Image Factorization for Inverse Rendering

Thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Engineering

by

Saurabh Saini
201250862
saurabh.saini@research.iiit.ac.in

International Institute of Information Technology, Hyderabad
(Deemed to be university)
Hyderabad - 500 032, INDIA
August 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, which is titled " Image Factorization for Inverse Rendering " by Saurabh Saini, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. P. J. Narayanan

Dedication

To (late) Harmohan Sharma and (late) Ilam Singh Saini, for their faith and motivation.

To my parents, for their understanding and support.

To my dearest Srishti and Shreshth, for their patience and love.

Acknowledgements

This thesis was possible only due to the direct or indirect support of so many people that acknowledging them all seems near impossible. I would certainly like to thank Prof. P J Narayanan , my guide, who always kept faith in me even when I did not. I would also like to thank all my other professors *i.e*. Prof. Vineet Gandhi, CV Jawahar, Anoop Namboodiri, Jayanthi Sivaswamy, Kamal Karlapalem, Madhav Krishna and Rajat Tandon who initiated my interests in various subjects and from whom I learned the fundamentals of research. I would like to acknowledge the consistent assistance (and reminders !) from the IIIT administrative staff specifically, Garu Y. Kishore (bureaucracy deobfuscator), G. Murlidhar (smiling sanctuary monk), R. S. Satyanarayana (helpful old jedi), Silar Shaik (friendly chatter), K. Satish (courteous custodian) and Madam B. Pushpalatha, Prathima Mandapati (field referees).

I also want to thank and cherish the personal relations which I gathered during my decade long association with IIIT which I proved more valuable than the final goal. Several of my labmates became close friends and supporters during this journey. This included both seniors like Nataraj and Anand, as well as graduated juniors and batchmates like Praveen, Yashaswi, Vidyadhar, Koustav, Pramod, Srinath, Vijay, Arunava, Falak, Lipika, Harit, Tejaswinee, Aditya (all of them !) and many more... Most important were my family away from family, which comprised of Aniket, Pritish, Parikshit, Rajvi and Revanth, who were there with me (even when I tried escaping) and accepted me, no matter my best or worst. Thank you all !

I can not measure the blessings bestowed, sacrifices made, patience shown, support provided and love given to me by my family through out this journey which involved frequent mental absenteeism. My father understood me and kept me focused on the goal, especially during the final duration where he is worthy of being a co-guide for his support. My mother kept the supporting even without understand the reasons which even more appreciable. I can not repay (Late) Harmohan Sharma (Sharma uncle) who shared his wisdom and his time during his end years with me which I will cherish throughout my life. Srishti, my beloved wife, thank you for your immeasurable tolerance, immense sacrifices, super-womanly fortitude and especially for your caring nature for me and our Shreshth. This would not have been possible without any of it.

Finally, thank you God, for setting me on this journey, helping me traverse through it and in the end, harboring me to the destination.

I am grateful for it all.

Abstract

Inverse Rendering is a core Computer Vision problem as it involves complete decomposition of an image into its constituting atomic components. These components can be stand-alone analyzed or suitably modified and recombined to solve the required image analysis task or achieve the required generative content. Rather than aiming for full decomposition, many applications only require decomposition into only a few factors which themselves are simple combinations of the underlying atomic components. This makes image factorization a critical first step in several computer vision and image processing applications. This factorization could either be optically motivated like reflectance-shading decomposition, white-balancing, illumination spectra-separation *etc*. or semantically motivated like style-content disentanglement, foreground-background matting *etc*.

In this thesis, we focus on the former and present several image factorization solutions with an aim to use it for a downstream image based rendering application. Initially, we assume Lambertian reflection only under the classical image formation model inspired from the Retinex theory. Our first solution in this category requires multiple images of the scene as input, which we then relax for our second solution which works on the single image input. Afterwards, we propose a novel image formation model based on the specularity of the image content and provide two solutions using the low light enhancement problem as the vehicle for empirical validation. Towards the end, a novel prior induction technique is also presented based on learnable concepts and its utility is shown by improving results of pre-existing state-of-the-art image decomposition networks. We conclude with a summary, limitations, future research directions and possible additional applications. The thesis is organized into four units respectively discussing the problem definition and significance; Lambertian reflection based Intrinsic Image Decomposition problem, specularity respecting novel illumination factorization methods and finally concept based model analysis and conclusion. We hope that with the problems and solutions discussed in this thesis we will be able to define and highlight the importance of image factorization step in multiple vision tasks and pique reader's interest in this research problem for image generation and beyond.

Contents

List of Figures

List of Tables

PART I

Overview

Chapter 1

Image Factorization

1.1 Definition

In this thesis we focus on the frequently encountered task of image factorization pertaining to several research problems arising in the subjects of Computer Vision, Computer Graphics and Machine Learning. Simply stated, *image factorization* can be understood as the disentanglement of an image matrix into two or more components which can be individually processed to benefit a given application. The components or the *factors* hence obtained, generally possess some associated semantic meaning and can be combined via some well-defined mathematical operation to reconstruct back the original image.

$$I \quad \rightleftharpoons \quad F_1 \odot F_2 \ldots \odot F_K, \tag{1.1}$$

where I is the image being processed, $F_i \in \{1, 2, \ldots K\}$ are two or more disentangled factors and $\odot$ is the associated binary operation (generally a simple addition $+$ or multiplication $*$). One simple example is from the image denoising application where assuming simple additive noise we have:

$$I \quad = \quad I_d + I_n, \tag{1.2}$$

with I and I_d as the noisy and denoised versions of the image and I_n as the extracted noise.

Note that this simplified definition can be extended to encompass many more scenarios by allowing complex varying operations between the factors and sacrificing the invertibility or interpretability of the above formulation (Eq. (1.1)) *e.g.* spatially and spectrally varying (homoscedastic) noise, style-content separation, foreground-background matting, image white-balancing *etc*.

Hence the task of image factorization lies at the heart of several research problems.

1.2 Image Factorization: A Computer Vision perspective

Computer Vision (CV) is that subject under the wider domain of Artificial Intelligence (AI), which specifically deals with visual data pertaining to different modalities *e.g*. images, focal/exposure bracketed stacks, stereoscopic images, multi-view images, panoramas, gifs, short clips, videos, depth maps, pose graphs, point clouds, surface meshes, volumetric 3D representations, multi-spectrum figures *etc*. Considering such a large number of such data representations, several CV research areas solve similar problems for these modalities *e.g*. image/video/3D segmentation/labeling, video/3D tracking, multi-view/RGBD shape estimation *etc*. This abundance of data variety brings a lot of challenges and opportunities in CV research as familiar questions need to be reevaluated for different modalities but still require separate adaptation for each task. Furthermore, several of these research areas are codependent and hence progress in one, opens newer solution possibilities in the associated tasks. Considering the commonality between these research topics, Malik et al. [193] summarize the essence of several research works in a meta study and propose following three broad mutually inclusive sets (Fig. 1.1):

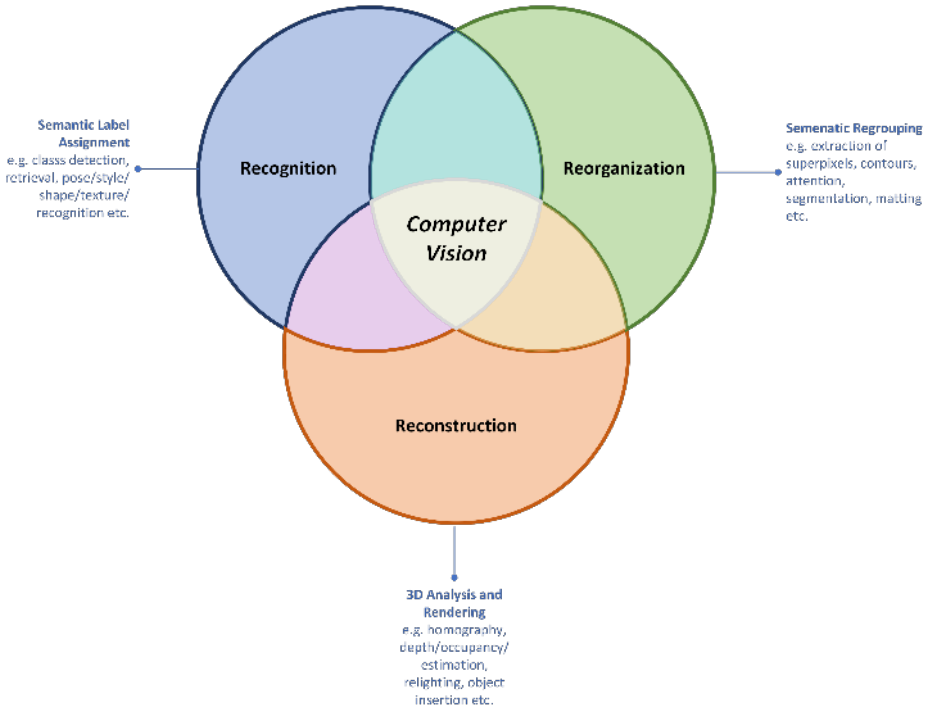


Figure 1.1: Computer Vision Research Categorization: The 3Rs of Computer Vision research [193] namely Recognition, Reorganization and Reconstruction with definition and examples from each category.

- **Recognition:** These problems primarily are concerned with *semantic labels estimation* which are defined based on hierarchies which are partonomic (part-whole *e.g*. fingers-hand, limbs-

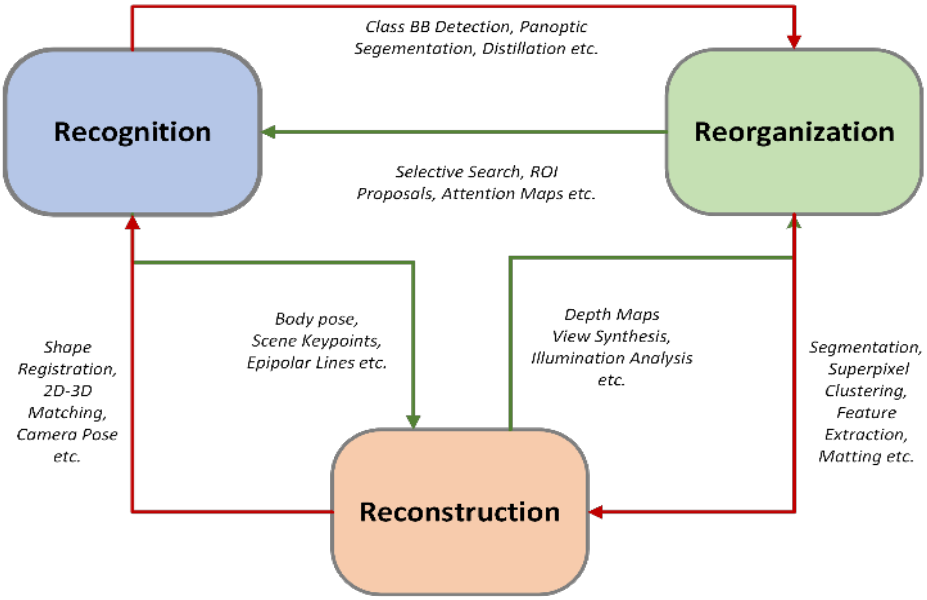


Figure 1.2: Computer Vision Research Categories Inter-relation: Illustration of mutual interactions between various research topics categorized under the 3Rs of Computer Vision [193] in Section 1.2.

body *etc*.), taxonomic (category-subcategory *e.g*. chair-table, pedestrian-road *etc*.) or ontological (concept-relations *e.g*. walkable spaces, graspable objects *etc*.).

- **Reorganization:** Problems in this category deal with *semantic data reorganization and analysis* based on perceptual or task specific input processing *e.g*. grouping, segmentation, saliency detection, layout estimation, contours/edge detection, matting, attention map generation *etc*.
- **Reconstruction:** These problems involve *visual result generation* by estimating the original constituting information of the 3D scene and using it to render new enhanced or altered visual results *e.g*. reflectance-shading estimation, mesh reconstruction, inverse light transport, illumination map estimation, image enhancement, style transfer *etc*.

As can be observed from Fig. 1.1, this division is not mutually exclusive as most of the real world applications use sub-systems from more than one of these categories but based on the focus of the underlying core research, one can categorize the problem in one or the other set. As shown in Fig. 1.2, solutions from one category are known to assist the systems in the other category *e.g*. reorganization helps recognition [95] and recognition helps reconstruction [134]. A more comprehensive report on this categorization and their mutual interaction is given in [193]. Based on the focus of the thesis. we restrict our discussion here pertaining only to the *reconstruction* problems restricted to 2D input data modality.

1.2.1 Image Reconstruction

To broadly classify, the topic of image reconstruction in computer vision refers to those problems and solutions which finally yield a visual output for the underlying scene *e.g*. low-light enhancement, denoising, deblurring, reflection removal, view synthesis, super-resolution, stylization *etc*. This definition distinguishes reconstruction from recognition and reorganization problems which focus on image class categorization (pixel, patches or whole image level) which are directly evaluated using numerical scores disregarding perceptual aspect of the generated results. Hence recognition and reorganization problems lay more emphasis on the semantic analysis aspect of the CV problems over visual quality, whereas reconstruction problems require emphasis on both (with often even a higher emphasis on the perceptual quality). Thus, owing to this nature of the reconstruction problems, perceptual quality of results is as crucial as numerical accuracy which introduces human judgement as a necessary factor during system design and evaluation. Hence both good qualitative and quantitative evaluation is needed and performance is estimated with relevant perceptual comparisons for proposing a state-of-the-art solution. This illustrates the additional complexity and challenge posed by reconstruction problems compared to the other two categories. Reconstruction problem and their applications encompass several CV analysis tasks like depth estimation, point matching, viewpoint localization, lighting estimation, white-color balancing, relighting, reflectance function computation, texture extraction, object insertion, structure from motion *etc*.

Several proposed solutions for the above mentioned tasks often require factorization of the input image into relevant components before enhancement *e.g*. shadow removal requires illumination mask for shadow mask estimation, white-color balancing requires both illumination map and color estimation, object insertion requires scene geometry estimation *etc*. The computer vision perspective for image factorization mainly deals with the utility of the obtained factors for the end applications. In this thesis, instead of image features factorization, we restrict our discussions to the reconstruction problems, which directly require input image decomposition. Note that this implies higher significance given to the perceptual quality of the processed results, with lesser focus on the interpretability and the optical accuracy of the transient factors.

1.3 Image Factorization: An Optimization Perspective

Instead of focusing on the CV downstream application as output, we can also discuss image factorization from the optimization perspective during data processing. Factorization is useful and many a times even essential, for proper optimization formulation and solution for a given task. One intuitive explanation is that disentangled variables after factorization allow separate manipulation resulting in

simpler optimization equations and more faithful approximations *e.g*. initial style-content disentanglement allows separate manipulation and simpler operations on each rather than direct image editing.

Mathematically, matrix and tensor decompositions are well studied problems but the factors are relatively simplistic without any higher-order meaning associated with them *e.g*. LU, QR, low-rank, sparse *etc*. decompositions. Research community has studied the effects and utility of some relatively complex mathematical factorizations like orthogonal function representations, eigenvectors estimation, Fourier phase-amplitude analysis, wavelet/Laplacian decompositions, general harmonic analysis *etc*. Each of these factorizations makes some minor assumption about the input image and exploits some core mathematical property of the extracted factors *e.g*. eigenvector analysis for face recognition, Fourier phase for image enhancement, matrix sparsity of denoising *etc*.

In addition to the analytical mathematical factorizations listed above, many machine learning based data-driven methods also benefit from image factorizations with improved performance in the end task. Several deep learning based neural networks use image factors derived from coarse-fine factorization of the input for efficient processing. Still others use techniques like gradients based detail extraction, edges/contours extraction, resolution based factorization into spatial pyramids, pixel frequency based segmentation *etc*. It has been noticed that such factorizations during learning or as a pre-processing step, leads to improved performance as it helps in introduction of prior information based on the previously proposed traditional model-based solutions. Even in recent time with the advent of large scale self-supervised vision models based on transformer architecture, factorization of image into spherical harmonics components is essential for positional encoding of data before learning.

In this thesis, we discuss image factorizations which have very high-order meaning associated with them like reflectance-shading decomposition. Such problems require controlled manipulation of the optimization process to infuse the desired properties in the respective factors. We provide solutions using both traditional model-based optimization formulation and contemporary data-driven deep-learning based neural networks. Former are helpful in understanding the mathematical properties of the factors, while latter helps in obtaining efficient systems.

1.4 Image Factorization: A Computer Graphics Perspective

The importance of image factorization, specifically the initial problem definition itself, can be better understood from the Computer Graphics (CG) perspective. Based on the input, output and focus of analysis being performed, CG research can be split into these two complementary directions:

- **Image Rendering (Forward Graphics):** Here inputs are scene parameters and output is the generated image of the scene from a given viewpoint, simulating camera captured output but in a synthetic setting. Both *rasterization*, which approximates image formation process by using

Figure 1.3: Inverse Rendering Categories: Complete Inverse Rendering, Image Based Rendering and Image Enhancement problems are all related. The categorization is done based on the extent of focus on the perceptual quality or optical accuracy of the final results (Section 1.4).

suitable shader maps, and *raytracing*, which strives to accurately simulate ray optics computationally, are two types of forward graphics problems. Various other sub problems like light transport, indirect lighting, environment lighting estimation, sub-surface scattering, atmospheric diffusion *etc*. can all be understood from rendering or raytracing perspective.

- **Inverse Rendering (Inverse Graphics):** As the name suggests, the definition of this research area is opposite of the forward graphics pipeline, *i.e*., input is a pre-rendered image and the goal is to obtain various scene parameters which lead to the formation of this image. A few sample research problems are inverse light transport, inverse tone mapping, texture extraction, BRDF estimation, illumination decomposition, intrinsic image decomposition, shape estimation, camera pose estimation *etc*. Based upon the extent of the inversion process and the significance given to the perceptual quality of the results, Inverse Rendering problems can be further categorized into three related sub-problems:

  - **Full Inverse Rendering:** Here the task is to obtain all the underlying factors separately.
  - **Image Based Rendering:** Here the image rendering process is only partially inverted and only a few essential factors are extracted which are then suitably modified and recombined to re-render the output image.
  - **Image Enhancement:** Here the focus is on directly improving the input image qualitatively without explicitly extracting the underlying factors.

The relationship between these sub-problems is illustrated in Fig. 1.3. Note that in each of the above mentioned category, image factorization plays a crucial role. Its role is straight-forward in the full inverse rendering problems which explicitly aim to decompose the image into constituent underlying factors. It is also important for image based rendering with partial image inversion and direct enhancement problems as extraction of relevant masks/maps to distinguish between the constant and variable parts of the image is necessary in such applications. In the discussions below, we pursue the graphics

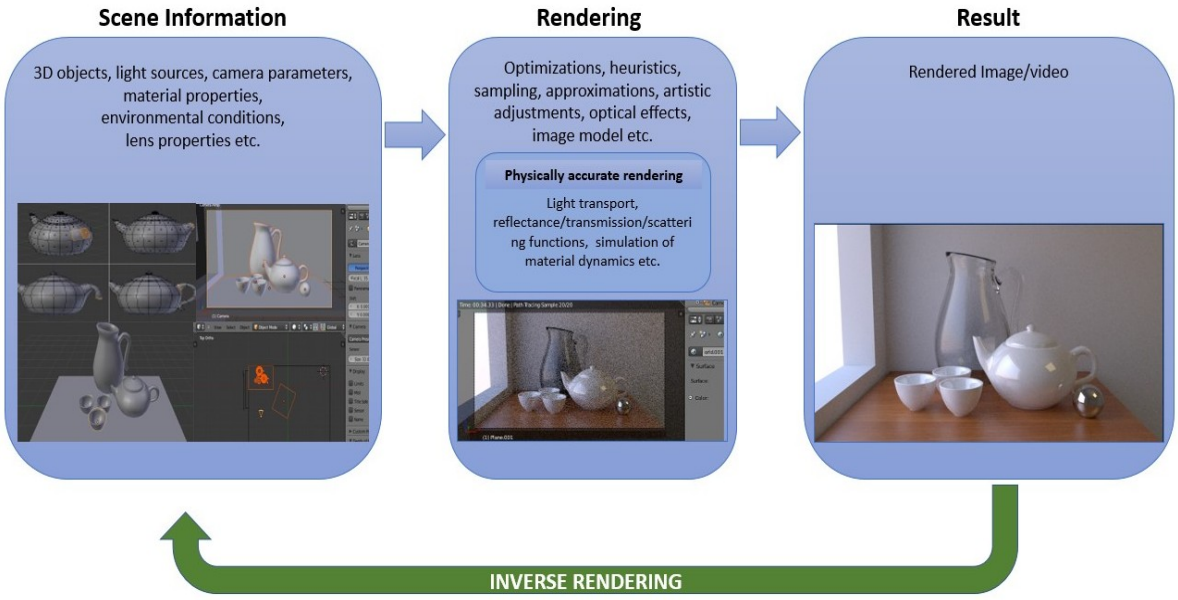


Figure 1.4: Abstract Rendering Workflow: Standard *forward graphics* pipeline and the corresponding *Inverse Rendering* workflow for synthetic scenes. Former uses parametric data to generate an image while the latter estimates these parameters from the rendered image. Note that various approximations of the physically accurate processes are necessary for feasible formulations and efficient optimization during the rendering process.

perspective for the definition of various image factorization tasks, and discuss Inverse Rendering in details in the subsequent sections.

1.5 Inverse Rendering

As discussed in the previous section, image factorization, whether partial (image based rendering/partial inverse rendering) or complete (full-inverse rendering) is an essential step in many reconstruction problems. In order to give an intuition about Inverse Rendering as a research problem, we first define its counterpart *i.e*. *Forward Rendering*. Standard forward computer graphics pipeline takes scene information like camera viewpoint, environmental lighting, material properties *etc*. as input and after assuming an appropriate image formation model, renders a visual representation of the scene in the from of image, video or 3D model as output. This forward workflow as shown in Fig. 1.4, involves several smaller reconstruction tasks like usage of various shaders for textures, visible points detection, light transport calculations, direct/indirect light estimation, shadow mapping, realistic/artistic camera effects mimicking *etc*. Apart from all of these processes which aim to increase the optical or perceptual realism of the final result, yet another set of computational optimization processes are generally

employed with the intent to improve the efficiency of the framework without significantly compromising the final perceptual result quality. Keeping this forward graphics workflow in mind, we can now define Inverse Rendering as:

Inverse Rendering is the reversed graphics workflow which takes in the rendered result as input and attempts to estimate the underlying latent scene information as output.

This definition covers the Inverse Rendering characteristics as discussed already in currently existing literature which either focus on the intrinsic parameters estimation ([248, 195, 87, 217]) or the associated applications ([186, 96, 304]).

1.5.1 Inverse Rendering: Challenges

Inverse Rendering is an ill-posed, under-constrained and challenging problem [39]. It is *ill-posed* as, many-a-times, there is no simple one-to-one mapping between the rendered output and underlying scene information. Multiple different scene configurations can result in the same rendered output. This leaves ample room for ambiguities in the problem definition and the expected results *e.g*. ambiguity between object surface reflectance color *vs*. source illumination color, light source distance *vs*. intensity, shadow *vs*. dark object color *etc*. This ill-posed nature of the inverse rendering problems requires the optimization of the solution algorithms to be shepherded by some task dependent guidance principles in order to purge the trivial solutions. Hence domain knowledge plays an important role in the inverse rendering problems.

Inverse rendering problems are also *under-constrained* in nature [39], especially in the case of a single image input. This is due to the fact that factorization of a single image into multiple constituents leads to more variables than available number of equations in the model. Naïve solutions for such an under-constrained system generates infinitely many solutions, only few of which are perceptually and optically sensible. This issue is often tackled by introducing additional constraints into the system in the form of *priors*. These prior terms are created using knowledge about the nature of the problem and/or real-world/perceptual image formation information and assumptions *e.g*. the Retinex prior, constant albedo constraint, color line hypothesis, chromatic gradient clustering *etc*. This technique drastically reduces the search space and guides the optimization towards the desired perceptually and optically sensible solution.

Additionally, physical accuracy of the finally rendered result is affected due to various approximations employed during the forward rendering optimization. Some of these aspects are discussed below:

- **Theoretical simplifications of light interaction:** The assumptions regarding number of light bounces, intensity spectrum, reflectance function distribution *etc*. are inherent in the mathematical formulation of the assumed image formation model. Many image formation models like Retinex, Lambertain reflection, dichromatic reflection, Fresnel transmission, Rayleigh and Raman scattering, anomalous diffraction, internal reflections *etc*. vary according to their optical accuracy. As natural scenes are rich in many light scattering phenomena (Fig. 1.7) co-occurring and co-dependent on each other, it is both analytically and empirically very difficult to exactly simulate scene lighting even in a high end graphics pipeline. A frequently employed workaround for this problem is to neglect any rare optical phenomena and use simplifying assumptions for the complex models. This brings the results perceptually closer to the real world scenes but provide no guarantees for the optical accuracy.

- **Imprecise parameterization of materials and camera:** In addition to non-linear optics and corresponding complex mathematical models mentioned above, some materials too have complex reflection properties like anisotropic reflection, subsurface scattering, dichroism, birefrigence, polychromatism, pleochroism *etc*. These effects are produced due to rich, non-uniform and complex nature of interaction between object material property and illumination. A few of such special optical effects like Tyndall scattering, edge diffraction, halos, light shafts, backscatter, airglow *etc*. are show in Fig. 1.8. Similar to rare optical phenomena, simplifying assumptions and rule based or scene specific heuristics are used to represent such complex material properties which again limit the optical accuracy of the rendering process.

- **Digitization residues:** Rendering is a digital process and hence it implicitly involves steps like quantization, compression and finite analysis during processing *e.g*. finite number of integral computed, limited sampling, limited parametric representation and precision, dynamic range clipping, color quantization, tone mapping *etc*. All such digitization errors arising both during the digital modeling and computation of the rendering process, further deviate the outputs from the optically correct results.

All of the above mentioned challenges compounded together makes it quite hard to realistically simulate the full rendering process with high optical accuracy. This separates the synthetically rendered results from the real world images, hence uniquely demarcating the synthetic *vs*. real-world data domains. This domain split has significant implications in the optimization processes used in the inverse rendering research. We further explore and discuss these implications in the next section.

1.5.1.1 Synthetic *vs*. Natural Domain Dichotomy

Due to the various issues in rendering mentioned in the last section, there is a clear divide between the data domains of synthetically rendered scenes and real-world images. Both forward and inverse

Figure 1.5: Domain Dichotomy: Synthetic (top) *vs*. natural (bottom) images. Note how there are significant differences between the two domains. Even when carefully rendered, compared to their synthetic counterpart, natural images have more complex materials, lighting, background, textures, depth effects, camera artifacts *etc*. Synthetic images taken from As Realistic As Possible dataset [39] which makes sure that the statistics of the rendered results are similar to real world images. Real images taken from wikimedia commons.

rendering research fields aim to bridge this gap either by realistically rendering synthetic scenes or by using synthetic scenes to train models which predict scene parameters for real world images. Specifically, forward graphics techniques use synthetic data domain as input and try to produce as realistic as possible renders. Contrarily, inverse graphics techniques take in natural images as input and factorize them into constituent parameters as understood by the standard synthetic rendering framework. Hence this domain split has a significant effect on both the graphics and the inverse graphics pipelines.

Compared to the realistic forward graphics, inverse rendering of real world images is a still more challenging problem. In addition to all the issues discussed previously, real world images are significantly more challenging due to additional perceptual complexity (see Fig. 1.5). Real scenes comprise of substantially *higher variety* in the terms of object shapes, textures, lighting sources, material reflectivity properties *etc*. Complex geometry of even the everyday simple objects is something quite hard to achieve in general synthetic scenes. Similarly the wide variety of textures like fur, hair, rust, *etc*. and the optical properties of materials like translucency, caustics, sub-surface scattering *etc*. are present in nearly all natural images but rarely so rendered properly. The number and type of illuminating sources and complex light scattering due to mathematically hard-to-model interactions between all the factors mentioned above, compound this issue further. As natural images comprise the input domain for an inverse rendering system, this input data choice selectively differentiates and complicates such research problems from the standard graphics research. Owing to this imprecise nature, the inverse rendering

research has interestingly evolved as a sequence of Computer Vision, Machine Learning and Graphics tasks.

Yet another significant challenge in applying inverse rendering to natural images is the lack of proper *ground truth* data. This is quite obvious as it is very difficult, if not utterly impossible, to collect all the required rendering information from a natural scene like high resolution object shapes, accurate material properties of all surfaces with corresponding textures, including all illumination sources with complete spectral and geometric information, optical properties of the transmission medium, representation capability of the acquisition equipment *etc*. This is due to the unavailability of proper sensors with sufficient resolution, lack of any good method to capture relevant material/texture data and difficulty in representing per 3D point omni-directional illumination and scattering information. This lack of sufficient good quality large synthetic datasets have slowed the progress of the inverse rendering research and has restricted the applicability of training based *i.e*. data-driven optimization frameworks for such problems.

1.5.1.2 Manual *vs*. Data-driven Optimization

In addition to the challenges faced due to the type of data domain (synthetic *vs*. natural), for inverse rendering research, further complexity arises based on the type of chosen processing or optimization. Due to all the issues discussed above in attempting inverse rendering for both synthetic and natural images, the type of solutions generally proposed in Inverse Rendering research comprise of several sub-systems and have been rarely a direct single end-to-end supervised framework. Most of the inverse rendering research involve direct or indirect usage of domain knowledge in the form of priors, theoretical model assumptions or implementation approximations. Hence earlier works were mostly manually made, model-based *optimization frameworks* containing none or a very few supervised subsystems. Hence, until recently, training based systems for such problems have seen limited feasibility. Although this trend is now changing now with the resurgence of data-driven frameworks and introduction of large datasets.

With the advent of powerful data-driven optimizers in the form of *deep learning frameworks* like Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs), Variational Auto Encoders (VAEs) *etc*., some inverse rendering solutions have been proposed providing end-to-end scene information in some limited sense. Although these methods using data-driven frameworks have been able to attempt solution for smaller tasks and applications, still finding a generic and complex inverse rendering solution is currently an open research problem. State-of-the-art solutions here are generally the ones which pragmatically redefine the problem limits and creatively encode the appropriate domain knowledge into their frameworks. Some of these solutions work by generating synthetic data and using them for supervision during training and then testing on real world scenes which are prone to the problem of *domain shift* between synthetic and natural images. One way to tackle this

issue is to use still larger quantity of diverse data thus increasing the cost of such solutions. Yet another way is to use appropriate *Domain Adaptation* or *Knowledge Transfer* techniques to adapt the framework for different environments than training dataset. Although several domain adaptation methods exit in the literature for recognition problems but lesser for reorganization and still lesser for reconstruction and inverse rendering tasks. This is due to the complex nature of the latter topics. As a result domain adaptation techniques have seen limited usage for inverse rendering problems leaving an interesting research direction to be explored in the future.

Yet another method is to forgo supervision by substituting it with *unsupervised learning* or *self-supervised learning*. Neural encoders like deep convolutional Auto-Encoders (AEs), Variational Auto-Encoders (VAEs) and transformers can bypass the requirement of complete supervision by learning an appropriate latent space of variables from unlabeled or partially labeled data. AutoEncoders with their encoder-decoder architecture can be used to disentangle underlying scene information by encoding the input data in a lower dimensional space representing latent generative factors. Similarly Variational AutoEncoders can be used to learn the latent space of underlying generating probability distribution variables. Transformers on the other hand employ pseudo labels with a pretext task learning based self-supervised strategy by either auto-regressing future image patches or doing masked content prediction. Although such methods are quite useful as they bypass the expensive step of collection of large quantities of annotated training data, but it is difficult to guide the learning process to make sure that the learned latent space represents an encoding of semantically meaningful scene factors. This can be helped somewhat by properly managing the training process and designing the loss functions appropriately so that the known properties of variables like color sparsity of scene albedo, illumination gradient distribution, perceptual quality of the generated result *etc*. are properly represented in the framework. Hence domain knowledge about the application in the term of priors and the underlying modeling assumptions in the inverse rendering problems from the classical literature is of great benefit in designing even while training advanced neural networks architectures.

Apart from using discriminative supervised deep learning paradigm, it is also possible to learn the *generative distribution* of underlying inverse rendering factors, conditioned on a given input image. This can be used to categorize the solutions using such generative deep learning architectures like Generative Adversarial Networks (GANs) and VAEs compared to the usual discriminative deep learning paradigms. GANs work by training two adversarial systems comprising of discriminating and generating subsystems, with the training and the loss function designed in such a way so as to enable simultaneous learning of both together [98]. The adversarial criterion of GANs guide the learning process so that the generator network gets better and better at generating the desired output while discriminator too improves to distinguish between the fake and the real outputs better. The training finally ends when the discriminator is unable to distinguish between the generated fake and the ground truth real output. VAEs on the other hand use the variational reformulation of the problem and train

the network in the parametric space representing the family of distributions from which the given input is assumed to have originated. At the test time both the GANs and VAEs can be conditioned by providing query input data in order to sample the desired output.

As inverse rendering formulation is inherently linked to the reconstruction problem, it is more intuitive to use a generative architecture rather than a discriminative one but the generative spaces learned by such methods need to uphold certain desired mathematical properties like *stable dynamism and orthogonal factorization* [144, 61, 188, 196, 183, 63, 71]. If the dynamic behaviour of a generative framework has to be stable then the system should be *well-behaved and non-chaotic* in nature *i.e*. subtle input perturbations result in only subtle output perturbations. This enables easier modification of the underlying factors for the final reconstruction application [63, 144]. This can be illustrated with the help of material synthesis problem. In material synthesis, the goal is to obtain a generative framework over the space of material parameters or features derived from them, so that the system can generate new materials by traversing the generative space in a predictable fashion for a given input conditioning *e.g*. if input represents a blue metallic smooth material, minor perturbations of the parameter representation in the generative space should lead to results closer in appearance to the input *i.e*. change in blue color or metallicity or smoothness of the spherical surface. Such predictable dynamic nature of the learned latent space is essential for inverse rendering problems as it is important during downstream reconstruction application. Similarly, another vied mathematical property is *orthogonal factorization* by which it means that generative factors should be disentangled in the learned generative space as much as possible [61, 71]. If the factor representations in the learned space are orthogonal to each other (*e.g*. by getting embedded along different latent dimensional axes or combination thereof), then each one of them can be manipulated independently. Without any additional effort such disentangling is not ensured in standard deep generative frameworks. Hence a good inverse rendering generative model must possess both predictable dynamism and orthogonal factorization as fundamental characteristics.

One more issue with using generative paradigm directly for this purpose is the limited resolution of the results. Naively increasing the resolution of the output by modifying the output layer size causes massive increase in the number of learnable parameters causing complications in the learning process. This issue is specifically pronounced in the case of 3D data (point clouds, voxels, multi-view images *etc*.) and still more so in case of 3D time series input. Furthermore while designing the frameworks, one needs to pay special attention to the requirement of *perceptual and semantic validity* of the results, as the basic criterion of adversarial learning and variational optimization do not focus on these as design principles. Due to this, generative deep neural inverse rendering solutions generally have additional post-processing networks, pre-processing or optimization steps to get the final reconstructed results [292, 201, 132, 259]. Similar intuition can also be employed at the input stage by pre-processing the image and feeding into the network as concatenated volume or as an input to a parallel network branch. This simplifies the function the network is expected to optimize by

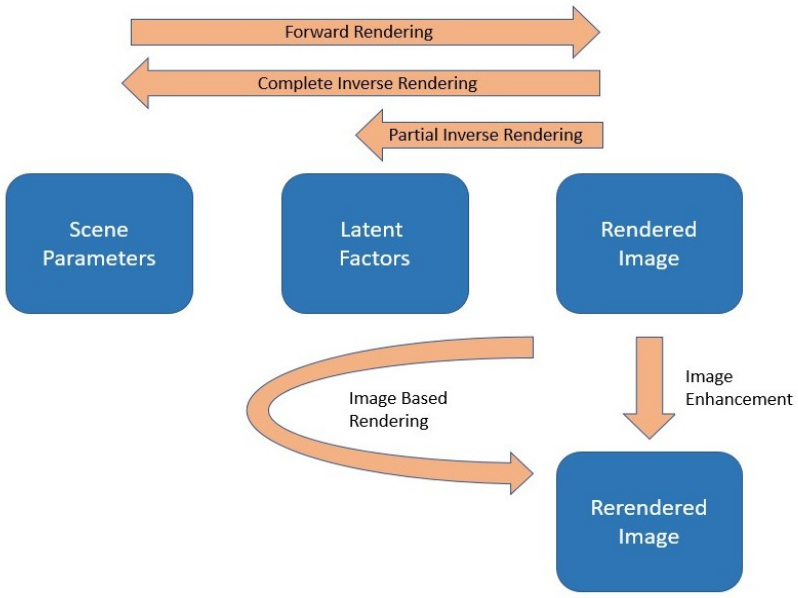


Figure 1.6: Inverse Rendering Categorization: Various subcategories of Inverse Rendering problems and their relationship with each another (Section 1.5.2).

providing additional input variables and by dividing the complex function into smaller and simpler learnable parts.

In our thesis, we merge the two paradigms: classic manual model-based optimization with emphasis on handcrafted cost terms encoding domain expertise as priors and data-driven frameworks of supervised or unsupervised nature with efficient feature representation for various image reconstruction tasks. The intuition behind focusing on optimization based methods is to understand the fundamentals of the inverse rendering problems better by manually designing the optimization framework, using explicit model assumptions, selecting proper loss functions, handcrafting novel prior cost terms using domain knowledge and employing suitable numerical algorithm.

1.5.2 Inverse Rendering Applications

As mentioned previously briefly, inverse rendering research encompasses several image reconstruction tasks. Based on the extent of the inversion carried out and the kind of output desired, these research problems can be broadly categorized under three headings which are described below:

- **Complete Inverse Rendering:** This category contains the research problems which strictly adhere to the inversion definition. Such problems fully invert the rendering pipeline mentioned in the Fig. 1.4 to obtain outputs in the form of completely factorized scene information. As

Airglow: Nocturnal faint emission of light by planetary atmosphere caused by upper atmospheric processes like recombination of solar photoionized atoms, cosmic ray luminescence and atmospheric chemiluminescence due to O_2, N_2 and OH radicals.

Gegenschein/Counterglow: Backscatter of sunlight by interplanetary dust, is a faintly bright spot in the night sky centered at the antisolar point.

Brocken spectre/Mountain spectre: Magnified shadow of an observer cast upon clouds opposite the Sun's direction caused when uniformly-sized water droplets in clouds refract and backscatter sunlight.

Heiligenschein/Cellini's halo: A bright spot around the shadow of the viewer's head in the presence of dew.

Mirage: Light rays bend via refraction near the surface due to atmospheric heat, to produce a displaced image of distant objects or the sky, giving a mirroring effect.

Afterglow: A broad arch of whitish or pinkish sunlight in the sky that is scattered by fine particulates, like dust, suspended in the atmosphere (different from Belt of Venus and appears in the geometrically opposite part of the sky).

Light Halo/Glory: Solar or lunar light interacting with ice crystals suspended in the atmosphere to generate a circular arcs of varying hues due to dispersion.

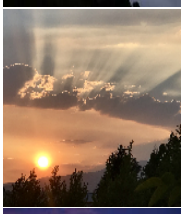
Anti-crepuscular rays: Radiating beams of sunlight that appears to converge geometrically opposite to the Sun's position during twilight.

Zodiacal light/False Dawn: Sunlight scattered by interplanetary dust, seen as a faint, diffuse, and roughly triangular white glow that is visible in the night sky and appears to extend from the Sun's direction and along the zodiac.

Cloud iridescence/Irisation: Colorful optical diffraction in a cloud that appears in the proximity of the Sun or Moon due to small water droplets or small ice crystals individually scattering light.

Parhelion/Sundog: A bright spot to one or both sides of the Sun in a 22° halo.

Earth shadow and Venus Band: Shadow of Earth casted onto its atmosphere. A related phenomenon in the same part of the sky is the Belt of Venus, a pinkish band visible above the bluish shade of Earth's shadow.



Sunbeam: Parallel beams of sunlight that appears to radiate from the Sun's position due to atmospheric scattering.

Light pillar: Vertical beam of light appears to extend above and/or below a light source caused by the interaction of light with atmospheric ice crystals.

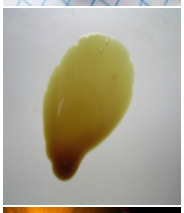
Figure 1.7: Complex Light Interaction: Challenging optical phenomena in real world images caused due to a combination of various complex atmospheric effects and nonlinear optics (source: wikimedia commons [3]).

Tyndall effect: Light scattering by particles in a colloid or in a very fine suspension. Longer wavelengths are more transmitted while the shorter wavelengths are more diffusely reflected via scattering.

Light polarization: Light of different wavelengths gets refracted by different amounts based on their polarization in some materials like translucent plastics.

