = b\}| + 1 \quad (5.20)$$

where h is the histogram binning function and b indexes the histogram bins. We add a pseudocount of 1 to every histogram bin to prevent undefined log-likelihoods later. Finally the histogram is normalized by the total number of pixels to get a probability distribution.

$$P_i(b) = \frac{H_i(b)}{\sum_{b'} H_i(b')} \quad (5.21)$$

Before inference can be made, this probability function is converted into a log-likelihood

5.3 Illuminant Estimation Method Implementations

function,

$$L_i(b) = \log P_i(b) \quad (5.22)$$

To estimate the illuminant of a test image, a new histogram is created using the same data transformations and histogram bounds as in the training phase. We call the set of transformed chromaticities from the test image \mathbf{Z}_{test} , and the histogram is defined as

$$H_{test}(b) = |\{\mathbf{z} \in \mathbf{Z}_{test} \mid h(\mathbf{z}) = b\}| \quad (5.23)$$

Given the precomputed log-likelihoods $L_i(b)$ (with b indexing the flattened histogram bins), a score is computed for each illuminant as

$$s_i = \sum_b H_{test}(b)L_i(b) \quad (5.24)$$

The estimated illuminant is then selected as

$$\hat{\mathbf{e}} = \mathbf{e}_i, \quad \hat{i} = \arg \max_i s_i \quad (5.25)$$

Differences from original paper

There are three main differences between our implementation and the original.

The first one is the way we compute chromaticities. The original paper works on three-channel RGB images and computes the chromaticities by dividing the red and blue channels by the green one. Then they only use those two dimensions to build the histograms. We opted for using L_1 normalization instead to maintain a more even chromaticity distribution across the space.

The second difference is how the histogram bounds are calculated. In the original paper, they assume 8-bit integer images so all values range from 0–255. After their normalization they get the natural bounds of $[\frac{1}{255}, 1]$. In our case, we are working with floating point numbers and varying data transforms, so we use empirically determined bounds from our actual dataset.

The third difference has to do with the test image histogram. The original method thresholds that histogram to a max of one entry in each bin. Making it a binary presence indicator of each histogram. We found that the method always performs better by using the full chromaticity counts.

Chapter 5. Methodology

5.3.4 FC4

To evaluate a representative learning-based illuminant estimation method and study learned dimensionality reductions, we use Fully Convolutional Color Constancy with Confidence-weighted Pooling (FC4) [16]. FC4 is a widely used baseline in the color constancy literature and is well suited to our setting due to its relatively lightweight design.

FC4 Architecture

FC4 is a fully convolutional model that predicts an illuminant white point estimate from spatial feature maps. It consists of a convolutional backbone, pretrained on ImageNet [12] followed by a small convolutional head. The head produces an illuminant estimate and confidence value for each spatial location in the final feature map. These estimates are aggregated by a confidence-weighted pooling layer to obtain a global estimate for the image. In our implementation, we follow a variant suggested by the original paper that predicts three channels and interprets the vector direction as chromaticity and its magnitude as confidence.

In the original paper, the convolutional parts of either AlexNet [21] or SqueezeNet [17] are suggested as the backbone. We use SqueezeNet as the backbone due to its smaller parameter count, which is appropriate for our relatively small dataset size. The exact layer configuration is provided in Table A.3 in the Appendix.

To investigate the effects of pretrained RGB features on illuminant estimation performance on hyperspectral data. We try using both a pretrained backbone like the original FC4 paper does, and a randomly initialized backbone for comparison.

Input representations for FC4

For a baseline we can train this model with our data rendered in camera RGB space using the RGB projection described in Section 5.2.1. To study whether hyperspectral information can improve performance, we additionally train FC4 on the hyperspectral data, using simple projections that the model can learn as well.

We consider two spectral projection methods that map per-pixel spectra from 31 dimensions to 3 dimensions before passing them to the FC4 backbone: (i) a linear projection implemented as a single $31 \rightarrow 3$ layer, and (ii) a shallow non-linear projection implemented as $31 \rightarrow 64 \rightarrow 3$ with a ReLU activation between layers.

5.4 Experiments

Now that we have defined the implementation details of all of our methods and data processing, we will introduce the experiments we conducted in this thesis. There are two sets of experiments, the Color by Correlation experiments which investigate the effects of different

dimensionality reduction methods and extent, and the deep learning experiments that aim to compare the illuminant estimation performance of the same image in RGB and hyperspectral forms.

5.4.1 Dimensionality Reduction Study with CbC

To study how much dimensionality-reduced spectral information can improve illuminant estimation we use CbC (Section 5.3.3) as a reference model. It is appropriate for this task as it inherently utilizes the multidimensional representation of the input data.

We aim to quantify the effects of three parameters: (i) the projection method, (ii) the number of components d' , and (iii) the number of histogram bins per dimension B .

Specifically, we tried the five different dimensionality reduction methods (Section 5.2):

$$T \in \{\text{RGB}, \text{PCA}, \text{ILL_PCA}, \text{NNMF}, \text{LDA}\}$$

with five choices for the number of components:

$$d' \in \{1, 2, 3, 4, 5\}$$

and 4 different bin counts:

$$B \in \{5, 15, 25, 35\}.$$

We evaluated the CbC method with the full combinations of parameters, except that RGB has a fixed number of components $d' = 3$. This results in $4 \times 5 \times 4 + 1 \times 1 \times 4 = 84$ evaluations.

All experiments were done using the same 80/20 train–test split and the illuminant subset consisting of all the D-series, F-series and the A illuminant (Sections 5.1.3 and 5.1.4).

Performance is measured using the angular error metric defined in Equation 3.7 between the estimated illuminant spectrum $\hat{\mathbf{e}} \in \mathbb{R}^d$ and the ground truth $\mathbf{e} \in \mathbb{R}^d$. Similarly, we calculate angular error between the corresponding camera RGB space white points $\hat{\mathbf{e}}_{wp}, \mathbf{e}_{wp} \in \mathbb{R}^3$. We compute white points as $\mathbf{e}_{wp} = S\mathbf{e}$ and $\hat{\mathbf{e}}_{wp} = S\hat{\mathbf{e}}$ where $S \in \mathbb{R}^{3 \times d}$ is the matrix of camera spectral sensitivity functions.