Total internal reflection: TIR occurs when waves in one (internal optically slower) medium reach the boundary with another (external optically faster) medium at a sufficiently slanting angle causing a reflection like a mirror, with no loss of brightness.

Birefringence: Caused when a material has a refractive index that depends on the polarization and propagation direction of light (might cause double refraction).



Dichromatism: A material or solution's hue is dependent on both the concentration of the absorbing substance and the depth or thickness of the medium traversed.

Lens flare: Light is scattered or flared in a lens system, often in response to a bright light due to internal reflection and scattering within the lens.

Subsurface scattering: Incident light penetrates the surface of a translucent object and is scattered by interacting with the material and then exits the surface at a different point.

Iridescence: Hue changes with the angle of observation and the angle of illumination. Caused by multiple reflections from semi-transparent surfaces in which phase shift and interference of the reflections modulates the incidental light frequencies.

Dichroism: When incident visible light is split up into distinct beams of different colors upon reflection (refraction for dispersion) or when rays having different polarizations are absorbed by different amounts.

Pleochroism: When a substance has different colors when observed at different angles, especially with polarized light caused due to varying optical properties with the direction of light.

Fresnel reflection: At near-grazing incidence, media interfaces appear mirror-like especially due to reflection of the a particular polarization component, despite being poor reflectors at normal incidence.

Edge Diffraction: Edge diffraction is generated by the interference between the flat wave front of the incident radiation from a collimated light source and the secondary spherical waves generated by the edge of the object.

Figure 1.8: Insufficient material and camera parameterization: Challenging optical phenomena in real world images caused due to a combination of various complex material properties, fringe optical phenomena and other acquisition anomalies (source: wikimedia commons [3]).

expected this is the most difficult of the all categories listed here but also allows the maximum flexibility to the subsequent applications as complete scene understanding is available in terms of rendering parameters. Some sample problems in this category are object material analysis,

image camera parameters estimation, image localization, image light source location, irradiance spectrum analysis, scene geometry estimation, light/radiance field estimation, *etc*.

- **Partial Inverse Rendering:** Several computer vision and graphics applications do not require complete inversion of the rendering process and can work with partially inverted parameters. Hence this category of IR research problems only performs partial rendering inversion and generate parameters which might be considered as transient factors during the forward rendering process *e.g*. Intrinsic Image Decomposition (IID) which separates the intrinsic image generation factors like reflectance and shading or Image background separation or Image layered representation where only partial factorization of the image is carried out instead of complete geometric and semantic understanding of the scene. Such transient parameters represent combination of several constituent factors which might be hard to disentangle further due to the complexity of the rendering process and implicit model assumptions during the partial inversion. Several such problems are posed as two factor disentangling problems with an aim to discover useful latent space representations amenable for limited recognition and reconstruction applications. Most of the Computer Vision and Image Based Rendering applications take advantage of such partial inversion directly or indirectly. Hence this is the most widely utilized category of inverse rendering solutions. Some more examples of partial inversion research problems are: alpha matting, green screen keying, style-content separation, shadow removal, specularity assessment, low light enhancement, plane fitting, lines and corners detection, symmetry detection, object detection, texture extraction *etc*.

- **Image Enhancement:** In this category we have all those research problems which are more closer to application rather than the optical accuracy of the decomposed factors. Unlike the first two categories, for these problems the output is generally a re-rendered image derived from the original input without proper factorization. The level of inversion is very limited if not negligible, and generally depends on the complexity of the data, expected quality of the results, optical accuracy and the desired level of perceptual photo-realism. A few sample applications belonging to this category are: color harmonization, object insertion, color constancy correction, image tone adjustment, style transfer, image colorization, content hallucination *etc*.

An abstract pictorial representation of these categories and their mutual relationships is shown in Fig. 1.6. Note how this expands the concept presented in Fig. 1.4 by further subdividing the inverse rendering workflow based on the extent of the inversion carried out and the whether the end goal is parameter estimation or image generation. Several existing inverse rendering image reconstruction applications are listed in Figs. 1.9 and 1.10.

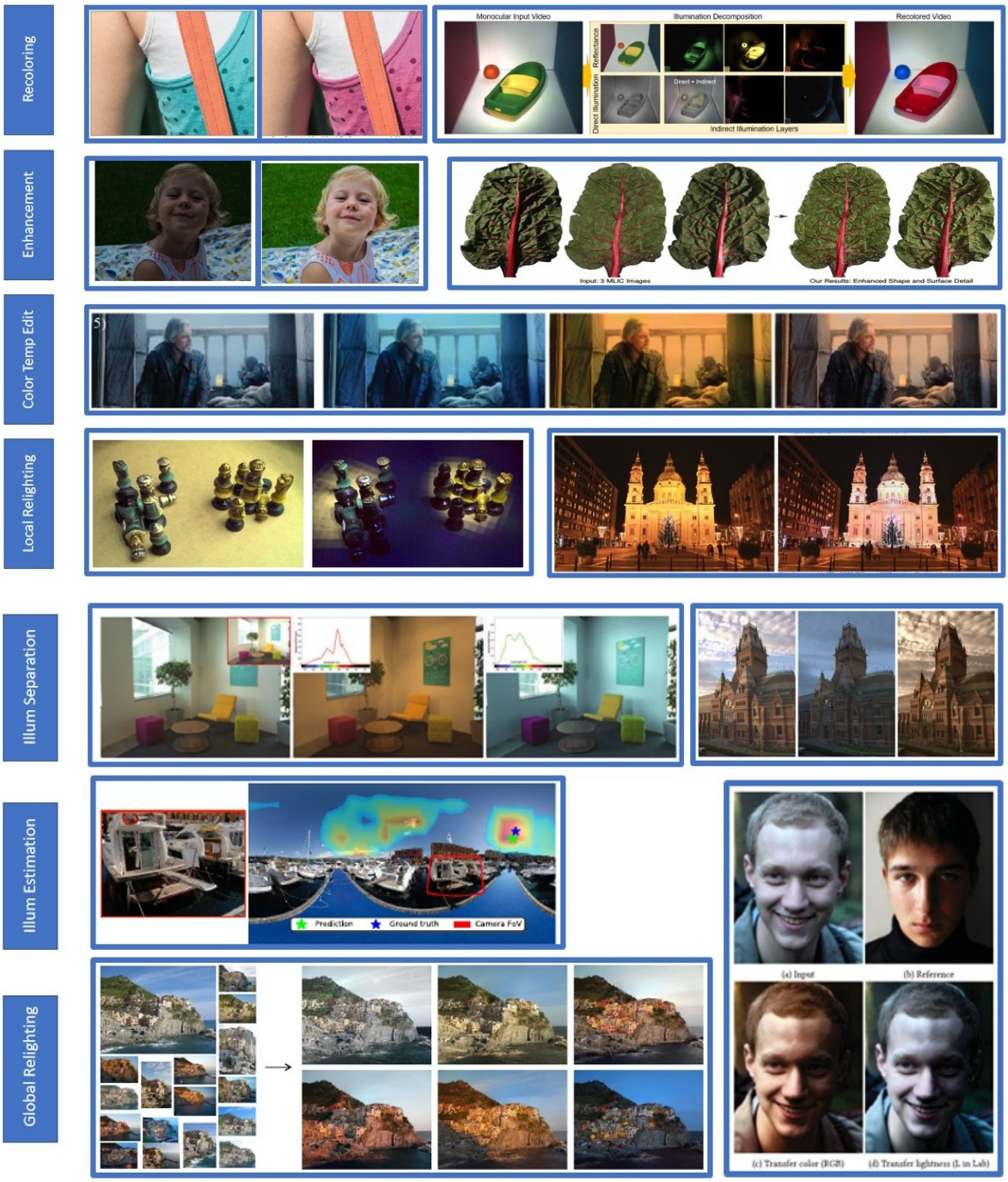


Figure 1.9: Inverse Rendering Tasks and Applications: Various examples of CV and Graphics applications using Inverse Rendering (Section 1.5.2) (images courtesy: cited papers in the bibliography).

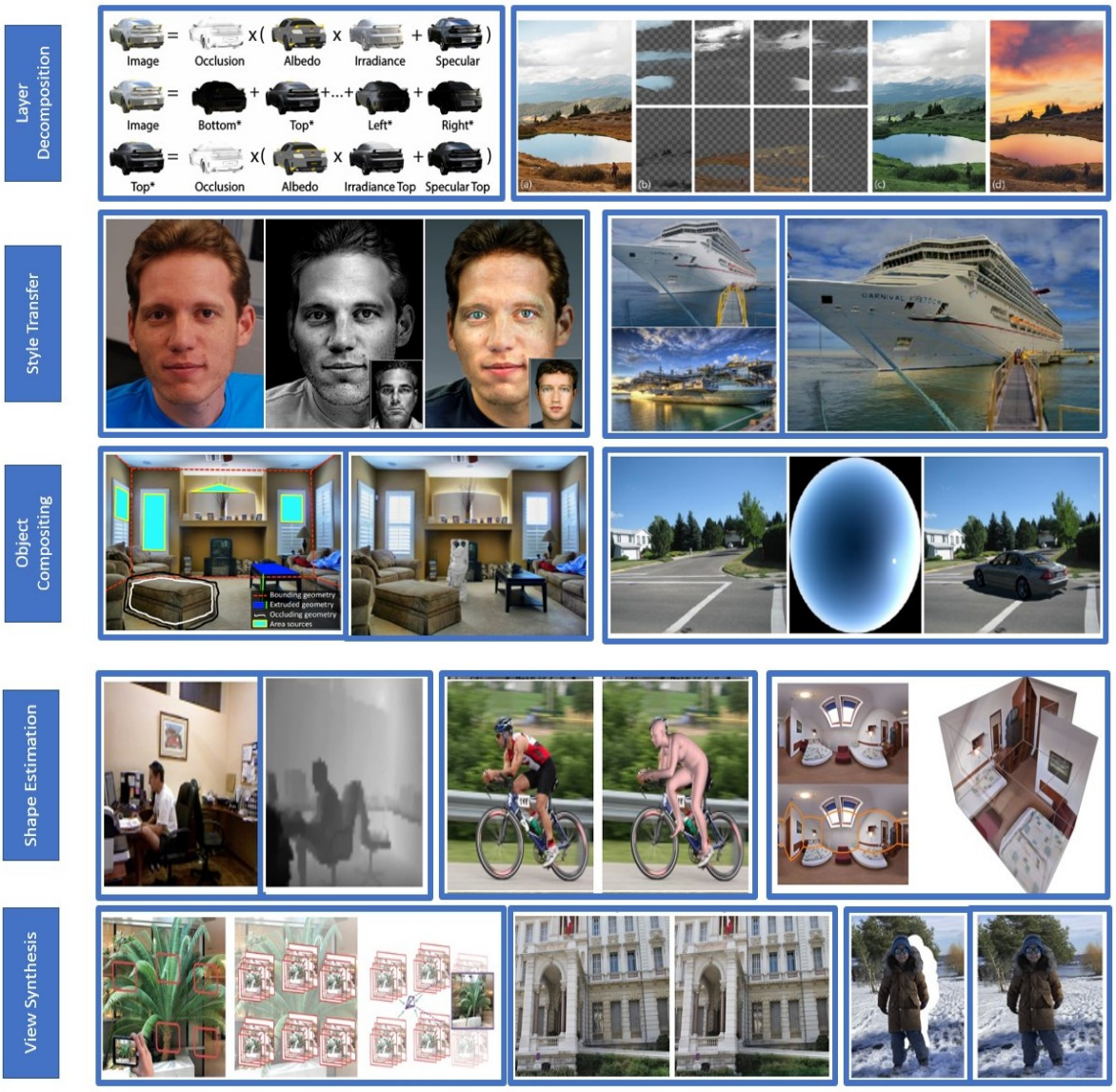


Figure 1.10: Inverse Rendering Tasks and Applications: Various examples of CV and Graphics research applications using Inverse Rendering (Section 1.5.2). (images courtesy: cited papers in the bibliography).

1.6 Thesis

1.6.1 Motivation

Considering the challenging but fundamental nature of the inverse rendering problem, we focus our attention on the core task of image factorization and hence the thesis is titled “Image Factorization

for Inverse Rendering". Note that this reflects our perspective of approaching this task with a focus on the problem formulation, instead of the optimization process being used (*i.e*. optimization perspective) or the importance of downstream application (*i.e*. computer vision perspective). Image factorization or image decomposition (used interchangeably) allows us to extract and study certain aspects of the image formation process like illumination, albedo, camera configuration *etc*. On a much wider scale it is essential and ultimately leads to the complete scene understanding. We study this task using two main research threads:

- Intrinsic decomposition, *i.e*. separation of scene intrinsic reflectance-shading components for a single scene with a focus on minimal acquisition cost and general applicability.

- Illumination factorization, which involves decomposition of image based on the scene lighting instead of object color or shape with a focus on low-light image enhancement as application.

The first problem is chosen as it provides a unique learning opportunity to start from any natural image in the wild and perform practical inverse rendering by extracting reflectance and shading components which respectively contain the object based material and scene illumination information. This alone has been shown to enable a lot of downstream computer vision applications. The second problem provides a new perspective of factorizing the input image based on the illumination information within the scene in-place of the standard color based analysis. We show how this inherently contains scene structural information and can enable diverse image enhancements.

Furthermore, to keep our proposed solutions practically relevant and easy to use, we take in only real world RGB images as input. In order to maximize understanding and learning, we have also restricted ourselves to interpretable optimization techniques. This means frequent use of tradition model-based energy optimizations and usage of explainable neural network design blocks instead of blindly using direct end-to-end black box learning frameworks. Additionally, most of the applications discussed in the subsequent research work require only partial inverse rendering, which hence forms the type of inverse rendering category we opt for in this thesis. All these design choices and constraints, along with previously mentioned challenges (Section 1.5) significantly burdens the process and restricts the allowed solution space. One major benefit though is that this provides a valuable learning opportunity and reveals deep insights about each task, which we believe is the fundamental purpose and motivation of this or any thesis.

1.6.2 Road map

The work done in this thesis is organized in four units:

- **Overview:** Our first unit (this one) consists of a single chapter wherein we introduce the topic of image factorization, localize subjects it pertains to, discuss related research areas, challenges, applications and define our problem/solution space restrictions and general design choices.

- **Intrinsic Decomposition:** The second unit discusses the problem of Intrinsic Image Decomposition (IID) which is a classic partial inverse rendering problem and a challenging task aptly portraying the essential characteristics of all inverse rendering problems. We present two IID solutions in two chapters representing two different scenarios. In the first scenario we have access to some auxiliary scene information in the form of a burst of variable focal length images of the scene. While the second one is where where we are provided only with a single input image.

- **Illumination Factorization:** In the third part, we focus on analyzing the scene illumination from a single image input by factorization it based on similar optical characteristics. Unlike IID, which outputs only two hard dense factors, this splits the image into multiple sparse soft factors each approximating various image regions with consistent illumination. We use Low Light Enhancement (LLE) as the driving application to highlight the utility of our proposed approach and present both traditional model-based and contemporary neural-network based solutions in two chapters.

- **Analysis and Summary:** In the last unit we discuss the observed issues with the evaluation of such solutions. We also provide a knowledge distillation technique to properly evaluate and infuse desired abstract solution qualities in the existing solutions. Finally in the last chapter we summarize all the work and paint out the future directions of research corresponding the observed but unanswered questions and paths not chosen.

PART II

Intrinsic Decomposition

Chapter 2

Intrinsic Image Analysis with Auxiliary Scene Inputs

In this chapter we discuss our proposed solution, RGBF-IID, for the first category of Intrinsic Image Decomposition (IID) problem *i.e*. we present a method which performs IID in the presence of auxiliary scene information. IID refers to binary factorization of image into its reflectance and shading components. Auxiliary information is required for IID as it is an under-constrained problem. The extra information we require is obtained in the from of a focal stack during the initial image acquisition stage via a focal length burst-bracketted photography. Similar to RGBD, we call this modality as RGBF where F represents varying focal distance. Such an input is relatively easy to obtain compared to other kinds of auxiliary data like depth, by using any standard commercially available digital camera and open source camera API. This keeps our technique generalizable, without incurring any significant acquisition cost. We use a robust focus measure and generalized random walk algorithm to compute dense probability maps across the stack. These maps are then used to define sparse local and global pixel neighbourhoods adhering to the structure of the underlying 3D scene. We use these neighbourhood correspondences with standard chromaticity assumptions as constraints in an optimization system. We present our results on both indoor and outdoor scenes using manually captured stacks of random objects under natural and artificial lighting conditions. We also test our system on a larger dataset of synthetically generated focal stacks from natural RGBD scenes and computer generated images with depth buffers. Our method provides a strong evidence for the potentiality of using RGBF modality in place of RGBD in computer vision problems and can estimate intrinsic images for any wild scene without any restrictions on the complexity, illumination or scale of the image. We define the problem setting, feature extraction, model assumptions, experiments and results in the following sections below.

2.1 Introduction

Intrinsic image decomposition (IID) is a classic problem in computer vision. IID involves the decomposition a given image into its constituent reflectance (albedo) which is based on the material properties of the scene and the illumination (shading) representing the effect of scene lighting on object geometry. Apart from being an interesting inverse rendering research problem in itself, IID proves to be useful in several other computer vision problems such as image colorization [181], shadow removal [148], image enhancement [72], image recoloring [301], image relighting [69] *etc*.

IID is an ill-posed problem [39, 129]. It involves decomposing an image I of a natural scene into its intrinsic components as:

$$I \quad = \quad R \cdot S, \tag{2.1}$$

where R represents reflectance, S represents scene shading and $\cdot$ represents element-wise multiplication *i.e*. Haddamard product. Note that Eq. (2.1) can have multiple possible valid combinations with a common scalar multiplier and divisor for either of the two components. Hence the equation has infinite possible solutions. Furthermore, for each pixel in image I, we need to estimate two variables which makes it an under-constrained setting. Solutions for Eq. (2.1) are thus proposed under various assumptions like color consistency, solution sparsity, Lambertian reflection *etc*. The *color consistency* assumption can be traced back to the Retinex algorithm [155] which assumes that objects with similar chromaticity indicate similar reflectance and that the image can be divided into the above two disentangled terms. Yet another commonly employed assumption is to *ignore all specular reflections* in the given scene to help simplify the mathematical image model.

In order to resolve these challenges, several approaches therefore propose to use auxiliary scene information in the form of multiple views, multiple illumination images, scene depth *etc*. Our proposed method here also falls into this category. Specifically, in this work we present a method to compute reflectance and shading images for a given scene using a set of input images as focal stack of the scene. We refer to our proposed method as '*RGBF-IID*' for the rest of the chapter. Formally defining:

A focal stack is a set of images of the scene captured at varying focal distances.

It is a relatively easier way to capture scene structure information compared to other methods which require careful camera positioning, specialized in-scene exposures settings, simple materials or a specialized depth sensor. Capturing a focal stack is possible without any extensive setup up or training. Focal stack capture is mostly an automated process and is possible on devices ranging from high-end DLSR cameras to everyday point-and-shoot smartphone cameras. This makes the idea of using focal stacks for IID quite useful and interesting. Moreover, since we directly use images and not any specialized sensors, our method can be used for a wide variety of both indoor and outdoor scenes under natural and artificial lighting conditions where depth sensors tend to give poor resolution due to interference.

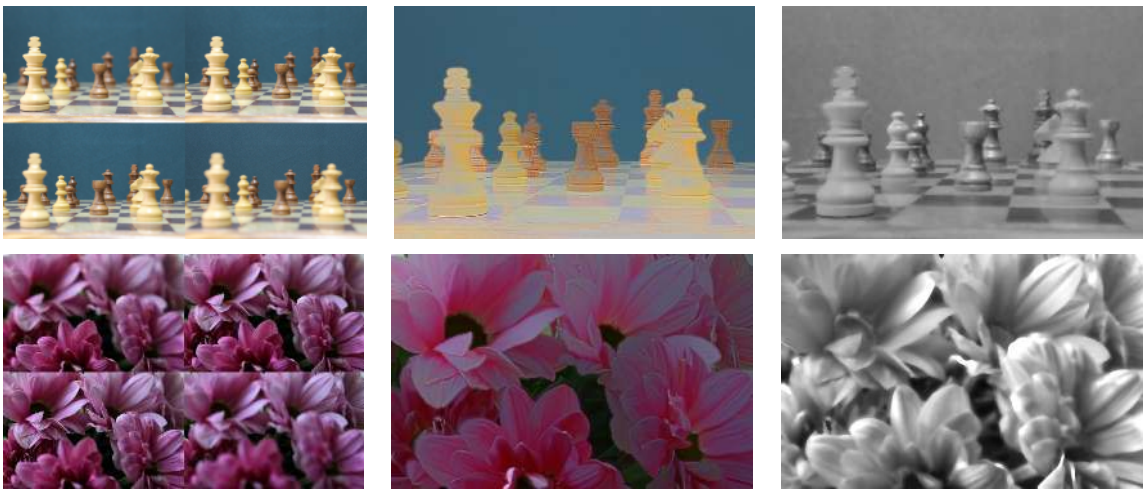


Figure 2.1: RGBF-IID: Our method takes a focal stack and computes intrinsic images of the scene. Left column shows four sample images from a sample focal stack with each image focusing at different depth in the scene. Middle column shows the resultant reflectance and right shows the shading component of the scene as estimated by our proposed method. Note how the shadows and highlights on the chess pieces are separated out in the shading component while reflectance component exhibits textures and uniform color values.

In a focal stack, the amount of defocus of a given scene point in a focal slice depends on its depth from the camera. We use this observation in conjunction with a robust focus measure and design features to be used as a proxy for depth. We define local and global point neighbourhoods based on this RGBF representation where F refers to the focus derived features. Specifically for RGBF-IID, we compute a dense probability map for each slice corresponding to the contribution of that slice towards the all-in-focus image. One minor assumption used in our image model is that we assume that pixels which achieve similar sharpness variation across the stack lie in a similar depth range in the scene. This is a standard assumption in many previous image processing papers dealing with focal distance. To summarize, our main contributions in this work can be summarized as given below:

- We present a novel way to obtain IID using focal stacks which are easy to capture and applicable for any scene.
- We show that an easy-to-compute RGBF representation can be used as a substitute for RGBD based solutions. This opens up several exciting research directions for other computer vision problems.
- We show comparisons with RGBD based IID methods on both real world and synthetic benchmarks and also show results on randomly captured images from a mobile phone camera.

2.2 Background

In this section we discuss the relevant literature for our proposed RGBF-IID method. We divide this discussion into two parts. In the first part we discuss the literature related to focal stack generation and its use in computer vision applications. The second part deals with various relevant solutions for the IID problem.

2.2.1 Focal stacks

As mentioned earlier focal stacks are images of a scene captured from a fixed viewpoint with different focus distances. Focal stacks are not a novel modality and have been used earlier in computer vision to compute the all-in-focus image of the scene [10, 206, 290]. This is especially useful for macro imaging where it is very difficult to capture an all-in-focus image in a single shot. Hasinoff and Kutulakos [114] shows that for any given depth range, generating an in-focus image from a focal stack is more time efficient than capturing a single narrow aperture image.

Different parts of the scene come into focus in different focal slices based on their 3D location in the scene. Measuring the amount of focus for each pixel across the stack can thus serve as a proxy for 3D scene information. Depth-from-focus using focal stacks has been attempted by [211] and [261]. Some Depth-from-defocus methods also make use of pairs of differently focused wide aperture images [59, 78].

For fusion of multi-focus images we adapt the system proposed by Shen et al. [255] for multi-exposure image fusion. They generate dense probability maps for each exposure slice based on the probability of each pixel belonging to that slice. Specifically, they use Generalized Random Walks (GRW) to obtain a high dynamic range output from several low dynamic range inputs. Their framework is useful for solving labeling problems where one desires candidate probabilities for each site. We discuss how we use their framework for our problem in more detail in Section 2.3.2.

2.2.2 Intrinsic Image Decomposition

We restrict our discussion here to automatic IID methods only which require no user annotations. Such methods fall under two categories: those which work on a single image and those which use auxiliary scene data. IID methods which work on a single image require several assumptions as stated earlier. For example, in Tappen et al. [263] image derivatives are used and classified to be caused by either change in reflectance or shading and then a generalized belief propagation based framework is used for final labeling. Barron [23] uses new priors on reflectance based on sparsity but results are shown only on simple images. In Barron and Malik [24], more priors are introduced on smoothness, entropy, absolute color and illumination in a multi-scale optimization framework and

improved results are reported. In Bell et al. [33] a new manually annotated dataset is introduced and fully connected conditional random fields are iterated over for reflectance estimation followed by shading optimization. Gehler et al. [91] assumes sparse basis for colors and estimates them as latent variables using a Gaussian distribution based random field model. Garces et al. [89] use k-means clustering to find group of pixels and estimate reflectance discontinuities which they use in a simple linear system for IID.

The second category of methods leverage extra information from auxiliary data to produce more constraints. Weiss [284] uses multiple images under different illumination conditions. They use natural scene statistics and a Laplacian distribution model and optimize using maximum-likelihood estimation. Liu et al. [181] estimate intrinsic components using multiple images of a given scene for grayscale to color conversion. They register the images using manual features like SIFT [187] and apply RANSAC [80] and median filtering over the overlapping segments with a maximum-likelihood optimization like Weiss [284].

Most relevant approach to ours is by Jeon et al. [129] and Chen and Koltun [60]. Both these works use RGBD information of the scene to compute IID. Chen and Koltun [60] decompose shading into three components caused by direct irradiance, indirect irradiance and illumination color. Different constraints are applied to each component based on pixel neighbourhood values obtained using the depth data. They pose the final optimization problem as simple linear least-squares framework. Similar to Chen and Koltun [60], Jeon et al. [129] also work on RGBD images. They first separate image textures using smoothing filters and then define local and global shading constraints using depth based point neighbourhood estimates. They reformulate these constraints as a Local Linear Embedding [232] of sparse features which is then propagated to rest of the image using separate propagation terms. Our method can be thought of as an extension of their technique to the focal length modality. We show below how our proposed method is inspired, yet different from theirs, in the subsequent sections.

2.3 Intrinsic Image Decomposition using Focal Stacks

Figure 2.2 shows block diagram of our proposed RGBF-IID method which can be divided into following three steps:

1. Input focal stack acquisition
2. Probability map estimation
3. Image factorization

We provide details on each of these steps in the three subsequent subsections below.

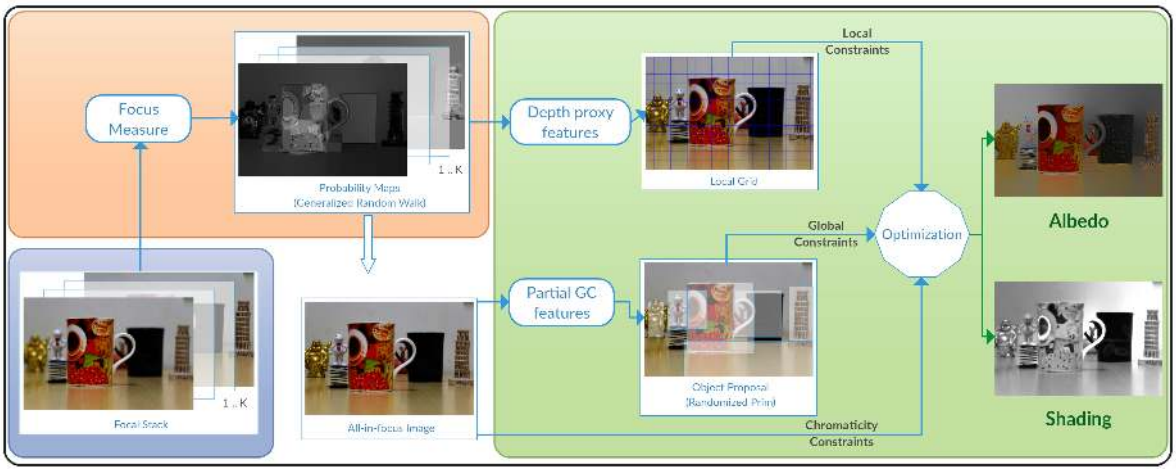


Figure 2.2: RGBF-IID Block diagram: Our framework comprises of three main steps: blue box indicates focal stack input acquisition Section 2.3.1, orange indicates probability map extraction using contrast as focus measure Section 2.3 and finally green indicates optimization for IID factorization Eq. (2.6) which consists of both feature extraction using both fixed overlapping grid and flexible bounding box patches.

2.3.1 Input Acquisition

Direct Capture: As described earlier, focus stacking is a technique where multiple images of the scene are captured with changes in focus distance between consequent shots. Focus stacking is becoming increasingly easy to use on modern day cameras. Several methods exist to automatically capture a focal stack given a rough range of depths in the scene. MagicLantern [2] is a camera firmware add-on for Canon DSLR cameras which enables several image stacking methods directly on these cameras. It allows for discrete control of the amount of focus-ring movement between shots and is a very useful tool for focus stacking. Focus stacking is also possible on mobile phones using either region based focusing such as by Sakurikar and Narayanan [236] or using direct focus distance control of the official Android Camera2 API [1]. We use MagicLantern to capture the real world focal stacks shown in our experiments. The focus-ring movement between consequent images is tuned to capture the entire scene in an exhaustive and exclusive manner *i.e.* each pixel is in-focus in one and only one focal slice. We also show results on an automatically captured focal stack using the Android Camera2 API on a Nexus 5X device.

Synthetic Focal Stacking: For evaluation purposes, we also experiment with synthetic focal stacks created over the MPI Sintel [46] and the NYUv2 [210] datasets. To this end, we use the depth maps available with each RGB image and first cluster the depth regions present in the scene. Then we synthesize a focal stack by setting the focus distance to each of the cluster centers in a sequential

manner. Since the associated depth data is available, we use standard lens optics to derive the amount of defocus for each pixel in each focal slice based on the focus distance for the slice. We apply spatially varying defocus kernels in a manner similar to Barron et al. [27] to synthesize these focal slices.

In order to synthesize a focal stack over a scene where the depth information for each pixel is available, we need to first estimate the correct focus distances for each of the slices in the focal stack. We do this by clustering the pixels in the depth space. We use the k-means clustering algorithm over the depth map to label pixels into depth clusters. The mean of each cluster refers to the mean depth value of the cluster and is thus used as the object-side focus distance for that focal slice. Using this known focus distance, we can compute the amount of defocus for all the other pixels in the scene based on their depth values. As shown in Fig. 2.3, the defocus radius for pixels focused at sensor

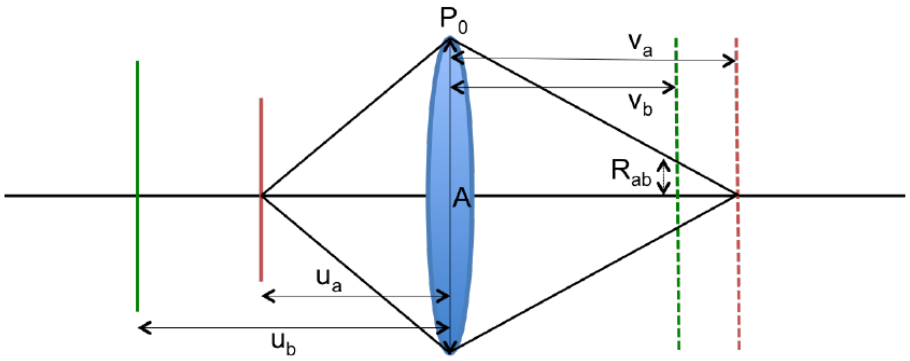


Figure 2.3: Synthetic focal stacking using depth maps: Defocus radius for scene objects at v_a when sensor is placed at u_b can be derived using basic optics and geometry.

location v_a when the sensor is moved to v_b is given by:

$$R_{ab} = \frac{A}{2}\frac{|v_a - v_b|}{v_a}, \tag{2.2}$$

where A is a the aperture of the camera, and F is the focal length. The values for A and F are assumed independent of the scene. Using the thin-lens equation governing u, v and F, Eq. (2.2) is re-written as:

$$R_{ab} = \frac{A}{2}\left|1 - \frac{u_b(u_a - F)}{u_a(u_b - F)}\right| \tag{2.3}$$

Since the depth data is known, the u values are known for all the pixels in the scene. Thus, we iterate over the previously defined cluster centers and fix the focus distance to be the mean depth value of the respective cluster center. For each focal slice, u_b is derived from the cluster center, as u_b is the distance where the sensor is currently placed (Fig. 2.3). For all the other pixels having a depth u_a, we find the defocus radius using Eq. (2.3) and apply a spatially invariant blurring operation as suggested by Barron et al. [27].

2.3.2 Focus Probability Maps

We compute the all-in-focus image from our focal stack F by adapting the method of Shen et al. [255] which uses generalized random walks for fusion of multiexposure images. They pose image fusion as a quadratic energy minimization problem and solve it by reformulating it as a Dirichlet equation for each label. They propose a probabilistic model by assigning color and contrast based node potentials for the image-label graph. We use similar formulation and adapt their framework for multi-focus fusion.

Consider graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \mathcal{L} \cup \mathcal{X}$ and $\mathcal{E} = \mathcal{V} \times \mathcal{V}$. Here $\mathcal{L} = \{l_1, l_2, ..l_K\}$ is the set of all labels with l_j referring to focal slice F_j. $\mathcal{X}$ is the set of variables x_i in the final fused image F^*. The fused image is composed as:

$$F^*(x_i) = F_j(x_i) \mid j = \underset{k}{\operatorname{argmax}}(P^k(x_i)), \tag{2.4}$$

where $P^k(x_i)$ is the k^{th} label assignment probability for x_i. Nodes $x_i \in \mathcal{X}$ form a 4-connected grid with each node sharing an edge with all nodes in $\mathcal{L}$. Also $\mathcal{E} = \mathcal{E}_{\mathcal{X}} \cup \mathcal{E}_{\mathcal{L}}$ where $\mathcal{E}_{\mathcal{X}}$ denote edges between grid nodes and $\mathcal{E}_{\mathcal{L}}$ denotes edges between grid and label nodes. We define compatibility functions y for $\mathcal{E}_{\mathcal{L}}$ edges and w for $\mathcal{E}_{\mathcal{X}}$ edges. This graph framework is similar to the one frequently used in systems which are posed as a Markov Random Field (MRF) [143] which can also be used as an alternative.

$P^k(x_i)$ is proportional to the node potential of x_i which is obtained by minimizing the energy associated with graph network by solving K Dirichlet problems. We use parameter values and definitions as given in [255]. The main difference is that we define $\mathcal{E}_{\mathcal{L}}$ edge compatibility functions as

$$y_k = \theta_k \left[erf(\frac{f^k}{\sigma}) \right]^K \tag{2.5}$$

where f^k is a per-pixel focus measure for k^{th} focal slice and is computed as the local variance in a 11×11 patch around each pixel. θ_k is the frequency of f^k and σ is standard deviation of f^k in k^{th} focal slice. We define pairwise functions w_k in a manner equivalent to [255] and solve the Dirichlet equations using generalized random walks.

We then concatenate the obtained probability maps P^k with normalized image coordinate values $\bar{x} = \frac{x}{m}, \bar{y} = \frac{y}{n}$ where m, n are image height and width respectively. The probability maps encode per-pixel focus variations across the stack and the normalized co-ordinates encode spatial locality. Thus our concatenated feature d_{xy} efficiently captures local neighborhood information required by the IID framework. The major advantage of using GRW instead of MRF here is that this fusion method gives continuous probability maps for each pixel across the stack unlike discrete labels in MRF. We show sample probability maps and the in-focus image in Fig. 2.4.

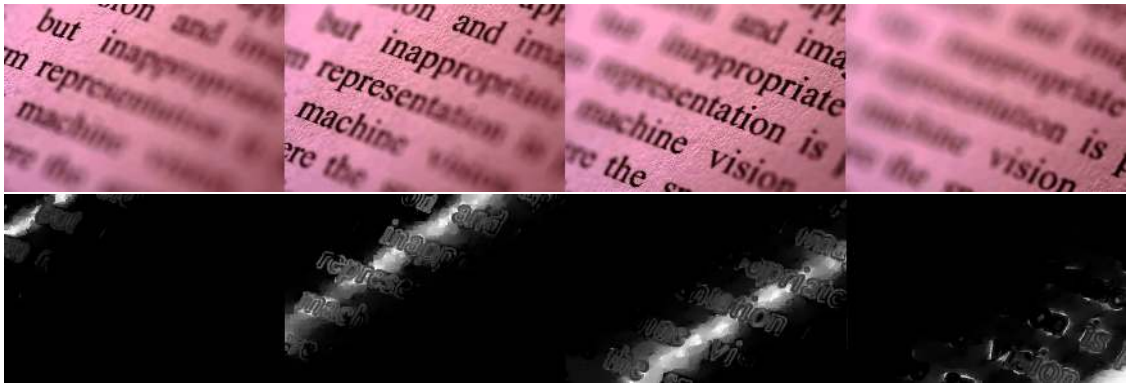


Figure 2.4: Probability Maps Visualization: Top row shows a few images from a sample focal stack with associated probability maps in the bottom row.

2.3.3 Image Factorization Optimization

We derive inspiration from depth based IID methods [60, 129] for our IID subsystem. These methods use depth and normals obtained from depth to define local and global point neighbourhoods which are then used as constraints in the optimization equation. We use our concatenated proxy-depth features defined in the previous section, to capture neighbourhood information. We build upon the framework of [129] which shows an improvement over [60] and is based on the separation of texture from the image. We briefly discuss the full system below and highlight the relevant extensions and modifications.

Let $I = B \cdot T$ where T is the texture and B is the textureless base image. Where B can be decomposed further as $B = R_B \cdot S_B$. Estimating the reflectance R_B and the shading S_B involves the minimization of the following energy function

$$f(S_B, R_B) = f^F(S_B) + \lambda^P f^P(S_B) + \lambda^R f^R(R_B), \tag{2.6}$$

where right hand side terms refer to proxy-depth based energy term, propagation term and reflectance term respectively. The λ parameters refer to the relative weights between the terms. The first term $f^F(S_B)$ refers to shading constraints such that

$$f^F(S_B) = f_l^F(S_B) + \lambda_g^F f_g^F(S_B), \tag{2.7}$$

where $f_l^F(S_B)$ is for local and $f_g^F(S_B)$ is for global shading consistency constraints. In [129], both these terms are reinterpreted using Local Linear Embedding [232] to define local $\mathcal{N}_l$ and global $\mathcal{N}_g$ neighbourhoods using depth based features. Similar to their approach, for RGBF-IID, we define $\mathcal{N}_l$ by computing κ nearest neighbours in LLE space using concatenated d_{xy} features. The underlying intuition is that scene points with similar variation in focus across the stack have a higher probability

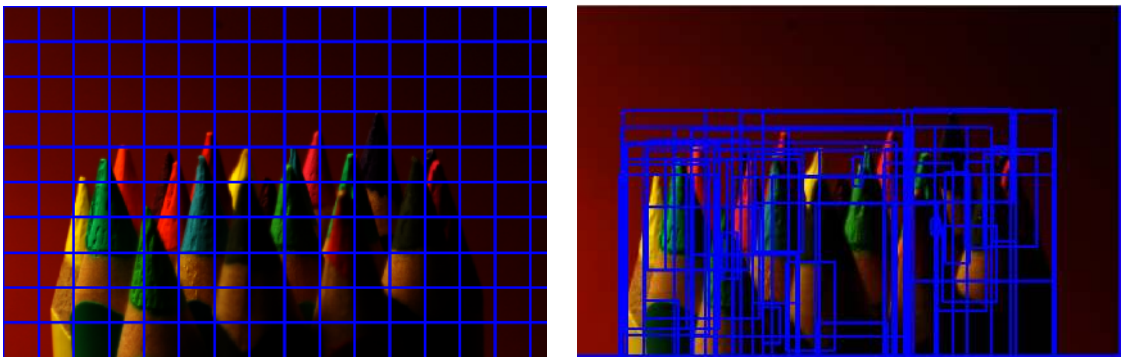


Figure 2.5: Local and global feature extraction: On left, non overlapping rigid grids for local constraints computation. On right, overlapping flexible sized grids for global constraints estimation.

of lying in the same depth range. Hence, combined probabilities derived from the focus measure and normalized image coordinates will encode local structural properties. For ease of computation we divide the image into rigid 11×11 grids and compute local linear embedding between these grids by choosing the feature with has minimum variance as a representative feature for that entire grid. The global shading term $f_g^F(S_B)$ aims to capture global consistency in spite of occlusion and color changes as an object can have multiple colors but shading component should be based on its shape and not texture or color. To enforce global consistency, we capture scene and object semantics by using partial Geometric Context (GC) features [121]. We use color and texture terms from GC features which capture statistics at smaller super-pixel scale. In order to use these features for global semantics we collect prime object proposals based on the method of Manen et al. [194]. This takes image super-pixel statistics into account and gives window proposals which have higher probability of containing an object. We use these proposals and compute candidate GC features per window based on minimum variance. We use these candidate features to compute local linear embedding based global constraints. This is illustrated in Fig. 2.5 which shows the two window types and sparsity pattern of the embedding weight matrix used for local and global shading constraints.