As baselines for comparison, we use spectral extensions of the Gray World and White Patch algorithms, as defined in Section 5.3. These methods produce continuous illuminant estimates, which are mapped to the closest illuminant in the set of possible output illuminants using the snapping procedure described in Section 3.4.3.

For each evaluation, we aggregate the errors of all the test images and report their mean.

The resulting performance as a function of T , d' and B is reported in Section 6.1.

Chapter 5. Methodology

5.4.2 Deep Learning Improvement Study with FC4

To study whether low-dimensional spectral information improves illuminant estimation in a deep learning setting, we train FC4 (Section 5.3.4) under multiple configurations. We vary the input representation, and whether the SqueezeNet backbone is initialized with pretrained weights or trained from scratch. This allows us to assess whether RGB-pretrained features transfer to reduced spectral inputs or whether they hinder learning.

We train with the Adam [20] optimizer, using a learning rate of 1×10^{-4} , batch size 8, and weight decay 5×10^{-5} . Before projection to the network input representation, each hyperspectral image input is normalized by its maximum value so that its peak intensity equals one. Data augmentation consists of random horizontal flips ($p = 0.5$) and random resized crops: the crop area is sampled uniformly from $[0.3, 1.0]$ of the original image area with an aspect ratio sampled from $[0.75, 1.3]$, and the crop is then resized back to 512×512 .

Training uses the 4-fold cross-validation strategy, introduced in Section 5.1.4, for 50 epochs, using the stratified random illuminant assignment described in Section 5.1.3. After training for 50 epochs, we select the checkpoint with the best validation performance and report test results obtained by evaluating this selected model on the test set.

We evaluate a total of six configurations, given by the combinations of two backbone initializations (pretrained vs random initialization) and the three input projections: (1) a fixed camera RGB space projection, (2) a learned linear projection, and (3) a learned shallow non-linear projection, with (1) and (2) learned via the model training (Section 5.3.4).

Each configuration is repeated with three different seeds controlling the non-deterministic parts of the pipeline, including the cross-validation splits, network weight initialization, illuminant sampling, data loader shuffling, and data augmentation. We repeat runs three times with different seeds to make the results less sensitive to randomness in the pipeline.

5.4.3 Learned Projection Transfer Study

We investigate whether the learned linear projection from training FC4 is transferable as a projection for CbC. For this experiment, we select the best-performing FC4 model with a linear projection. The learned projection weights are extracted and used as a fixed projection basis for CbC. We compare the resulting CbC performance against the other dimensionality reduction methods at $d' = 3$.

6 Results

This chapter presents the experimental results of all studies in this thesis. We first report the dimensionality reduction experiments using the Color by Correlation (CbC) [13] method, analyzing how performance changes with projection type, number of components, and histogram discretization. We then present the deep learning experiments with FC4 [16], comparing input projections and backbone initialization, and finally evaluating whether a learned projection transfers to the statistical setting.

6.1 Dimensionality Reduction Study with CbC

In this section, we present the results of the dimensionality reduction study with CbC. The results of all 84 experiments are shown in Figure 6.1. The full numeric results are shown in Table A.1 in the Appendix.

The results show that all projection methods produce useful low-dimensional representations of the illuminated pixel spectra. One-dimensional representations ($d' = 1$) generally perform worse than, or comparable to, the simple baselines (GW and WP). With two components and $B \geq 15$, all methods outperform the baselines. At $d' = 3$, PCA and Ill. PCA begin to outperform the other methods (Figure 6.2a). For $d' \in \{4, 5\}$, performance differences narrow, especially at higher bin counts. Figure 6.2b compares all methods for $d' = 5$. At $B = 35$, PCA, Ill. PCA, and LDA perform similarly, while the best overall setting is obtained with $B = 25$, $T = \text{Ill. PCA}$, and $d' = 5$ with mean angular error of 2.21° .

At two components, all methods perform similarly to the RGB representation, while at three components they consistently outperform RGB. This shows that spectral data is more informative for illuminant estimation than a typical RGB camera sensor.

In all cases, except for NNMF where the 4- and 5-dimensional representations perform almost identically, higher-dimensional representations yield better performance. However, the gains diminish after 3 or 4 dimensions. Figure 6.3 shows the mean angular error across all methods, excluding RGB, as a function of d' .

Chapter 6. Results

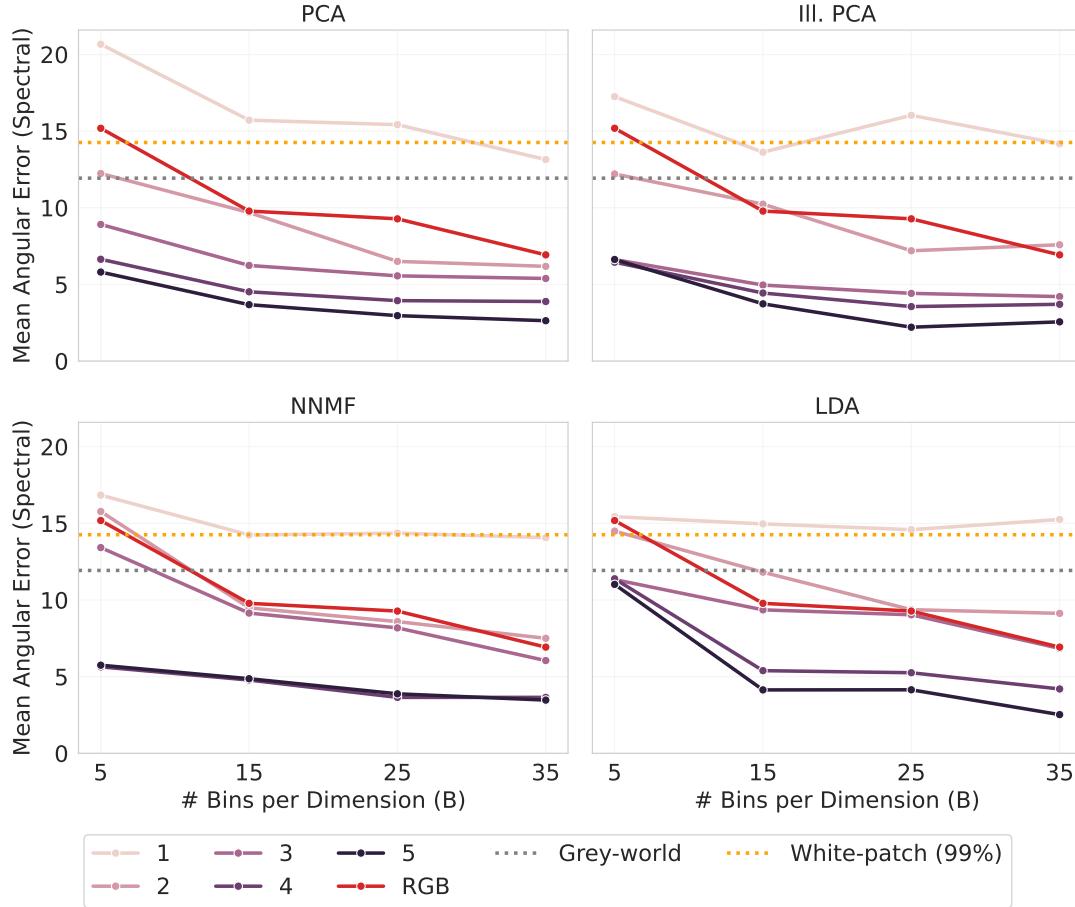


Figure 6.1: Mean spectral angular error of all CbC experiments. Each panel shows the results from a single projection method (PCA, III. PCA, NNMF, LDA). Curve colors denote the number of components $d' \in \{1, \dots, 5\}$. Gray World, White Patch and RGB projection (fixed $d' = 3$) are shown on each panel for reference. Lower is better.

It is also clear from our data that increasing the per-dimension bin count B generally provides better performance, but gains start to diminish after $B = 15$, especially for higher dimensions. This can be seen in Figure 6.4. $d' = 1$ is an interesting case, as it does not seem to benefit from increased discretization beyond $B = 15$, indicating that a small histogram is able to represent all the information inherent in the single component representation. Another notable thing about $d' = 1$ specifically for the Ill. PCA projection method (Figure 6.1) is that the angular error is worse for $B = 25$ than $B = 15$. This points to some unexplained instability in the method.

6.1.1 Analysis of Learned Bases

Here we analyze the learned bases for our four learned projections. Figure 6.5 plots each component for every method. From these plots we can see some interesting patterns.

6.2 Deep Learning Improvement Study with FC4

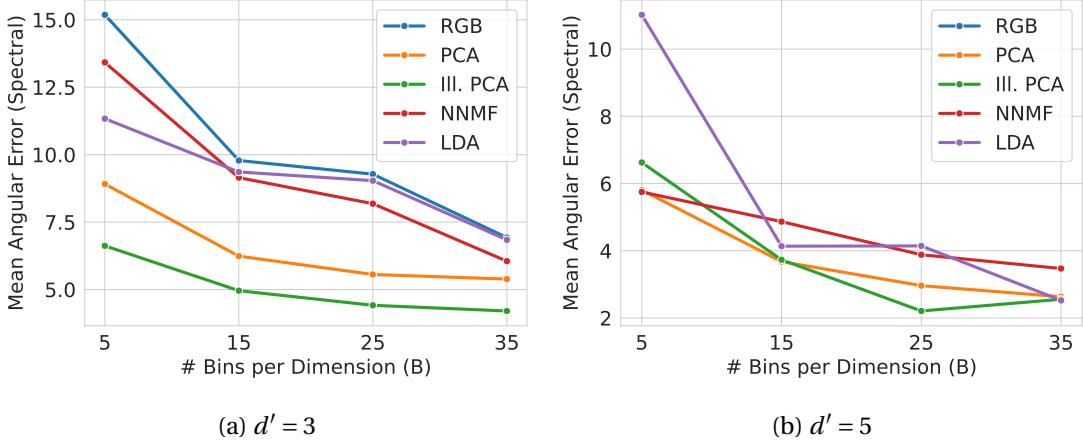


Figure 6.2: Comparison of projection methods for $d' = 3$ and $d' = 5$. The plots show CbC mean spectral angular error (degrees) as a function of the histogram discretization B for each projection method. Lower is better.

The first observation is that the two PCA-based methods learn very similar components. The first component is nearly identical, and the second has the same overall shape but with opposite sign. The remaining three components differ more, but still share the same general structure in terms of which wavelength bands have high and low weights. These later components are smoother under PCA, which may reflect the typically smooth structure of reflectance spectra compared to the spiky structure of some illuminant spectra. Overall, the similarity suggests that the learned PCA components are driven primarily by illuminant variation. Notably, Illuminant PCA never sees reflectance data, so this indicates that PCA learns illuminant-specific structure even when fitted to illuminated reflectance images. This is consistent with the Gray World intuition: when aggregating over many scenes, reflectance variation tends to average out, leaving illuminant variation dominant in the learned components.

Unlike other projection methods, NNMF does not yield a projection basis in the same sense. Instead, it learns non-negative components V used for reconstruction (Section 5.2.3), which can be viewed as additive building blocks. The NNMF components contain more entries that are zero or close to zero. Component 2 is especially interesting because it is mostly zero but has two spikes at locations where many of the F-series SPDs also have spikes. For example, the F11 illuminant shown in Figure 4.2.

The LDA components are used as a projection basis. However, they represent discriminative directions learned to separate the illuminant classes. This makes them difficult to interpret.

6.2 Deep Learning Improvement Study with FC4

In this section, we discuss the results of the deep learning improvement study with FC4. Table 6.1 aggregates all training runs across four folds and three seeds per configuration.