The second term $f^P(S_B)$ in Eq. (2.6) is used to propagate sparse grid based information computed in the previous step to rest of the pixels. This term comprises of two smoothing factors:

$$f^P(S_B) = f_{lap}^P(S_B) + \lambda_F^P f_F^P(S_B), \tag{2.8}$$

where $f_{lap}^P(S_B)$ is the matting Laplacian [162]. In our approach, we define $f_F^P(S_B)$ as a smoothing term with the local propagation weights derived from the concatenated d_{xy} features rather than surface normals as in [129]. The weights are defined as:

$$w_{pq}^F = \exp\left(-\frac{1 - \langle d_{xy}(p), d_{xy}(q)\rangle^2}{\sigma_n^2}\right). \tag{2.9}$$

Method	MSE			LMSE			DSSIM		
	reflectance	shading	average	reflectance	shading	average	reflectance	shading	average
RGBF-IID (Ours)	0.0298	0.0679	0.0488	0.0191	0.0390	0.0291	0.2771	0.2762	0.2766
RGBD-IID-Jeon [129]	0.0263	0.0527	0.0395	0.0151	0.0311	0.0231	0.2513	0.2642	0.2577
RGBD-IID-Chen [60]	0.0307	0.0277	0.0292	0.0185	0.0190	0.0188	0.1960	0.1650	0.1810

Table 2.1: **Quantitative results:** Considering the absence of any previously known focus based IID method, we show comparison of our results against two similar depth based IID solutions by Jeon et al. [129] and Chen and Koltun [60]. Note that even in the absence of complete depth data with only a few focal stacks images (<20 per scene), our RGBF-IID method is able to give comparable performance to the two similar depth based IID solutions.

The third term corresponding to the reflectance constraint and is defined as:

$$f^R(R_B) = \sum_{\mathcal{P}} \sum_{\mathcal{N}_p} w^R_{pq}(R_B(p) - R_B(q))^2, \tag{2.10}$$

over the set of all image pixels $\mathcal{P}$ and current pixel neighbourhood $\mathcal{N}_p$. Weights are computed based on the angular dissimilarity in the chromaticity between the pixels. Here w^R_{pq} is the angular distance based inter pixel chromaticity difference vectors.

$$w^R_{pq} = \exp\left(-\frac{1 - \langle C(p), C(q)\rangle^2}{\sigma_c^2}\right)\left(1 + \exp\left(-\frac{B(p)^2 + B(q)^2}{\sigma_i^2}\right)\right). \tag{2.11}$$

Here $C(p)$ represents normalized RGB value for three channels, $B(p)$ is the actual RGB value and σ's are free parameters which are preset by the user.

2.4 Experiments and Results

In this section we discuss the three datasets used and the experiments performed for evaluation of our technique. Most of the datasets in this problem domain consist of well lit simple objects [103] and do not capture real world scene complexity and variety. Owing to the lack of any available dataset for IID using focal stacks we manually collected some data for a few scenes. Apart from this we also show qualitative results using two standard depth datasets: NYUv2 [210] which consists of natural images captured in sync with the corresponding depth data from a Kinect sensor and MPI Sintel dataset [46] which consists of several video clips from a computer generated movie along with corresponding depth information.

As this is the first usage of focal stack modality for the IID problem, we compare our results against two similar depth based IID frameworks of [129] and [60]. We show that our focal stack based IID framework achieves comparable results with these two methods even in the absence of complete depth information. For the experiment with the focal stack we show only our results as the other two methods can not be executed without depth information. We show qualitative comparisons for the other two experiments. For quantitative comparison, we use synthetic focal stacks for our method and the provided depth data for the other two methods. We report results using three standard IID metrics: Mean Squared Error (MSE) and Local Mean Squared Error (LMSE) [102] and Dissimilarity Structural Similarity Index Metric (DSSIM) as defined in [60].

2.4.1 Results

Here we discuss our results on multiple datasets. There are no direct comparisons possible considering that our proposed technique is first of its kind using focal stacks for IID. Still for a quantitative comparison we evaluate our method against similar depth based IID solutions. Furthermore, we show results on both natural images by acquiring focal stacks using a Canon DSLR camera ourselves and by generating synthetic focal stacks using available depth maps from standard natural image dataset *i.e*. NYUv2 depth dataset [210]. Additionally, we also compare using completely computer generated images and depth data from the MPI Sintel benchmark [46]. Both qualitative and qualitative results are presented for each of the three datasets and discussed in detail below:

Focal Stack Dataset: We run our first set of experiments on the manually collected focal stacks of the real word scenes captured as discussed in Section 2.3.1. All these images are of varying resolutions and vary in the number of the stack images captured (minimum 10 and maximum 25 images). Also there is no restriction on the type of scene (indoor and outdoor) or the illumination condition (artificial or natural). We captured approximately 30 focal stacks and we show results on a few of those images in Fig. 2.7. Notice in 'classroom' scene how the well lit parts of the scene especially the specular regions separates into shading component. Notice the highlights formed due to indoor lights, natural lighting near the window, reflections on the floor and on the desk. Contrary to this, the reflectance component captures scene albedo as smooth color values in the remaining areas. The reflectance image for 'desk' scene captures the color of 'pisa tower' and 'buddha' figurines correctly after removing the dark regions and the lighting reflections respectively in these two cases. In 'leaves' even the dark regions under shadow have suitable green reflectance. For the 'bench' scene uniform color of the grass blades and background bushes can be seen in the reflectance image. In the 'outdoor' scene the shadow of the tree is absent in the reflectance image and is captured in shading. One possible failure case is noticeable here as the white flowers on the bush get captured in the shading image owing to being devoid of color and being smaller in size to get accounted in the local or global patches properly. This also

leads to bush being brightly lit while the corresponding color in the reflectance image becomes darker shade of green. We believe this happens because the shading constraints were sparsely computed. In the 'flower' image the color of leaves shadowed by the flower petals is visible in the reflectance image.

NYU dataset: For comparison with the RGBD based IID methods we show results on the artificially generated focal stacks from the test-split of NYUv2 dataset [210] which has 654 RGB images with 480×640 resolution and depth-maps captured using Kinect sensor. We use the depth only to generate the focal stacks for input in our framework as mentioned above Section 2.3.1. For Jeon et al. [129] (RGBD-IID-Jeon) and Chen and Koltun [60] (RGBD-IID-Chen) we use the entire data with depth and obtain results using the official code provided by authors with the default parameters. Chen and Koltun [60] divide image into four components namely, albedo, direct irradiance, indirect irradiance and illumination color. For comparison we combine the direct, indirect irradiance and illumination color channels to generate the shading and we combine albedo and illumination color to generate reflectance image. As can be observed from Table 2.1, our results are comparable to both the methods without using the dense depth values. This shows that our depth proxy features are strong enough to capture the scene structure information using only a focus measure. Corresponding qualitative results are shown in Fig. 2.8.

MPI Sintel dataset We also show results on synthetic focal stacks from the Sintel dataset [46]. It is composed of 24 single camera views from a naturalistic open source animated movie. In this 50 frames for each scene are provided with ground truth depth and albedo in both 'clean' pass without particle effects added and a 'final' pass with particles. We use the 'clean' pass to construct our focal stacks and show the output in Fig. 2.9. As reported by [129], MPI Sintel is not created for IID benckmarking and has several issues but in the absence of any other focal stack based IID benchmark dataset, we report quantitative results using this dataset only for a rough comparison. For [129] we run the code provided by the authors on their official project homepage with default parameters. Following Jeon et al. [129], the results are obtained on a every fifth frame of MPI Sintel dataset (89 images) and the final subset is selected for which both the methods converged. For Chen and Koltun [60], we directly report the numbers as mentioned in their main paper. We can see that our method achieves comparable performance against [129] even though it sees substantially low input information compared to them as only a few focal stacks images are fed (approximately 20 per scene) to the system instead of a complete dense depth map. For [60] their numbers are reported on a larger test subset and has additional model complexity as multiple decompositions going on with a larger objective function and mathematical framework.

2.5 Conclusion

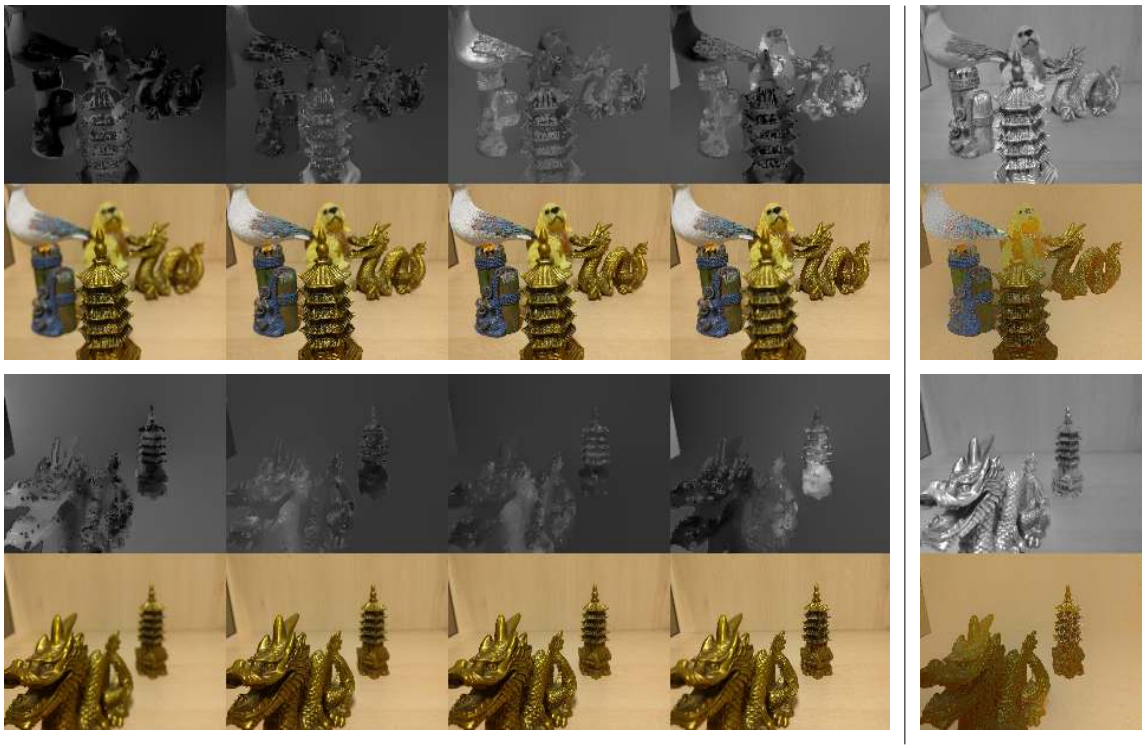


Figure 2.6: Qualitative results on a mobile device: First row for each scene: Sample focal slices from a cellphone device. Bottom row: Corresponding probability maps. Right column: Shading on top and Reflectance in bottom from our RGBF-IID method.

In summary, we present a novel method (RGBF-IID) to obtain intrinsic images of a given scene by capturing a set of images at varying focal distances. Compared to RGBD based methods focal stacks are easy to capture and can work on outdoor scenes also. This allows our system to be easily used without any restrictions on scene type. Also in our intrinsic decomposition formulation we capture scene semantics by generating novel object localization based global constraints. Additionally we also provide support to the hypothesis that focus based RGBF modality can be used in place of RGBD for some computer vision applications. In future we would like to explore this idea further and test it on other related problems.

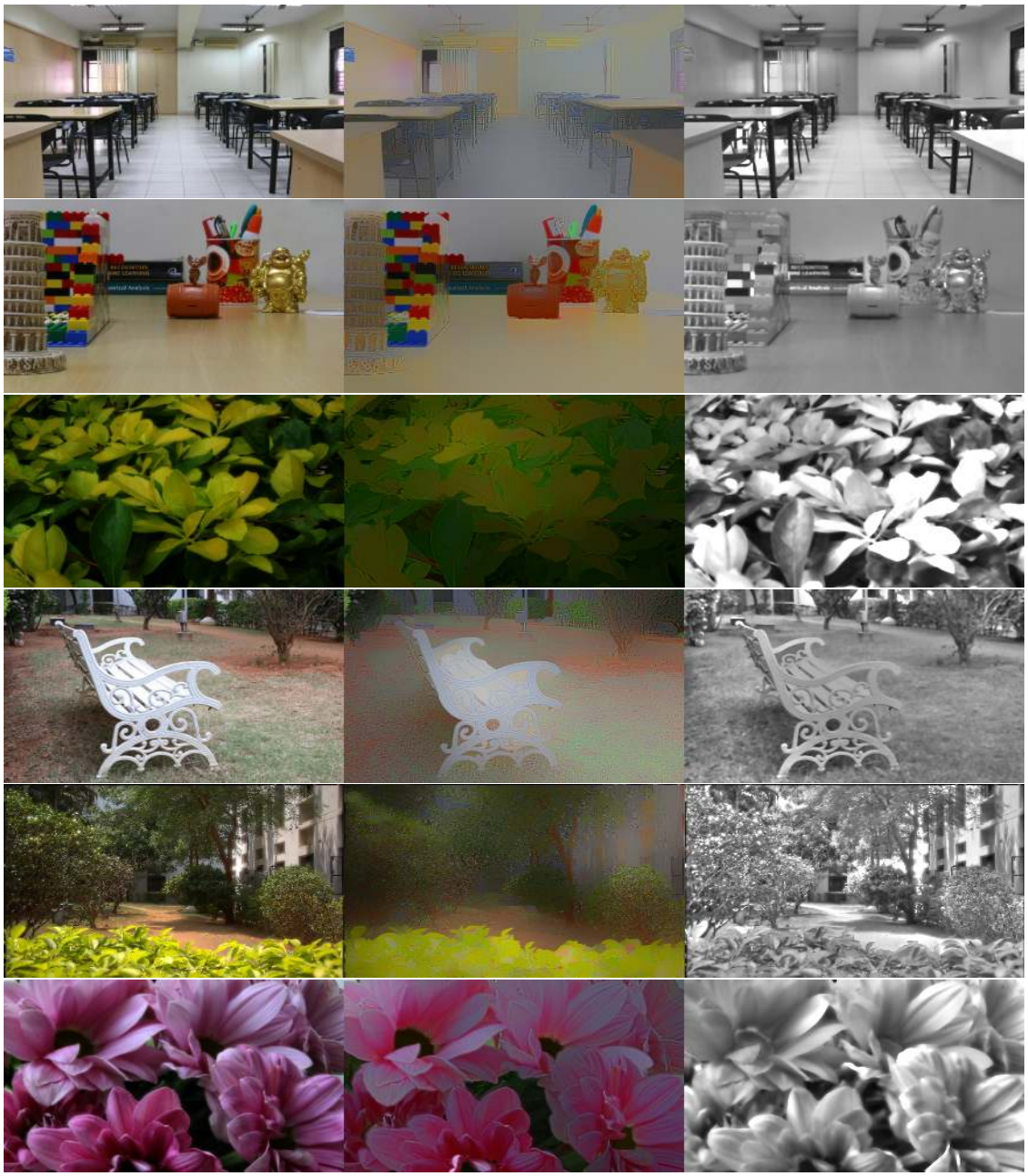


Figure 2.7: Results on real images: First row is the computed all-in-focus image followed by reflectance and then shading component obtained by our RGBF-IID method. Scenes names (top to bottom): 'classroom', 'desk', 'leaves', 'bench', 'outdoor', 'flowers'. The number of focal stack images used per scene respectively are: 12, 10, 11, 9, 11 and 15. Note how all highlights and shadows are part of the shading component while color is highlighted in the reflectance component even in the dark regions like shadows.

Figure 2.8: Qualitative results on NYUv2 dataset [210]: Our RGBF-IID results on NYU synthetic focal stacks. For each scene top row from left-to-right is original image, RGBF-IID reflectance, RGBD-IID-Jeon [129] reflectance and RGBD-IID-Chen [60] reflectance. Bottom row is depth (blue closer, red farther) and corresponding shading results. Note how our shading results (second column) are able to capture scene highlights and shadows well without significant reflectance gradations (third column) or over-smoothing (fourth column).

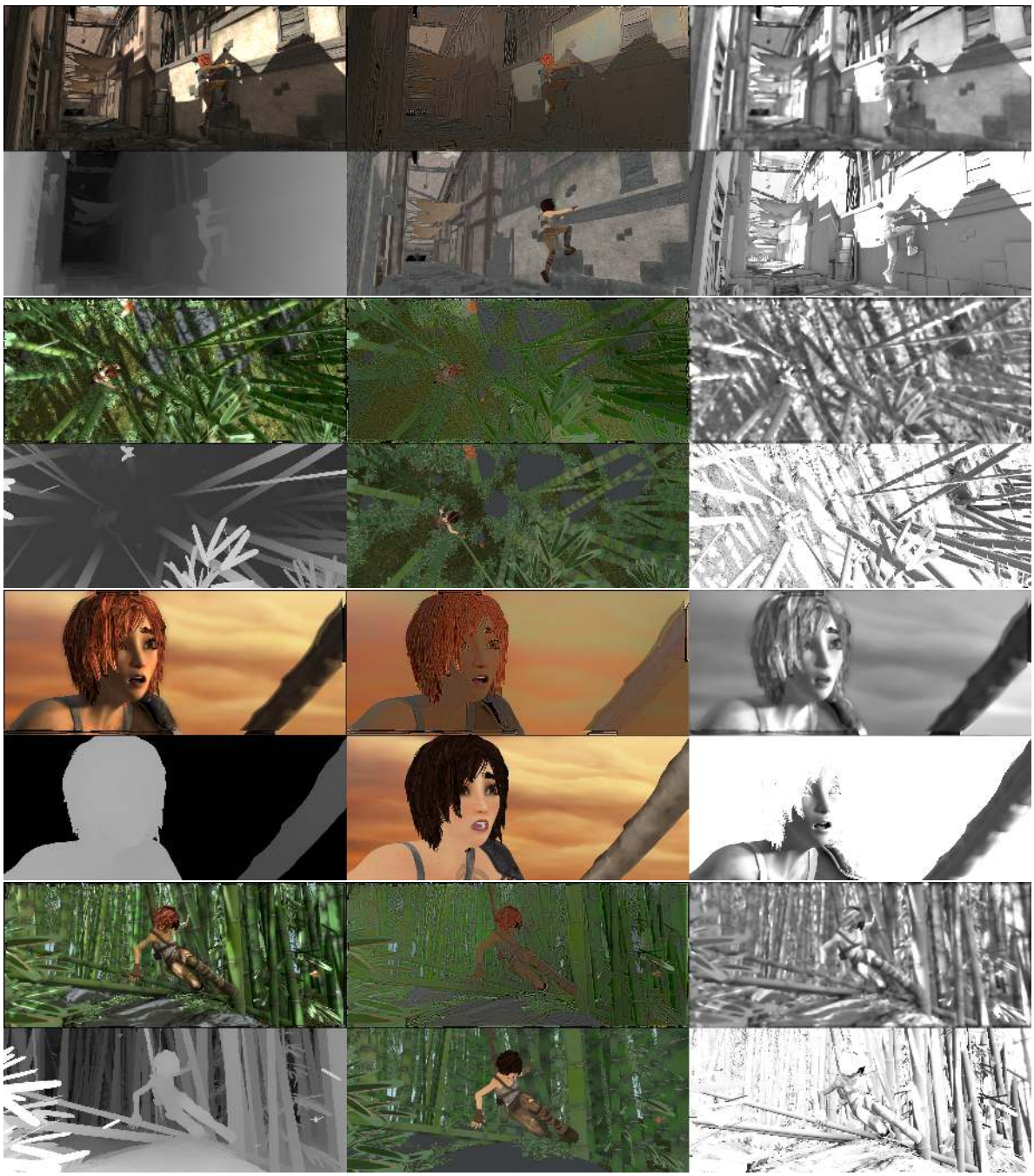


Figure 2.9: Qualitative results on MPI Sintel dataset [46]: For each scene the all-in-focus image, RGBF-IID reflectance and shading are shown in the first row, while ground truth depth map, reflectance and shading are shown in the second row. Note that how all the illumination highlights and shadows are extracted as part of the shading component while the reflectance component mostly consists of the color information. Note that apart from the global scalar factor multiplicative difference, the relative order of illumination is consistent in shading.

Chapter 3

Intrinsic Image Analysis with Single Image Input

Intrinsic Image Decomposition, in the context of single image input, is significantly more challenging but practically more useful compared to the scenario with provided auxiliary scene information like discussed in Chapter 2. In order to answer the issue of under-constraining of IID formulation, most of the single image IID methods pose the task in a simplified and restricted setting. Additionally, they might also take advantage of already established facts about the problem and encode such domain-expertise in the form of prior knowledge in the system. Naturally, this introduces *prior* likelihoods and other approximation assumptions in the framework. These factors are posed as mathematical constraints or *cost terms* in the optimization equation. Yet another way of introducing such priors is by using direct data-driven supervision but it is not a simple undertaking considering paucity of well labeled datasets in this domain. In this chapter we focus on the analytical formulation of the single image IID task and present our energy minimization based iterative optimization algorithm comprising of carefully constructed cost terms based on hierarchical IID priors. Specifically, we present novel semantic priors and an integrated approach for single image IID that involves analyzing image at three hierarchical context levels. *Local* context priors capture scene properties at each pixel within a small neighbourhood. *Mid-level* context priors encode object level semantics. *Global* context priors establish correspondences at the scene level. Our semantic priors are designed on both fixed and flexible regions, using selective search method and Convolutional Neural Network features. Our IID method is an iterative multistage optimization scheme and consists of two complementary formulations: L_2 smoothing for shading and L_1 sparsity for reflectance. We optimize our formulations using Split-Bregman iterations in a single integrated algorithm and generate competitive results with lesser artifacts. Experiments and analysis of our method indicate the utility of our semantic priors and structured hierarchical analysis in an IID framework. Finally, we highlight that proper choice and encoding of prior knowledge can produce competitive results even when compared to end-to-end deep learning IID methods which themselves can benefit from the insights and techniques presented in this chapter and hence will be useful in the future IID research.

3.1 Introduction

Before jumping into our proposed solution, we comprehensively discuss and reiterate some of the points made earlier for the IID problem. Humans are good at visual understanding of several aspects of a scene. We can detect and recognize various objects, do semantic associations and guess structural properties in a scene. We also have the capacity to make inferences about the object-dependent and the scene lighting-dependent visual effects like highlights, soft-shadows, light caustics *etc*. Intrinsic Image Decomposition (IID) as a research problem is motivated by this observation enables the computers to distinguish light-based and object property-based image effects. This helps in improving scene understanding and image rendering research and hence would be useful from both computer vision and computer graphics perspective.

As mentioned previously, IID is a classic problem first proposed by [155] and studied by both computer vision and graphics research communities and can be categorized under the broad *inverse rendering* field of research Marschner [195], Ramamoorthi and Hanrahan [223]. Restating, in IID we split a given image (I) into two underlying components:

$$I = R \cdot S, \tag{3.1}$$

where R (reflectance) captures the object dependent properties like colour, textures, *etc*. and S (shading) represents direct and indirect lighting in the scene. These components could further be reorganized into still smaller factors by using more detailed image formation models which take into consideration optical effects like specular-Lambertian lighting, sub-surface scattering, material reflectivity, translucency, volumetric scattering *etc*. Though such complex image decompositions might be needed in specific scenarios, a simple object-lighting dichotomy based definition of IID enables many interesting applications in computer vision and image editing applications like image colourization [181], shadow removal [148], re-texturing [69], scene relighting [70], *etc*.

IID is an ill-defined and under-constrained problem [33]. It is ill-defined as in the presence of general real world complex lighting and material reflective properties, final appearance of an object in an image cannot cleanly be separated by using reflectance and shading components only. IID problem is under-constrained as we have to estimate two variables per pixel from a single intensity value from the given image. Moreover, IID solutions are inherently ambiguous as there can be multiple valid reflectance and shading decompositions differing by a positive scalar multiplicative factor [39] as:

$$I = \alpha_i R \cdot \frac{S}{\alpha_i}, \qquad where\ \{\alpha_i \in \mathrm{Re} \mid \alpha_i > 0\}. \tag{3.2}$$

All these issues make IID a challenging and interesting research problem.

Previous IID solutions can be categorized under two classes:

1. Solutions which assume auxiliary input data from the scene in the form of depth, user annotations, optical flow, multi-view, multiple illuminations, photo collections, *etc*.

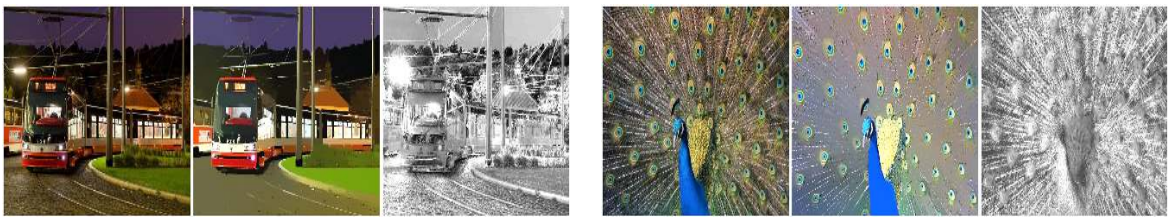


Figure 3.1: Single Image Intrinsic Image Decomposition (IID): IID decomposes a given image (I) into intrinsic reflectance (R) and shading (S) components such that $I = R \cdot S$ with R containing object colour properties and S capturing scene lighting information. From left to right: I, R and S for two images. Notice the colour consistency in the reflectance and separation of lighting and shadows into the shading.

2. Solutions which work directly on single images and are more dependent on priors and scene assumptions.

We have already discussed the first category in the previous chapter (Chapter 2 and here we focus on the second type of solutions. We utilize weak semantic information from the scene for building novel priors for IID. This is inspired from the observation that scene semantics, even if weak, give us an idea about the underlying scene structure and the object level association between various image pixels. We harness this information to establish constraints between various pixels to tackle the under-constrained nature of the IID problem. We present two simple techniques for weak semantic feature extraction computed on both flexible (segmentation masks) and fixed (overlapping patches) splitting of image regions. We use these features to build priors at three hierarchical contextual scales in our model. In summary, three main contributions of this chapter are:

- We introduce a technique for capturing weak scene semantic information for both fixed and flexible region definitions using CNN and selective search features for IID.
- We analyze scene at three context levels: *local context* where optimization weights are based on a small pixel neighbourhood; *mid-level context* which tries to capture object level semantics and *global context* where various regions of the image are linked based on their shared characteristics at the scene level.
- We present a new iterative integrated IID framework based on Split-Bregman iterations [97] using two competing formulations and generate results with fewer artifacts.

We perform experiments to analyze the effect of our semantic priors at various context levels and illustrate the decompositions generated by our competing formulations over successive iterations. Finally, we present improved qualitative and competitive quantitative results with respect to the contemporary IID methods on challenging IIW dataset [33] and 'wild' images from the Internet. We believe

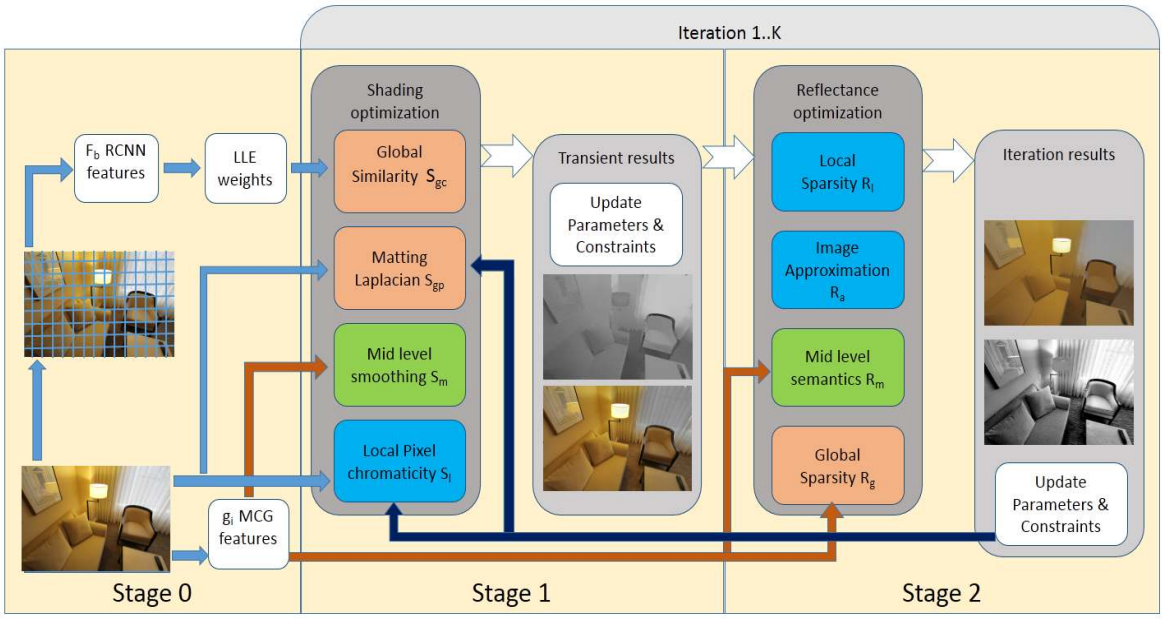


Figure 3.2: SP-IID Block Diagram: Our method (Semantic Priors IID) can be understood in three stages. After semantic features extraction (Stage 0), in each iteration our method alternates between the L_2 shading (Stage 1) and L_1 reflectance optimization (Stage 2) with energy terms computed for both the formulations at three hierarchical context levels: local, mid-level and global. Finally after the convergence of the optimization iterations, reflectance and shading images are generated as output.

this is the first IID solution with explicitly encoded semantic priors. The major takeaway message of this work is that meaningful priors are very useful to solve an ill-posed problem like IID. Supervised methods employing end-to-end deep learning have recently shown promise on the IID problem, as datasets and loss functions have started to improve. We believe rich and meaningful priors, based on scene semantics or other image properties, will have strong roles on both supervised and unsupervised systems of the future.

3.2 Background

In this sections we discuss the two IID solution categories relevant in this context and other associated topics like available IID datasets, evaluation metrics and learning paradigms.

Auxiliary Data IID: Several IID methods depend on auxiliary scene data in various forms. Jeon et al. [129], Chen and Koltun [60] and Barron and Malik [25] take an RGBD image input and use depth to establish structural correspondences between image pixels. Bousseau et al. [40] require user

annotations in the form of scribbles marking constant reflectance regions, as auxiliary information. For videos, Kong et al. [146] use optical flow and enforce temporal reflectance consistency constraint between frames. Similarly, Laffont et al. [151] use multiple views and enforce spatial reflectance consistency by identifying corresponding scene points across images. This idea is also employed by Weiss [284] for reflectance consistency between multiple illumination images. Along similar line, our previous work Saini et al. [235] use focal stacks as auxiliary information by substituting it for depth data. Laffont et al. [150] and Liu et al. [181] use diverse photo collections to establish correspondences between image regions to build constraints. The common idea behind these methods is to approximate textural and shape similarities using the auxiliary information. The necessity to acquire additional input data is a major drawback of all such methods.

Single Image IID: A second category of IID methods work directly on single images. These methods require several assumptions and priors, as it is hard to gather sufficient information about geometry, material property and illumination of the scene from a single image. Many such methods work on simple images containing a single object with no background like Barron [23], Barron and Malik [26, 24]. Other methods which work on natural scenes utilize priors like *Retinex* constraints [155], reflectance sparsity [91, 253], long *vs*. short tailed gradient distribution separation [168], spatio-chromatic clustering [89], *etc*. These methods encode interesting insights for the IID problem but are limited when generalizing to 'wild' cases with varying lighting and complex textures. Results vary based on how much significance is given to a prior and the type of optimization framework. Moreover some of these priors have adversely competing goals. Smoothness prior on shading removes texture details from S as opposed to reflectance sparsity assumption which simplifies colour details in R. Recent methods try to solve this issue by sequentially employing two separate optimizations for shading and reflectance estimation [33, 321, 36].

Based on these insights, our algorithm combines these two types of optimizations in a single integrated algorithm by alternating between the two competing formulations: smoothness for shading and sparsity for reflectance. We use L_2 cost terms based optimization for shading and L_1 cost terms for reflectance. We alternate between these two formulations and adapt Split-Bregman iterations [97] for achieving the final decomposition.

Datasets: A major challenge associated with IID research is lack of diverse large datasets and proper evaluation metrics [39]. This arises mainly due to subjective nature of the problem and difficulty in collecting dense annotations. MIT intrinsic images dataset introduced by Grosse et al. [103] is limited to a handful of single object images on a black background. As-Realistic-As-Possible dataset by Bonneel et al. [39] tries to capture complexity of natural scenes but is also not large enough for supervised training. Synthetic datasets like MPI Sintel by Butler et al. [46], provide dense annotation but lack sufficient diversity and complexity compared to the natural scenes. Bell et al. [33] provide a large

manually annotated dataset called Intrinsic Images in the Wild (IIW) but have only sparse relative annotations. This limits the utility of such datasets in learning based approaches which aim to work on complete scenes under unrestricted illumination and material property settings.

Evaluation Metric: Yet another challenge in IID research is lack of proper evaluation metric which reflects both quantitative and qualitative performance. *Local Mean Square Error* (LMSE) and *Structural Similarity Index Metric* (SSIM) are used for synthetic scenes [60, 129] but these metrics require dense ground truth annotations. Bell et al. [33] suggest a new performance evaluation metric based on their IIW dataset: *Weighted Human Disagreement Rate* (WHDR). WHDR gives relative error rate within a given threshold based on the sparse annotations in the IIW dataset. Hence it is a dataset specific metric. Lack of a proper evaluation metric which could properly evaluate results both perceptually and objectively for all application scenarios, makes comparison between different IID solutions a difficult task.

Supervised *vs*. Unsupervised IID: Some IID methods use supervised learning to solve related subproblems using gradient classifiers [263], Bayesian graphical models [57] and deep neural networks [321, 208, 252, 142]. Zhou et al. [321] and Zoran et al. [325] learn Convolutional Neural Network (CNN) priors using sparse IIW annotations which they propagate to other pixels using a traditional dense Conditional Random Field (CRF) or flood fill the super-pixels in a separate post-processing system. Yet another approach is to use dense ground truth from synthetic scenes like from MPI Sintel dataset [46]. Such approaches either use the underlying depth information like Narihira et al. [208] or use previously proposed RGBD based IID solutions to generate ground truth for supervision like Kim et al. [142]. Synthetic datasets like Sintel do not represent true reflectance and shading of natural scenes as the dataset was not originally curated with the intention of IID benchmarking as pointed by Jeon et al. [129]. Recently, two new IID specific synthetic datasets introduced by Bi et al. [37] and Li and Snavely [172] have helped in training supervised learning frameworks like CNNs. Still the domain shift between synthetic *vs*. real world scenes has an effect on the performance of these methods. Due to the limited supervision data and significant domain shift, such end-to-end CNNs are prone to over-fitting and dataset bias [268, 138]. This observation was also highlighted by Nestmeyer and Gehler [212] who showed how a simple post processing using guided filters could improve results from several deep learning IID solutions, suggesting that such solutions are not able to capture the insights of the previously known IID priors effectively.

These issues concerning datasets and performance evaluation, along with it ill-defined nature, make IID a challenging problem to solve using supervised learning frameworks. On the other hand, several older IID methods were unsupervised in nature. Weiss [284], Gehler et al. [91] and Bousseau et al. [40] relied on intelligently designed priors and designed their method as optimization schemes. Such earlier models take advantage of prior knowledge, but either work in restricted settings based upon

the assumptions in the models or have scope for performance improvement compared to deep learning based schemes. CNNs have been widely used in computer vision and machine learning literature as black box feature extractor [249, 303, 66]. Donahue et al. [66] directly use pre-trained CNNs as a feature extractor and prove the generality and cross domain applicability of such features on varied tasks like scene recognition, fine-grained recognition and domain adaptation. Along similar lines, Sharif Razavian et al. [249] and Yosinski et al. [303] also use these features on increasingly different tasks and datasets, highlighting their task agnostic characteristics.

In our model, we absorb the advantages of both the supervised and unsupervised approaches by combining the generality of supervised deep learning methods with prior domain knowledge. We employ an 'off-the-shelf' pre-trained deep neural network as a black-box to obtain generic features. Additionally, we also use an unsupervised technique to provide yet another set of semantic features. We use both these features to introduce new context priors in an unsupervised optimization algorithm by posing it as a standard total variational optimization problem [97]. Recently, great improvements have been made in IID with the use of supervised methods primarily utilizing new large new training datasets such as: time lapse dataset [171], multi-illumination dataset [37] and synthetic datasets [172]. Bigger datasets improve the performance of supervised learning systems, but we can still improve these solutions by explicitly encoding problem specific insights. We believe our IID specific semantic priors, encode crucial domain knowledge and would further help such methods.

3.3 Semantic Hierarchical Priors for Intrinsic Image Decomposition

Our method is as an iterative algorithm alternating between shading and reflectance formulations (see Fig. 3.2). Optimizing for reflectance sparsity alone leads to loss of textures in reflectance while focusing on shading smoothness leads to non-sparse reflectance (see 3.7). We tackle this adversarial nature of the two formulations by estimating IID in two stages for shading smoothness and reflectance sparsity separately. Such an iterative scheme has earlier also been used by Bell et al. [33] and later adapted by Zhou et al. [321]. Our framework differs from theirs as we present a single integrated algorithm without requiring additional steps for building a dense CRF like Bell et al. [33] or separate additional optimization frameworks for post-processing like Zhou et al. [321]. We take inspiration from Bi et al. [36] who employed [97]'s Split-Bregman L_1-L_2 optimization method for image flattening and edge-preserving smoothing, and we adapt it to directly estimate IID. We show that this cleaner integrated approach leads to lesser artifacts in the results while maintaining good quantitative performance. We discuss these two formulations and the new priors used in our framework below.

Figure 3.3: Visualizing selective search features: From left-to-right: Original image; four sample mask images from Multi-scale Combinatorial Grouping [20] (binary masks overlaid on the image for visualization) and dimensionality reduced image of selective search features (g_i) used for encoding class agnostic semantic information. Notice how regions belonging to an object get grouped together representing approximate semantic information in the scene.

3.3.1 Semantic Features

Semantics could provide crucial object and scene information which could help the IID process. Based on on this intuition, we propose two simple techniques to represent semantic information for IID. Semantics in images could be either obtained using bounding box annotations or dense segmentation masks. Both of these problems are separate challenging computer vision research problems in themselves. Bounding boxes give us weak semantic information whereas dense segmentation mask extraction is a still harder computer vision task and results are often noisier and less accurate compared to the former option. In order to avoid solving either of these tasks directly, we use approximate semantics to build our IID features. Additionally, this also makes our features class and task agnostic, unlike object detection or segmentation frameworks, which are limited by the number of classes that are assumed during training. This significantly improves the generality of our framework. We extract two different kinds of features using two complimentary region definitions: fixed patches and flexible regions. We approximate semantic information over these region definitions using two separate techniques as explained below:

RCNN features (f_b): Using the fixed region definition, we divide the input image I into B patches using a fixed grid of a constant size. In order to extract features from these patches, we pass them through a Region-based Convolutional Neural Network (RCNN) by Girshick et al. [95]. We pre-train the RCNN on ImageNet dataset [65] and extract 4096 dimensional features f_b for each patch with $b \in \{1, 2, \dots B\}$ from the last fully connected layer ($fc7$) of the network. We assign this to the center pixel of the patch to obtain a sparse set of regional features for the image. Long et al. [185] show that such features, despite having weak label training over the entire scene and large receptive

fields, encode fine correspondences between regions similar to traditional structure encoding features like *SiftFlow* [178]. Hence these features could be used in tasks requiring precise localization like intra-class alignment and keypoints classification. Furthermore, as *SiftFlow* has earlier been used for estimating scene structural information by Karsch et al. [135], it presents a good case for applicability of RCNN features for designing IID semantic priors. Hence we use f_b to approximate shape similarity and estimate correspondences between image patches.

Selective search features (g_i): Complimentary to fixed patches, we also extract approximate semantic information from flexible region definitions using selective search techniques. Selective search or detection proposals give interesting image regions which have higher probability of containing an object. This improves object detection by avoiding sliding-window search. Hence selective search results could be used as an indicator of presence of an object ('objectness') in a given region (please see the survey paper by [123] for further information on selective search techniques). Selective search is simpler and faster compared to training a Conditional Random Field (CRF) for a finite number of classes for dense pixel associations [36, 321]. Furthermore selective search has off-the-shelf implementations available and does not require separate training. We use Multiscale Combinatorial Grouping (MCG) by [20] for capturing object semantics following the conclusions based on recall and detection quality from the survey by Hosang et al. [123]. MCG is a bottom up segmentation method based on fast normalized cuts which are then efficiently assembled into object proposal regions based on an efficient grouping strategy. MCG generates dense binary region masks and scores for each detection proposal $c \in \{1, 2, \ldots, P\}$ for a total of P proposals.