Chapter 6. Results

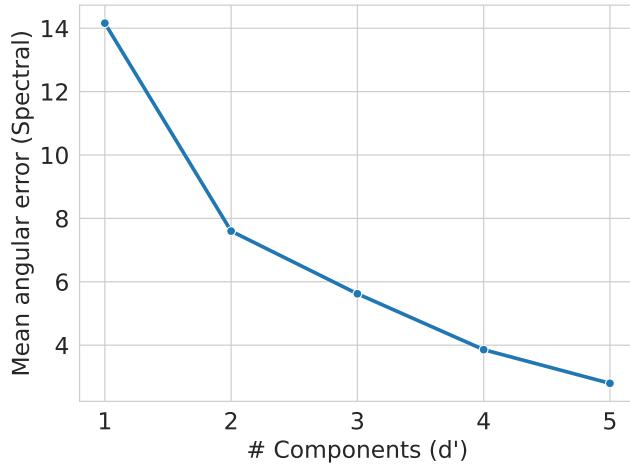


Figure 6.3: Mean spectral angular error as a function of the number of components. Results are averaged across all projection methods, excluding RGB. Lower is better.

Figure 6.6 shows the results for each fold, together with the mean and standard deviation for each seed.

Initializing the backbone with ImageNet-pretrained weights improves performance for the RGB and linear projection inputs, but performs worse for the non-linear projection. The non-linear pretrained setting includes one extreme outlier fold with angular error 9.14, which strongly affects the mean and standard deviation. If this outlier is omitted, the difference between pretrained and randomly initialized backbones in the non-linear case becomes negligible. Table 6.1 also reports aggregated statistics with the outlier omitted (in parentheses). Overall, the differences between backbone initializations are small and do not support strong conclusions about consistent benefits from pretraining in this setting.

Comparing input projections, neither learned spectral projection provides a consistent improvement over the fixed RGB projection, performing slightly worse in all cases. This suggests that the additional flexibility introduced by learned projections does not translate into increased performance in this setting.

Taken together, these results suggest that, within the FC4 architecture with the dataset considered here, low-dimensional spectral inputs do not offer a clear advantage over RGB for deep learning-based illuminant estimation, and that making the spectral reduction more complex does not provide any benefits either. This result is somewhat surprising given that the learned linear projection did provide improvements when applied to the CbC method, as is discussed in Section 6.3.

6.3 Learned Projection Transfer Study

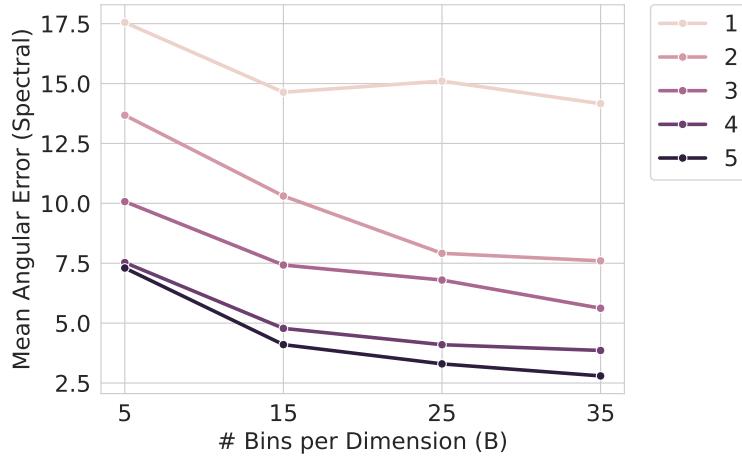


Figure 6.4: Mean spectral angular error as a function of the number of bins per dimension B . Results are averaged across all projection methods, excluding RGB. Curve colors denote the number of components $d' \in \{1, \dots, 5\}$. Lower is better.

Table 6.1: FC4 illuminant estimation performance across input representations and backbone initializations. Results are aggregated over three random seeds and four cross-validation folds (12 runs total) and reported as [median | mean \pm standard deviation] of white point angular error (in degrees). Lower values indicate better performance. Best result shown in bold. For the pretrained non-linear setting, values in parentheses are recomputed after excluding one extreme outlier run.

	RGB	Linear	Non-Linear
Pretrained	3.83 3.87 \pm 0.56	3.93 3.94 \pm 0.48	4.36 4.79 \pm 1.45 (4.28 4.39 \pm 0.48)
Random init	4.34 4.20 \pm 0.53	4.56 4.63 \pm 0.63	4.34 4.44 \pm 0.40

6.3 Learned Projection Transfer Study

We select the best-performing FC4 run among those using the jointly learned linear projection. This run uses a pretrained backbone and achieves a mean white point angular error of 3.20° on its test split. The corresponding learned projection basis is visualized in Figure 6.7. Qualitatively, the basis resembles the LDA projection shown in Figure 6.5, which is consistent with both methods learning discriminative directions, in contrast to more descriptive projection methods such as PCA.

To evaluate whether this learned projection transfers to a classical method, we use the learned weights as a fixed projection basis for CbC and compare it against the other dimensionality reduction methods at $d' = 3$, including the RGB baseline. The results are shown in Figure 6.8. Despite being learned within the FC4 training setup and under a different illuminant assignment, the transferred projection outperforms the RGB projection and, for higher histogram bin counts, achieves performance comparable to the strongest projections from the CbC

Chapter 6. Results

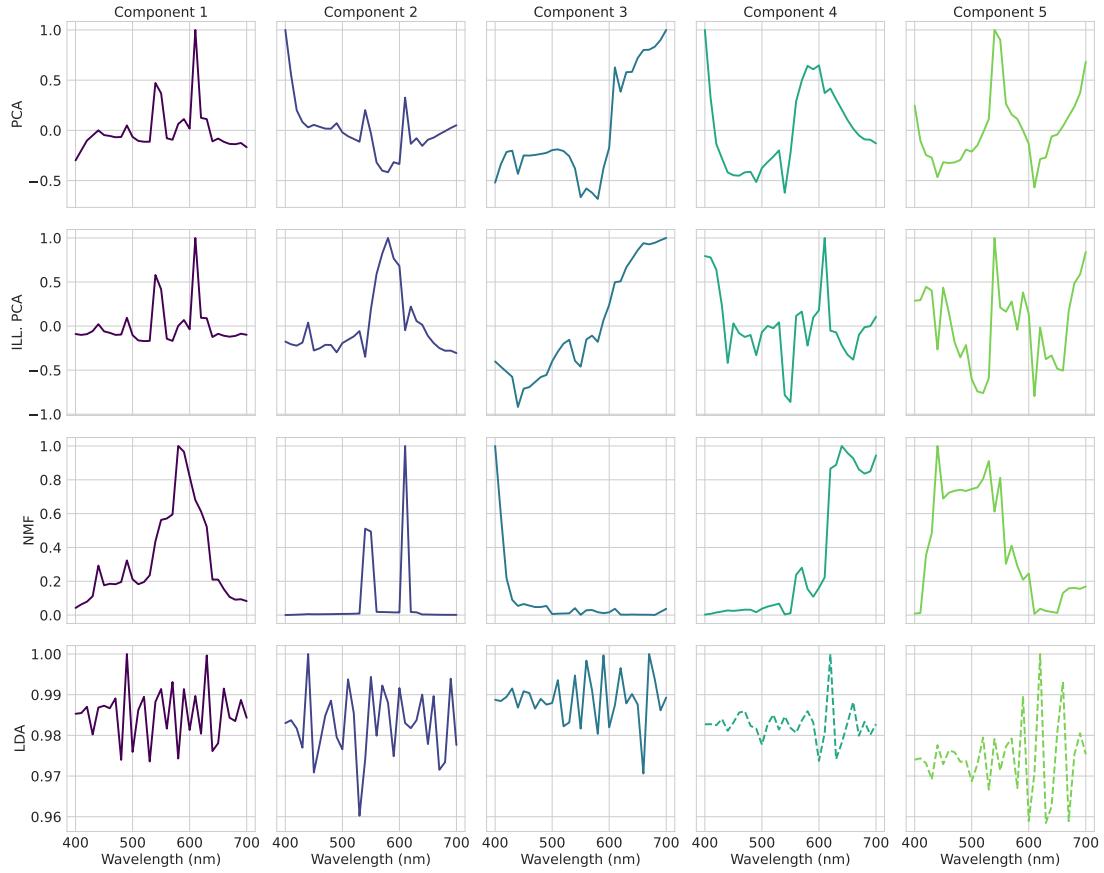


Figure 6.5: Learned projection bases. This figure shows the five learned components for each projection method. Each component is scaled so its maximum absolute value is 1. Dashed lines indicate negative values, which are plotted as positive for convenience.

study.

This confirms our hypothesis that learned projections can be successfully transferred to simpler statistical methods. The learned projection does not, however, outperform simpler projection methods such as PCA and Ill. PCA.

Overall, these results provide evidence that projections learned in a modern deep learning pipeline can transfer to simpler statistical methods. However, the transferred projection does not outperform the best-performing classical projections (e.g., PCA and Ill. PCA) in this setting.

6.3 Learned Projection Transfer Study

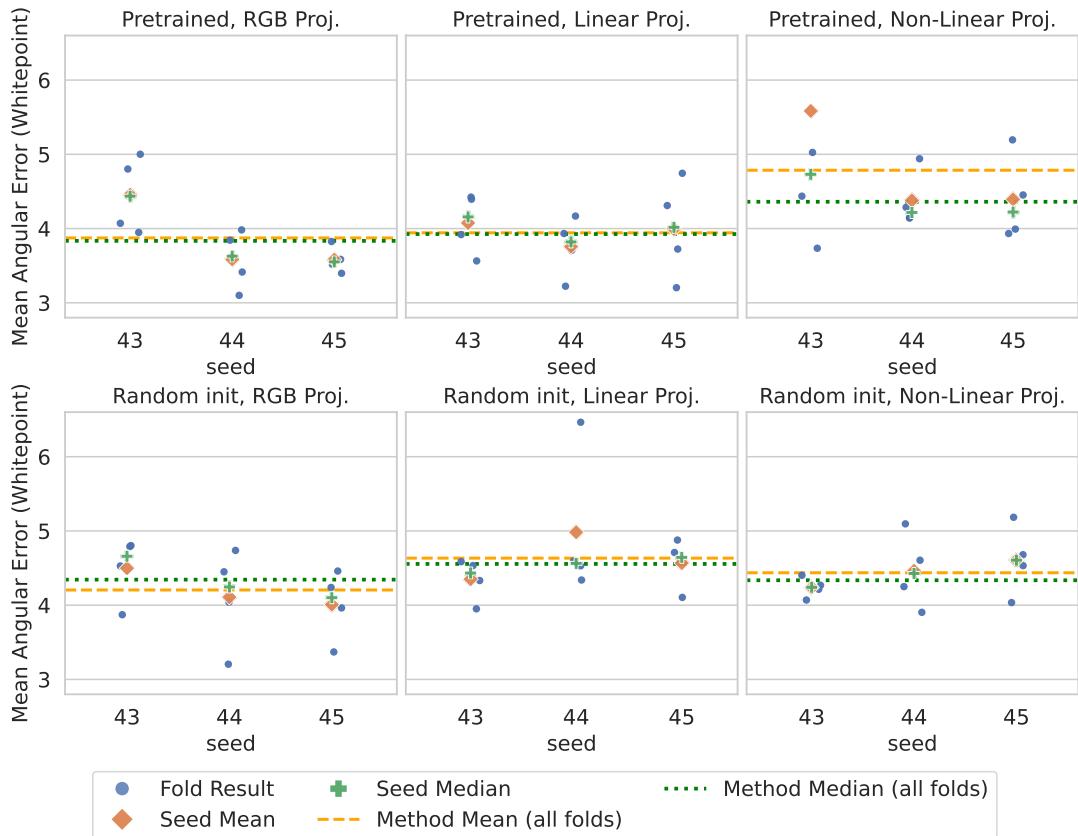


Figure 6.6: FC4 illuminant estimation performance across input projections and backbone initializations. Each point corresponds to a single cross-validation fold for a given random seed. For each seed and configuration, the mean and median across folds are indicated. Horizontal lines show the overall mean and median for each configuration across seeds. The top row corresponds to pretrained backbones and the bottom row to randomly initialized backbones. An outlier for seed 43 with angular error 9.14 is omitted from the top right plot. Lower is better.