Our selective search features are formed by concatenating various mask values at a particular pixel, weighted by MCG [20] 'objectness' score. We form a concatenated feature vector g_i of proposal masks weighed by proposal score at each pixel and normalize it using L_2 norm. We do dimensionality reduction on these features using PCA for efficient computation during reflectance formulation. We use dimensionality reduced features in *Stage 2* of the framework unlike *Stage 1* as the mid-level priors are iteratively recomputed only in this stage. Figure 3.3 shows a few sample sample masks (overlaid over the image for visualization) and the 'PCA-image' (formed by reducing the dimensions to 3) for an input image. Note how in the regions belonging to the same object get clustered together illustrating how our selective search features, g_i encode mid-level scene semantics from the image.

3.3.2 Shading Formulation

Our shading formulation assumes monochromatic Lambertian illumination and piecewise constant reflectance and is inspired from Jeon et al. [129] like in the previous chapter Chapter 2 wherein authors use depth maps to define pixel neighbourhoods. We generalize their system for a single image by modifying the priors using RCNN and selective search features. The intermediate IID results as

shading (σ) and reflectance (ρ), are estimated by minimizing the following energy function:

$$\Psi = \lambda_g S_g + \lambda_m S_m + \lambda_l S_l. \tag{3.3}$$

Here S_g, S_m and S_l are respectively global, mid-level and local shading priors and λ_g, λ_m and λ_l are the corresponding weights.

Global context (S_g): Our global shading prior S_g is a combination of a sparse neighbourhood consistency term S_c and a weight propagation term S_p: $S_g = S_c + S_p$. In [129] authors show that under the assumption of Lambertian model, shading at a point can be approximated using a weighted linear combination of surface normals where the weights are computed using local linear embedding of these features in the neighbourhood $\mathcal{N}$. But unlike Jeon et al. [129] we do not have depth information and therefore we approximate structural similarity using our pre-computed RCNN features f_b as:

$$S_c = \sum_b \big(\sigma_b - \sum_{a \in \mathcal{N}_b} w^c_{ab} \sigma_a\big)^2. \tag{3.4}$$

Here $\mathcal{N}_b$ represents the set of 10-nearest neighbours for patch b computed using f_b features and w^c are linear combination weights computed using the local linear embedding representation of b over $\mathcal{N}_b$. These are sparse constraints as we assume the center pixel to be the representative of the entire patch and assign the constraint to it. In order to propagate these constraints to the rest of the pixels, we do structure-aware weight propagation using a Laplacian matting matrix [162]. This approximates shading by an affine function over a base image in a small local window ($\mathcal{N}_{3\times3}$). Our propagation term is defined as:

$$S_p = \sum_i \sum_{j \in \mathcal{N}_{3\times3}} w^p_{ij} \big(\sigma_i - \sigma_j\big)^2. \tag{3.5}$$

Here weights w^p are computed using the matting Laplacian with reflectance result of the previous iteration as the base image. For the initial iteration, the base image for the Laplacian is taken as Gaussian smoothened version of I. In their work, Bell et al. [33] propagate global constraints using a dense CRF whereas Zhou et al. [321] devised a Nyström approximation to integrate their proposed CNN reflectance prior for message passing during CRF inference. In comparison, Laplacian matting term has a closed form solution and is easy to compute than explicit CRF optimization.

Mid-level context (S_m): For mid-level prior we use selective search features g_i which encode object semantics. Similar to the weight propagation term S_p, we define this prior as:

$$S_m = \sum_i \sum_{j \in \mathcal{N}_{3\times3}} w^m_{ij} \big(\sigma_i - \sigma_j\big)^2, \tag{3.6}$$

where $w^m_{ij} = \exp\big(\text{-}\frac{(1-\langle g_i, g_j\rangle)^2}{t_m^2}\big)$ which penalizes dissimilar g_i and g_j. This captures the intuition that in a local neighbourhood if two pixels are predicted to belong to a common object proposal, then they

should have similar shading. This causes shading smoothness within each detection proposals and preserves texture in the reflectance component.

Local context (S_l): Local context prior is defined following the Retinex model (*i.e.* change in chromaticity implies change in reflectance). Similar to Jeon et al. [129], we use this prior in the logarithmic form and substitute $\log \rho = \log I - \log \sigma$ to obtain:

$$S_l = \sum_i \sum_{j \in \mathcal{N}_{3\times 3}} w_{ij}^l \big((\log p_i - \log \sigma_i) - (\log p_j - \log \sigma_j) \big)^2, \tag{3.7}$$

where $w_{ij}^l = \exp\left(\text{-}\frac{(1-\langle \overline{p_i}, \overline{p_j} \rangle)^2}{t_c^2}\right) \left[1 + \exp\left(\text{-}\frac{p_i^2 + p_j^2}{t_b^2}\right)\right]$. Here $\overline{p_i}$ is pixel chromaticity computed as normalized RGB vector. The first term in the product awards higher value to similarly coloured pixel pairs. The second term gives higher weight to pairs with very low intensity values. This reduces colour artifacts by suppressing chromatic noise in the dark regions. t_m, t_c and t_b are fixed deviation parameters for weight estimation. We solve this quadratic optimization problem ($\sigma^* = \text{argmin}_\sigma \, \Psi$) using gradient descent and set $\rho^* = I - \sigma^*$.

3.3.3 Reflectance Formulation

Unlike our shading formulation (Section 3.3.2) which enforces smoothness using L_2 terms, our reflectance formulation enforces colour sparsity using L_1 terms. The backbone of this stage is inspired from image flattening work by Bi et al. [36] which uses Split-Bregman method [97] for optimization. For IID, they use flattened image as input and perform a series of steps like self-adaptive clustering, Gaussian mixture modeling, boosted tree classification, CRF labeling and L_2 energy minimization. We show that we can use Split-Bregman iterations directly for IID by using proper context priors and alternating between shading and reflectance formulations. In addition to being a direct approach, our method is more robust to clustering artifacts (Fig. 3.6). Our reflectance formulation is given as:

$$\pi = \gamma_g R_g + \gamma_m R_m + \gamma_l R_l + \gamma_a R_a. \tag{3.8}$$

Here R_g, R_m, R_l and R_a are global, mid-level, local and image approximation terms respectively and γ_g, γ_m, γ_l and γ_a are the associated weights. We use a similar definition for local and global prior weights (v^l and v^g) and have a fixed deviation parameter (t):

$$v_{ij} = \exp\Big(- \frac{(\overline{r_i} - \overline{r_j})^2}{2t^2} \Big). \tag{3.9}$$

Here $\overline{r_i}$ is channel normalized CIELab colour value with a suppressed luminance [36]. Note that unlike Bi et al. [36], we re-estimate priors in each iteration which gradually leads to IID directly instead of image flattening.

Local context (R_l): We define local reflectance energy term by enforcing the piece-wise local image sparsity similar to Bi et al. [36]:

$$R_l = \sum_i \sum_{j \in \mathcal{N}_{11\times 11}} v_{ij}^l \|R_i - R_j\|_1 = \|\mathbf{Az}\|_1, \tag{3.10}$$

where R_i represents the reflectance to be computed at pixel position i. This term enforces sparsity on reflectance values using local colour information in the form of weights v_{ij} in a 11×11 neighbourhood. This term can be rewritten in matrix form by linearizing the colour channels as a single column (z) and assembling a block matrix A of associated pixel weights.

Mid-level context (R_m): As R_l enforces sparsity based only on colour similarity in a small local neighbourhood, for mid-level context we enforce sparsity at object level using our selective search features (g_i). For ease of computation, we reduce the dimensions of g to get $\mathring{g}$ using PCA and redefine the weights as:

$$v_{ij}^m = \exp\Big(-\frac{(\overline{r_i} - \overline{r_j})^2}{2t^2}\Big)\Big(-\frac{(\mathring{g}_i - \mathring{g}_j)^2}{2t^2}\Big). \tag{3.11}$$

This prior enforces reflectance sparsity at object level which leads to colour constancy within an object. This captures object level semantics better compared to the local reflectance sparsity constraints which might lead to over flattening due to ambiguity between edges, textures and noise in an image. The complete mid-level reflectance prior is given as:

$$R_m = \sum_i \sum_{j \in \mathcal{N}_{11\times 11}} v_{ij}^m \|R_i - R_j\|_1 = \|\mathbf{Bz}\|_1. \tag{3.12}$$

Global context (R_g): The global reflectance prior encodes reflectance similarity at the scene level which is useful in enforcing colour constancy for various instances and occlusion disconnected parts of an object in the scene. We write R_g as:

$$R_g = \sum_{i \in Q} \sum_{j \in Q} v_{ij}^g \|R_i - R_j\|_1 = \|\mathbf{Cz}\|_1. \tag{3.13}$$

We define Q as the set of representative pixels obtained from each MCG segmentation [20] by ranking all the pixels in a segmentation according to minimum distance from the mean.

Image approximation (R_a): This term enforces continuity between the two stages by forcing the reflectance estimate from the current stage to be similar to the intermediate reflectance solution from the previous shading formulation stage. More concretely, we use this equation:

$$R_a = \|R_i - \rho\|_2^2 = \|\mathbf{z} - \boldsymbol{\rho}^*\|_2^2 = \|\mathbf{D}\|_2^2. \tag{3.14}$$

3.3.4 Iterations and Updates

Using Eq. (3.10), Eq. (3.12), Eq. (3.13) and 3.14 we can restate the main reflectance equation *i.e*. Eq. (3.8) in matrix form as:

$$\pi = \|\mathbf{Az}\|_1 + \|\mathbf{Bz}\|_1 + \|\mathbf{Cz}\|_1 + \|\mathbf{D}\|_2^2. \tag{3.15}$$

This is an L_1-L_2 minimization problem and can be solved by adapting the Split-Bregman iterations [97] by introducing intermediate variables $\mathbf{b}$ and $\mathbf{d}$ which reformulates the equation as:

$$\mathbf{z} = \underset{\mathbf{z}}{\operatorname{argmin}} \Big(\|\mathbf{D}\|_2^2 + \theta\big(\|\mathbf{d_1} - \mathbf{Az} - \mathbf{b_1}\|_2^2 + \|\mathbf{d_2} - \mathbf{Bz} - \mathbf{b_2}\|_2^2 + \|\mathbf{d_3} - \mathbf{Cz} - \mathbf{b_3}\|_2^2\big)\Big). \tag{3.16}$$

Here θ balances the contribution from reflectance sparsity priors *vs*. prior for shading consistency from previous stage. We recompute priors after each iteration for the two formulations based on the current values of σ^* and ρ^* and gradually update the contribution of various weighing parameters (λ, γ and θ), increasing the effect of mid-level priors, global priors and the previous solution, while reducing the effect of local priors over the course of iterations. It is challenging to decide the convergence of the iterations like in a general Split-Bregman method as there is no IID metric which can give us an estimate of the quality of the transient decompositions without ground truth. We cannot directly use reconstruction error as convergence criterion as it does not convey information about the perceptive quality of the decomposition. Hence we empirically estimate the total number of required iterations ($k = 5$) like other model parameters by manually tuning for optimal results over a small subset of images using hyper-parameter grid search.

3.4 Results

All our results are generated using a 5^{th} generation Intel i7 3.30 GHz desktop processor. Most of our prototype implementation is in Matlab [197] with a few sections in backend C++ suggesting a significant scope of improvement for runtime efficiency if code optimization is performed. We show the results of our method on the IIW dataset in Fig. 3.4 and Fig. 3.5. Notice separation of shadows and illumination from light sources to the shading component and the colour consistency in the reflectance component. To explore the generality of our method beyond IIW dataset (only indoor scenes), we also experimented with diverse images from the Internet (Fig. 3.12). Our method can work on several scene types (indoor, outdoors, natural, cityscapes *etc*.) with varying complexity (single object *vs*. multiple objects) and diverse lighting (single *vs*. multiple light sources, natural *vs*. artificial lighting, day *vs*. night lighting *etc*.).

We compare our method quantitatively with other contemporary IID methods which encode scene information in terms of IID priors *i.e*. Bi et al. [36], Zhou et al. [321], Bell et al. [33]. The results

Figure 3.4: Qualitative results on IIW: We show results from our SP-IID method. In each set from L to R we show: Original image, reflectance and shading on sample images from IIW dataset [33]. Notice separation of shadows and highlights in shading and colour sparsity in reflectance component.

are shown in Fig. 3.5 for the entire IIW dataset (green) and the test-split used by Narihira et al. [208] (blue). As Zhou et al. [321] use most of IIW dataset for training, we show their results only on the test-split. The scores are reported as mentioned in the respective papers or downloaded from the respective

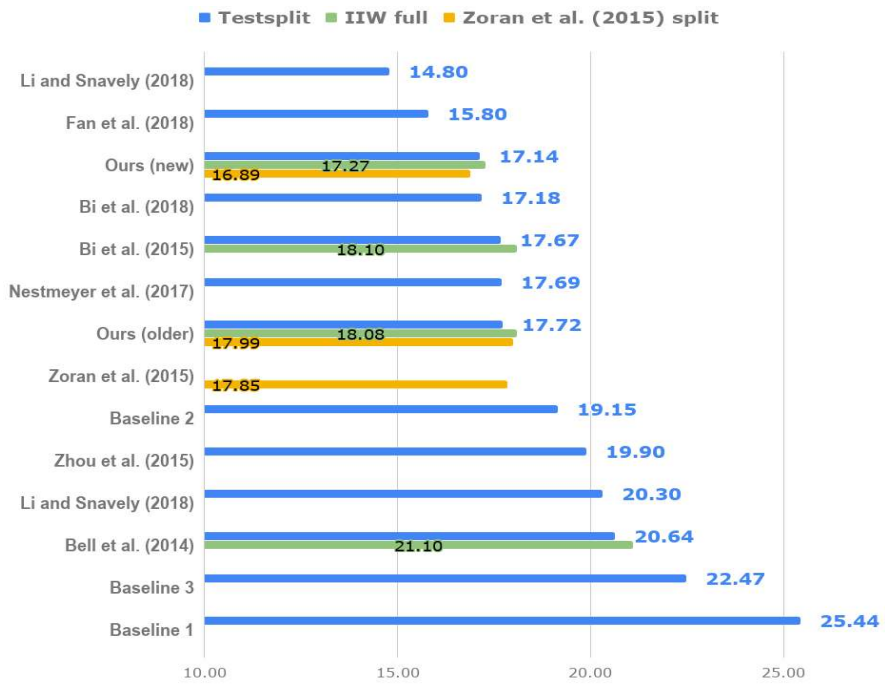


Figure 3.5: Quantitative results on IIW: Numerical performance comparison between our method and other contemporary IID solutions. WHDR scores are reported on the IIW dataset [33] where lower is better. Our score of 17.14 is above several baselines and previous optimization based IID solutions listed under. Only Fan et al. [77] and Li and Snavely [172] who propose large supervised deep learning based IID solutions on newly introduced large datasets are able to perform better than ours. Our results are better than all other optimization and unsupervised methods listed underneath.

project homepages. We also compare our method with three baselines and on the test-split by Zoran et al. [325] (orange) which we explain below:

- *Baseline 1*: only shading formulation is optimized.
- *Baseline 2*: only with the reflectance optimization.
- *Baseline 3*: edge preserving smoothing results as reflectance.

Notice that our *Baseline 2* performs better than both Zhou et al. [321] and Bell et al. [33] which highlights the strength of our reflectance priors. Also in order to show how different it is from the underlying image flattening framework, we have *Baseline 3* which is computed directly on the results of edge preserving smoothing from Bi et al. [36]. As can be seen from the graph in Fig. 3.5, our method achieves significant error reduction in comparison to both Bell et al. [33] and Zhou et al. [321] on both

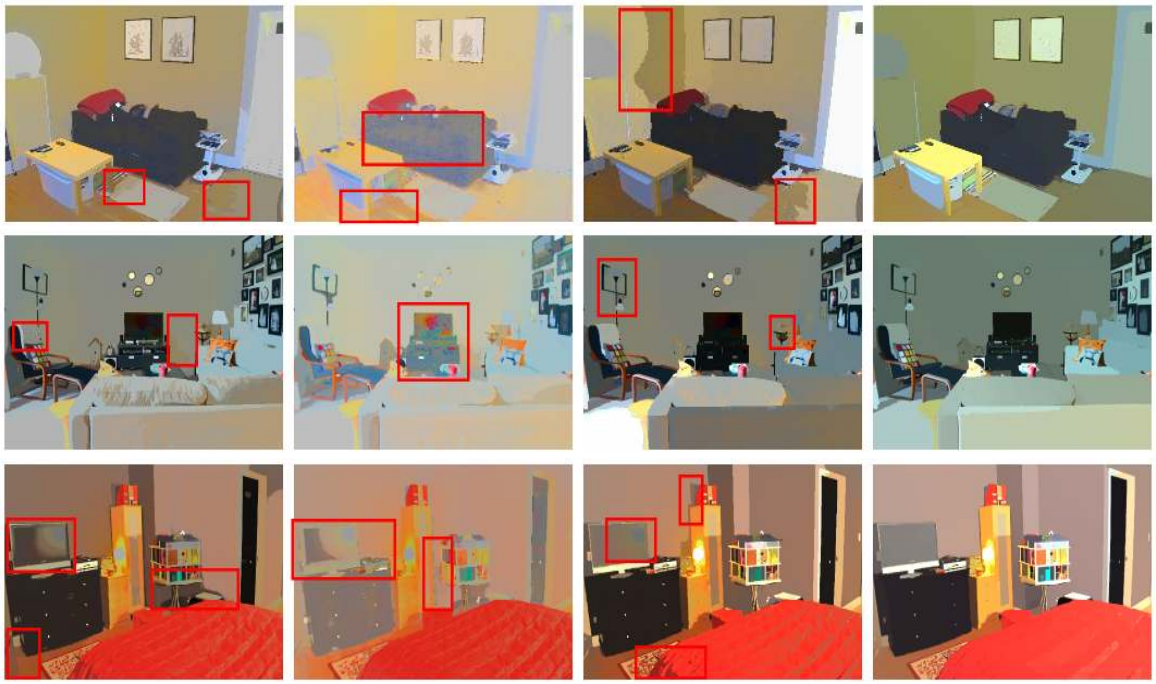


Figure 3.6: Qualitative comparisons: (L to R) Reflectance from [33], [321], [36] and our method. Compared to the other methods shown, our framework produces results with fewer artifacts.

the test-split and the full dataset (WHDR of $\mathbf{17.14}$ *vs*. 20.6 and 19.9 respectively on the [208] test-split). Our method is competitive in performance to both Bi et al. [36] and Nestmeyer and Gehler [212] (with WHDR 17.67 and 17.69 respectively) but with lesser artifacts in reflectance results as can be seen in Fig. 3.6. Additional comparisons with still older IID methods by Zhao et al. [317] and Garces et al. [89], with WHDR as 23.20 and 25.46 respectively, are not shown in graph for the sake of clarity. Also note that in our direct method, we do not need to perform separate clustering, classification or CRF labeling steps unlike other methods. Our semantic priors lead to consistent reflectance values with lesser patchy artifacts. Furthermore our results handle chromatic noise much better as can be seen in the reflectance of dark regions.

In parallel to our work presented here, there were a few recent direct deep learning solutions by Bi et al. [37], Fan et al. [77] and two works from Li and Snavely [172, 171]. The respective WHDR scores on the test-split split are 17.18, 15.80, 20.3 and 14.80. In the papers by Li and Snavely [171] and Bi et al. [37], authors have introduce new datasets for training. They use the illumination invariant property of reflectance from time lapse videos or synthetically rendered scenes as a prior for IID. Fan et al. [77] take inspiration from Nestmeyer and Gehler [212] and perform guidance filtering within the CNN framework rather than a separate post processing step which leads to significant error reduction.

Fixed grid parameter selection		
Grid size (pix.)	Stride (pix.)	Mean WHDR
30×30	30	18.19
45×45	45	18.09
45×45	30	18.17
60×60	60	18.39
$\mathbf{60 \times 60}$	**30**	**17.65**
60×60	15	18.59

Table 3.1: **Patch size ablation:** Experimentation for choosing fixed region feature extraction design parameters by varying grid size and overlap region percentage.

Based on this observation, we think that properly incorporating semantic information (perhaps in the form of region proposals or masks) within the deep network architecture would further improve the IID performance of such networks. Even with our current framework, if we allow for manual tuning of k parameter for each image, chosen based on image complexity (textures, colours, lighting), the error could be further reduced. Our observations are in-line with the conclusions provided by Nestmeyer and Gehler [212] that using explicit prior knowledge could improve IID performance and future end-to-end deep learning IID solutions could harness such priors for improved results as done by Fan et al. [77] by using insights from Nestmeyer and Gehler [212].

3.5 Analysis

Feature analysis: In order to analyze the effect of varying the design parameters involved during our weak semantic feature stage, we conducted various experiments using the standard test-split mentioned previously. To decide the grid size to be used for feature computation, we conducted several experiments and the results are reported in the Table 3.1. As we can observe from this table, the size of the grid and the overlap percentage between them, have a significant effect on the overall performance. Smaller grid don't capture enough contextual information whereas large grids are too ambiguous. Similarly too much overlap leads to most of the nearest neighbours getting picked from the same region, reducing the patch diversity and ability of the system to establish global constraints. Following our empirical observations, we used the 60×60 grid size with a sliding window stride of 30 for feature extraction.

Semantic features based prior estimation			
Prior strategy	Feature type	LLE approx.	Mean WHDR
p1	RGB	kNN	17.62
p2	RGB	random	17.54
p3	f_b	random	17.42
p4	f_b	kNN	**17.14**

Table 3.2: **Feature extraction strategy ablation:** Experimentation with various prior computation strategies, using our patch based weak semantic features f_b *vs*. mean appearance based RGB features.

We also experimented with four different strategies for our global prior term computation (p1, p2, p3 and p4). We estimate the effect of using our weak semantic features f_b *vs*. normal RGB appearance based cues. Additionally, we also analyze the effect of establishing constraints based on local linear embedding approximations computed using k-nearest neighbours (kNN) or randomly chosen patches. The results of these experiments are shown in Table 3.2. As can be observed from the 3.2 using mean RGB value based features alone in place of weak semantic features f_b gives higher error score. Also as RGB values only capture appearance cues and might not indicate correct structural similarity, even randomly choosing patch neighbours (p2) performs better than kNN based linear approximation strategy (p1). RCNN based weak semantic features are able to capture the structural similarity much better than only mean RGB values with kNN strategy improving performance (p4) over random chosen neighbours strategy (p3).

Framework analysis: In Fig. 3.7 we show qualitative performance of our method for a sample image over successive iterations. Notice how as per the intended design of our framework, the reflectance component from our second formulation gradually gets more 'flattened' while shading from the first formulation becomes smoother. Split-Bregman method uses reconstruction error as the stopping criterion [97, 36] but in our case it cannot be directly used to quantify IID performance. Hence we empirically estimate the value of k. Considering various scene and lighting settings we observed that overall our algorithm achieves peak perceptual and quantitative performance for $k = 5$ which can be seen in the WHDR *vs*. iterations graph in Fig. 3.8. Better performance could be obtained if IID quality could be approximated for each image separately without ground truth information. But devising such a metric is non-trivial and beyond the scope of this work. From our experiments we observed that manually selecting optimum k for each image separately can reduce the error.

Ablation study: In order to highlight the significance of various context priors, we conducted an ab-

Figure 3.7: Optimization iterations visualization: From left to right we show Shading formulation results (σ^*) and reflectance formulation results (R^*) for iterations $k = 1, 3, 5, 7$. Notice how shading gets 'smoother' while reflectance becomes 'flatter'.

lation study (Table 3.3) using different variants of our framework formed by combining different prior terms. The study was conducted on the standard test-split of 1046 images defined by Narihira et al. [208]. Variant v1 is essentially iterative Retinex model based smoothing followed by image flattening. Similarly v4 is only local L_1 flattening performed on top of L_2 shading formulation. These two variants capture the effect of the two stages with inclusion of basic 'smoothening' and 'flattening' steps. Addition of other context priors on top of these basic variants successively improves the performance proving the significance of these priors. In v2 and v5, we introduce the global context priors, leading to improvement in performance over v1 and v4 respectively. The large error drop from v1 to v2 is due to our global semantic priors based on RCNN features (f_b) computed on a fixed grid. In v3 and v6 we introduce mid-level context priors using selective search features (g_i) computed using flexible regions, which further leads to significant error reduction. This shows the utility of our semantic priors at various context levels. Overall the combination of all these priors gives the best IID results which can be observed from comparisons from v2 and v6 *vs*. v7 which gives the best qualitative and quantitative performance. The qualitative results obtained using these variants are shown in Fig. 3.9. Note, v1 has

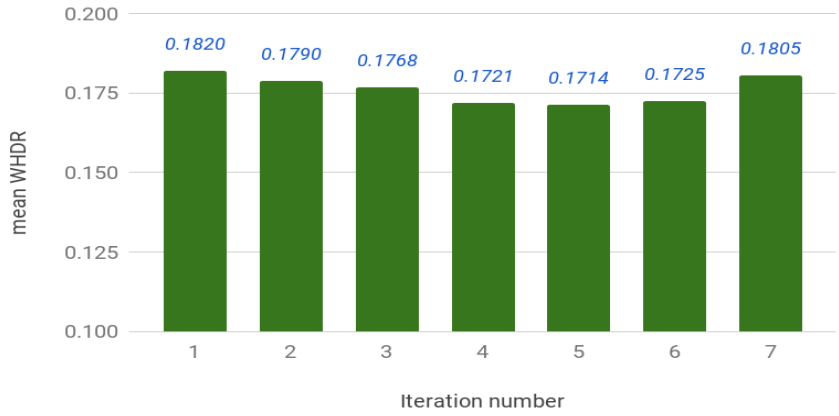


Figure 3.8: Optimization iteration analysis: The graph shows iterative WHDR reduction for the image with minimum at $k = 5$.

Ablation Analysis			
Variant	Shading priors	Reflectance priors	Mean WHDR
v1	S_l	$R_l + R_m + R_g$	24.34
v2	$S_l + S_g$	$R_l + R_m + R_g$	18.70
v3	$S_l + S_m$	$R_l + R_m + R_g$	17.16
v4	$S_l + S_g + S_m$	R_l	22.12
v5	$S_l + S_g + S_m$	$R_l + R_g$	22.09
v6	$S_l + S_g + S_m$	$R_l + R_m$	16.88
v7	$\mathbf{S_l + S_g + S_m}$	$\mathbf{R_l + R_m + R_g}$	**16.86**

Table 3.3: **Framework abalation:** Ablation analysis and our results on challenging Internet images highlighting generality of our method for a variety of scene types and light settings.

very little structural information as most of the shading priors are missing and hence derives results mainly based on colour information. This causes incorrect IID reflectance as shown in column 1. Variant v2 brings scene level structural information in the form of S_g but in a few cases is unstable as no mid-level semantic information is present. v3 gives significantly better results compared to previous two as it has nearly all the priors but for a few cases might lead to incorrect global reflectance tone due to lack of global shading information. v4 and v5 give good reflectance results but do not handle

Figure 3.9: IID using different variants of our framework: In each scene from left-to-right: Results using variant v1, v2, v3, v4, v5, v6 and v7. Variants v3 and v6 give good results as they contain all except one prior. Variants v4 and v5 lack mid-level reflectance sparsity term and is unable to remove the highlights from the scene (in 4^{th} and 5^{th} columns light gradients and shadows are not properly removed). Finally v7 gives overall the best qualitative and quantitative results.

shadows and lights well and contain some artifacts. These are better handled by v6 due to our semantic prior R_m. Finally v7 though looks similar to v6 but also gives overall best quantitative performance.

We reuse the values of most of the parameters in Split-Bregman iterations as provided by Bi et al. [36] and empirically estimate the remaining parameters over a small subset of images. All analysis and results in this chapter are generated using these fixed set of parameter values: $\lambda_g = \lambda_m = 0.02$, $\lambda_l = \gamma_g = 2$, $\gamma_m = \gamma_l = 20$, $\theta = 40$, $\tau = 1.2$, $t_c = 0.0001$, $t_m = t_b = t = 0.05$, $k = 5$

3.6 Applications

In the previous sections we have discussed IID modeling, experiments, analysis and results as a standalone computer vision problem. In this section we employ the results obtained from our framework to present two novel IID applications. For this, we utilize the results from the final iteration from both the formulations *i.e.* ρ and σ from the shading optimization and similarly R and S from the reflectance optimization. Although our IID modeling is based on a simple image model (Section 3.1), but here we show that we can approximate more complex lighting components for new applications by combining the results from our two optimization formulations. Note that the two methods presented here are sample applications implemented as automatic but fixed parameter systems and require no user intervention during execution. These applications could be further improved with an interactive interface which allow user input to enable more specific effects.

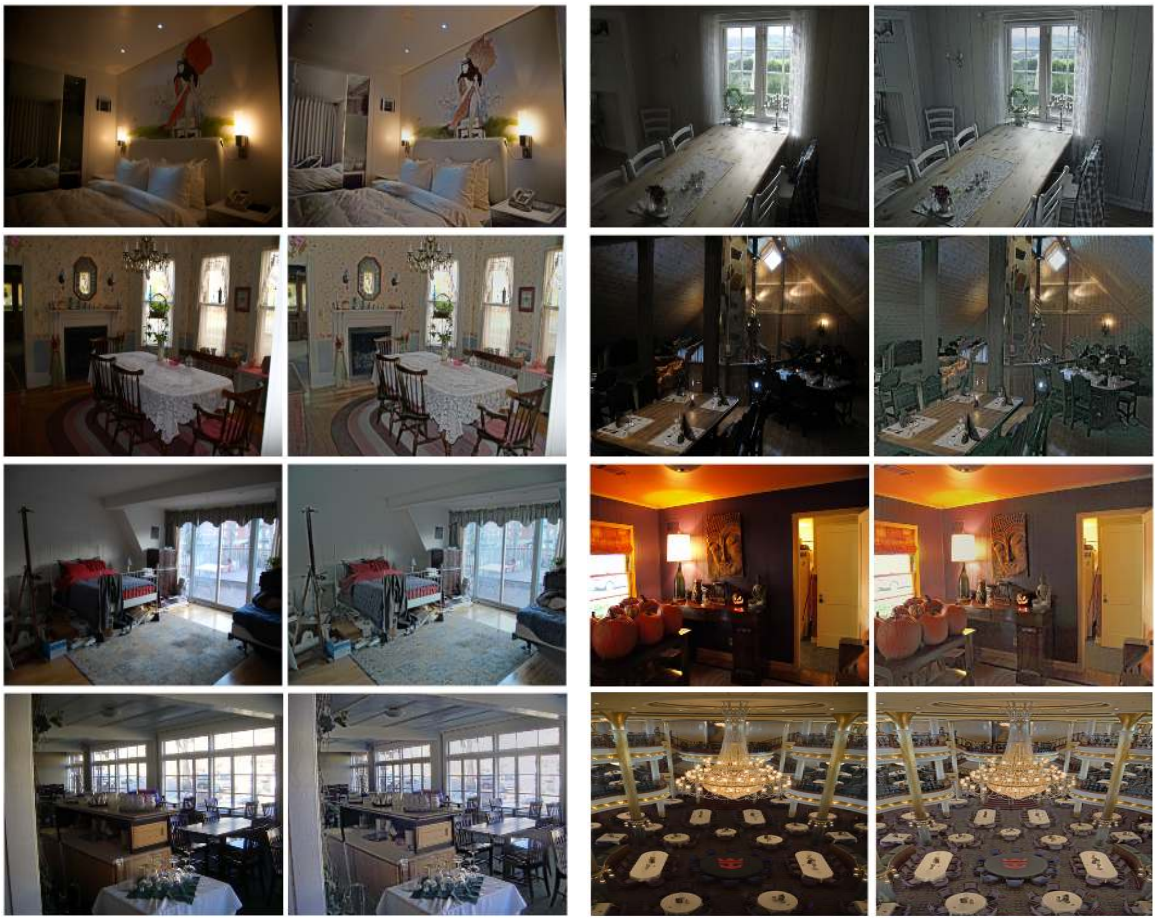


Figure 3.10: Single image LDR to HDR conversion: We increase the contribution of our approximated indirect light component without altering the direct component to enhance the visibility of the dark regions in the image. Here for each scene original image and the relit versions are shown.

Single image LDR to HDR conversion: As the dynamic range of cameras and display devices is limited, some poorly or improperly lit images are excessively dark or bright in certain regions. This is due to extreme intensity variation between bright light source regions *vs*. some dark unlit regions in a given scene. Such images have Low Dynamic Range (LDR) of intensity compared to properly lit or intensity remapped images called High Dynamic Range (HDR) images. We use our IID results to relight dark regions in a given image achieving the effect of single image LDR to HDR conversion. We use the insight that our first formulation results have smooth shading component whereas second formulation has flat sparse reflectance component by design. Hence by disassociating additive and multiplicative residual information using flat and smooth components from their denser counterparts, we can estimate the regions which are hard to decompose due to low light. In order to estimate the

residual information we compare the two results components as:

$$E_1 = mean((\rho - R), (S - \sigma)) \quad \text{and} \tag{3.17}$$

$$E_2 = mean(\frac{\rho}{R}, \frac{S}{\sigma}). \tag{3.18}$$

We add the Gaussian filtered estimates back to the original image (*i.e.* I+Gaussian(E_i)) and rescale the results between normal image intensity values for visualization. By this we obtain the final well lit HDR image as shown in Fig. 3.10. Notice how the dark regions are highlighted but in the well lit regions the intensity is maintained, achieving a shading sensitive intensity normalization. This presents a simple yet novel method for IID based single image LDR to HDR conversion application.

Illumination colour based image tone manipulation: In order to change the illumination colour for image tone manipulation, we need to approximate illumination chromaticity and regions. As our original IID light transport model assumes monochromatic illumination, we approximate the low and high frequency smooth component of illumination color as:

$$C_1 = \frac{I}{I_{R\sigma}} \quad where\ I_{R\sigma} = R \cdot \sigma \quad \text{and} \tag{3.19}$$

$$C_2 = \frac{I}{I_{\rho S}} \quad where\ I_{\rho S} = \rho \cdot S \quad \text{respectively.} \tag{3.20}$$

We take Gaussian filtered mean of these two estimates to obtain our illumination colour approximation C. We update our shading component by dividing C from S and multiplying it to σ and taking the mean of both the results. In order to estimate the light source regions in the image, we find the pixels within the high percentile set in our updated shading component. We change the intensity colour of shading component in CIELab space based upon the distance of pixels from the estimated light source regions. This gives us a illumination colour modified shading component. Recombining this modified shading with reflectance gives us the illumination colour changed image as shown in Fig. 3.11. In Fig. 3.11 we show illumination recolouring for several scenes using red and green tone modification. Notice that the modified colour of the light source regions in the original image and the shading intensity based tone adjustment of the surrounding regions. The objects farther away from the estimated light source regions, retain their original colour. Hence shading sensitive tone adjustment achieves a much more subtle and realistic effect, unlike putting the entire image through a red or green filter. This illustrates a novel method for IID based illumination colour manipulation.

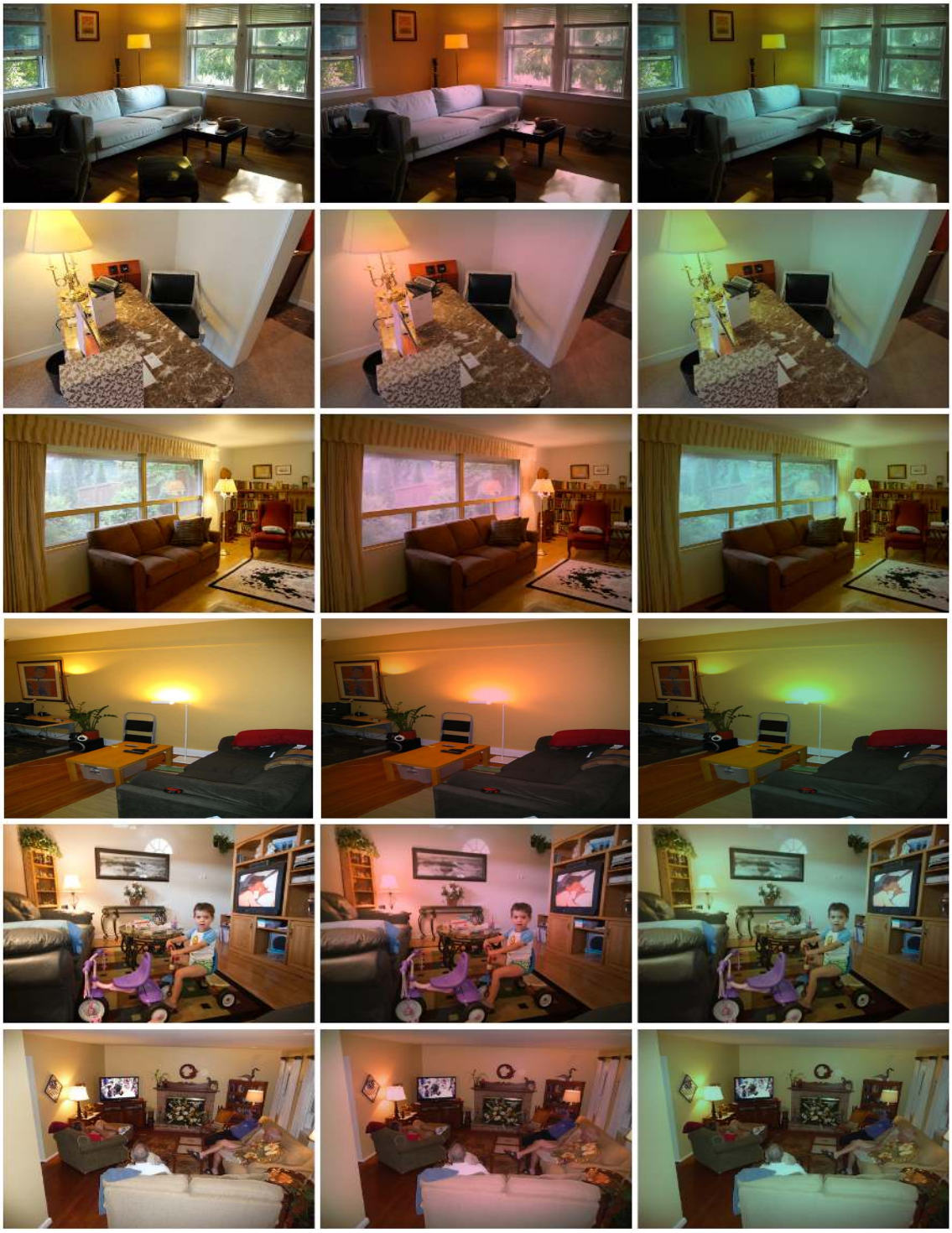


Figure 3.11: Image tone manipulation: Using our approximated lighting region and colour estimates, we can modify the illumination colour in the scene, thereby altering the tone of the image. Here each scene is shown in original, red illumination and green illumination respectively.

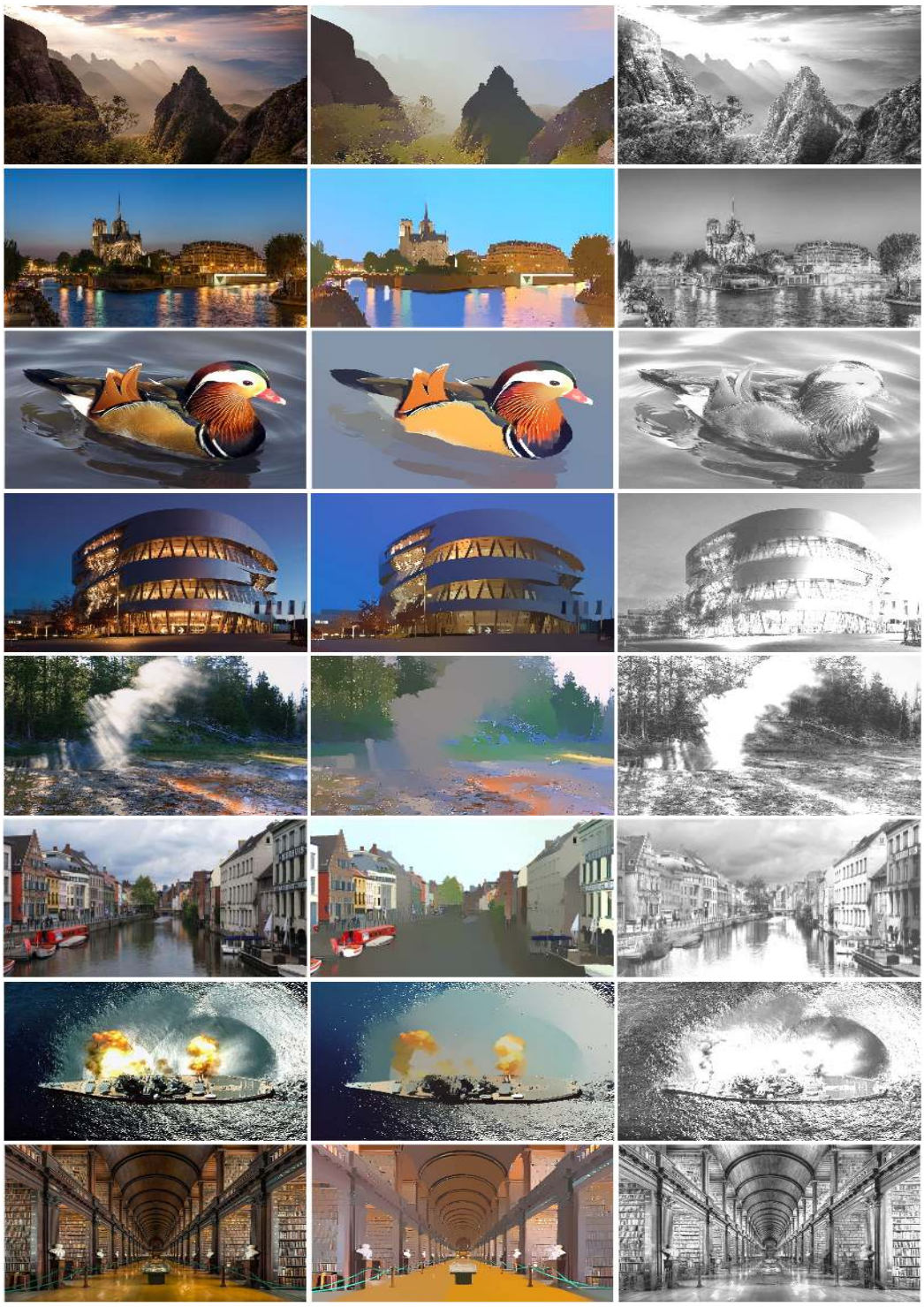


Figure 3.12: Result visualization on wild images: Our results on challenging internet images. The scenes shown contain variations like indoors/outdoors, artificial/natural lighting, day/night environment, complex/simple subjects, multiple/single illuminant and dynamic/static content (L to R : input, reflectance and shading images).