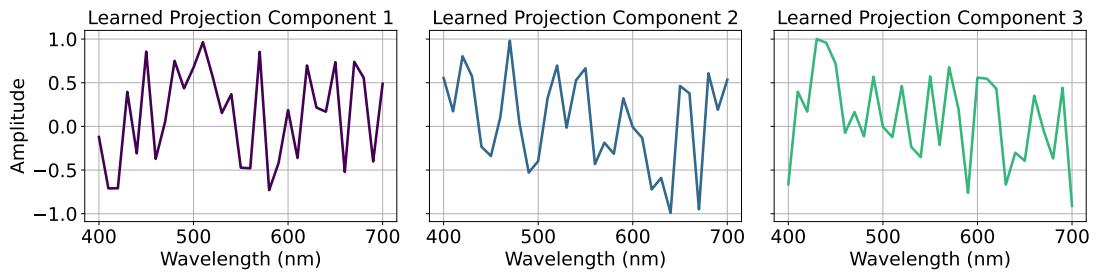


Figure 6.7: Basis functions of the learned linear projection. Each component is scaled so its maximum absolute value is 1.

Chapter 6. Results

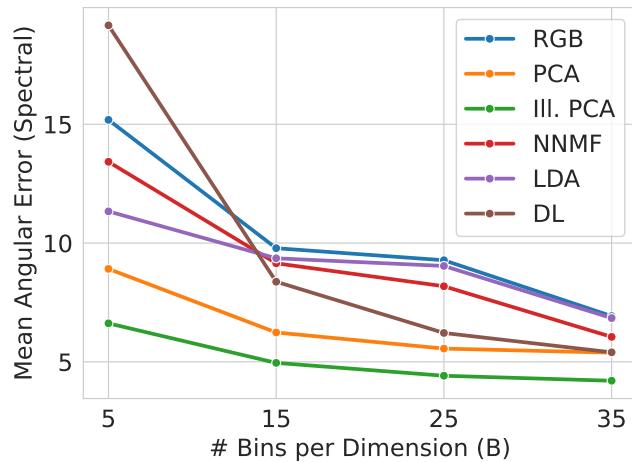


Figure 6.8: CbC performance at $d' = 3$ for different projection methods, including the transferred deep-learning projection (DL). The plot shows mean spectral angular error (degrees) as a function of the histogram discretization (number of bins B). Lower is better.

7 Conclusions

In this thesis, we presented a controlled study of illuminant estimation using hyperspectral reflectance data, with synthetic RGB observations generated from the same scenes for direct comparison. Experiments were carried out using both statistical and deep learning methods. In the statistical setting, reduced hyperspectral representations consistently improved performance compared to RGB. In the deep learning setting, replacing RGB inputs with learned three-channel projections of the hyperspectral data did not yield consistent improvements.

7.1 Answers to Research Questions

RQ1: How much does illuminant estimation performance improve when moving from RGB representations to hyperspectral representations?

In our Color by Correlation experiments, hyperspectral data consistently outperforms RGB, even when reduced to low-dimensional representations ($d' = 2, 3$). This suggests that reduced hyperspectral inputs retain more useful information for illuminant discrimination than RGB with the same number of channels, and that the gains are not solely explained by increased dimensionality.

We did not observe the same trend in the deep learning experiments. When training with learned three-channel projections of hyperspectral data, we did not see consistent improvements over training on RGB inputs. Whether this is due to limited dataset size or because the spatial features exploited by convolutional networks are more important than the precise scene representation remains unclear.

RQ2: How does illuminant estimation performance change as the dimensionality of hyperspectral data is varied?

In the CbC experiments, performance improves substantially as the number of retained dimensions increases ($d' = 1, 2, \dots, 5$). A single component is generally insufficient, while $d' = 2$ already outperforms RGB. Improvements diminish around $d' = 4-5$. This is broadly

Chapter 7. Conclusions

consistent with prior work showing that real-world reflectance spectra can be represented well using a small number of components [26], although that objective differs from representing *illuminated* spectra for illuminant estimation.

RQ3: How do different dimensionality reduction methods compare in preserving the information needed for illuminant estimation?

In the CbC experiments, PCA and especially Illuminant PCA perform best overall. These methods are also among the simplest and fastest to compute. LDA is competitive at higher dimensions but produces basis functions that are harder to interpret and needs training on labeled data. NNMF generally performs worse than PCA-based methods in our setting, suggesting that the non-negativity constraint trades off reconstruction capability for interpretability.

RQ4: Under pretrained and randomly initialized RGB backbones, do learned three-channel projections of hyperspectral data improve deep illuminant estimation?

We found no consistent improvement from training the FC4 model on learned three-channel hyperspectral projections compared to using the RGB projection. This holds for both linear and non-linear learned projections. Initializing the backbone with pretrained weights provided only small and inconsistent gains over random initialization, and did not change the overall conclusion that the learned three-channel hyperspectral inputs do not reliably outperform RGB in this setting. A possible explanation is that the convolutional network's ability to extract spatial features is a dominating factor compared to the exact data representation.

RQ5: Can a learned linear projection from a deep learning setting be transferred to improve statistical illuminant estimation?

The learned projection transfer study shows that a learned hyperspectral linear projection improves CbC compared to the RGB projection. This transferred projection reaches performance comparable to the PCA projection. However, it does not consistently exceed Illuminant PCA performance. This suggests that learned projections can be useful and transferable, but do not necessarily provide a unique advantage over simpler data-driven reductions in our current experimental setup.

7.2 Limitations

Several limitations should be kept in mind when interpreting the results of this thesis. First, the assumptions of a single global illuminant and Lambertian surface reflectance are strong and do not generally hold in real scenes, where spatially varying illumination, interreflections, and non-Lambertian materials may be present. Our synthetic relighting procedure, as well as the corresponding illuminant removal used to create the reflectance image dataset, relies on these assumptions.

Second, in the classical Color by Correlation experiments, we only evaluate reduced hyperspectral representations up to $d' = 5$. Higher-dimensional inputs are difficult to evaluate with histogram-based methods due to the curse of dimensionality, so this work does not show performance trends for larger dimensions.

Third, the deep learning experiments exhibit noticeable variability across runs. This is likely influenced by the limited dataset size and by the fact that hyperparameters were not tuned individually for each configuration. While we reduce sensitivity to randomness by evaluating multiple seeds and cross-validation folds, the reported trends should be interpreted appropriately.

Finally, the RGB observations used throughout the thesis are synthetic. Although we employ measured camera spectral sensitivities, these are sampled at 31 wavelengths and RGB values are computed via a discrete inner product. This will differ slightly from the continuous spectral integration performed by real camera sensors and may introduce small discrepancies relative to real RGB images.

7.3 Implications

The experiments suggest several practical implications. First, hyperspectral information can be beneficial for illuminant estimation even when compressed into low-dimensional representations, and performance gains in the classical setting are not solely explained by increased dimensionality. Second, simple data-driven projections such as PCA provide strong baselines. Third, projections learned in the deep learning setting can be reused in CbC, indicating that learned projections can transfer to simpler statistical methods.

7.4 Future Work

Several directions could extend this work. Collecting larger hyperspectral reflectance datasets would enable more reliable evaluation of deep learning methods and may reduce training variability. For the deep learning setting, it would be valuable to study more advanced fine-tuning strategies, such as freezing certain backbone layers. Finally, architectures that accept higher-dimensional inputs (beyond three channels) would allow studying the effect of increased representation dimensions of hyperspectral data on a deep learning method.

A Numeric Results & Model Architecture

Appendix A. Numeric Results & Model Architecture

Table A.1: Mean spectral angular error (degrees) for all projection methods as a function of the number of components d' and histogram bin count B . Lower is better. Best result per (d', B) is shown in bold.

# Components (d')	Projection Method	$B = 5$	$B = 15$	$B = 25$	$B = 35$
1	PCA	20.67	15.72	15.42	13.15
	Ill. PCA	17.25	13.62	16.03	14.17
	NNMF	16.84	14.24	14.36	14.07
	LDA	15.44	14.96	14.59	15.25
2	PCA	12.24	9.69	6.50	6.18
	Ill. PCA	12.20	10.24	7.20	7.59
	NNMF	15.78	9.49	8.59	7.50
	LDA	14.50	11.81	9.37	9.13
3	PCA	8.91	6.24	5.56	5.39
	Ill. PCA	6.62	4.96	4.42	4.20
	NNMF	13.42	9.15	8.18	6.05
	LDA	11.33	9.36	9.04	6.84
	RGB	15.19	9.78	9.28	6.93
	DL	19.16	8.37	6.22	5.40
4	PCA	6.64	4.52	3.94	3.88
	Ill. PCA	6.45	4.44	3.55	3.70
	NNMF	5.64	4.78	3.65	3.65
	LDA	11.39	5.39	5.26	4.20
5	PCA	5.81	3.68	2.96	2.63
	Ill. PCA	6.63	3.73	2.21	2.56
	NNMF	5.75	4.87	3.88	3.47
	LDA	11.02	4.14	4.14	2.52

Table A.2: Per-fold and per-seed FC4 illuminant estimation results (white point angular error in degrees) across backbone initialization and input projection. Mean and standard deviation are computed across the four cross-validation folds for each seed. Lower is better.

Init	Input	seed	Fold 0	Fold 1	Fold 2	Fold 3	Mean	Std
Pretrained	RGB	43	3.95	5.00	4.80	4.07	4.46	0.52
		44	3.10	3.98	3.84	3.41	3.59	0.40
		45	3.58	3.40	3.52	3.83	3.58	0.18
	Linear	43	3.92	4.42	3.56	4.39	4.08	0.41
		44	3.93	3.71	4.17	3.22	3.76	0.40
		45	4.31	3.20	3.72	4.74	4.00	0.67
	Non Linear	43	4.44	5.03	3.73	9.14	5.58	2.43
		44	4.29	4.14	4.94	4.15	4.38	0.38
		45	3.99	5.19	3.93	4.45	4.39	0.58
Random init	RGB	43	4.79	4.53	3.87	4.80	4.50	0.44
		44	4.45	4.74	4.04	3.21	4.11	0.67
		45	4.46	3.96	4.24	3.37	4.01	0.47
	Linear	43	4.59	4.33	3.95	4.53	4.35	0.29
		44	4.34	4.60	6.46	4.53	4.98	0.99
		45	4.10	4.71	4.58	4.88	4.57	0.33
	Non Linear	43	4.07	4.21	4.27	4.40	4.24	0.14
		44	4.61	3.90	5.10	4.25	4.46	0.51
		45	4.53	4.68	4.04	5.18	4.61	0.47

Appendix A. Numeric Results & Model Architecture

Table A.3: Architecture of the FC4 model used in this thesis with a truncated SqueezeNet v1.1 backbone. Each Fire block is parameterized by its squeeze channels and its two expand branches (1×1 and 3×3). The network outputs an RGB illumination estimate via spatial summation and ℓ_2 normalization. Convolution parameters are denoted as k (kernel size), s (stride), p (padding), and P dropout probability.