Figure 3.13: Failure cases: Note incorrect decomposition in marked regions with challenging sharp highlights, shadows and fine textures in a colour similar to object. This is an indirect result of our initial shading smoothness and reflectance flatness modelling assumption (L to R : input, reflectance and shading images).

3.7 Limitations and Future Work

The Fig. 3.13 shows a few failure scenarios of our proposed framework. An often observed challenging case is that of images with sharp shadow and highlight regions. Owing to the lack of depth data or some similar additional structural information, most single image IID methods struggle in this task of disambiguation of such gradients from sharp object boundaries. Yet another issue is distinguishing fine local textures in the same colour as object reflectance and lighting variation. Our method is able to handle mid-level and large textures well due to our semantic priors but in a few cases such textures get decomposed into the shading layer. Finer textures of similar colour as that of the object, persist in shading component due to ambiguity in differentiating local illumination changes with such textures (this is not an problem with differently coloured textures). Notice how in the last image the textures on the table cloth are correctly decomposed into the reflectance layer but the textures on the wood owing to the same colour as that of the object itself, are shifted to the shading component. Also while our priors work on varied scenes and generate lesser artifacts, in few cases it is difficult for us to distinguish sharp shadows and highlights from reflectance variations. These issues are not unique

to our method and are also observed in several other solutions [39]. Still our object semantic priors and alternating iterative model design leads to perceptually better decompositions for a large variety of scene and diverse lighting settings (Fig. 3.12). Discounting the training time, deep learning based solutions generally run faster during testing in comparison to energy based optimization methods. Hence the unoptimized prototype implementation of our method is slower compared to other methods (few seconds *vs*. minutes) but this could be significantly improved with better implementation and parallelization.

In order to automatically assign the value of total number of iterations k based on the lighting and scene complexity, we would like to explore the problem of learning a performance metric for IID respecting both perceptual and quantitative assessment without ground truth information. It would also be interesting to see the effect of explicitly introducing semantic information in current deep learning IID solutions. We believe that properly encoding semantic and contextual information as an additional information, either as collated input, a separate network branch and/or as a loss function, would help improve the performance of the new IID deep learning solutions [37, 77, 172, 171]. Additionally, it will also be interesting to see the utility of our and other recent IID solutions in novel applications like automatic video-editing, object insertion, machine learning dataset augmentation, style-content disambiguation, *etc*. In future we would like to explore these questions in the context of IID and in the broader context of inverse rendering and inverse light transport research in general.

3.8 Conclusion

In this chapter we present new priors which encode class agnostic weak object semantics using selective search and pre-trained region-based Convolutional Neural Network features. We encode these priors by analyzing scene at three hierarchical context levels and use an integrated optimization framework for single image intrinsic image decomposition without requiring any additional optimization steps. Our system has two alternating optimization formulations with competing strategies: first focusing on shading smoothness and the second on reflectance sparsity. We highlight the effectiveness of our strategy and semantic priors with supporting qualitative and quantitative experimentation and results. We hope our work will draw attention of wider research community towards the utility of semantic priors and hierarchical analysis for the problem of intrinsic image decomposition and in the future will lead to better end-to-end deep learning architectures and optimization frameworks.

PART III

Illumination Factorization

Chapter 4

Robust Principal Component Illumination Analysis

In this unit we move beyond the intrinsic image decomposition task and focus on the task of illumination analysis which shares similar modelling functions to the IID. In this work specifically, we focus our attention on the image illumination content and propose a new technique to factorize the image into multiple additive components by decomposing the image iteratively into a low-rank reflectance and a sparse shading term. The core idea behind this to be able to disentangle various underlying illumination sensitive factors and mark the regions in the image corresponding to them. Similar to the IID components, this allows better control of the process by inverting the rendering process.

Image Fusion maximizes the visual information at each pixel location by merging content from multiple images in order to produce an enhanced image. Exposure Fusion, specifically, fuses a bracketed exposure stack of poorly lit images to generate a properly illuminated image. Given a single input image, exposure fusion can still be employed on a 'simulated' exposure stack, leading to direct single image contrast and low-light enhancement. In this work, we present a novel 'Quaternion Factorized Simulated Exposure Fusion' (QFSEF) method by factorizing an input image into multiple illumination consistent layers. To this end, we use an iterative sparse matrix factorization scheme by representing the image as a two-dimensional pure quaternion matrix. Theoretically, our representation is based on the dichromatic reflection model and accounts for the two scene illumination characteristics by factorizing each progressively generated image into separate specular and diffuse components. We empirically prove the advantages of our factorization scheme over other exposure simulation methods by using it for the low-light image enhancement task. Furthermore, we provide three exposure fusion strategies which can be used with our simulated stack and provide a comprehensive performance analysis. Furthermore as our proposed layered representation is intuitive and directly usable, we also present a simple graphical interface for user-guided image contrast manipulation. Finally, in order to validate our claims, we show extensive qualitative and quantitative comparisons against relevant

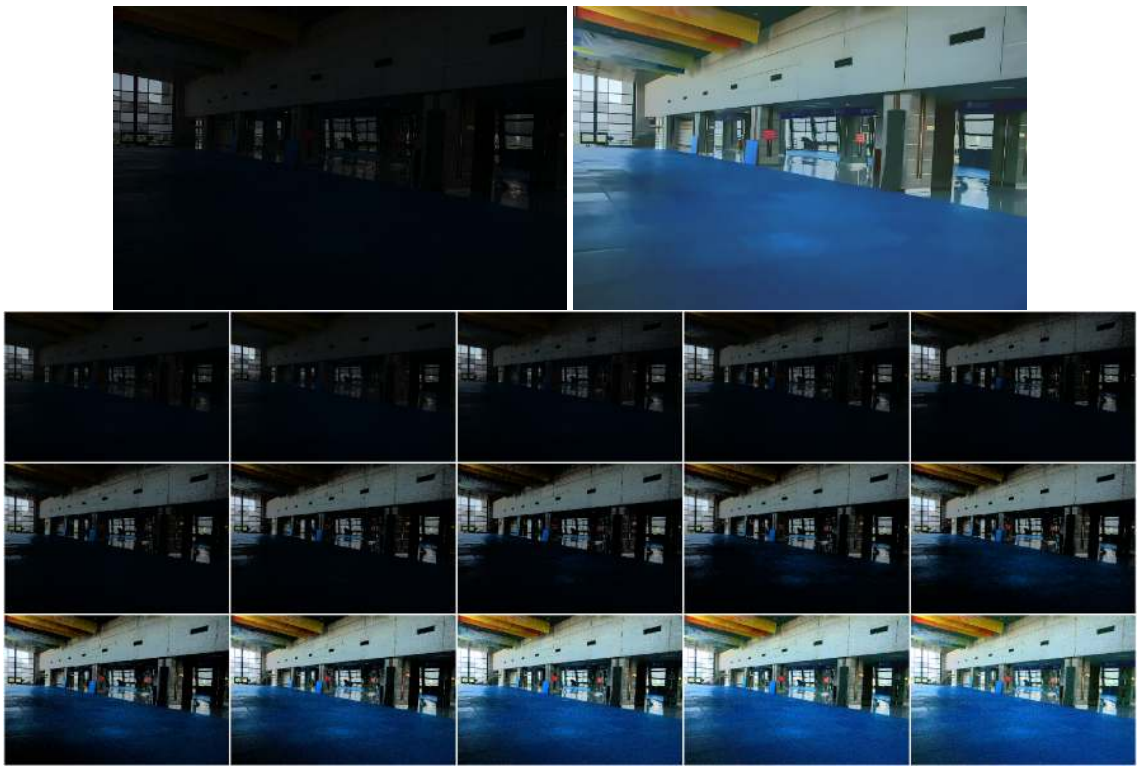


Figure 4.1: Quaternion Factorized Simulated Exposure Fusion (QFSEF): Given a poorly lit image as input (top-left), we factorize it into multiple illumination consistent layers using a pure quaternion matrix factorization scheme, which we then use to simulate an exposure stack (bottom 15 images) and fuse to obtain an enhanced image (top-right). Note how the simulated exposure stack slowly improves the image brightness while the final enhanced image is able to balance the over and the under exposed regions in the image properly.

state-of-the-art solutions on multiple standard datasets along with relevant ablation analysis to support our proposition. Our code and data are publicly available for easy reproducibility and reference. [1]

4.1 Introduction

Image enhancement is a classic Computer Vision application involving problems like image denoising, super resolution, contrast enhancement, deblurring *etc*. Low light image enhancement is also a well studied research problem with utility in high-level applications such as image classification, recognition, reconstruction *etc*. Several solutions have been proposed to this end using a variety of input modalities like flash-noflash image pairs, bracketed exposure sequences, light field data, raw

[1] `https://github.com/sophont01/QFSEF`

sensor signals *etc*. Flash-noflash images enable a range of applications like material estimation, light spectra estimation and illumination map estimation but the with the core limitation that scene areas far away from the flash illumination are not resolved properly in the result. On the other hand, models which assume raw images from the camera or light field data as inputs are restrictive in everyday applications. Comparatively, it is easy to obtain a bracketed sequence of images with varying shutter speed (*i.e*. an exposure stack) using already available camera modes and external install-able softwares. We focus on this category of low-light image enhancement methods and propose a technique to simulate an exposure stack from a single poorly illuminated input image.

Several stacked image representations have been used in the literature for multiple applications. Apart from the exposure stacks, bracketed sequences of images with varying depth-of-field have also been used in several papers [127, 235, 78, 10]. Several other methods also exist which use layered depth information for view synthesis and scene understanding [270]. Semantic information has also been factorized into multiple layers using soft segmentation, color un-mixing and image matting in [15, 13]. The main advantage of layered representations is that the similar image regions are grouped together which allows easy global manipulation without requiring adaptive local adjustments. Additionally, compared to patch based processing, our estimated layers have an intuitive semantic meaning and can be directly utilized by end users for preferential adjustments.

Single image low-light enhancement is a challenging task considering the limited amount of information available. This is further exacerbated due to camera sensor noise prevalent in the dark image regions, which, if not handled properly will lead to colored artifacts in the results. Earlier solutions for this task were mostly optimization based involving histogram equalization, tone curve adjustment, Retinex theory based illumination map estimation *etc*. Recent advances in the field mostly employ deep learning techniques and train priors using supervision on large datasets. Current state-of-the-art methods are inspired by the fundamentals of the the problem and try to encode domain expertise in their systems *e.g*. decomposing the problem using Retinex theory [283], constructing multi-scale Laplacian pyramids [9], adaptable tonal curve adjustment [105] *etc*.

In this work, we present a new insight into the problem by proposing to factorize an image into a sequence of specular factors. We propose a method to progressively remove the specular content from the image by performing iterative sparse matrix factorization which can be used to render a virtual exposure stack. This converts the difficult single image exposure correction problem into a simpler exposure fusion task thereby allowing us to directly use existing exposure fusion strategies on a single image. In order to make this simple idea work, we harness the power of *Quaternions* by representing the image pixel color values on a unit norm sphere of pure quaternions (*i.e*. no real component). This enables better inter-color spectral representation across channels compared to the traditional concatenated color channel approach and allows for a simple yet elegant and interpretable

solution.
To summarize, following are our main contributions in this work:

- We present a novel single image exposure fusion method by simulating an exposure stack for low light image enhancement application.

- We present an iterative matrix processing scheme for illumination factorization by using Robust Principal Component Analysis.

- We pose the illumination factorization for the low light enhancement problem in the quaternion space and show the benefit of this algebraic mapping.

- We provide performance scores and comparison with the current state-of-the-art solutions via extensive experiments and ablation analyses.

4.2 Related Work

4.2.1 Quaternion Image Processing

Apart from the wider utility of quaternions in robotics and graphics for 3D manipulation, they have also been used in several image processing tasks. Quaternion counterparts to several crucial techniques like Fourier transforms [73], Singular Value Decomposition (SVD) [240], Principal Component Analysis (PCA) [38], derivatives like Gradients and Hessians [288], and optimizations like least square algorithm [149] *etc*. have been proposed in the literature and newer techniques are still being developed. Following this, several methods based on the application of these techniques have also come up. Initially, Sangwine [239] used quaternion image representation for the task of color image edge detection and proposed additional quaternion filters in their later work [241]. Afterwards, Ell and Sangwine [73] formulated vector Fourier transform for quaternions which was then used for image saliency prediction [244], texture estimation [21], motion detection [17], image smoothing [260] *etc*. Recently, quaternions have also been used for face recognition [133], image inpainting [130] and in image forensics [302]. Here we use one such existing quaternion optimization technique and propose a novel low-light illumination application. Specifically, we use the quaternion Robust Principal Component Analysis (RPCA) formulation proposed by Chan and Yang [55]. The authors in [55] use quaternion RPCA for the task of separating relatively sparse human voice signals from the background musical score in a short music clip. Taking inspiration from their work, we propose to use this technique for image factorization with an aim to use it for the low light enhancement problem.

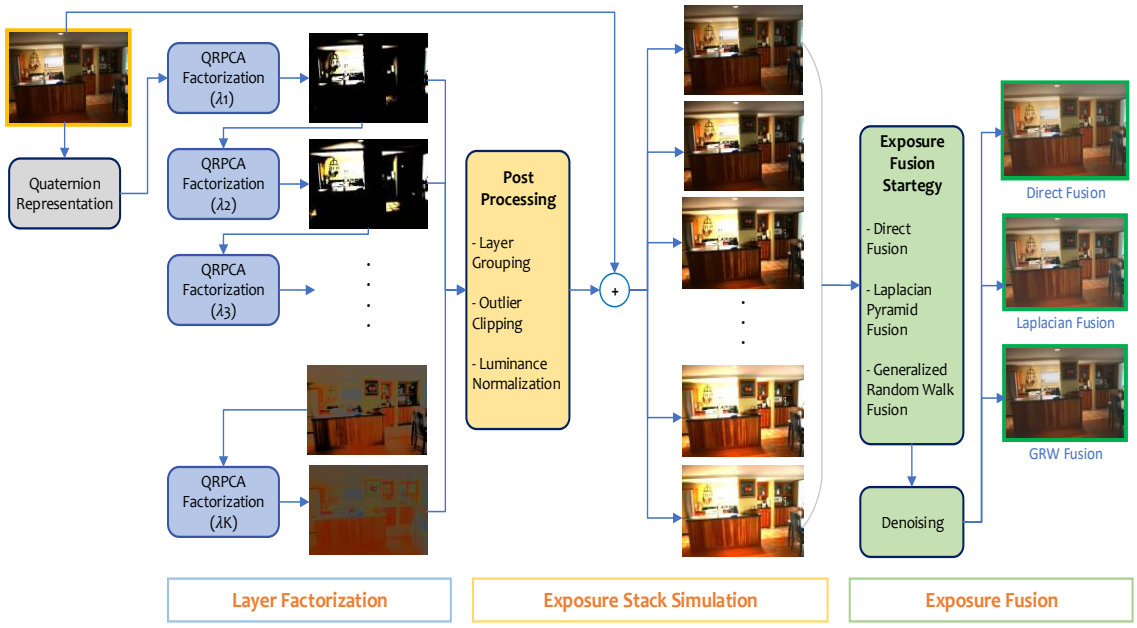


Figure 4.2: QFSEF Conceptual Overview: An abstract block diagram of our proposed approach shows our three system sub-modules: *Layer Factorization*, applies gradually relaxed iterative RPCA on Quaternion representation of the input image, followed by *Exposure Stack Simulation*, where we combine factors with original image to render controlled illumination image sequence and finally *Exposure Fusion* where we merge the stack information via three strategies to obtain enhanced results.

4.2.2 Robust Principal Component Analysis

RPCA as used in this work, was first proposed by Candès et al. [51] in which the authors' presented a closed form solution to the sparse *vs*. low-rank matrix decomposition problem using a convex optimization algorithm named Principal Component Pursuit (PCP). The method is named RPCA because it is able to recover matrix principal components even in the case of corrupted or missing values. Candès et al. [51] illustrated the utility of their algorithm by using it in two computer vision tasks: video background removal and specularity estimation. Since its initial proposition, RPCA problem has been solved using a variety of other optimization methods beyond PCP [41] and used in a variety of applications [42] in several domains like low level image analysis, medical imaging, 3D computer vision and video processing [104, 124, 58, 322, 271]. We take inspiration from the initial application of RPCA by its authors and use it for image specularity estimation. Note that the solution proposed by Candès et al. [51] was proposed only for real number matrices and the extension to quaternion space was later proposed by Chan and Yang [55] by defining appropriate quaternion projection operator.

4.2.3 Illumination Analysis

From image intrinsics perspective *i.e*. Retinex theory [154, 28] as discussed in the two chapters in the previous unit Chapters 2 and 3, there are two fundamental components in the image formation process *i.e*. object material dependent reflectance and scene illumination dependent shading. Either or both can be used for various image based rendering applications which involve generating a new image directly from a given input image. Controlling the first reflectance component enables applications like re-texturing, material modification, object recolouring, palette extraction *etc*., while the shading component can be manipulated to perform shadow removal, glare removal, low light enhancement, image relighting *etc*.

Illumination manipulation can be done at either global or local level. Global analysis involves techniques like histogram equalization, white balancing, gamma correction, light source direction computation, illumination spectra estimation *etc*. Local methods do neighbourhood analysis and involve methods performing spatially varying illumination and environmental map estimation techniques. Illumination analysis has also been performed for applications like shadow, haze, underwater blur and glare removal. It is also carried out for image based rendering task like harmonization after compositing an object inside image and image relighting. In this chapter, we specifically focus on low light image enhancement methods and discuss them in detail in the next section.

4.2.4 Low Light Image Enhancement

One crucial category of illumination analysis is performed for the task of exposure correction in poorly lit images. Earlier methods in this field were based on either pixel intensity manipulation (statistically via histogram equalization [220, 218, 160], or individually via intensity curve manipulation [305, 115]) or tried to estimate illumination map based on Retinex theory [154, 278, 84]. Recently, several deep learning based methods have been proposed which have been trained on paired image datasets with low light images and their enhanced counterparts. Some of these solutions have been proposed for raw inputs or require other camera parameters as input. We restrict our discussion to the works which assume standard $sRGB$ image as input.

Gharbi et al. [93] proposed an image enhancement architecture by introducing deep bilateral filtering. Wei et al. [283] took inspiration from the Retinex decomposition and proposed RetinexNet by first splitting the image into separate reflectance and illumination components. Later, Zhang et al. [314] extended this idea by focusing on dark regions. Guo et al. [105] introduced a light weight zero-shot network which predicts global gamma correction parameters for the image which they apply iteratively to achieve enhancement. Afifi et al. [9] introduced a new dataset by changing exposure values from the information obtained from the raw images and train a sequential model on the basis of image Laplacian pyramid decomposition. Recently, Yan et al. [294] proposed a multi-branch model for

separately handling the overexposure caused by the light sources in dark images. We use several such recent state-of-the-art methods for comparison in Section 4.4.

4.3 Quaternion Factorized Simulated Exposure Fusion

Figure 4.2 illustrates our QFSEF framework that can be divided into three sub-modules which sequentially are:

1. Layer Factorization Module
2. Stack Simulation Module
3. Exposure Fusion Module

We individually discuss each of these modules in detail in the following sub-sections.

4.3.1 Layer Factorization

Our first sub-module performs factorization of the given input image into multiple layers. Factorization is carried out in such a way, so that the illumination characteristics are consistent in each layer. Specifically, we follow the *Dichromatic reflection model* [266] under which, for a given image I, the total irradiance at any pixel position x is the sum of diffuse and specular reflection components ($I_{diffuse}$ and $I_{specular}$):

$$I = R_d \cdot S_d + R_s \cdot S_s = I_{diffuse} + I_{specular}, \tag{4.1}$$

where R_d and S_d are diffuse material reflectance and scene shading respectively. Similarly, R_s and S_s refer to specular reflectance and shading components.

4.3.1.1 Robust Principal Component Analysis for Specularity

Specular component of the irradiance is sparse in nature and can be extracted from I via matrix factorization methods [6, 11, 108, 106, 313]. To this end, we use Robust Principle Component Analysis (RPCA) by Candès et al. [51] which has been used previously for various computer vision problems [42] like background separation, denoising, specularity removal, tracking *etc*. RPCA has a tractable convex approximation for the original NP-hard formulation:

$$I = \min_{A,E} A_{rank} + \lambda E_{sparse} \approx \min \ \|A\|_* + \lambda \|E\|_1 \quad \textit{s.t. } I = A + E \tag{4.2}$$

Here E and A are estimations of the sparse specular and the remaining low-rank diffuse components respectively. Also $\|\|_*$ and $\|\|_1$ respectively stand for nuclear and L_1 norms which are used to approximate the original low-rank A_{rank} and sparse E_{sparse} matrices. λ is a positive constant for controlling

the sparsity (= specularity) of the result. Equation (4.2) can be solved using various optimization techniques as mentioned in Bouwmans et al. [41]. Specifically, we employ the Augmented Lagrangian Method (ALM) [176] which uses Principal Component Pursuit (PCP) [51] for the optimization.

4.3.1.2 Quaternion Representation

We use the quaternion extension of PCP introduced by Chan and Yang [55] for instrument and voice signal separation in music clips. Quaternions are hyper-complex numbers which can be understood as an extension of the regular complex numbers in three dimensions with one scalar (r) and three vector components (i, j, k) such that:

$i^2 = j^2 = k^2 =$ -1 and $i.j = k$; $j.k = i$; $k.i = j$; $i.k =$ -j; $j.i =$ -k and $k.j =$ -1.

We first represent the normalized image I as a pure Quaternion matrix (*i.e.* real part r=0) by encoding each pixel as a *Versor* (*i.e.* a unit norm quaternion) with the color channels (RGB) encoded as the three quaternion vectors:

$$R \mapsto i, \qquad G \mapsto j, \qquad B \mapsto k. \tag{4.3}$$

The advantage of optimizing in quaternion space compared to concatenated RGB matrices is that the quaternion PCP preserves both spectral and inter-channel phase information unlike Real (no phase) and Complex (spectral phase) variants [74, 55]. Thus the color channel information is better represented in this manner leading to improved factorization. This claim is empirically verified later in Section 4.4.

4.3.1.3 Iterative Factorization

In order to generate more than two simulated images we propose to apply the RPCA splitting iteratively on the factorized low-rank component by gradually reducing the sparsity constraint λ from $\lambda_{max} \rightarrow 1$. This can be understood as progressive subtraction of the specularity content from the factorized components till the total signal energy limits to zero. To achieve this, we gradually relax k in Eq. (4.2).

$$I \quad = \quad E_1 + A_1 \quad = \quad E_1 + (E_2 + A_2) \quad = \quad E_1 + E_2 + E_3 + A_3 = ... \quad = \quad \textstyle\sum_i^K E_i. \tag{4.4}$$

Figure 4.4 shows an example of such factors and their successive differences (*i.e.* $E_{i+1} - E_i$) for visualization. Note how the initial factors contain predominantly scene highlights and light sources, followed by differently lit regions and finally shadows (loosely representing: sharp highlights $\rightarrow$ soft highlights $\rightarrow$ direct $\rightarrow$ indirect $\rightarrow$ ... $\rightarrow$ soft shadows $\rightarrow$ dark shadows *etc.*). One can also look at these factors, especially the successive differences, as illumination consistent super-pixel regions which are similar in their shading content instead of color. The scene hence gets factorized into multiple illumination consistent layers which can be globally processed with simple enhancement operations.

Input (I)

Iterative factors (E_i)

$E_{i+1} - E_i$

Simulated Exposure Stack (S_i)

Figure 4.3: Layer Factorization: Input image shown on top is iteratively split using quaternion RPCA into multiple factors as shown in the center-left grid. Corresponding factor differences shown in center-right grid, highlight additional information captured in each successive estimated factor. Finally, the last row shows the exposure stack simulated using these factors. All the images have been luminance normalized as discussed in Section 4.3.2, for visualization. Note how various image regions are identified separately in each successive factor in the fator difference images and how the simulated exposure stack is completely natural looking without color or details degradation.

4.3.2 Stack Simulation

As the image layers obtained from the previous step posses similar optical characteristics, we can directly edit them with simple global image manipulation operators without introducing significant illumination artifacts. Based upon this idea, we simulate a virtual exposure stack from a single image for the purpose of low-light image enhancement task. We first factorize the input low-light image into K factors (E_i) by following Eqs. (4.2) and (4.4). We post-process thus obtained layers by following three steps one after another:

1. **Layer grouping** *i.e.* based on layer signal energy merge with the next layer if lower than a set threshold ($\tau = 1\%$). This increases efficiency by simplifying the stack and removing very small factors.

2. **Outliers removal** *i.e.* clip the pixels above and below certain percentiles (99.9 and 0.1) to the corresponding cutoff values. This controls extremely high and low valued noisy pixels thereby cleaning each factor for processing.

3. **Luminance normalization** *i.e.* rescale the luminance of layers $\in [0, m_k]$ by normalizing the value in *HSV* color space. This helps in only rescaling of intensity without impacting the color information.

In the third step above, m_k is the signal energy in each factor which is estimated as the sum of all pixel values in that layer divided by sum of all m_ks *i.e.*:

$$m_k \quad = \quad \frac{\sum_x E_k}{\sum_k \sum_x E_k}. \tag{4.5}$$

This enhances contrast while preserving to the original order of luminosity values of various image regions.

After post-processing, we simulate our virtual exposure stack (S_i) by linearly combining the layers with I progressively:

$$S_{i+1} = (1 - \alpha)S_i + \alpha E_i, \qquad \textit{where} \quad i \in [0, K] \quad \textit{and} \quad S_0 = I. \tag{4.6}$$

Linear combination in Eq. (4.6) helps in the gradual introduction of information from the successive layers into the transient low-lit image. This avoids sudden illumination jumps which might generate unnaturally lit images. Furthermore, it simulates the effect of slowly increasing the exposure time (or reducing shutter speed), leading to progressively brighter images with varying exposure values as shown in Fig. 4.4.

4.3.3 Exposure Fusion

The fundamental task in exposure fusion is to assess the amount of information at each pixel in the stack and use it to merge all the images to render an enhanced image. It is a well understood problem with multiple solutions proposed in the current literature [41]. We build our framework by adapting multiple existing exposure fusion strategies which we discuss below:

4.3.3.1 Direct Fusion

As a simple baseline we process our last simulated image which contains the information from all the layers. Similar to Section 4.3.2, we first percentile clip the outliers and then normalize the luminance. In order to remove the image noise arising from the sensor errors in the low-lit regions and rescaled in the dark layers, we use Block-Matching and 3D collaborative filtering (BM3D) [205]. BM3D is a non-local transform based denoising method which first spatially (2D) and then spectrally (1D) transforms similar image patches which are then denoised in this 3D domain via shrinkage. After denoising, we rescale the image and use it as our directly fused enhanced result I_D.

4.3.3.2 Laplacian Pyramid Fusion

In their seminal work, Mertens et al. [199] introduced the concept of exposure fusion via a Laplacian pyramids from a bunch of Low Dynamic Range (LDR) to create a High Dynamic Range (HDR) output. They first estimate the image quality at each pixel using three metrics:

- Contrast $C_k(x)$
- Saturation $S_k(x)$
- Well exposed-ness $W_k(x)$

These terms respectively measure the spatial gradient magnitude, pixel color standard-deviation and Gaussian curve distance from the mid-tone value of 0.5 as provide them as a matrix of weights. For each image the combined weight matrix is constructed by simply multiplying the three quality maps as:

$$w_k(x) \quad = \quad C_k(x)^{\lambda_c} \,.\, S_k(x)^{\lambda_s} \,.\, W_k(x)^{\lambda_w}, \tag{4.7}$$

with various λ_js as tune-able hyper-parameters. The input images are then fused by merging their Laplacian pyramids weighed by the Gaussian pyramids of their respective normalized weights [45]. We ignore the contrast term as it is already handled by the Luminance normalization step during simulation in Section 4.3.2 and empirically fix the hyper-parameters values at $\lambda_c = 0, \lambda_s = 1$ and $\lambda_w = 2$. We denoise and rescale and denote the hence obtained enhanced result as I_L.

Methods →	**RetinexNet**	**DCENet**	**LPNet**	**UNIE**	$\mathbf{I_D}$ (Direct)	$\mathbf{I_L}$ (Laplacian)	$\mathbf{I_G}$ (GRWF)
Trainset →	Lolv1	SICE Part 1	Adobe5k	Lolv1	-	-	-
Lolv1 [283]	16.77 - 0.46	14.86 - 0.59	15.30 - 0.56	**21.52** - 0.76	20.39 - **0.77**	19.28 - 0.75	17.72 - 0.70
Lolv2 [300]	15.47 - 0.56	20.54 - 0.78	16.38 - 0.53	**25.53** - **0.88**	19.12 - 0.67	18.16 - 0.67	19.01 - 0.69
SICE Part2 [49]	15.99 - 0.53	16.57 - 0.59	14.55 - 0.50	13.72 - 0.46	16.82 - **0.62**	**17.75** - 0.60	15.64 - 0.56
Adobe5k UnderExp.	11.06 - 0.60	11.02 - 0.52	**19.35** - **0.74**	16.93 - 0.66	17.90 - 0.71	15.60 - 0.65	18.71 - 0.71
Adobe5k OverExp.	12.49 - 0.62	14.96 - 0.59	19.69 - **0.74**	15.64 - 0.60	**19.87** - 0.72	17.94 - 0.69	19.34 - 0.70
Adobe5k Full [8]	11.63 - 0.61	12.60 - 0.55	**19.48** - **0.74**	16.41 - 0.64	18.69 - 0.71	16.78 - 0.67	18.96 - 0.71
Average (S_a)	12.19 - 0.60	13.17 - 0.56	**18.87** - **0.71**	16.26 - 0.62	18.49 - 0.70	16.92 - 0.66	18.58 - 0.69
Generalizability ($\mathbf{S_g}$)	12.18 - 0.60	12.74 - 0.55	14.77 - 0.51	16.42 - 0.62	18.49 - **0.70**	16.92 - 0.66	**18.58** - 0.69

Table 4.1: **Quantitative comparison:** We evaluate our simulated exposure stack generation scheme over 5 datasets using results from our 3 exposure fusion strategies (I_D, I_L, I_G). Each tuple represents PSNR-SSIM scores (higher is better). S_a is average score weighted by test-set size and S_g is method's *Generalizability* score computed as weighted average leaving out the test-set corresponding to its supervision dataset. Best score is in **boldfaced** and second best is underlined.

4.3.3.3 Generalized Random Walk Fusion

The third strategy that we experiment with is the Generalized Random Walk Fusion method by Shen et al. [254] which we used previously in Chapter 2. The authors' in Shen et al. [255] fuse multiple LDR images to construct a HDR image which is then tone mapped to produce the enhanced result. They perform a global optimization using local contrast and color smoothness as two quality measures. They setup a Dirichlet problem [67] per stack image and solve them via global Generalized Random Walk algorithm [100]. They estimate dense probability maps representing significance of each pixel in the stack towards generating a combined enhanced image. We apply their algorithm on our simulated exposure stack and label our enhanced result as I_G, post denoising and re-scaling.

4.4 Experiments and Results

4.4.1 Implementation Details

We implement our framework using the *Matlab Quaternion Toolbox* [238]. Codes for various fusion strategies and BM3D denoising were adapted from their respective official releases [199, 254, 205]. We show the utility of our framework by comparing with the previous state-of-the-art single

image low-light enhancement methods. We restrict ourselves to the solutions which directly work on $sRGB$ images and do not require RAW inputs. Specifically we compare with RetinexNet by Wei et al. [283], DCENet by Guo et al. [105], LPNet by Afifi et al. [9] and UNIE by Yan et al. [294]. RetinexNet and LPNet are trained via direct supervision on LOL dataset [283] and simulated exposure stack from Adobe5k dataset [47] respectively. Whereas, DCENet and UNIE follow zero-shot and unpaired unsupervised learning adapted on SICE [49] and LOL [283] datasets respectively. We use the code and pretrained weights as shared by the authors' and reference the quantitative scores as reported in the literature.

4.4.2 Quantitative Results

We show both quantitative (PSNR-SSIM scores Q_i in Table 4.1) and qualitative results (Fig. 4.6) on four datasets ($\mathcal{D}_i$) comprising of varying number of test images: LOLv1 testset (15) [283], LOLv2 testset (100) [299], SICE Part 2 testset (767) [49] as described by Guo et al. [105] and Adobe5k testset (5905 images) [9] . Adobe5k contains both over and under exposed images and provides 5 expert ground truth annotations. Afifi et al. [9] report both separate and combined results on these two categories for all the annotations. We follow the same procedure and show our overexposed, underexposed and combined scores averaged over all the expert annotations (for individual expert ground truth evaluations refer to the supplementary material).

For overall performance, we report average scores (S_a) weighted by test-set cardinality *i.e.*

$$S_a = \frac{\Sigma_i(|\mathcal{D}_i|Q_i)}{\Sigma_i|\mathcal{D}_i|},$$

thus assigning equal importance to all test images across datasets. Furthermore in order to gauge the cross-dataset generalizability of learning based methods, we also calculate the average scores by leaving out the test images of the dataset on which the respective model was trained (*e.g.* leaving out LOLv1 test-set for RetinexNet and computing weighted mean over the rest). We report this metric as the *Generalizability Index* (S_g) of the method and list them in the last column. As our method is a non-data driven optimization algorithm, last two columns are same in our case *i.e.* S_a = S_g. We provide evaluation scores for all three of our fusion strategies in the three last rows (I_D, I_L and I_G).

As can be observed from Table 4.1, our results are frequently ranked best or second best on multiple datasets even without having learned any data-driven prior via training. As expected, all four learning based methods perform well on the datasets which are similar to their respective training sets but suffer significant performance degradation when dataset domain shifts. The effect pronounced in the case of smaller models like RetinexNet and DCENet (note that datasets LOLv1 and LOLv2 are similar than the rest). In overall performance S_a our method ranks second just behind LPNet [9]. Note that LPNet was trained on simulated images from the raw data in Adobe5k. Hence their performance on

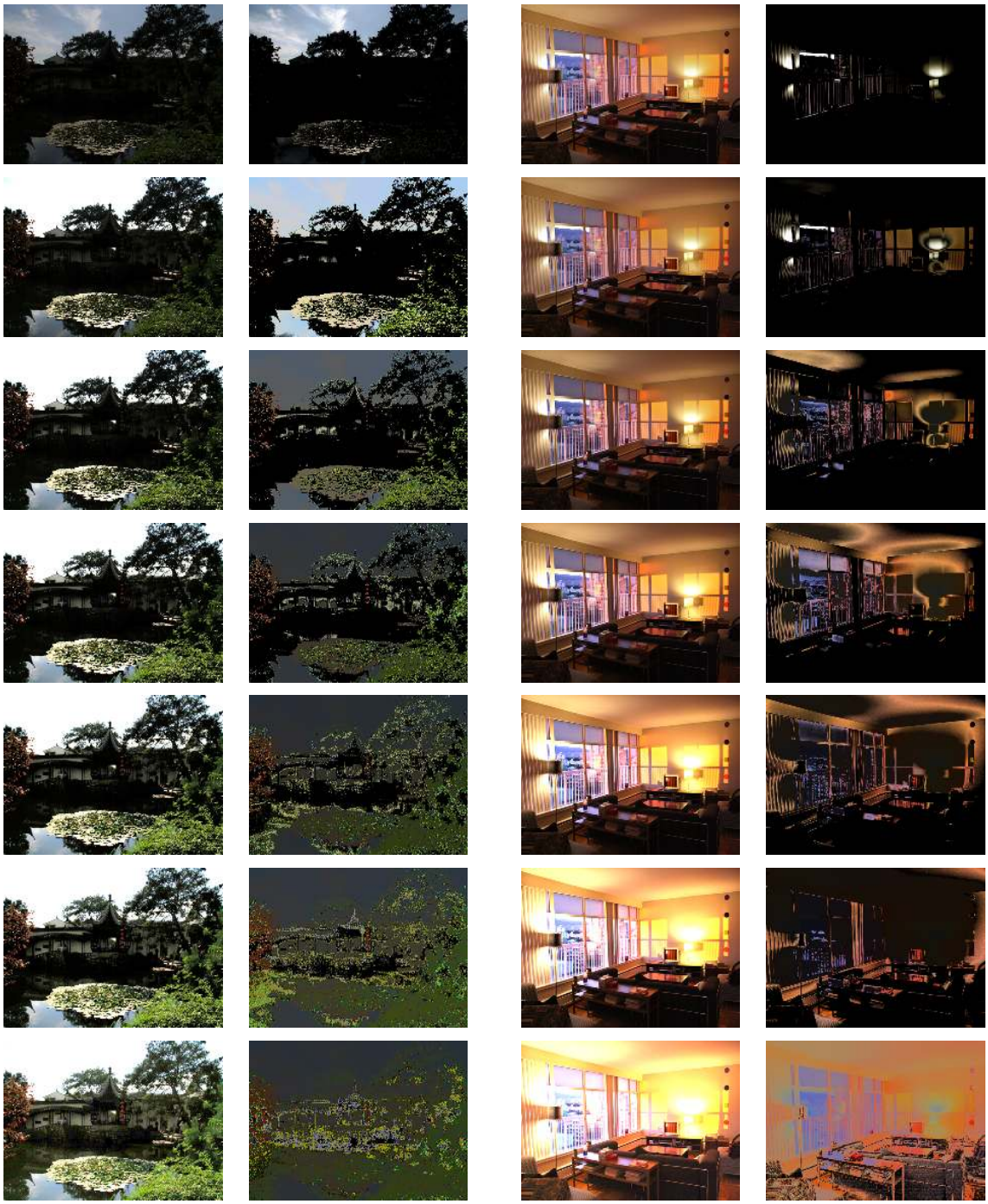


Figure 4.4: Stack Simulation: This figure shows Simulated Exposure Stack and the underlying quaternion RPCA decomposed factors for two types of scenes: outdoor naturally lit scene with single light source (left) and indoors artificially lit room with multiple illuminants (right).