Idx	Layer	Parameters	Channels (in → out)
SqueezeNet Backbone			
0	Conv2D + ReLU	$k=3, s=2, p=0$	$3 \rightarrow 64$
1	MaxPool2D	$k=3, s=2$	—
2	Fire2	squeeze=16, expand=(64,64)	$64 \rightarrow 128$
3	Fire3	squeeze=16, expand=(64,64)	$128 \rightarrow 128$
4	MaxPool2D	$k=3, s=2$	—
5	Fire4	squeeze=32, expand=(128,128)	$128 \rightarrow 256$
6	Fire5	squeeze=32, expand=(128,128)	$256 \rightarrow 256$
7	MaxPool2D	$k=3, s=2$	—
8	Fire6	squeeze=48, expand=(192,192)	$256 \rightarrow 384$
9	Fire7	squeeze=48, expand=(192,192)	$384 \rightarrow 384$
10	Fire8	squeeze=64, expand=(256,256)	$384 \rightarrow 512$
FC4 head			
11	MaxPool2D	$k=2, s=2$	—
12	Conv2D + ReLU + Dropout	$k=6, s=1, p=0, P=0.5$	$512 \rightarrow 64$
13	Conv2D (semi-dense map)	$k=1, s=1, p=0$	$64 \rightarrow 3$
14	Spatial sum pooling	$\sum_{h,w}$	$3 \rightarrow 3$
15	ℓ_2 normalization	$\hat{\mathbf{p}} = \mathbf{p} / \ \mathbf{p}\ _2$	$3 \rightarrow 3$

Bibliography

- [1] Afifi, M., Barron, J. T., LeGendre, C., Tsai, Y.-T., and Bleibel, F. (2021). Cross-camera convolutional color constancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1981–1990.
- [2] Ahmad Khan, H., Thomas, J. B., Hardeberg, J. Y., and Laligant, O. (2017). Illuminant estimation in multispectral imaging. *Journal of the Optical Society of America A*, 34:1085–1098.
- [3] Barron, J. T. (2015). Convolutional color constancy.
- [4] Barron, J. T. and Tsai, Y.-T. (2020). Fast fourier color constancy.
- [5] Berns, R. S. (2019). *Billmeyer and Saltzman's Principles of Color Technology*. John Wiley & Sons, 4 edition.
- [6] Bianco, S. and Cusano, C. (2015a). Color constancy using cnns.
- [7] Bianco, S. and Cusano, C. (2015b). Single and multiple illuminant estimation using convolutional neural networks. *IEEE Transactions on Image Processing*, PP.
- [8] Buchsbaum, G. (1980). A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310(1):1–26.
- [9] Cheng, M.-F., Mukundan, A., Karmakar, R., Valappil, M., Jouhar, J., and Wang, H.-C. (2025). Modern trends and recent applications of hyperspectral imaging: A review. *Technologies*, 13:170.
- [10] CIE (2018). Colorimetry, 4th edition. Technical Report CIE 015:2018, Commission Internationale de l’Éclairage, Vienna, Austria.
- [11] CIE (2019). Colour-matching functions of cie 1931 standard colorimetric observer.
- [12] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Bibliography

- [13] Finlayson, G., Hordley, S., and Hubel, P. (2001). Color by correlation: a simple, unifying framework for color constancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1209–1221.
- [14] Finlayson, G. D., Drew, M. S., and Funt, B. V. (1994). Color constancy: generalized diagonal transforms suffice. *J. Opt. Soc. Am. A*, 11(11):3011–3019.
- [15] Forsyth, D. A. (1990). A novel algorithm for color constancy. *International Journal of Computer Vision*, 5:5–35.
- [16] Hu, Y., Wang, B., and Lin, S. (2017). Fc4: Fully convolutional color constancy with confidence-weighted pooling. pages 330–339.
- [17] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. (2016). SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size.
- [18] Jiang, J., Liu, D., Gu, J., and Süsstrunk, S. (2019). Camera spectral sensitivity database - jiang et al. (2013).
- [19] Jiang, J., Liu, D., Gu, J., and Süsstrunk, S. (2013). What is the space of spectral sensitivity functions for digital color cameras? In *2013 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 168–179.
- [20] Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.
- [21] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.
- [22] 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.
- [23] Land, E. H. (1977). The retinex theory of color vision. *Scientific American*, 237(6):108–128.
- [24] Li, Y., Fu, Q., and Heidrich, W. (2021a). Dataset for multispectral illumination estimation using deep unrolling network.
- [25] Li, Y., Fu, Q., and Heidrich, W. (2021b). Multispectral illumination estimation using deep unrolling network. pages 2652–2661.
- [26] Malone, L. T. (1986). Evaluation of linear models of surface spectral reflectance with small numbers of parameters. *J. Opt. Soc. Am. A*, 3(10):1673–1683.
- [27] Oh, S. and Kim, S. (2016). Approaching the computational color constancy as a classification problem through deep learning. *Pattern Recognition*, 61.

Bibliography

- [28] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- [29] Qian, Y., Chen, K., Kamarainen, J.-K., Nikkanen, J., and Matas, J. (2016). Deep structured-output regression learning for computational color constancy.
- [30] 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.
- [31] Shi, W., Loy, C. C., and Tang, X. (2016). Deep specialized network for illuminant estimation. volume 9908, pages 371–387.
- [32] Slater, D. and Healey, G. (1998). Analyzing the spectral dimensionality of outdoor visible and near-infrared illumination functions. *Journal of the Optical Society of America A*, 15:2913–2920.
- [33] Su, T., Zhou, Y., Yu, Y., Cao, X., and Du, S. (2018). Illumination separation of non-lambertian scenes from a single hyperspectral image. *Optics Express*, 26:26167–26178.
- [34] von Kries, J. (1970). Chromatic adaptation. In MacAdam, D. L., editor, *Sources of Color Science*. MIT Press, Cambridge, MA. English translation of von Kries (1902) from *Festschrift der Albrecht-Ludwig-Universität* (Freiburg), 1902.
- [35] Xu, B., Liu, J., Hou, X., Liu, B., and Qiu, G. (2020). End-to-end illuminant estimation based on deep metric learning. pages 3613–3622.
- [36] Yamashina, H., Fukushima, K., and Kano, H. (1996). White balance in inspection systems with a neural network. *Computer Integrated Manufacturing Systems*, 9(1):3–8.
- [37] Yu, H., Chen, K., Wang, K., Qian, Y., Zhang, Z., and Jia, K. (2019). Cascading convolutional color constancy.
- [38] Zheng, Y., Sato, I., and Sato, Y. (2015). Illumination and reflectance spectra separation of a hyperspectral image meets low-rank matrix factorization. pages 1779–1787.