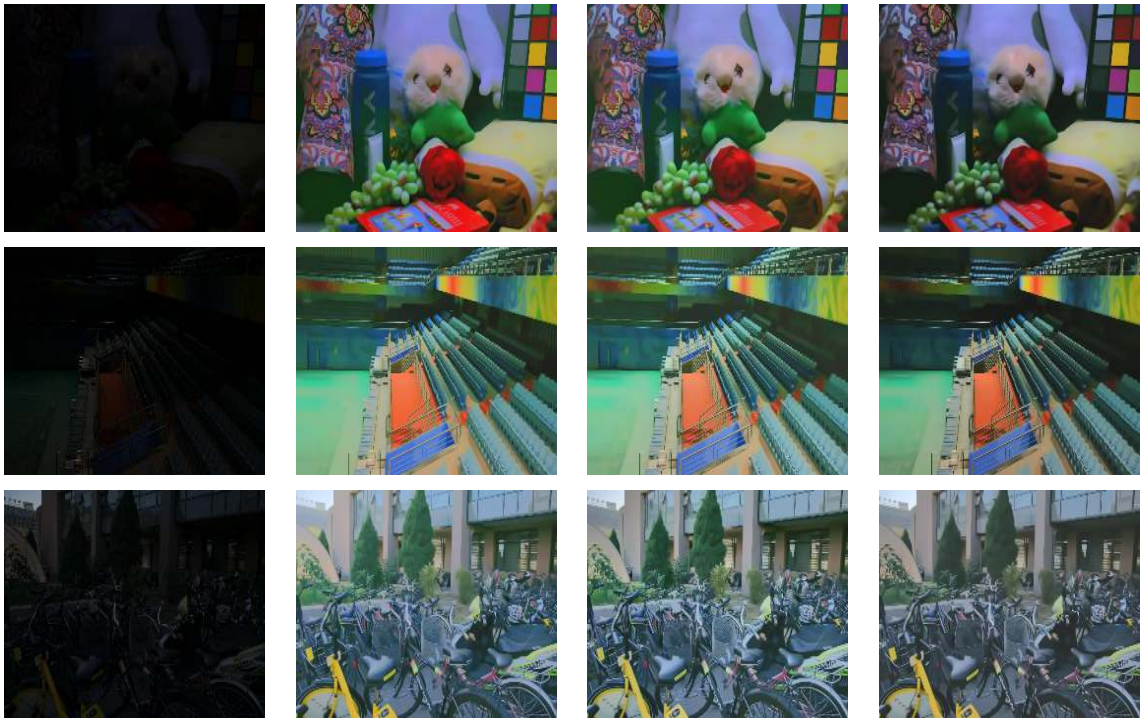


Figure 4.5: Our QFSEF results visualization: Our low light image enhancement results on three scenes using our three exposure fusion strategies (Section 4.3.3). For each scene we show our three results for input image I (left) sequentially as: Direct fusion I_D, Laplacian Pyramid Fusion I_L and Generalized Random Walk fusion I_G. Test images taken from LOL dataset [283, 299].

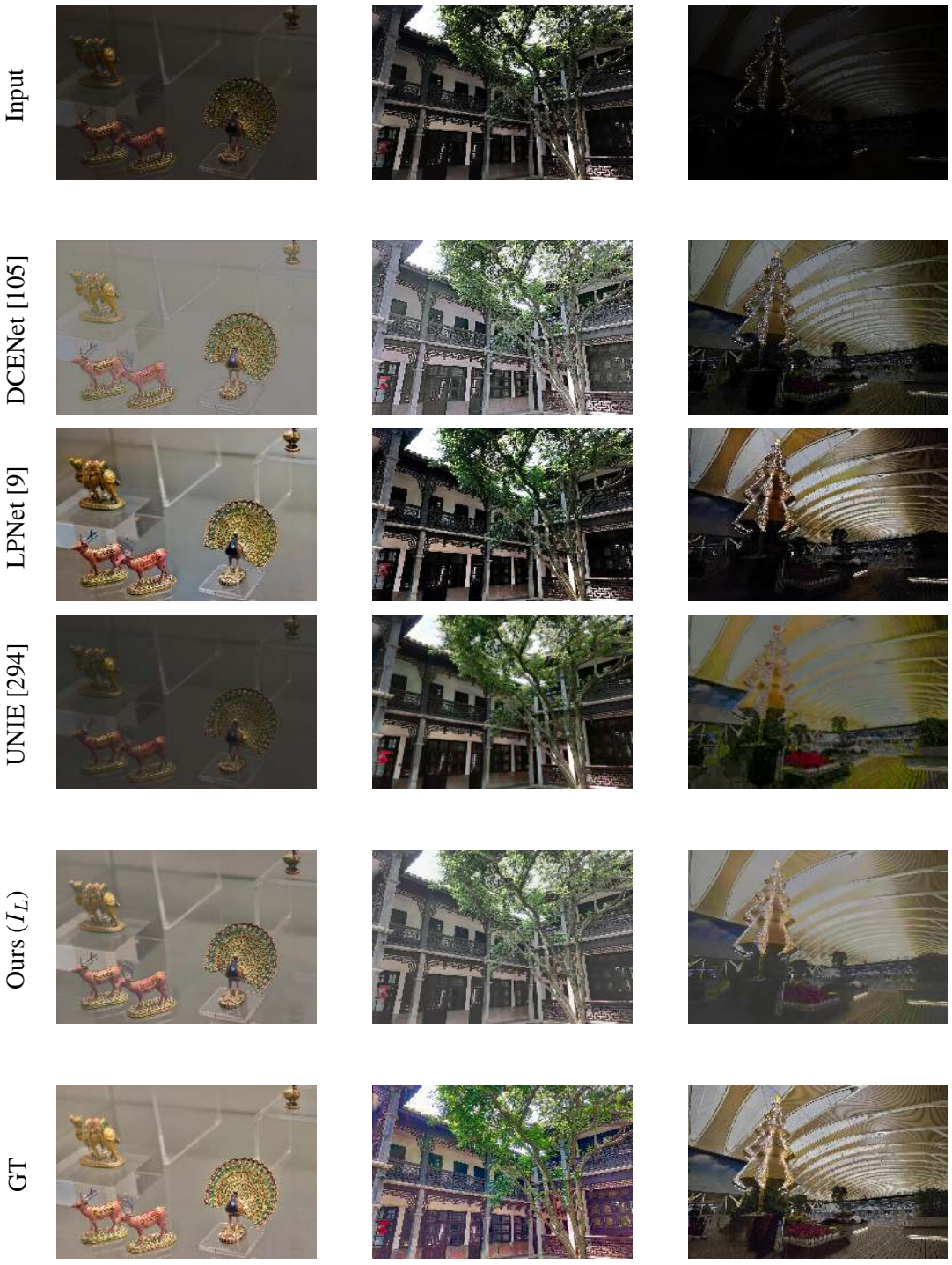


Figure 4.6: Qualitative Comparison: LLE results comparison between DCENet [105], LPNet [9], UNIE [294] and our method I_L. DCENet [105] leads to de-saturated colors while LPNet [9] and UNIE [294] fail to properly illuminate some regions. Our method achieves good enhancement without significant color degradation.

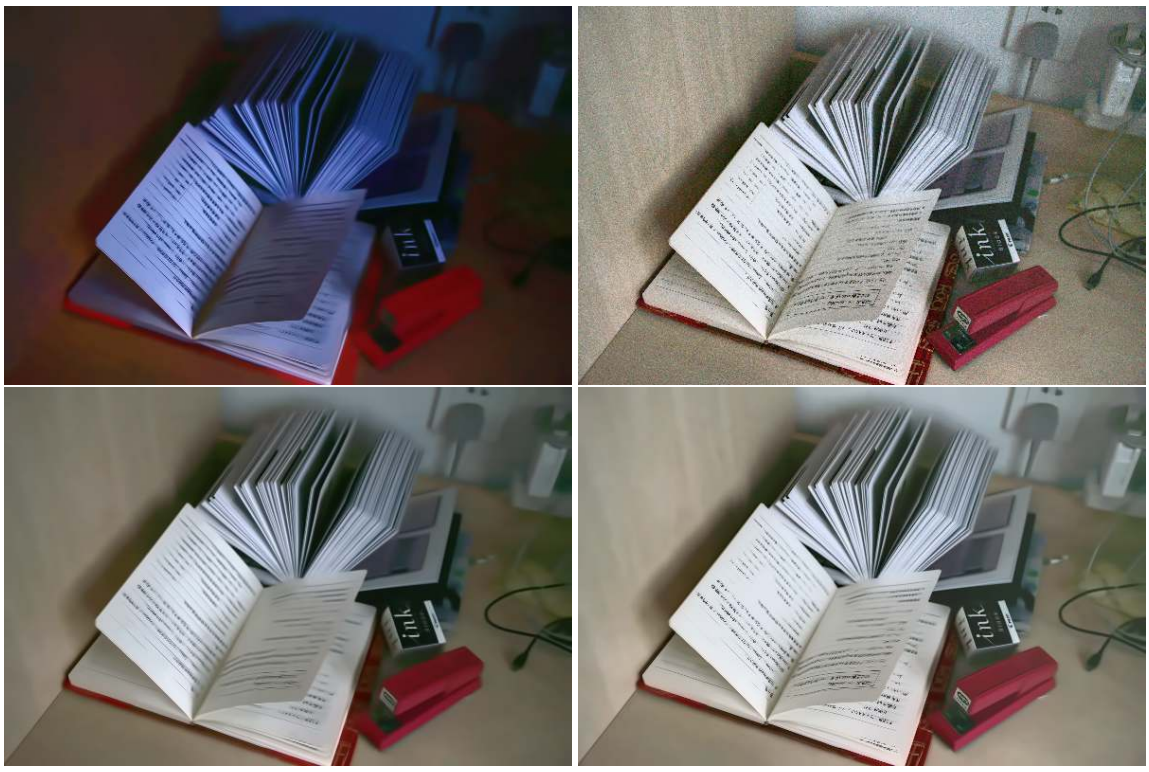


Figure 4.7: Ablation: Illustrative low light enhancement results for our four variants v_1, v_2 (top left and right) and v_3, v_4 (bottom left and right) in two rows respectively. Variant v_1 optimizes in real space and hence does poor color preservation. Without denoising, v_2 contains salt-pepper noise. Note slight over smoothing of the text in the notebook and dark corners in v_3 results compared to our final design choices in v_4.

the corresponding test-set is also higher which contributes a high weight to the overall score S_a. In order to ameliorate this, we use the Generalizability Index score S_g by leaving out significance of respective training dataset test images. Under this metric all of our strategies perform better than the current state-of-the-art solutions indicating our wider generalizability and utility.

4.4.3 Qualitative Results:

The first step of our method *i.e*. Quaternion RPCA factorization, is not limited to dimly lit scenes and can be applied on a variety of images as shown in Fig. 4.4. We show our resultant simulated exposure stack from a single under-, well- and over- exposed image input. In Fig. 4.5 we show three sample enhancement results for a low light image using all three of our strategies (I_D, I_L and I_G). Out of these three, I_D is the fastest but does not integrate contrast from the initial layers quite well. On the other hand, I_G is the slowest but has better contrast. I_L though has relatively lesser quantitative score

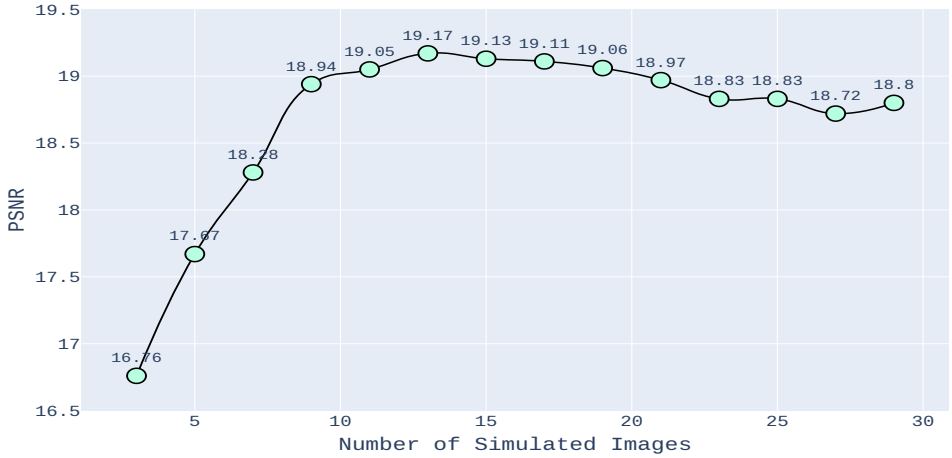


Figure 4.8: Ablation on k parameter: The graph above shows the effect of different number of simulated images on the mean PSNR of the three strategies. After swift increase initially, performance plateaus after $k = 10$ with a slight degradation towards the end. We choose the middle plateau value of $k = 15$ for all our experiments.

but strikes a balance between both the scenarios. Any of these strategies can be used for the task based on user preferences. We also gauge the perceptual quality of our results (I_L) by comparing against previous state-of-the-art in Fig. 4.6. It can be observed that DCENet [105] brightens the image but dulls the color information and the resultant images are grainy. LPNet [9] leaves some dark shadows and over saturates colors in some regions. UNIE [294] results have good color but are blurred and dark. They are also inconsistent depending on the scene type. Our method achieves balanced enhanced results with good contrast and are reported in the last column. This effect is particularly pronounced in larger datasets (SICE and Adobe5k) where learning based methods suffer significant performance degradation highlighting method's limitations in generalizing to out of domain cases.

4.4.4 Ablation Analysis

We validate our design choices by performing two different ablation analyses on LOLv1 test-set images. First we compare four different variants of our system. Variant v_1 is performed by swapping quaternion RPCA with real RPCA on the color channels concatenated representation of the image. Variant v_2 is designed to signify the impact of denoising and is formed by skipping the CMB3D denoising step in the original pipeline. v_3 is very similar to v_4 only with luminance normalization step replaced with normal range normalization whereas v_4 represents our complete system. As seen in Table 4.2 and Fig. 4.7, v_1 fails to capture the color information properly, whereas v_2 contains amplified dark channel noise prevalent in the low-light images. Variant v_3 performs quite well quantitatively but

Variants $\longrightarrow$	real RPCA v_1	w/o Denoise v_2	w/o LNorm. v_3	Full v_4
I_D	12.83 - 0.5	18.35 - 0.6	20.86 - 0.75	20.39 - 0.77
I_L	14.54 - 0.56	18.14 - 0.59	20.11 - 0.76	19.28 - 0.75
I_G	12.66 - 0.48	15.83 - 0.53	17.48 - 0.69	17.72 - 0.70

Table 4.2: **Ablation using system variants:** We show PSNR-SSIM (↑) scores on 15 LOLv1 [283] test-set images for our four system variants *i.e.* using real RPCA (v_1), without denoising (v_2), without luminance normalization (v_3) and our complete version (v_4). The performance gradually improves for each step empirically validating our design choices. Note that although we achieve higher scores with v_3 but it leads to over-smoothing of edges especially in high frequency regions (see Fig. 4.7).

does over smoothing in high frequency parts in certain scenes. Overall v_4 presents a balanced approach with good numerical and perceptual accuracy.

Our second ablation experiment is done to analyze the effect of number of simulated images K. We gradually increase the value of K $(3, 5, 7, ...29)$ and observe the mean PSNR scores computed on our ablation set. As can be observed from Fig. 4.8, the scores after increasing drastically for first few values, plateaus in the range $K = [10\text{-}15]$. This happens because the fewer number of factors do not represent illumination consist regions in the image properly and hence fail to properly enhance the poorly lit pixel values. On the other hand higher values of K lead to several dimly lit factors which contain sensor noise, thus degrading the quality of stack input in the exposure fusion step and thereby reducing the quality of generated results. To enable better user control over the scene, we fix $K = 15$ in all our experiments.

4.5 Conclusion and Future Directions

To summarize, in this work we have presented a novel exposure stack simulation method and applied it to the task of low-light image enhancement. We extend the Retinex theory by proposing a Robust Retinex Decomposition formulation. To this end we present a new way of image factorization by first representing the image pixels in the quaternion space as a pure quaternion vector and then apply Robust Principal Component Analysis to obtain sparse factors representing the specular information in the image. We propose a scheme to simulate a virtual exposure stack from a single image by iterative

factorization and adapt existing exposure fusion strategies to generate an enhanced well-lit image. Our results show good qualitative and quantitative performance on multiple datasets, especially exhibiting marked generalization performance even when compared with the contemporary state-of-the-art deep learning based solutions for this task.

In the end, we would like to point out the our proposed factorization technique can assist several related image based rendering problems beyond the low light image enhancement task like selective relighting, shadow removal, white balancing, object compositing, image harmonization *etc*. Furthermore, we can extend our method by assembling an end-to-end deep model for the three sub-modules *i.e*. factorization, simulation and fusion. In future we would like to pursue these directions.

Chapter 5

Recursive Specular Illumination Analysis

In this chapter we continue our discussion on image factorization for the purpose of image illumination analysis. Unlike previous chapter (Chapter 4) where we proposed a Robust Principal Component Illumination Analysis and used a complex low-rank *vs*. sparse matrix decomposition in quaternion space as our core technique, here we simplify the optimization significantly by only focusing on the matrix sparsity ignoring the reflectance low-rank assumption. This significantly simplifies the optimization as the expensive singular value decomposition step required for low-rank optimization is replaced by a simpler shrinkage operator. We show how this approximation allows porting the technique to the real number space and hence can be learned via unrolling into a normal neural network for a seamless integration with the current deep learning based fusion frameworks. Here too we mainly experiment with the single image Low Light Enhancement (LLE) problem but also provide evidences for the utility of our technique to other image enhancement tasks.

LLE is an important step to enhance images captured with insufficient light. Several local and global methods have been proposed over the years for this problem. Decomposing the image into multiple factors using an appropriate property is the first step in many LLE methods. In this chapter, we present a new additive factorization that treats images to be composed of multiple latent specular components that can be estimated by modulating the sparsity during decomposition. We propose a model-driven learnable RSFNet framework to estimate these factors by unrolling the optimization into network layers. The factors are interpretable by design and can be manipulated directly for different tasks. We train our LLE system in a *zero-reference* manner without the need for any paired or unpaired supervision. Our system improves the state-of-the-art performance on standard benchmarks and achieves better generalization on multiple other datasets. The specularity factors can supplement other task specific fusion networks by inducing prior information for enhancement tasks like de-raining, deblurring and de-hazing with negligible overhead as shown in this chapter.

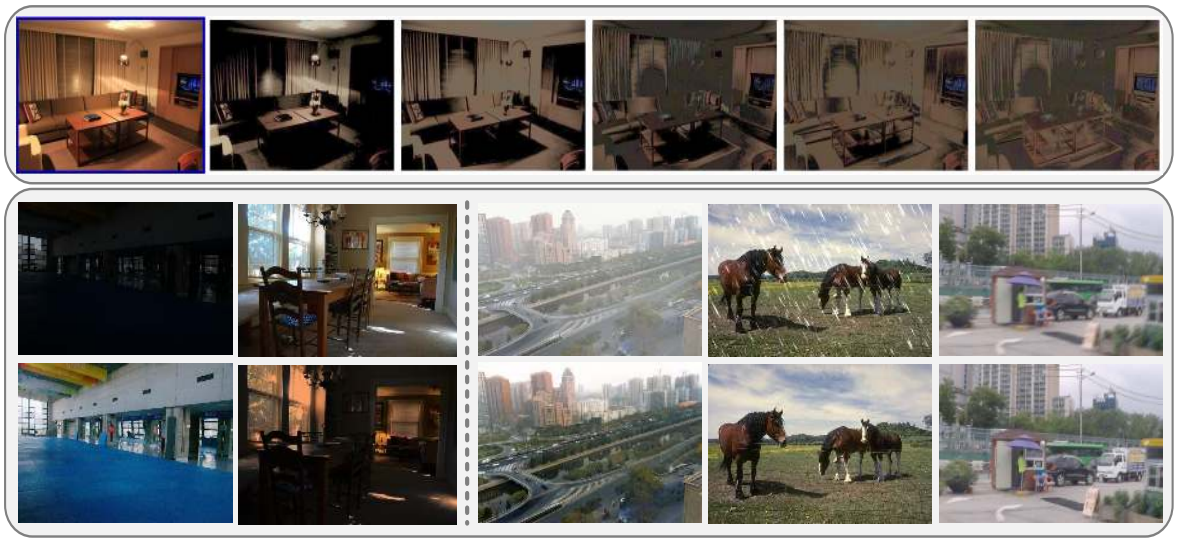


Figure 5.1: Specularity Factorization: We factorize a single input image (blue box, top row) into multiple *soft* specular factors (rescaled for visualization) based on their similar illumination characteristics (note table shadow and lamp reflection). Our factors directly enable zero-reference low-light enhancement and user controlled image relighting (bottom left). Additionally, they can also be used as a plug-and-play prior for various supervised image enhancement tasks like de-hazing, de-raining and de-blurring.

5.1 Introduction

A low-light image has most regions too dark for comprehension due to low exposure setting or insufficient scene lighting which makes images highly challenging for computer processing and aesthetically unpleasant. Low-Light Enhancement (LLE) aims to recovers a well-exposed image from a low-light input [165]. LLE can be a critical pre-processing step before the downstream applications [184, 191]. Core LLE challenge lies in modeling the degradation function which is spatially varying and has complex dependence on multiple variables like color, camera sensitivity, illuminant spectra, scene geometry, *etc*.

Most LLE solutions decompose the image into meaningful latent factors based on a relevant optical property (Table 5.1). This allows individual manipulation of each factor which simplifies the enhancement operation. A common factorization is based on the Retinex approximation [153, 198], which assumes a multiplicative disentanglement of image I into two intrinsic factors: illumination-invariant, piece-wise constant *reflectance* R and color-invariant, smooth *shading* S as $I = R \cdot S$. Other factorization criteria include frequency [289, 126], spatial scale [8, 175], spatio-frequency representation

[76, 221], intensity [117], reflectance rank [225, 234], *etc*. Fixed number of factors [283, 126, 225] and variable number that allow better representation [117, 8, 175] have been used. Some decompose image multiplicatively like Retinex [283, 126], while others split into additive factors which are numerically more stable [116, 221, 82]. Pixel segmentation could be soft or hard based on the membership across factors, with the former introducing fewer artifacts [14]. LLE solutions can be *global* or *local*. Global methods use whole image level statistics like gamma [105], histograms [306], *etc*., to enhance the images. Local methods employ spatially varying features like illumination maps [283], intensity/segmentation masks [117, 213], *etc*., for the same. Global methods are simpler but local ones can capture scene semantics better.

Traditional LLE methods used manually-designed model-based optimisation by deriving specific priors from the image itself [109, 85, 310], needing no training. Data-driven, machine learning based solutions have done better recently. They use training datasets to tune the model which generalizes to other images [291, 8, 283]. *Supervised* methods require annotated input-output pairs of images [314, 286, 291]. *Unsupervised* methods require annotated training data but not necessarily paired [131, 294]. *Zero-reference* methods do not need annotated data and approach the problem by explicitly encoding the domain knowledge from training images [105, 229, 191]. They generalize better and are simpler, lighter, and more interpretable.

In this chapter, we present a zero-reference LLE method that outperforms prior zero-reference methods on the average. The core of our method is a novel image factorization strategy based on image specularity. We decompose an image into additive specular factors by thresholding the amount of sparsity of each pixel recursively. Successive factor differences mark out newly discovered image regions which are then individually targeted for enhancement. The factorization is model-driven, task-agnostic, and light-weight, needing very few (< 200) trainable parameters. The image factors are fused using a task-specific UNet-based module to enhance each region appropriately. We call our method *Recursive Specularity Factorization Network (RSFNet)*. Our factorization is useful to other applications when combined with other task-specific fusion modules.

Overall, our major contributions in this work are:

- A novel image factorization criterion and optimization formulation based on recursive specularity estimation.
- A model-driven framework to learn factorization thresholds in a data-driven fashion using algorithm unrolling.
- A simple and flexible zero-reference LLE solution that surpasses the state of the art on multiple benchmarks and in average generalization performance.
- Extension to other enhancement tasks like de-hazing, de-raining and de-blurring showing the power of our specularity factorization as a prior.

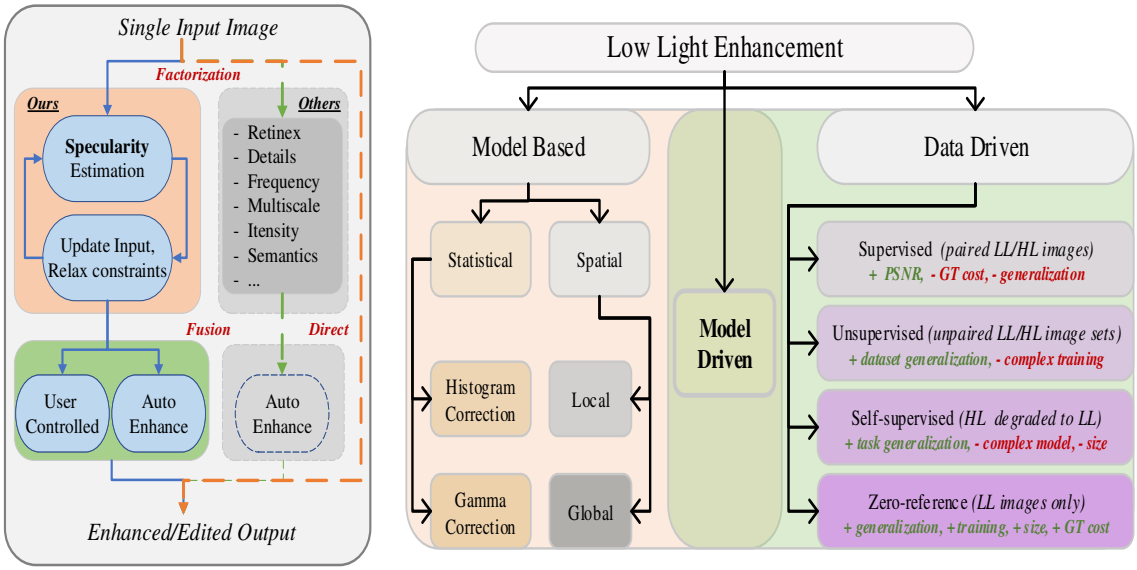


Figure 5.2: Concept Diagrams: In this figure we highlight various LLE solution types conceptually. We show how many solutions have a common factorization and fusion architecture with various strategies used for factorization. Also these methods can be categorized based on the learning paradigm each of which have associated advantages and disadvantages as listed underneath.

5.2 Background

5.2.1 Low Light Enhancement

In this section we discuss various LLE solutions, their categorization, their internal learning paradigms and corresponding factorization strategies adopted for them. Conceptually we can understand LLE background in two ways:

- The overall pipeline followed *i.e.* factorization-fusion scheme or direct output regression.
- Basedon the type of optimization carried out *i.e.* model-based, data-driven or hybrid model-driven.

Both of these categorizations are illustrated in Fig. 5.2.

5.2.1.1 Model-based LLE:

As discussed earlier, early LLE solutions used traditional optimization models using either global statistics [219, 226, 54, 159] or spatially varying illumination maps for local editing [109, 277, 83,

309]. They were more interpretable but required hand-crafted algorithms and heuristics. This makes them easily generalizable but less efficient as a new optimization problem needs to be solved for every input image separately. In spite of this issue, such traditional model-based methods encode interesting domain insights in the form of priors or optimization constraints which are valuable for subsequent learning based methods. For a more detailed discussion please refer Section 4.2 or survey by Li et al. [165].

5.2.1.2 Data-driven LLE:

Modern solutions take inspiration from traditional techniques and induce domain knowledge via loss terms or designed within the network architecture which are learned from large datasets in a data-driven fashion. They belong to one of the five training paradigms [165]. *Supervised* LLE methods require both low-light and well-lit *paired images* like Wei et al. [283], Zhang et al. [315], Yang et al. [300], Sharma and Tan [250], Xu et al. [291]. On the other hand, *unsupervised* methods like Jiang et al. [131], Ni et al. [214], Zhang et al. [306], require only unpaired low-light and well-lit *image sets*. *Semi-supervised* methods combine the previous two techniques and use both paired and unpaired annotations *e.g*. Yang et al. [298], Robert et al. [230]. The other two paradigms do not use any external annotated data for supervision. *Self-supervised* solutions like Liang et al. [173], Nguyen et al. [213] generate their own annotations using pseudo-labels or synthetic degradation. Contrary to all of these, *zero-reference* methods do not use ground truth reconstruction losses and assess the quality of output based upon encoded prior terms like Guo et al. [105], Li et al. [166], Risheng et al. [229], Ma et al. [191], Zhang et al. [308], Zhu et al. [323]. These methods, like ours, possess improved generalizability due to explicit induction of domain knowledge and reduced chances of over-fitting [105]. Zero-reference insights also provide direct valuable additions to the subsequent solutions in other paradigms.

Specifically, RetinexNet by Wei et al. [283] follows simple decomposition of reflectance and shading from input before rescaling and denoising. This was improved upon by Zhang et al. [314, 315] by focusing on dark regions. Later, other works like Yang et al. [300], Sharma and Tan [250], Lv et al. [189], Xu et al. [291] provided additional performance gains by regularizing gradients, suppressing glares, using spatial attention and noise maps respectively. In contrast, very few unsupervised LLE solutions exist that use *unpaired* low-light and well-lit image sets [131, 306, 294] Light-weight zero-shot solution was proposed to directly predict intensity curve correction parameters which are iteratively applied for enhancement [105, 166]. Our proposed method here belongs to the zero-reference category of solutions. Please note that zero-reference does not necessarily imply zero-shot supervision. The main difference is that zero-shot methods do not require any label information for a new class of inputs and the terminology is generally used in a classification setting. Contrary to this zero-reference methods do not require any annotated ground truth and use several images during training. Such so-

lutions derive the required supervision information directly from the input images using the encoded domain and task priors.

5.2.1.3 Unsupervised *vs*. Zero-reference LLE:

Although similar, there is a crucial difference between the unsupervised and zero-reference LLE paradigms [165]. As mentioned previously, unsupervised LLE solutions like Jiang et al. [131], Zhang et al. [306], Yan et al. [294], Liang et al. [174], Yang et al. [296], Fu et al. [86] require both poorly lit and well illuminated image sets for supervision though they need not be paired. On the other hand, zero-reference LLE solutions Zhang et al. [308], Guo et al. [105], Li et al. [166], Risheng et al. [229], Ma et al. [191], Nguyen et al. [213], Fei et al. [79] do not need any well-lit examples for training and purely use domain/task dependent loss terms and models for enhancement. In addition to making the methods more inexpensive, this also allows for better generalizability due to low domain dependence. Furthermore, due to explicitly encoded expert knowledge as domain priors, zero-reference solutions are smaller in size with simpler architectures and training curriculums than their unsupervised counterparts. This enables easy adoption of such techniques to other tasks beyond LLE. Although fair comparison is possible only between the methods of the same paradigm [79], still we report our comparison with various unsupervised solutions in later sections. Note that our method beats several unsupervised LLE solutions and is competitive against the best two unsupervised solutions: Yang et al. [296] and Zhang et al. [306]. Yang et al. [296] uses a complicated architecture comprising of pretrained multi-modal Large Language Models, multiple generator-discriminator pairs, implicit neural representation, collaborative mask attention modules *etc*. Relative to ours, this is significantly complex training process without direct interpretability/utility of intermediate results or possible extension to other enhancement tasks. In our method, we have focused on encoding the fundamental aspects of the image formation process and represented it as in a recursive specularity factorization model. Still our method surpasses [296] on 4 out of 6 and [306] on 5 out of the 6 reported metrics individually.

5.2.2 Model-Driven Networks

Data-driven solutions have good performance but lack interpretability, whereas model-based methods are explainable by design but often compromise with lower performance. Model-driven networks [204] are hybrids which bring the best of both together. Such networks *unroll* optimization steps as differentiable layers with learnable parameters, inducing data-driven priors in place of hand-crafted heuristics. Although data-driven solutions are plenty, only a few model-driven solutions exist for low-level vision tasks like image restoration [156, 157], shadow removal [324], dehazing [182], deraining

Criteria	No.	Type	Map	Seg	Example
Retinex	2	$*$	global	soft	[286, 229]
Frequency	2	$+$, low/high	global	hard	[289]
Spectral	2	$*$, fourier	global	soft	[126]
Low Rank	2	$+$	global	soft	[225, 234]
Wavelets	2^n	$+$, pyramid	global	soft	[221, 76]
Multiscale	2^n	$+$, pyramid	global	soft	[8, 175]
Glare/Shadow	$3, 4$	$*, +$	local	hard	[30, 250]
Intensity	var.	$+$, bands	local	hard	[117, 116]
Specularity	var.	$+$	local	soft	RSFNet

Table 5.1: **Image Factorization:** Various LLE factorization criteria, with number of components (var. implies variable), type of factorization ($+$ additive/$*$ multiplicative), types of output maps (local/global), pixel segmentation across maps (soft/hard) and corresponding examplar methods. Our RSFNet proposes a novel specularity based factorization which allows a variable number of local soft-segmented factors.

[274], denoising [224] and super-resolution [35, 34]. Such solutions are concise and efficient due to the underlying task specific analytical formulation.

Model-driven LLE solutions are very recent. UretinexNet [286] and UTVNet [318] are both supervised methods which respectively unroll the Retinex and total variational LLE formulations. RUAS [229] and SCI [191] are closest to our approach as they both propose model-driven zero-reference LLE solutions. RUAS unrolls illumination estimation and noise removal steps in their optimization and compliment it with learnable architecture search, towards a dynamic LLE framework. SCI on the other hand propose a residual framework wherein reflectance estimation is done by a self-calibration module which is then used to iteratively refine illumination maps. In contrast, our method is inspired directly by image formation fundamentals and presents a novel factorization criterion which provides better interpretability, performance and flexibility.

5.2.3 Image Factorization for LLE

5.2.3.1 Retinex:

Using Retinex reflectance and shading decomposition, degraded input image I^d and corresponding enhanced image I^e can be disentangled into reflectance and shading components as $\mathbf{I}^e = \mathbf{I}^e_{\mathrm{R}} \cdot \mathbf{I}^e_{\mathrm{S}}$ such that:

$$\mathbf{I}^e_{\mathrm{R}} = \mathcal{H}^{-1}_{\mathrm{R}}(\mathbf{I}^d_{\mathrm{R}} - \mathbf{n}_{\mathrm{R}}) \quad \text{and} \quad \mathbf{I}^e_{\mathrm{S}} = \mathcal{H}^{-1}_{\mathrm{S}}(\mathbf{I}^d_{\mathrm{S}} - \mathbf{n}_{\mathrm{S}}), \tag{5.1}$$

where reflectance degradation operator $\mathcal{H}^{-1}_{\mathrm{R}}$ is identity, and $\mathbf{I}^e_{\mathrm{R}} = \mathbf{I}^d_{\mathrm{R}} = \mathbf{I}_{\mathrm{R}}$ due to illuminance-invariance and shading noise $\mathbf{n}_{\mathrm{S}}$ is negligible due to the smoothness assumption. Hence Eq. (5.1) can be simplified and restated as:

$$\mathbf{I}_{\mathrm{R}} = \mathbf{I}^e_{\mathrm{R}} = \mathcal{D}_{\mathrm{R}}(\mathbf{I}^d) \quad \text{and} \qquad \mathbf{I}^e_{\mathrm{S}} = \mathcal{D}_{\mathrm{S}}(\mathbf{I}^d), \tag{5.2}$$

where $\mathcal{D}$ is the combined estimation and denoising function. Note that this allows us to represent 'illumination map' I^e_S directly as a function of the input image. Retinex is the most widely used LLE factorization strategy [165]. One major limitation here is due to the Lambertian reflection [152] approximation which assumes all surfaces are purely diffuse, thereby ignoring prevalent non-Lambertian effects in a real scene like specularity, translucency, caustics *etc*. Another issue is that pixel-wise multiplicative nature of Retinex factors is cumbersome to handle numerically (especially in LLE with near zero pixel values) and the obtained illumination maps require further semantic analysis for downstream applications. Extensions of Retinex like dichromatic model [267] and shadow segmentation, separate one extra component each in addition to diffuse R and S *e.g*. Sharma and Tan [250] and Baslamisli et al. [30] used glare and shadow image decomposition respectively. From this perspective our recursive specular factorization can be understood as an extension of the same idea with continuously varying illumination characteristics starting from bright glares and ending with dark shadows (see Fig. 5.5 and Section 5.3 for details).

5.2.3.2 Others Factorization Strategies:

Apart from Retinex, other factorization techniques are listed in Table 5.1. As mentioned in Sections 5.1 and 5.2 and shown in Table 5.1, various LLE solutions adopt different factorization strategies. We have provided a non-exhaustive list in the Table 5.1 but still others are possible. The *Frequency* strategy [289] here refers to the low and high pass filtering of the input to extract coarse and fine image details, which are then processed separately. On the other hand, *spectral* strategy [126] refers to decomposition into phase and amplitude using Fourier representation where phase is assumed to encode the entire structural information of the scene. *Low rank* strategy based methods specifically exploit low rank structure of the reflectance component of the scene and are hence somewhat related

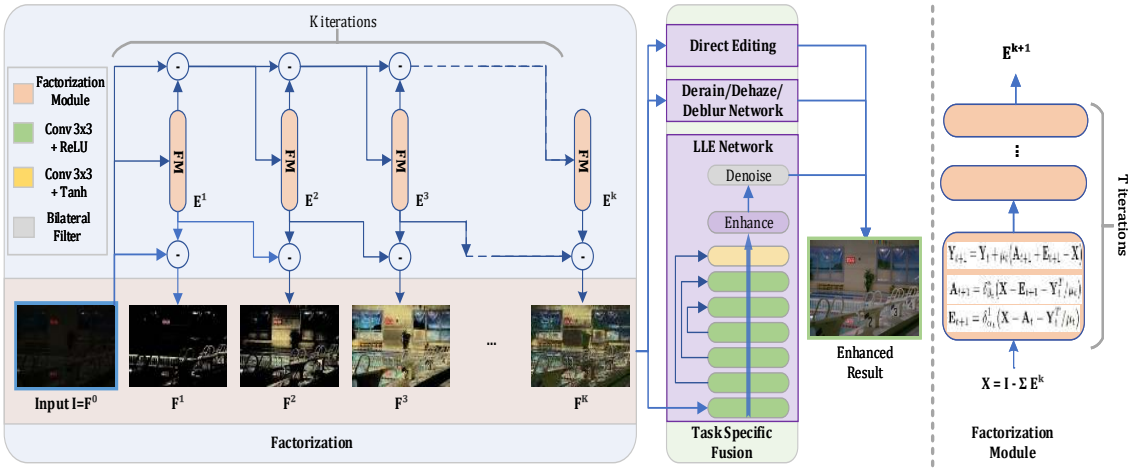


Figure 5.3: RSFNet Block Diagram: Overview of our proposed RSFNet. Factorization module splits a given image into multiple specular components using model-driven unrolled optimization steps. Fusion module combines all the factors to generate the enhance output.

to the Retinex division. [225] focuses on hyper-spectral images, whereas [234] uses a complicated quaternion based robust PCA optimization strategy [48] with no unrolled learning or generalization to other applications. *Wavelets* and *Multiscale* decompositions [76, 8] build factors like image pyramids and can be considered to be an extension of the *frequency* strategy. Decomposing input into extra glare or a shadow component [30, 250] along with the Retinex factorization has yielded better results and our method can be understood as the extreme case of such divisions. Similarities and differences with the often used *intensity* based factorization strategy [117, 116] has already been discussed previously. Note that the global/local categorization here refers to whether the factors and the subsequent processing is limited to local image regions. Mertens et al. [200], Xu et al. [289], Afifi et al. [8], Lim and Kim [175] employ spatial or frequency based image decomposition. Recently, Yang et al. [298] used recursively concatenated features from a supervised encoder and Huang et al. [126] proposed a Fourier disentanglement based solution. Apart from these supervised factorizations, Zheng and Gupta [320] proposed semantic classification based ROI identification using a pretrained segmentation network. [105, 213] predict multiple gamma correction maps for enhancement. [117, 116] simulate single image exposure burst using piece-wise thresholded intensity functions whereas [225] uses low-rank decomposition for reflectance. Each factorization strategy harnesses crucial underlying optical observations and adds valuable insights to the low-level vision research. To the best of our knowledge, our proposed method here is the first to use recursive specularity estimation as a factorization strategy for LLE and other enhancement tasks.

5.3 Recursive Specularity Factorization Network

Our *Recursive Specularity Factorization Network* (RSFNet), consists of two parts. We first decompose the image into K factors using our *factorization network*. We use multiple factorization modules (FM) for this with each optimization step encoded as a differentiable network layer. In the second part, we fuse, enhance, and denoise the factors using our *fusion network*, which is built on task dependent pre-existing architectures. This modular design allows easy adoption of our technique in several other tasks and learning paradigms (Section 5.4).

5.3.1 Factorization Network

5.3.1.1 Specularity Estimation

Specularity removal is a well studied problem. Most specularity removal methods [5, 107, 251] exploit the relative sparsity of specular highlights and use pre-defined fixed sparsity thresholds to isolate the specular component. According to dichromatic reflection model [267] image consists of a diffuse $\mathbf{A}$ and a specular $\mathbf{E}$ term: $\mathbf{X} = \mathbf{A} + \mathbf{E}$ for input X where specular component can be estimated by minimizing the L_0 norm approximated as:

$$\underset{\mathbf{E},\mathbf{A}}{\operatorname{argmin}} \|\mathbf{A}\|_* + \lambda\|\mathbf{E}\|_1 \quad \text{s.t. } \mathbf{X} = \mathbf{A} + \mathbf{E}, \tag{5.3}$$

where L_1 is relaxation of L_0, $*$ is Frobenius norm regularizer and λ is the sparsity parameter with higher values encouraging sparser results. Equation (5.3) can be restated as augmented Lagrangian [43] using dual form and auxiliary parameters $(\mathbf{Y}, \mu)$, which are then solvable using iterative ADMM updates ($t \in [0, T]$) [44] as given below:

$$\begin{aligned}
\mathbf{E}_{t+1} &= \delta^1_{\alpha_t}\big(\mathbf{X} - \mathbf{A}_t - \mathbf{Y}_t^T/\mu_t\big) && \text{where } \alpha : \mathcal{F}(\lambda, \mu),\\
\mathbf{A}_{t+1} &= \delta^*_{\beta_t}\big(\mathbf{X} - \mathbf{E}_{t+1} - \mathbf{Y}_t^T/\mu_t\big) && \text{where } \beta : \mathcal{F}(\mu),\\
\mathbf{Y}_{t+1} &= \mathbf{Y}_t + \mu_t\big(\mathbf{A}_{t+1} + \mathbf{E}_{t+1} - \mathbf{X}\big) && \text{where } \mu : \mathcal{F}(\mathbf{X}).
\end{aligned} \tag{5.4}$$

Here δ^p_α is element-wise soft-thresholding operator [216]:

$$\delta^p_\alpha(x) = max(1 - \alpha/|x|_p,\ 0) \cdot x. \tag{5.5}$$

We can back-propagate through updates in Eq. (5.4) [286, 318] and hence can unroll them as neural network layers with three types of learnable parameters $\boldsymbol{\alpha} : \{\alpha\}_0^T$, $\boldsymbol{\beta} : \{\beta\}_0^T$ and $\boldsymbol{\mu} : \{\mu\}_0^T$.

5.3.1.2 Relation with ISTA

Analyzing the structure of Eq. (5.4), we can draw parallels with the ISTA problem [64], which seeks a sparse solution to $\mathbf{E}$ for the condition:

$$\mathbf{X} = \mathcal{G}\mathbf{E} + \epsilon, \tag{5.6}$$

with $\mathcal{G}$ as a learnable dictionary and negligible ϵ. In contrast, we have a non-negligible residue and identity dictionary. LISTA by Gregor and LeCun [101] showed how $\mathbf{E}$ update step can be represented as a weighted function which can then be approximated as finite network layers *i.e.*:

$$\mathbf{E}_{t+1} = \delta_{\alpha_t}(\mathbf{w}_t^1\mathbf{E}_t + \mathbf{w}_t^2\mathbf{X}), \tag{5.7}$$

with learnable parameters $(\alpha_t, \mathbf{w}_t^1, \mathbf{w}_t^2)$ for each iteration $t \in [0, T]$. Based on the weight coupling between $\mathbf{w}^1$ and $\mathbf{w}^2$, Chen et al. [62] simplified Eq. (5.7) by deriving both $\mathbf{w}^1$ and $\mathbf{w}^2$ from a single weight term, thereby halving the computation cost. A major simplification was further proposed by Liu et al. [179] as ALISTA, who proved how all weight terms could be analytically obtained for a known dictionary, thereby leaving only step sizes and thresholds *i.e.* μ and α_t to be estimated. Later on this idea was extended to other similar optimization formulations and improved upon by additional simplifications and guarantees *e.g.* Cai et al. [48] unrolled their ADMM updates into a network for robust principal component analysis.

5.3.1.3 Recursive Factorization

Drawing parallels from ALISTA [179] and its applications [48], we propose to learn the analytically reduced sparsity thresholds and step sizes via unrolled network layers. After optimizing the above mentioned objective Eq. (5.3) we obtain one specular factor $\mathbf{E}^k$ where index $k \in [1, K]$ indicates the factor number. For multiple factors, we recursively solve Eq. (5.3) by resetting the input X after removing the previous specular output and relaxing the initial sparsity weight. We initialize variables for each factor at $t = 0$ as:

$$\begin{aligned} \mathbf{X}^{k+1} = \mathbf{X}^k - \mathbf{E}^k, \qquad \mathbf{Y}^k = \mathbf{X}^k / \left\| \mathbf{X}^k \right\|_2 \\ \alpha^k = (1 - \nu^k)\hat{X}^k, \quad \beta^k = \nu^k \hat{X}^k, \quad \nu^k = k/K, \end{aligned} \tag{5.8}$$

where $\hat{X}$ indicates input mean and $\mathbf{X}^0$=$\mathbf{I}$. Intuitively, this can be understood as progressively removing specularity (E^k) from the original image by gradually relaxing the sparsity weight ($\alpha^{k+1} < \alpha^k$). This lets us split the original image into multiple additive factors as:

$$\mathbf{I} = \mathbf{E}^1 + \mathbf{E}^2 + \ldots + \mathbf{E}^K = \textstyle\sum_{k=1}^{K} \mathbf{E}^k \tag{5.9}$$

Figure 5.4: Shadow Dataset Factors: Top images show one data point from CHUK dataset [125] with mask, processed shadow/highlight regions and extracted factors.

5.3.1.4 Unrolling

Based upon above discussion, we propose an unrolled network collecting all parameters in a single vector $\boldsymbol{\theta}$. In each iteration t, we estimate three scalars: thresholds for both components (α_t, β_t) and the step size (μ_t). Hence for a factor k, we have $3T$ parameters $\boldsymbol{\theta}^k := (\alpha^k, \beta^k, \mu^k)$ and overall we have only $3KT$ parameters $\boldsymbol{\theta} := \{\boldsymbol{\theta}^k\}_1^K$. Hence our model-driven factorization module is extremely light-weight compared to other decompositions (Table 5.4). We propose the following novel factorization loss:

$$L_f = \lambda_f \sum_{k=1}^{K} L_f^k \qquad \text{where} \qquad L_f^k = \left| \hat{E}^k / \hat{X}^k - \nu^k \right|. \tag{5.10}$$

This constraints the ratio of signal energy in the k^{th} factor compared to the input, to ν^k. As ν^k increases for higher factors, our factorization loss relaxes the sparsity constraint, thereby gradually increasing the number of pixels in the specular component. After training, we are left with K specular factors which sum to I. As shown in Fig. 5.1 and Fig. 5.5, each one of these factors highlights specific image regions with similar illumination characteristics which can be individually targeted for enhancement. Initial factors are comprised of highlights and direct light sources, while the latter are comprised of soft and dark shadows with others in the middle.

The entire factorization process is summarized in Algorithm 1.

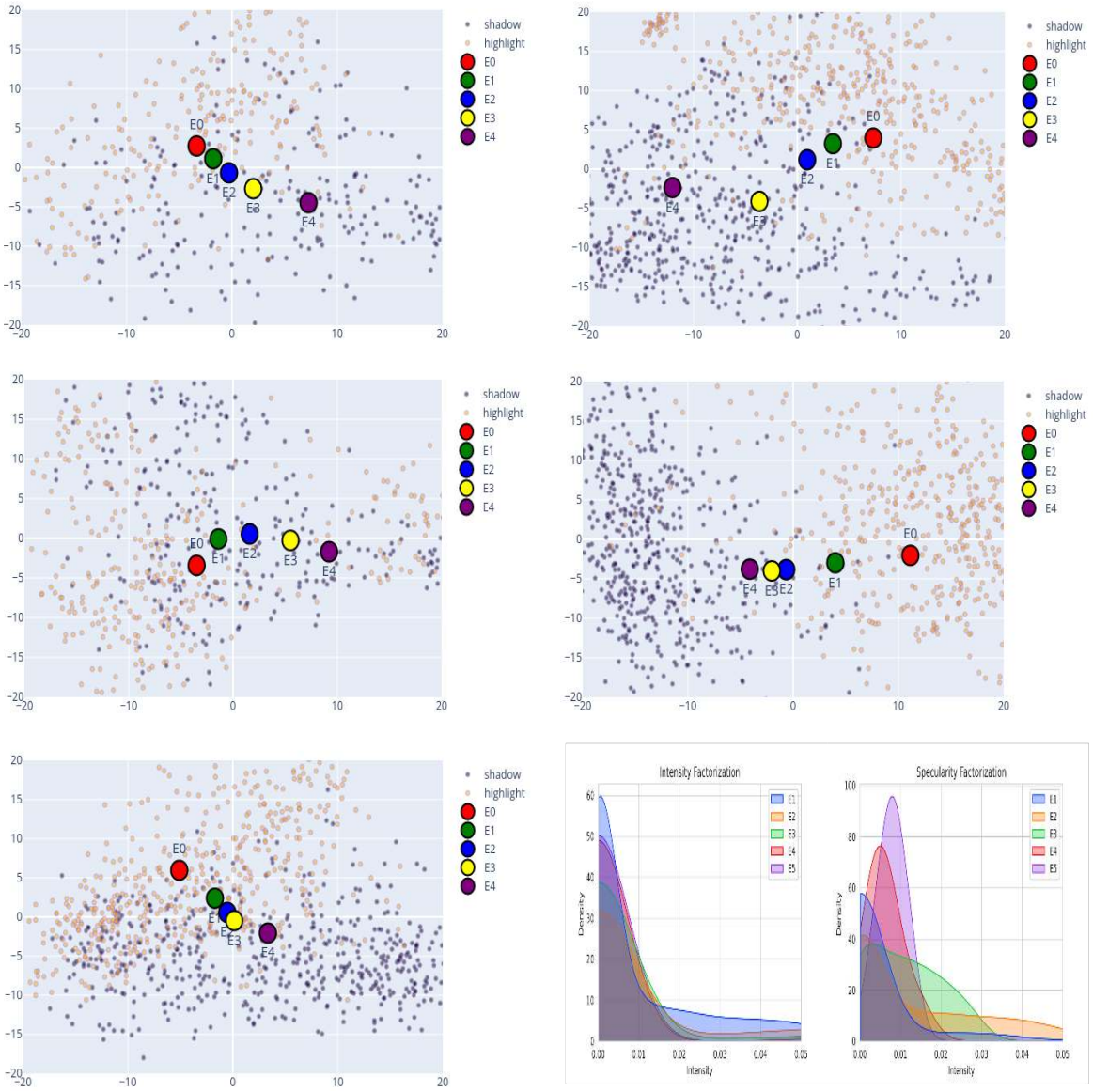


Figure 5.5: RSFNet Interpretation: Five scatter plots show the relationship between five factor cluster centers w.r.t each other and the background comprising of shadow/non-shadow regions estimated using PCA dimensionality reduced DINO features [53]. Gradual progression of feature cluster centers from highlight region to shadow region indicates their capability to capture various illumination regions in an image. Final distribution plots distinguish our specular fuzzy factors from intensity thresholding based binary division, with ours allowing more diverse distributions and richer representation.

5.3.2 Interpretability

The core assumption behind our factorization is that an image can be split into multiple specular factors with each representing specific illumination characteristic. To validate this hypothesis, we performed a toy experiment using shadow detection dataset [125] which contains binary shadow masks in complex real world images (Fig. 5.5). We extract semantics-rich DINO image features [53] after masking shadow and non-shadow image regions and visualize them in 2D using PCA. This marks separation of feature space between shadowed and highlighted regions in the background. The regions with progressively degrading illumination characteristics (glare, direct light, indirect light, soft shadow, dark shadow, *etc*.) are expected to gradually lie between the two extremes. Next we factorize each image into five factors using our approach and plot the cluster mean for each factor feature distribution on the same graph. We can observe in Fig. 5.5 that successive factors gradually shift from the non-shadow towards the shadowed feature space region mirroring the expected illumination order. This confirms that our factorization decomposes the pixel values across fundamental illumination types like glare, direct light, indirect light, shadow, *etc*.

Being a model-driven unrolled network, our entire framework is easily interpretable as each optimization step is clearly represented. This allows direct user intervention and better analysis of the intermediate latent factors as done in Fig. 5.5. We also repeat the same analysis with other parts of the shadow dataset [125]. [125] dataset consists of manually marked dense shadow regions in images taken from several standard datasets. Specifically there are five categories of such images with test split size mentioned in the parenthesis: shadow_ADE (226), shadow_KITTI (555), shadow_MAP (319), shadow_USR (489) and shadow_WEB (511). The analysis using shadow_ADE testset images were discussed previously. Here we similarly plot the factor features over the background of shadow and non-shadow PCA reduced feature space, for other sets. For feature extraction we use pretrained DINOv2 vits_14 backbone [53] and factors were computed using direct optimization using Eq. (5.3), Eq. (5.4) and Eq. (5.5). These plots are shown in Fig. 5.5. Note how in each case, the extracted features from the factors lie sequentially over the background of shadow and highlight image regions starting from highlight regions for the first factor (indicating glares and specular regions) to complete shadow regions for the last factor (indicating complete dark pixels). The other illumination types are expected to lie in between the two extremes and can be observed from the graph to follow the same. This helps us interpret the extracted factors as approximations of illumination types at each pixel into glare, direct light, indirect light, soft shadow, hard shadows *etc*.

We also plot the respective factor distribution densities of intensity factorization [117, 116] and our specularity factorization (Fig. 5.5, bottom right). Intensity factorization allows little variation in the underlying factor distributions and imposes hard segmentation constraints with binary pixel masks. Our specular factors, on the other hand, permit higher variability and soft masks, with each pixel

value spread across multiple factors. This provides more flexible representation and better optical approximation.

5.3.3 Fusion Network

In order to adhere to the zero-reference paradigm, we choose our fusion module to be a small fully-convolutional UNet like architecture with symmetric skip connections similar to other zero-reference methods [105, 213, 320]. One fundamental difference is that we modify the architecture to harness multiple factors and simultaneously perform fusion, enhancement and denoising. Specifically, it comprises of seven 3×3 convolutional layers with symmetric skip connections. We first pre-process all of our factors by subtracting the adjacent factors to discover the additional pixel values allowed in the current factor compared to the previous one as a soft mask:

$$\mathbf{F}^k = \mathbf{E}^k - \mathbf{E}^{k-1} \text{ where } \mathbf{F}^1 = \mathbf{E}^1. \tag{5.11}$$

These factors are weighted if required using fixed scalar values and are then passed as a concatenated tensor into the fusion network. The output gamma maps $\mathbf{R}^k$ rescale different image regions differently and are applied directly on the original image inside the curve adjustment equation [105] for the fused result:

$$O = \Phi(\sum_{k=0}^{K} I + R^k.\big((I)^2 - I\big)). \tag{5.12}$$

The fused output is finally passed through a differentiable bilateral filtering layer Φ [227] for the final enhanced result O. Note that all the parameters from both factorization and fusion networks are trained together in end-to-end manner.

Algorithm 1: RSFNet Training using Specularity Factorization

Input: Lowlight:I; Hyperparams: $\lambda_{c|e|s}, K, T$

Output: Enhanced: O; Params: $\boldsymbol{\theta}$=$\{\boldsymbol{\alpha}\}_0^K, \{\boldsymbol{\beta}\}_0^K, \{\boldsymbol{\mu}\}_0^K$

for $e \leftarrow 0$ **to** *num of epochs* **do**

  // Train Factorization Module

  for $k \leftarrow 0$ **to** K **do**

    for $t \leftarrow 0$ **to** T **do**

      Initialize E_0^k, A_0^k, Y_0^k ; *// Eqn. 5.8*

      $E_t, A_t, Y_t \leftarrow$ ADMM updates ; *// Eqn. 5.4*

    end

    $F^k \leftarrow E^k - E^{k-1}$; *// Eqn. 5.11*

  end

  Compute L_f ; *// Eqn. 5.10*

  // Train Fusion Module

  if $e >$ *freeze epoch* **then**

    Freeze all $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}$;

    $L_f \leftarrow 0$;

  end

  $I_{fuse} \leftarrow$ *Concatenate* $[I, F^1, ..., F^K]$;

  $O \leftarrow$ *Forward* (I_{fuse}) ;

  Compute L ; *// Eqn. 5.16*

  Backpropagate L;

end

5.3.3.1 Loss Terms

We use two widely employed zero-reference losses for enhancement [105, 213, 306] and one image smoothing loss for denoising. First *color loss* L_c [105, 306] is based on the gray-world assumption which tries to minimize the mean value difference between each color channel pair:

$$L_c = \sum_{(i,j)\in C} \left(\hat{O}^i - \hat{O}^j\right)^2, \quad C \in \{(r,g),(g,b),(b,r)\}. \tag{5.13}$$

Second is the *exposure loss* L_e [200, 105, 117], which penalizes grayscale intensity deviation from the mid-tone value:

$$L_e = \frac{1}{|\Omega|} \sum_{\Omega} \big(\phi(O) - 0.6\big)^2 \text{ where } \Omega \in \{c \times h \times w\}, \tag{5.14}$$

where ϕ represents the average value over a 16×16 window. Our third loss is the pixel-wise *smoothing loss* which controls the local gradients $\nabla_{x|y}$ in the final output:

$$L_s = \frac{1}{|\Omega|} \sum_{\Omega} \big((\nabla_x O)^2 + (\nabla_y O)^2\big), \tag{5.15}$$

Note that this differs from the previous works who focus on total variational loss of the gamma maps instead. Our final training loss with λ's as respective loss weights, is given as:

$$L = \lambda_f L_f + \lambda_c L_c + \lambda_e L_e + \lambda_s L_s. \tag{5.16}$$

5.4 Experiments

5.4.1 Setup

We implement our combined network end-to-end on a single Nvidia 11 GB GPU in PyTorch. We directly use low-light RGB images as inputs without any additional pre-processing. We first train factorization module for 25 epochs which we freeze and then optimize the fusion module for next 25 epochs. We use stochastic gradient descent for optimization with batch size of 10 and 0.01 as learning rate. Model hyper-parameters are fixed using grid search and the entire training take less than 30 minutes. Training time of our RSFNet is quite fast. For any Lol dataset [283, 300, 180], it takes approximately 30 minutes on a single 1080Ti GPU machine for the complete 50 epochs. We first train the factorization and fusion modules together for 25 epochs using Eq. (5.10) and then freeze the factorization parameters for next 25 epochs to train the fusion module with Eq. (5.16). Initial versions of the system involved slow decay of factorization learning rate without abrupt freezing but the current setting was adopted to clearly ascertain the effect of each module training. Hence we do not use any learning rate decay during our training but the reader is welcome to experiment with the same for their own datasets.

5.4.1.1 Initialization

During training instead of using any hard coded initialization value for thresholds, we allow per instance initialization. Specifically, we use 0.9 ratio of learned threshold values and 0.1 fraction of the image mean for initialization with initial threshold values set to dataset mean. This setting is also followed during inference and all the results reported in this chapter.

Config	Factorization		Fusion		Experiment
	Trad.	Deep	Trad.	Deep	
C_{11}		✓		✓	Main RSFNet framework (Fig. 5.3)
C_{10}		✓	✓		Ablation (w/o Fusion) (Table 5.5)
C_{01}	✓			✓	Extension Applications (Fig. 5.13)
C_{00}	✓		✓		User Applications (Fig. 5.7)

Table 5.2: **System Configurations:** Various possible configurations of our proposed technique. Two central steps of our method, factorization and fusion, could each be either traditionally estimated with manual model-based optimization or using deep data-driven methods. This gives rises to four possible configurations all of which are used in one or the other experiment in the main paper

Several optimization methods are sensitive to initialization conditions and when they are unrolled into model layers [179, 48]. During implementation sources of randomness can be corrected by properly seeding the random number generators of the deep learning and the numerical algorithm libraries using:

$$\texttt{np.random.seed}(c)$$
$$\texttt{torch.random.seed}(c)$$

where c is some fixed integer constant. We use $c = 2$ in our LLE experiments and the values of all the hyper-parameters can be provided in a configuration file.

5.4.1.2 Configurations

Our proposed method can be used for various applications in one of four possible configurations as shown in Table 5.2. This is dependent on whether the factorization and fusion steps are carried out via traditional model-based optimization or learned using data-driven deep networks. Model-based solutions are better generalizable but slower with lesser performance than data-driven solutions. We have used all four configurations in one or the other experiment as listed in the Table 5.2. For traditional factorization we use solution to the direct specularity estimation optimization equation Eq. (5.3) using Eq. (5.4), whereas for deep solution we use the unrolled layers Fig. 5.3 to learn the associated optimization thresholds using our Factorization Modules which are learned form the dataset in a data-driven fashion. Fusion is either task specific deep network or simply the running average as described in Eq. (5.21). This highlights the flexibility and versatile nature of our proposed technique

which allows easy integration with pre-existing fusion methods with observed improvement in all scenarios.

5.4.2 Datasets

We evaluate our method using multiple LLE benchmark datasets (Lolv1 [283], Lolv2-real [300], Lolv2-synthetic [300] and VE-Lol [180]) with standard train/test splits (Table 5.4). These datasets comprise of several underexposed small-aperture inputs and corresponding well-exposed ground-truth pairs. Here we report results on two datasets: Lolv1 and Lolv2-real and finally show the mean scores on all four datasets combined in the last sub-table Table 5.4 and in Fig. 5.6. Furthermore, we report generalization results (Table 5.6) on five additional no-reference datasets which have significant domain shift: DICM [158], LIME [109], MEF [190], NPE [277] and VV [273]. Note that our method achieves the best generalization across datasets compared to contemporary similar methods.

The details of five no-reference (Table 5.6) and four Lol datasets Table 5.4 are given below:

- Lolv1 [283]: It contains 500 low light and well lit image pairs of real world scenes with 485 for training and 15 for testing in the standard split. Each image is 400×600 in resolution with mean intensity = 0.05 (*i.e*. very low light).

- Lolv2-real [300]: It is an extension of Lolv1 dataset with 689 images in training and 100 in testing set. Mean intensity of images is 0.05 and resolution is same = 400×600. Note that majority of the images in the testing set of Lolv2 are present in the training set of Lolv1 and hence Lolv1 trained models should not be evaluated directly on Lolv2 testset.

- Lolv2-synthetic [300]: As Lolv1 mostly contains only indoor scenes with heavy dark channel noise, Lolv2-synthetic presents a significant domain shift with mean intensity=0.2 and resolution= 384×384. The scenes are both indoors and outdoors and the supervision data is obtained by synthetically reducing the exposure by using the raw image data and natural image statistics.

- VE-Lol [180]: Vision Enhancement in LOw Level vision dataset (VE-LOL-L-Cap) consists of 1500 image pairs with 1400 *vs*. 100 training to test split. The trainset here consists of multiple under-exposed images of the same scene but the test set is similar to Lolv2-real. Dataset image resolution=400×600 and mean intensity=0.07. Multiple exposure settings here help ascertain model's robustness to input perturbations.

  Other five datasets [165] are no-reference (*i.e*. without any ground truth well lit image) and are used for perceptual quality evaluation and generalization assessment. Although varying number of images have been reported in the previous literature for a few of these datasets [105, 8, 165],

$Type$	PSNR$_y$	SSIM$_y$	PSNR$_c$	SSIM$_c$	NIQE	LPIPS
$w\emptyset$ weights	19.73	0.843	19.39	0.745	3.701	0.278
weighted	20.22	0.884	17.23	0.815	4.286	0.159

Table 5.3: **Factor Weights:** Our improved results on Lol-synthetic dataset if we additionally allow the user to configure factor weights before concatenation and input to the fusion module. To be understood in the wider context of Table 5.4.

we use the download links provided by Li et al. [165] with the following brief description of each dataset:

- DICM [158]: 69 images, mean=0.32, mixed exposure settings, variable resolutions, real scenes, varying scene including macros, landscapes, indoors, outdoors *etc*.
- LIME [109]: 10 images, mean=0.15, varying resolutions, real scenes, varying scene types.
- MEF [190]: 17 images, mean=0.15, resolution=512 $\times$ 340, relatively darker images, varying scene types.
- NPE [277]: 85 images, mean=0.31, varying resolution, both over and under exposed image regions, mostly outdoor scenes.
- VV [273]: 24 images, mean=0.26, resolution=2304 $\times$ 1728, large images, both over and under exposed image regions, both indoor/outdoor scene types.

These results are listed in Table 5.6 and Table 5.4. As can be observed in the tables, our method achieves best score over all with best or second best performance on several benchmarks across multiple metrics.

5.4.3 Testing

For inference, we can edit the weights of the factors before concatenation and input into the fusion module to allow varying results. Although all results discussed till now are obtained without any weight manipulations (*i.e*. all factors are equally important with each the weight vector corresponding to E_0 to E_5 set to [1,1,1,1,1,1]), better results are possible if dataset specific finetuning is allowed. If this is followed our scores on Lol-synthetic dataset in the main quantitative results table Table 5.4 can be updated to Table 5.3 by using $w = [1, 4, 4, 4, 4, 4]$. Yet another setting which can be configured is related to the bilateral filtering step which includes window size, color sigma and the spatial sigma in

both of the horizontal directions. The values can be chosen based on the expected noise in the input datasets but we keep them constant as window size=5, color sigma=0.5 and spatial sigma=1 for all our experiments in Table 5.4

5.4.4 Metrics

We report both single channel (Y from YCbCr) and multichannel (RGB) performance scores. As full-reference metrics (which require ground truth), we use Peak Signal to Noise-Ratio (PSNR), Structural Similarity Index Metric (SSIM) [280] and Learned Perceptual Image Patch Similarity (LPIPS) [311]. For no-reference assessment (without ground truth), we report Naturalness Image Quality Evaluator (NIQE) [202] and Lightness Order Error [276]. Note while PSNR and SSIM gauge performance quantitatively, other three metrics estimate perceptual quality.

Most frequently reported metric for LLE task is PSNR (peak signal to noise ratio). Although traditional usage of PSNR has been for denoising of grayscale images with only single channel but now it also has been extended to multichannel scenarios for various tasks. PSNR for a predicted enhanced output $\hat{y}$ is given as:

$$p = 10 \log \left[\frac{\frac{1}{N} \sum_{i}^{N} (\hat{y}_i - y_i)^2}{M^2} \right], \tag{5.17}$$

where N is total number of pixels and M is the peak pixel value which depending upon the situation is either 1.0 or 255. Eq. (5.17) is straightforward in case of single channel image but there is slight ambiguity in case of multichannel prediction. Different results are obtained depending upon whether per channel mean is considered inside the logarithm or outside. Correct way of multichannel PSNR definition is to consider it inside the logarithm *i.e.* to take mean square error over all the channels simultaneously instead of individually and then averaging it as shown below:

$$p = 10 \log \left[\frac{\frac{1}{N*C} \sum_{c} \sum_{i}^{N} (\hat{y}_{i,c} - y_{i,c})^2}{M^2} \right]. \tag{5.18}$$

Yet another issue is during the YCbCr to rgb conversion for PSNR evaluation of Y only channel. Most of the codes directly use the in-built functions from the available libraries like opencv or PIL. The conversion involves applications of a transformation matrix which differs from library to library depending upon whether the input signal is assumed to be analog or digital *e.g.* opencv applies the following transformation assuming analog input:

$$Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B, \tag{5.19}$$

whereas Matlab prefers the digital transformation as:

$$Y \leftarrow 0.2568 \cdot R + 0.5041 \cdot G + 0.0979 \cdot B. \tag{5.20}$$

This leads to variability in results (approximately 1 PSNR difference) depending upon the conversion library chosen. In our opinion Eq. (5.20) should be chosen and the PSNR tables should clearly highlight that it is a single Y channel evaluation.

Paradigm	**Traditional Model Based**			**Zero-reference**							
Method	LIME [109]	DUAL [310]	SDD [112]	ECNet [308]	ZDCE [105]	ZD++ [166]	RUAS [229]	SCI [191]	PNet [213]	GDP [79]	**RSFNet** (Ours)
#Params	-	-	-	16.5M	79.4K	10.6K	3.43K	**0.26**K	15.3K	552K	2.11K
Lolv1 [283] (dataset split: 485/15, mean≈ 0.05, resolution: 400 × 600)											
PSNR$_y$ ↑	16.20	15.97	15.14	18.01	16.76	16.38	18.45	16.45	19.85	17.68	**22.17**
SSIM$_y$ ↑	0.695	0.692	0.754	0.644	0.734	0.645	0.766	0.709	0.718	0.678	**0.860**
PSNR$_c$ ↑	14.22	14.02	13.34	15.81	14.86	14.74	16.40	14.78	17.50	15.80	**19.39**
SSIM$_c$ ↑	0.521	0.519	0.634	0.469	0.562	0.496	0.503	0.525	0.550	0.539	**0.755**
NIQE ↓	8.583	8.611	3.706	8.844	8.223	8.195	5.927	8.374	8.629	6.437	**3.129**
LPIPS↓	0.344	0.346	0.278	0.358	0.331	0.346	0.303	0.327	0.340	0.375	**0.265**
Lolv2-real [300] (dataset split: 689/100, mean≈ 0.05, resolution: 400 × 600)											
PSNR$_y$ ↑	19.31	19.10	18.47	18.86	20.31	19.36	17.49	19.37	20.08	15.83	**21.46**
SSIM$_y$ ↑	0.705	0.704	0.792	0.613	0.745	0.585	0.742	0.722	0.691	0.627	**0.836**
PSNR$_c$ ↑	17.14	16.95	16.64	16.27	18.06	17.36	15.33	17.30	17.63	14.05	**19.27**
SSIM$_c$ ↑	0.537	0.535	0.678	0.459	0.580	0.442	0.493	0.540	0.539	0.502	**0.738**
NIQE ↓	9.076	9.083	4.191	9.475	4.191	8.709	6.172	8.739	9.152	6.867	**3.769**
LPIPS↓	0.322	0.324	**0.280**	0.360	0.310	0.340	0.325	0.294	0.340	0.390	**0.280**
Lolv2-synthetic [300] (dataset split: 900/100, mean≈ 0.2, resolution: 384 × 384)											
PSNR$_y$ ↑	19.16	17.16	17.93	18.21	19.65	19.81	14.91	17.09	18.29	13.26	**19.82**
SSIM$_y$ ↑	0.843	0.812	0.787	0.842	0.884	0.882	0.720	0.825	0.849	0.602	**0.893**
PSNR$_c$ ↑	17.63	15.61	16.47	16.75	**17.76**	17.58	13.40	15.43	16.62	11.97	17.13
SSIM$_c$ ↑	0.787	0.742	0.725	0.769	0.814	0.811	0.640	0.744	0.773	0.481	**0.816**
NIQE ↓	4.685	4.741	4.335	4.311	4.357	**4.257**	5.092	4.652	4.308	–	4.404
LPIPS↓	0.174	0.194	0.235	0.178	**0.142**	0.157	0.365	0.203	0.160	0.311	0.157
VE-Lol [180] (dataset split: 1400/100, mean≈ 0.07, resolution: 400 × 600)											

Continued on next page

Table 5.4 – continued from previous page

Method	LIME [109]	DUAL [310]	SDD [112]	ECNet [308]	ZDCE [105]	ZD++ [166]	RUAS [229]	SCI [191]	PNet [213]	GDP [79]	**RSFNet** (Ours)
PSNR$_y$ ↑	19.31	19.10	18.47	18.72	20.31	19.36	17.49	19.37	<u>20.39</u>	16.29	**21.18**
SSIM$_y$ ↑	0.705	0.704	<u>0.792</u>	0.610	0.745	0.585	0.742	0.722	0.715	0.628	**0.817**
PSNR$_c$ ↑	17.14	16.95	16.64	16.15	**18.06**	17.36	15.33	17.30	<u>17.64</u>	14.42	**18.06**
SSIM$_c$ ↑	0.537	0.535	<u>0.678</u>	0.457	0.580	0.442	0.493	0.540	0.557	0.498	**0.714**
NIQE ↓	9.076	9.083	<u>4.191</u>	9.482	8.767	8.709	6.172	8.739	9.073	7.027	**3.782**
LPIPS↓	0.322	0.324	**0.275**	0.418	<u>0.310</u>	0.340	0.390	0.355	0.368	0.444	0.397
	<u>Mean Scores</u> (**Lolv1** [283], **Lolv2-real** [300], **Lolv2-syn** [300] and **VE-Lol** [180])										
PSNR$_y$ ↑	18.50	17.83	17.50	18.45	19.26	18.73	17.09	18.07	<u>19.65</u>	15.88	**21.16**
SSIM$_y$ ↑	0.737	0.728	<u>0.781</u>	0.677	0.777	0.674	0.743	0.745	0.743	0.634	**0.854**
PSNR$_c$ ↑	16.53	15.88	15.77	16.25	17.19	16.76	15.12	16.20	<u>17.35</u>	14.15	**18.45**
SSIM$_c$ ↑	0.596	0.583	<u>0.679</u>	0.538	0.634	0.548	0.532	0.587	0.605	0.504	**0.758**
NIQE ↓	7.855	7.880	<u>4.106</u>	8.028	6.385	7.468	5.841	7.626	7.791	6.826	**3.763**
LPIPS↓	0.291	0.297	**0.266**	0.329	<u>0.273</u>	0.296	0.346	0.295	0.302	0.379	0.276

Table 5.4: **RSFNet Quantitative Results:** Qualitative results of our method RSFNet with other **traditional and zero-reference** solutions on multiple lowlight benchmarks and six evaluation metrics. Shown here are scores for four datasets Lolv1, Lolv2-real, Lolv2-synthetic and VE-Lol with mean value across all datasets in the last sub-table (key: ↑ higher better; ↓ lower better; **bold**: best; <u>underline</u>: second best; '-': NaN error computing value).

Variants	$w/o\ L_e$	$w/o\ L_c$	$w/o\ L_s$	$w/o\ Denoise$	w/o <u>Fusion</u>	**Full**
PSNR$_y$ ↑	8.12	16.05	20.13	19.51	<u>19.32</u>	**22.17**
SSIM$_y$ ↑	0.238	0.724	0.846	0.756	<u>0.830</u>	**0.860**

Table 5.5: **Ablation Results:** Ablation analysis on five variants of our RSFNet (Section 5.4).

NIQE↓ & *LOE*↓	ECNet [308]	ZDCE [105]	ZD++ [166]	RUAS [229]	PNet [213]	SCI [191]	**RSFNet** (Ours)
DICM [158]	3.37—*676.7*	3.10—*340.8*	**2.94**—*511.9*	4.89—*1421*	3.00—*590.3*	3.61—*321.9*	3.23—***303.1***
LIME [109]	**3.75**—*685.1*	3.79—*135.0*	3.89—*332.2*	4.26—*719.9*	3.84—*223.2*	4.14—*75.5*	3.80—***68.3***
MEF [190]	3.30—*863.3*	3.31—*164.3*	3.18—*458.5*	4.08—*784.2*	3.25—*363.0*	3.43—***95.0***	**3.00**—*100.7*
NPE [277]	**3.24**—*936.1*	3.52—*312.9*	3.27—*532.2*	5.75—*1399*	3.29—*601.1*	3.89—*239.8*	3.31—***221.5***
VV [273]	2.15—*292.4*	2.75—*145.4*	2.53—*222.9*	3.82—*583.7*	2.56—*260.2*	2.30—***109.0***	**1.96**—***109.0***
Mean	3.16—*690.7*	3.29—*219.7*	3.16—*411.5*	4.56—*981.7*	3.19—*407.5*	3.47—*168.2*	**3.06**—***160.5***

Table 5.6: **RSFNet Qualitative Results:** Qualitative comparison using naturalness preserving metrics (NIQE ↓ — LOE ↓) on five no-reference benchmarks: DICM, LIME, MEF, NPE and VV (**best** scores in bold, lower is better).

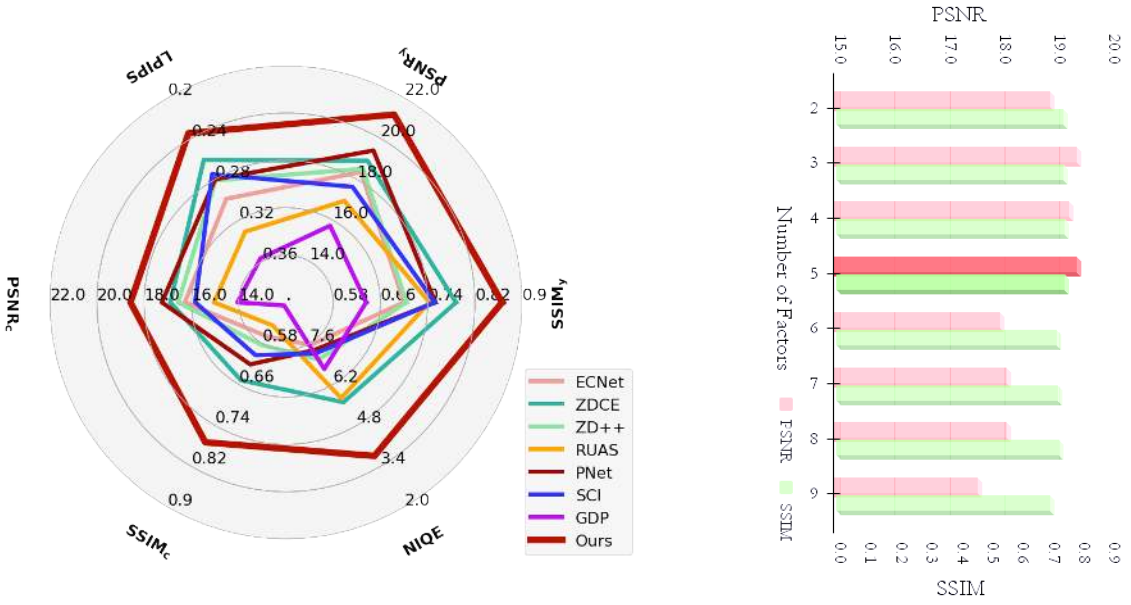


Figure 5.6: Analysis: On left, our average score on all datasets *vs*. other methods (more area implies better). On right, ablation analysis with varying number of factors.

Figure 5.7: User-controlled Edits: Here we show high resolution version of our user controlled results using our factors. For three scene from top to bottom we show modification of illumination specularity, indoor lighting color and outdoor lighting intensity respectively. All edits were carried out in GIMP [265] using our factors as layers and only global layer operations like curve adjustments, blurring, layer blending *etc*. were used without any local selection or modifications. Notice how our factors seamlessly merge to render such edits preserving the naturalness of the original image and without any additional artifacts. Note that these are only three representative applications and several other edits are possible with appropriate masking, color adjustments and even cross image layers harmonization.

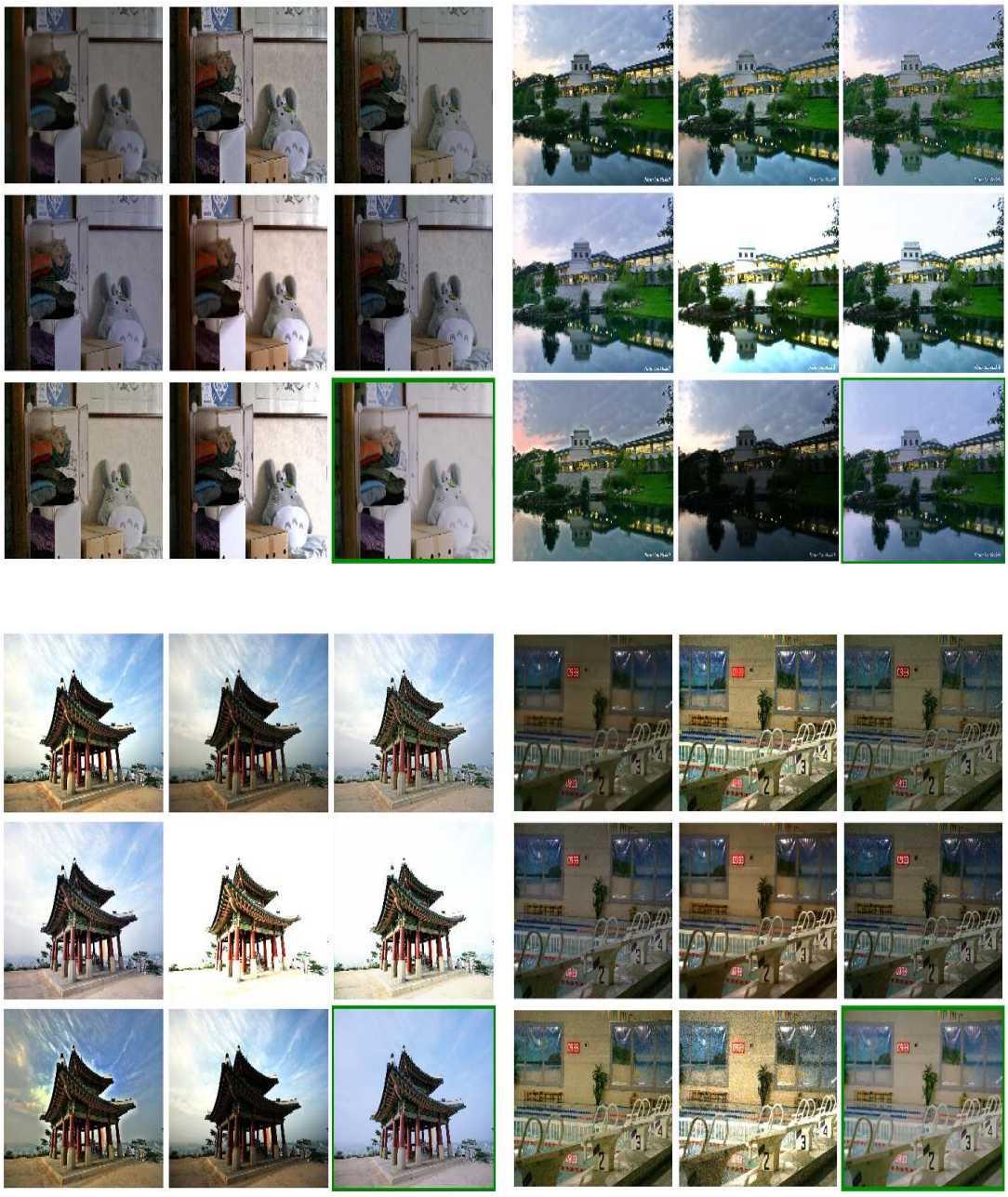


Figure 5.8: Qualitative Comparisons: Additional low light enhancement comparisons. Each set row in the grid contains results from: [SDD[112], ECNet[308], ZDCE[105]]; [ZD++[166], RUAS[229], SCI[191]]; [PNet[213], GDP[79], RSFNet(Ours, green box)]. Our results preserve the naturalness of the original scene without over/under exposing intensity or color saturation, which is also quantitatively supported by our overall better NIQE/LOE scores in Table 5.6 and Fig. 5.6.

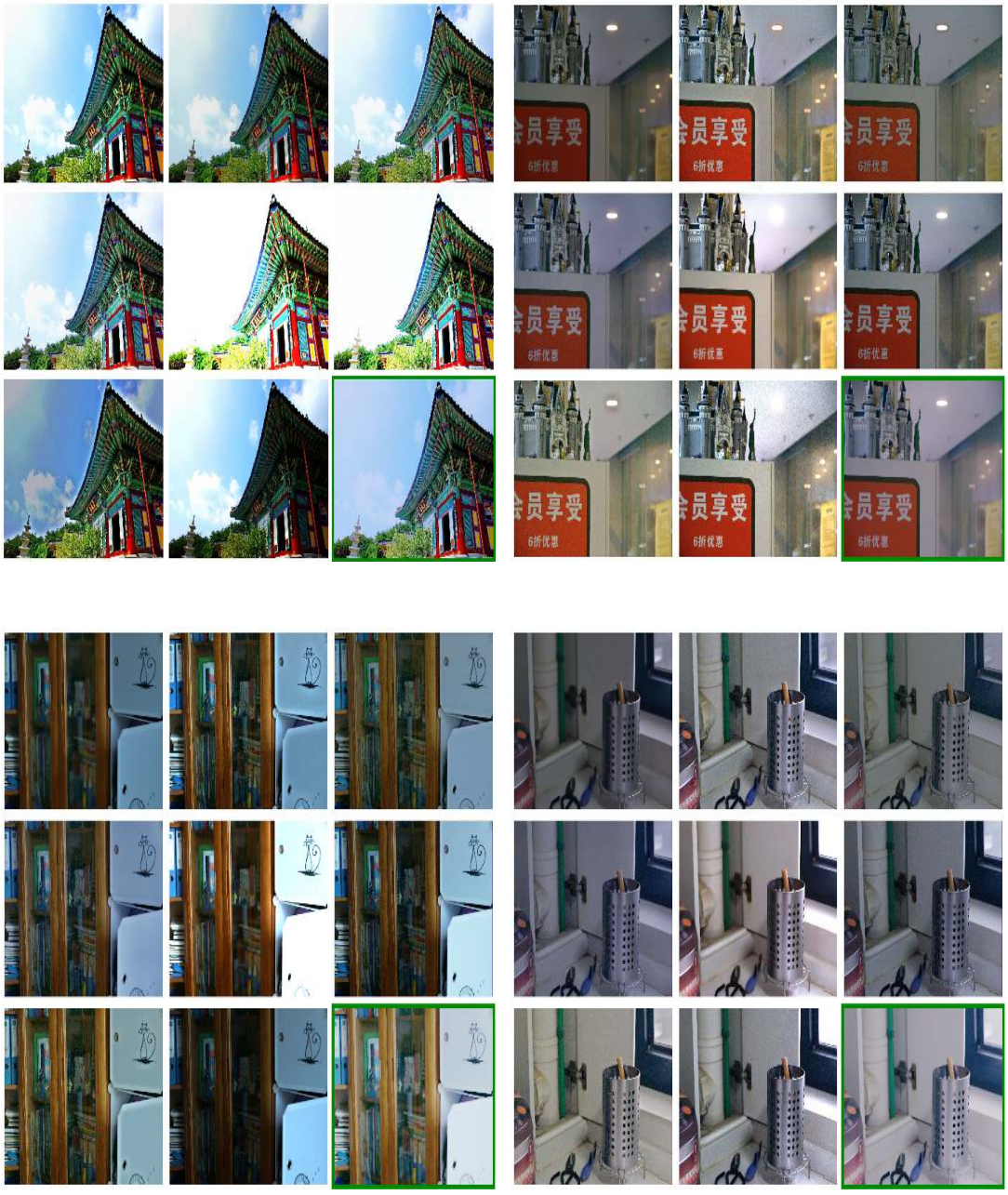


Figure 5.9: Qualitative Comparisons: Additional low light enhancement comparisons. Each set row in the grid contains results from: [SDD[112], ECNet[308], ZDCE[105]]; [ZD++[166], RUAS[229], SCI[191]]; [PNet[213], GDP[79], RSFNet(Ours, green box)]. Our results preserve the naturalness of the original scene without over/under exposing intensity or color saturation, which is also quantitatively supported by our overall better NIQE/LOE scores in Table 5.6 and Fig. 5.6.

5.4.5 Comparisons

We compare against three model-based traditional optimization methods: LIME [109], DUAL [310] and SDD [112] (others ignored due to low performance). For data-driven methods we use seven recent zero-reference methods (chronologically ordered): ECNet [308], zeroDCE [105], zeroDCE++ [166], RUAS [229], SCI [191], PNet [213] and GDP [79]. We use the official code releases with pretrained weights and default parameters for results generation. Quantitative and qualitative performance comparison is shown in Table 5.4 and Figs. 5.8 and 5.9 respectively. Qualitatively, our method is cleaner with fewer artifacts and natural illumination (Figs. 5.8 and 5.9). This is validated by perceptual metrics like NIQE, LPIPS and LOE scores (Tables 5.4 and 5.6). Our method outperforms other similar category contemporary solutions on multiple metrics and achieves the best generalization performance across datasets. For a generalized performance, we take mean of all the scores across benchmarks and graphically show them in the polar plot in Fig. 5.6. Each polygon represents a separate LLE method with higher area inside indicating better performance. As can be observed from the figure, our RSFNet encompasses other methods indicating overall better cross-dataset generalizability.

5.4.6 Ablation

To validate our design choices, we conduct ablation study on several variants of our methods using Lolv1 dataset. The effect of different number of factors K on the final PSNR and SSIM scores are shown on right in Fig. 5.6. We choose the best observed hyper-parameter settings $K{=}5$ for all our experiments. The effect of various loss terms after removing them one at a time (*i.e.* $w/o\ L_{e|c|s}$) and the effect of the final denoising step are shown in Table 5.5. The last variant (*w/o* Fusion) represents an especially interesting setting where the fusion network is totally removed and inference uses only $3KT$ (=3*5*3=45) parameters. Fusion now reduces to a running average of the current image and the next factor, weighted by the normalized mean:

$$O^{k+1} = (1 - w^k)O^k + w^k F^k, \ \text{ where } \ w^k = \hat{F}^k / \textstyle\sum_k \hat{F}^k. \tag{5.21}$$

Even without any other zero-reference losses and using only a simple linear fusion, this method performs well, which demonstrates the effectiveness of our factors. Note here we have an order of magnitude smaller network size than SCI (0.045 *vs*. 0.26 thousand parameters in Table 5.4).

Figure 5.10: Factor Visualizations: We show visualizations of our extracted five specular factors for various scenes. Input images (blue box) are taken from [125] dataset and factors are rescaled for visualization. Note how various regions are captured in the respective factors depending upon whether they are illuminated by directly, indirectly or in shadows.

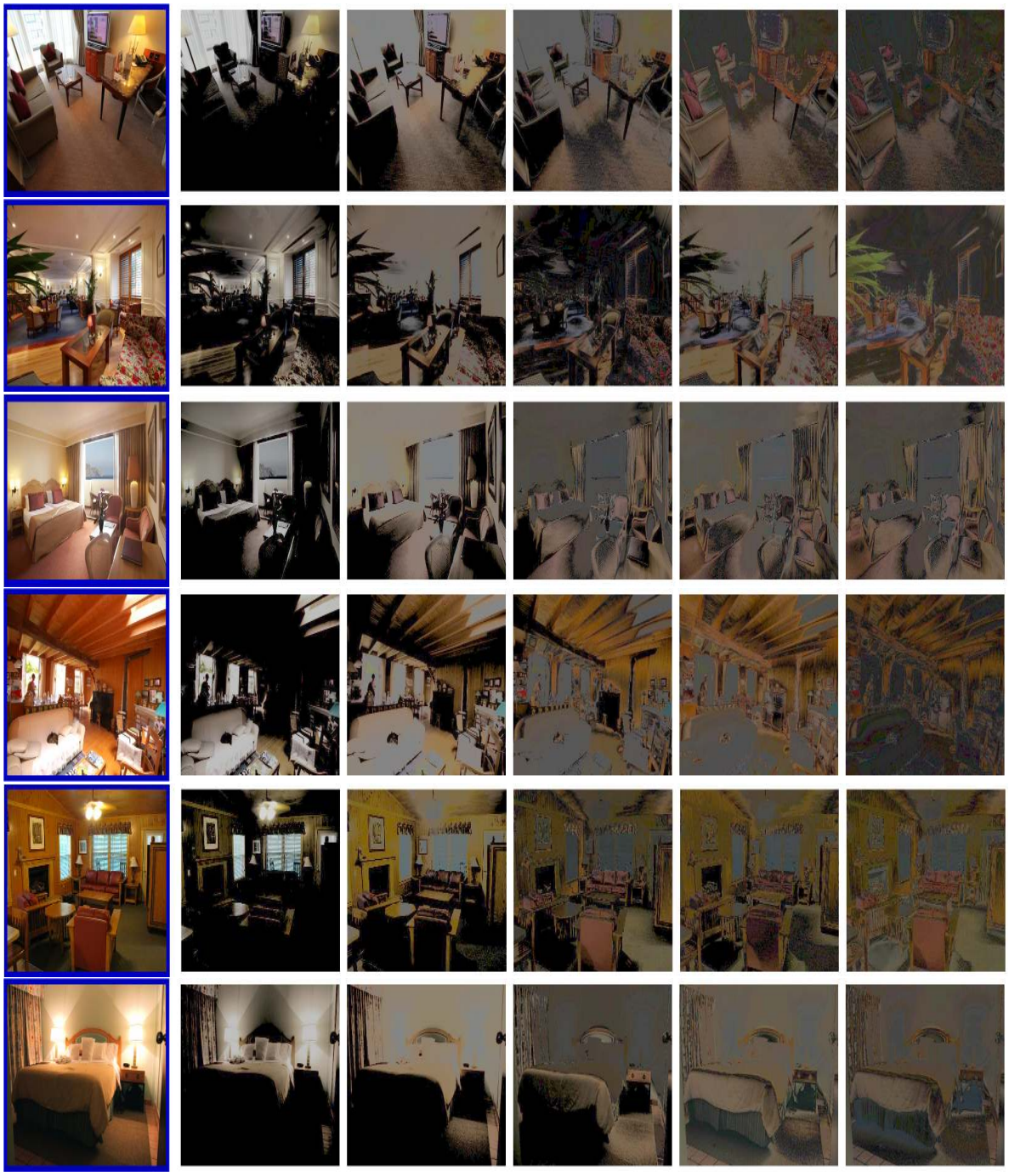


Figure 5.11: More Factor Visualizations: We show visualizations of our extracted five specular factors for various scenes. Input images (blue box) are taken from [32] dataset and factors are rescaled for visualization. Note how various regions are captured in the respective factors depending upon whether they are illuminated by directly, indirectly or in shadows.

Figure 5.12: Our LLE Results: Additional low light enhancement results from multiple Lol-x datasets [283, 300]. Each set contains input image (blue box), ground truth (red box) and our result (green box).

Figure 5.13: Extensions: Image enhancement applications using our specular factors as inputs on the AirNet [164] base model. Shown here left-to-right are our results for Dehazing [163], Deraining [297] and Deblurring [207] tasks respectively using AirNet [164] as base model.

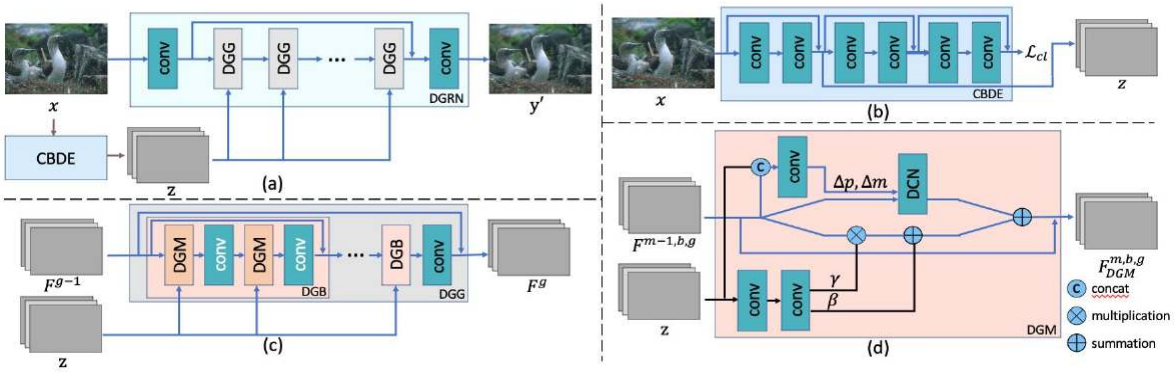


Figure 5.14: AirNet: (a) Block diagram from [164]. CBDE (b) refers to Contrastive-Based Degradation Encoder, DGG (c) means Degradation Guided Groups and DGM (d) is Degradation Guided Module. For complete details refers to [164]. For our usage, we alter first conv layer (first deep blue block on top-left (a)) and the first conv layer in CBDE (first deep blue block on top-right (b)).

5.4.7 Extensions

Our specular factors are easily interpretable and can be used directly for image manipulation as image layers in standard image editing tools like GIMP [265], Photoshop [7], *etc*. We show an image relighting example by varying the color and blending modes of factors in (Fig. 5.1 bottom left, Fig. 5.7). This indicates the potential of our factorization to complex downstream applications. We explore three diverse image enhancement tasks: dehazing, deraining and deblurring. Here our goal is to evaluate the use of specularity factorization as a pre-processing step on an existing base model.

We chose the recent AirNet [164] as it allows experimentation on multiple image enhancement tasks with minor backbone modification. To induce our factors as prior information, we concatenate them along with the original input and alter the first convolutional layer input channels. Note that we do not introduce any new loss or layers and directly train the model for three tasks one by one:

1. Dehazing on RESIDE dataset [163]
2. Deraining on Rain100L dataset [297]
3. Debluring on GoPro dataset [207]

Specifically, we use AirNet [164] (Fig. 5.14) and alter the input tensor from a single 3 channel input to a tensor comprising of the concatenated input image and other factors by simply modifying the in-channels of the first convolutional layer in both the main AirNet backbone and the CBDE module. We train for 500 epochs for each task separately (with additional 50 epochs for initial warm up) and keep the default learning rate and decay parameters. We found no significant difference in training from scratch or finetuning over the multi-task pre-trained checkpoint. We also provide full comparison table using the values as provided by [307] for various tasks in the multitask configuration. For uni-task configuration (*i.e*. one task at a time), we report the values as provided in the main AirNet paper itself or compute them ourselves by retraining with default parameters (for deblurring). Note that the we have chosen AirNet over others due to its overall better performance than others (except IDR). IDR [307] was not used as the public code is not available at the time of writing of this work. As can be observed from the table, even straightforward introduction of our factors as priors without any loss or major architecture modifications can improve the existing performance consistently for all reported tasks. There is 4.7 % PSNR and 2.3 % SSIM performance boost of the mean scores with our prior induction. As seen in Fig. 5.13 and Table 5.7, our results are perceptually more pleasing and improve the previously reported scores from multi-task methods consistently [307, 164]. We believe this is due to the induction of structural prior in the form of illumination based region categorization as the intensity and order of illumination at a pixel depends on the scene structure.

5.4.8 Generalization

Additionally, we also provide generalization performance comparison of various LLE solutions, including recent supervised and unsupervised methods, on the unseen data using images from standard no-reference LLE benchmarks (*i.e*. without any ground truth) in Table 5.9 . We report NIQE scores [202] to assess the overall perceptual quality and the naturalness of the generated results. As can be seen from the Table 5.9, our method, being a zero-reference solution, generalizes better due to low dependence on the training dataset compared to the supervised and the unsupervised counterparts. This

TASK →	**DEHAZE** [163]		**DERAIN** [297]		**DEBLUR** [207]		**Mean**	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DL	20.54	0.826	21.96	0.762	19.86	0.672	20.78	0.753
Tweather	21.32	0.885	29.43	0.905	25.12	0.757	25.29	0.849
TAPE	22.16	0.861	29.67	0.904	24.47	0.763	25.43	0.843
AirNet (multi-task)	21.04	0.884	32.98	0.951	24.35	0.781	26.12	0.872
AirNet (uni-task)	23.18	0.900	34.90	0.966	26.42	0.801	28.17	0.889
AirNet + Ours	**24.96**	**0.929**	**36.19**	**0.972**	**27.29**	**0.827**	**29.48**	**0.909**
% Improvement	+7.68	+3.22	+3.70	+0.60	+3.29	+3.25	+4.65	+2.25

Table 5.7: **Prior Induction:** Our factors can induce structure prior in an existing base model [164] and improve performance for multiple enhancement tasks. Here we show are results on the three enhancement tasks: deraining, dehazing and deblurring and compare them with several other image enhancement methods.

generalization across unseen datasets, along with generalization to other applications like deraining, dehazing *etc*., proves the advantage of zero-reference methods over other types of solutions.

5.4.9 Limitations

Our method is sensitive to initialization conditions like the underlying algorithms [179, 48]. As a heuristic we use dataset mean for initialization. Another idea, to be explored in future, is to dynamically adapt to each input which is expected to further increase the performance.

5.5 Conclusions

In this chapter, we presented a zero-reference LLE method that uses a novel image factorization strategy based on specularity. We learn optimization hyper-parameters in data-driven fashion by unrolling the stages into a small neural network. The factors are combined into the enhanced result using a fusion network. We also demonstrate the use of our factors for image relighting as well as for image enhancement tasks like dehazing, deraining and deblurring. In future, we want to extend our specularity priors to applications like image harmonization, foreground matting, white-balancing, depth estimation, *etc*., and extend the technique to other signals beyond the visible spectrum.

Paradigm	***Supervised LLE***				**Unsupervised LLE**					**Zero Reference**
Method	*URetinex* [286]	*CUE* [319]	*SNR* [291]	*RFormer* [50]	EGAN [131]	HEP [306]	Pair-LIE* [86]	CLIP-LIT [174]	NeRCo* [296]	**RSFNet** (Ours)
					Lolv1 [283] (dataset split: 485/15, mean≈ 0.05, resolution: 400 × 600)					
PSNR$_y$ ↑	*22.16*	*24.57*	*28.33*	*28.81*	19.69	20.82	20.51	14.13	25.53	22.15
SSIM$_y$ ↑	*0.900*	*0.852*	*0.910*	*0.914*	0.764	0.874	0.840	0.659	0.860	0.860
PSNR$_c$ ↑	*19.84*	*21.67*	*24.16*	*25.15*	17.48	20.23	18.47	12.39	22.95	19.35
SSIM$_c$ ↑	*0.824*	*0.769*	*0.840*	*0.843*	0.652	0.790	0.743	0.493	0.784	0.755
NIQE ↓	*3.541*	*3.198*	*4.016*	*2.972*	4.889	3.295	4.038	8.797	3.538	3.146
LPIPS ↓	*0.168*	*0.277*	*0.207*	*0.193*	0.327	0.223	0.290	0.359	0.243	0.265
					Lolv2-real [300] (dataset split: 689/100, mean≈ 0.05, resolution: 400 × 600)					
PSNR$_y$ ↑	*22.97*	*24.48*	*23.20*	*24.80*	21.27	20.87	–	17.03	–	21.59
SSIM$_y$ ↑	*0.900*	*0.848*	*0.893*	*0.888*	0.770	0.860	–	0.696	–	0.843
PSNR$_c$ ↑	*21.09*	*22.56*	*21.48*	*22.79*	18.64	18.97	–	15.18	–	19.39
SSIM$_c$ ↑	*0.858*	*0.799*	*0.848*	*0.839*	0.677	0.808	–	0.533	–	0.745
NIQE ↓	*4.010*	*3.709*	*4.141*	*3.594*	5.503	3.618	–	9.220	–	3.701
LPIPS ↓	*0.147*	*0.270*	*0.199*	*0.228*	0.321	0.218	–	0.328	–	0.278
					Lolv2-synthetic [300] (dataset split: 900/100, mean≈ 0.2, resolution: 384 × 384)					
PSNR$_y$ ↑	*20.35*	*18.48*	*25.89*	*27.66*	18.18	17.69	21.13	17.65	18.55	20.15
SSIM$_y$ ↑	*0.888*	*0.803*	*0.957*	*0.962*	0.843	0.828	0.866	0.840	0.745	0.895
PSNR$_c$ ↑	*18.25*	*16.49*	*24.14*	*25.67*	16.57	15.62	19.07	16.19	16.07	17.18
SSIM$_c$ ↑	*0.821*	*0.734*	*0.927*	*0.928*	0.772	0.752	0.794	0.772	0.673	0.817
NIQE ↓	*4.338*	*4.165*	*3.969*	*3.939*	3.831	4.692	4.946	4.690	3.735	4.404
LPIPS ↓	*0.195*	*0.283*	*0.065*	*0.076*	0.188	0.283	0.224	0.177	0.378	0.159
					Mean Scores (**Lolv1** [283], **Lolv2-real** [300], **Lolv2-syn** [300])					
PSNR$_y$ ↑	*21.83*	*22.51*	*25.81*	***27.09***	19.71	20.46	20.82	16.27	**22.04**	**21.30**
SSIM$_y$ ↑	*0.896*	*0.834*	*0.920*	***0.921***	0.792	**0.854**	0.853	0.732	0.803	**0.866**
PSNR$_c$ ↑	*19.73*	*20.24*	*23.41*	***24.54***	17.56	18.27	18.77	14.59	**19.51**	**18.64**
SSIM$_c$ ↑	*0.834*	*0.767*	***0.872***	*0.870*	0.700	**0.783**	0.769	0.599	0.729	**0.772**

Continued on next page

Table 5.8 – continued from previous page

Method	*URe-tinex* [286]	*CUE* [319]	*SNR* [291]	*RFormer* [50]	EGAN [131]	HEP [306]	Pair-LIE* [86]	CLIP-LIT [174]	NeRCo* [296]	**RSFNet** (Ours)
NIQE ↓	*3.963*	*3.691*	*4.042*	***3.502***	4.741	3.868	4.492	7.569	**3.637**	**3.424**
LPIPS ↓	*0.170*	*0.277*	***0.157***	*0.166*	0.279	**0.241**	0.257	0.288	0.311	**0.234**

Table 5.8: **Quantitative comparison** of our method RSFNet with five other **Unsupervised LLE** solutions [131, 306, 86, 174, 296] and four recent Supervised LLE solutions [286, 291, 319, 50] for reference. Note that the latter two categories require both low-light and well-lit images, either unpaired or paired, for supervision during training. The final average scores are presented in the last sub-table. (* For PairLIE [86] and NeRCo [296], training set includes Lolv2 test images, hence the results are not estimated for Lolv2 and average computed using other two scores. Even with zero-reference training requirements, our method (last column) is able to perform competitively against all unsupervised solutions. For [296] and [306], our method beats both of them separately on 4/6 and 5/6 metrics. Note that supervised solutions require significantly more supervision information during training and can not be compared directly with other categories. Here they are shown only for reference (Best score in each category here is in **bold** in the last sub-table. Our method in the last column gives the best mean results among Zero-Reference methods as shown elsewhere.).

NIQE ↓	**SNR** [291]	**RFormer** [50]	**HEP** [306]	**NeRCo** [296]	**RSFNet** (Ours)
DICM [158]	3.622	3.076	4.064	3.553	3.230
LIME [109]	3.752	3.910	3.981	3.422	3.800
MEF [190]	3.917	3.135	3.648	3.152	3.000
NPE [277]	3.535	3.63*	2.986	3.241	3.310
VV [273]	2.887	2.183	3.596	3.169	1.960
Mean	3.543	<u>3.187</u>	3.655	3.307	**3.060**

Table 5.9: **Generalized Performance:** Performance generalization comparison (Table 5.6 extension) of best ranking (Table 5.8) two supervised LLE solutions (first two columns: SNR [291], RFormer [50]) and two unsupervised LLE solutions (last two columns: HEP [306], NeRCo [296]) *vs*. our zero-reference RSFNet method on five no-reference benchmarks namely: DICM [158], LIME [109], MEF [190], NPE [277] and VV [273]. Our method is able to generalize better to unseen data compared to others as observed from the overall lowest NIQE scores [202] in the last row. (SNR, HEP and NeRCo results computed using provided pretrained weights with Lolsyn checkpoint where ever applicable and all images resized to 512x512 before processing to avoid dataloader errors. For RFormer, results downloaded from their official homepage. * refers to the incomplete NPE dataset results as available).

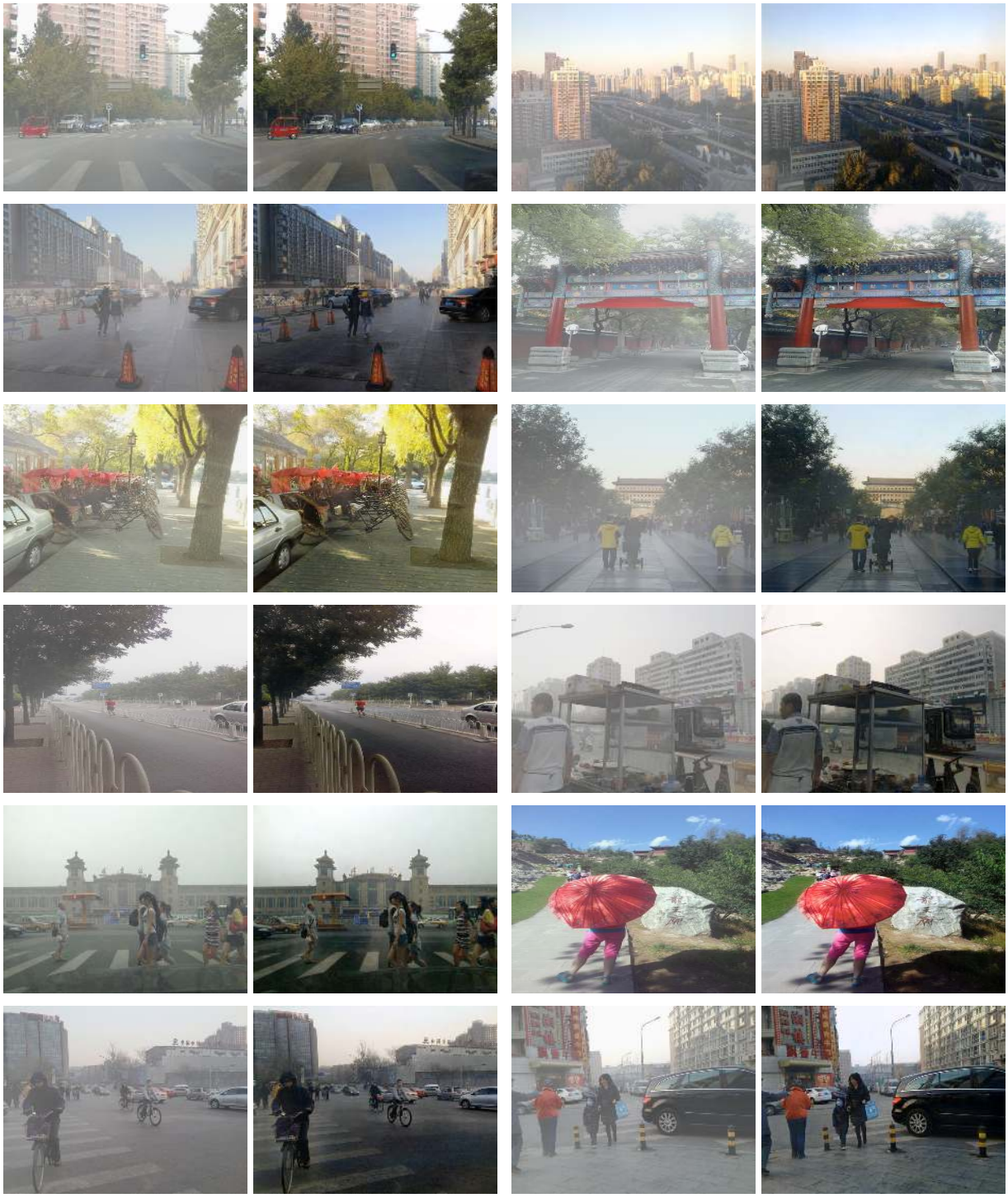


Figure 5.15: Our Dehazing Results: Additional results (Fig. 5.13 extension) for the dehazing application on the RESIDE dataset [163].

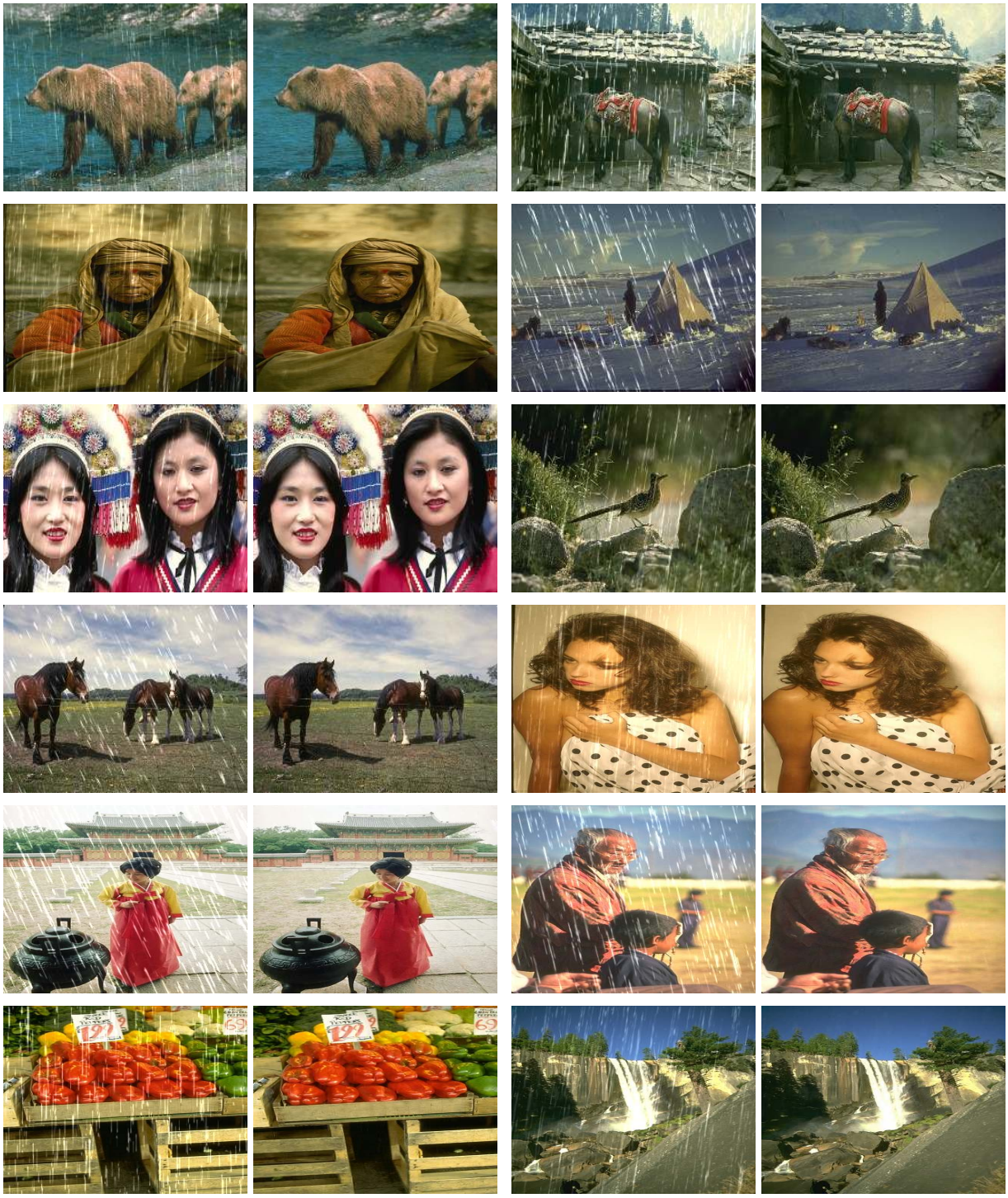


Figure 5.16: Our Deraining Results: Additional results (Fig. 5.13 extension) for the deraining application on the Rain100L dataset [297].

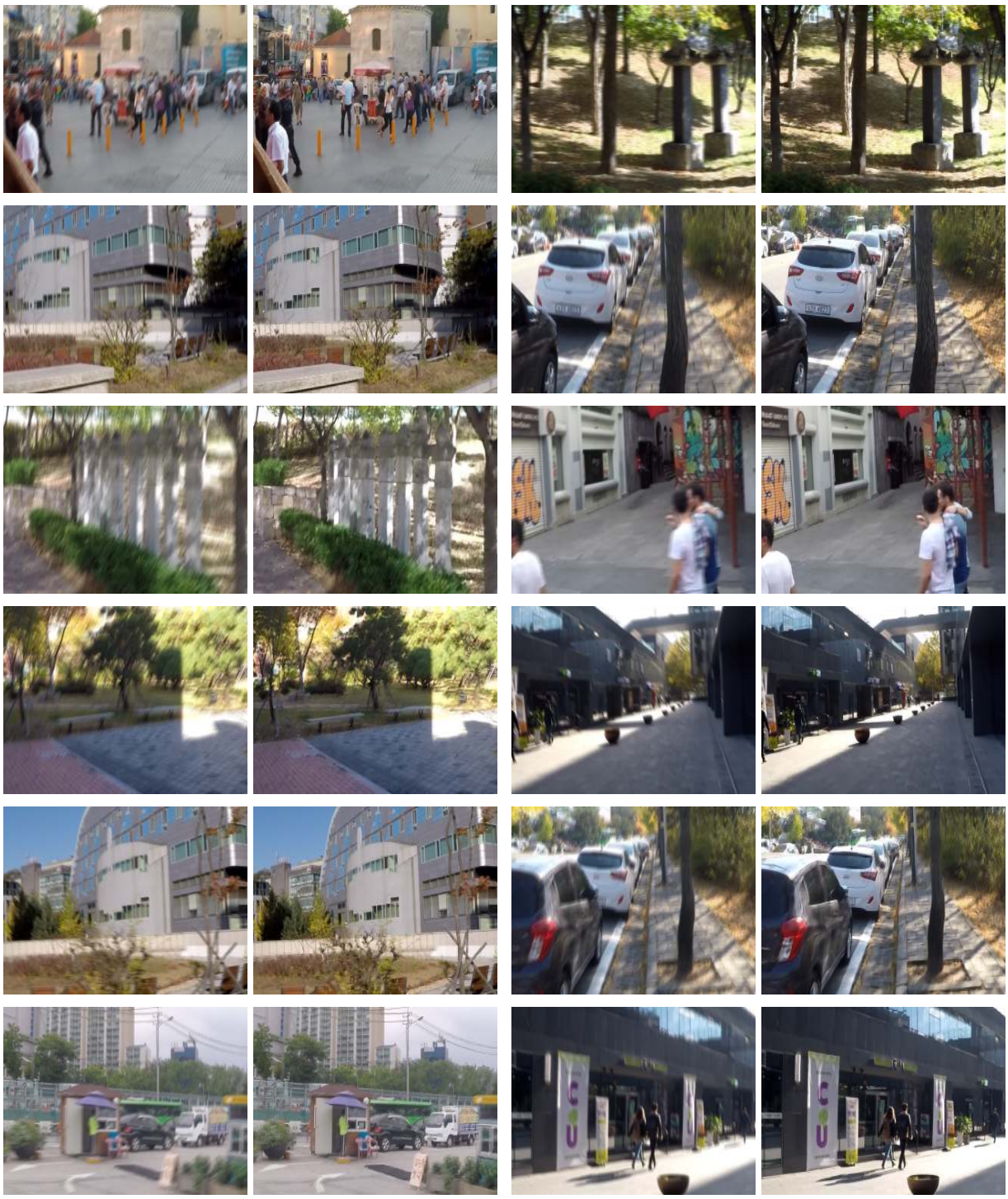


Figure 5.17: Our Deblurring Results: Additional results (Fig. 5.13 extension) for the deblurring application on the GoPro dataset [207].

PART IV

Analysis and Summary

Chapter 6

Concept Distillation for Prior Induction

In the last two units comprising of previous four chapters we have proposed multiple solutions from both tradition model-based and contemporary model-driven paradigms. The two Inverse Rendering problems discussed earlier *i.e*. Intrinsic Image Decomposition (IID) and Low Light Enhancement (LLE), transit through ill-defined disentangled components like reflectance, shading, iterative specularity *etc*. This abstract nature of the latent factors makes it difficult to directly evaluate the quality of factorization. Hence, the often used performance evaluation is done either by estimating success in a downstream application or by using synthetic benchmarks. This type of evaluation is sub-optimal as the factorization components have wider applicability beyond the single chosen application.

In this chapter, we present *Concept Distillation*, a novel method to enable concept-sensitive training of neural networks to induce human-centered conceptual knowledge in pre-existing frameworks. 'Concepts' are defined as human-interpretable groupings of entities which are inherent to a particular class [120]. Such concepts present necessary condition for that particular class and are often a crucial distinguishing variable during classification. Concepts are often not defined explicitly but is often implicitly captured in good feature extractors [66, 53]. In order to solve the problem of factorization quality assessment, we propose to use appropriately defined learn-able concepts. This is helpful as humans interpret think in terms of these abstract concepts instead of hard features. More concretely, we identify fundamental concepts associated with the disentanglement and learn Concept Activation Vectors (CAVs) [139], which represent the concept as a vector in the activation space of the learned model. CAVs indicate the direction in the activation vector space which maximizes the sensitivity of the model for that particular concept. CAVs can be learned using a small representative set of images embodying the concept as provided by the user. This exemplar based definition makes it easy to estimate the CAV and interpret the model. CAVs can be used to determine the sensitivity of the model towards a particular concept and identify learned shortcuts or biases. Though originally intended only for post-hoc (*i.e*. after training) model analysis, we extend the usage of CAVs for ante-hoc (*i.e*. during training) knowledge distillation. We distill the conceptual knowledge from a pretrained knowledge-

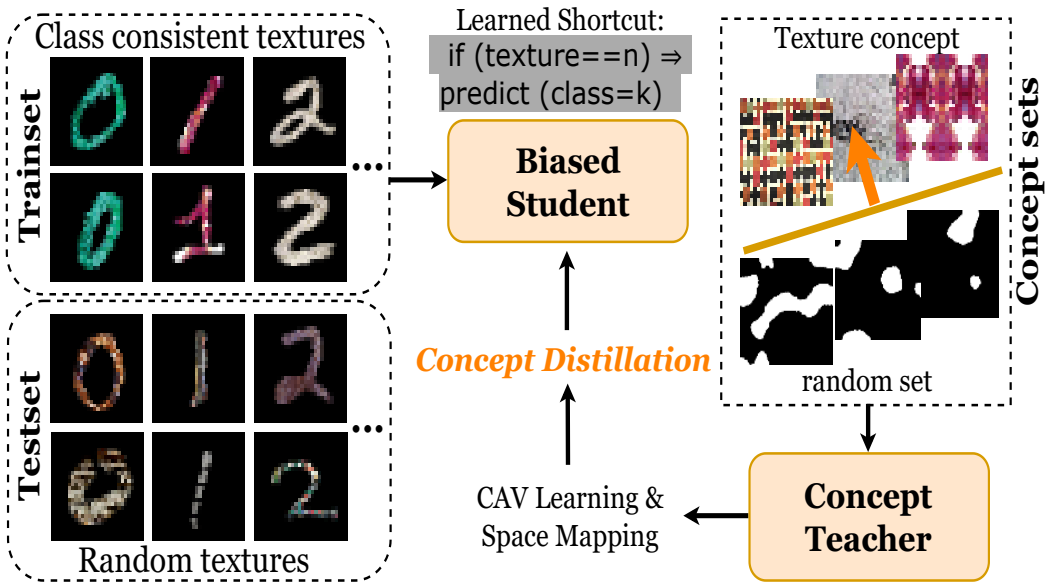


Figure 6.1: Concept Distillation overview: The generic conceptual knowledge of a capable teacher can be distilled to a student for performance improvement through bias removal and prior induction.

able teacher model, well-versed in various concepts, to a smaller student model, focusing on a single downstream task. Our method enables us to sensitize or desensitize the student model toward specific concepts. We show applications of our proposed concept-sensitive training in mitigating model bias in simple classification problems and inducing prior knowledge in a complex computer IID reconstruction problem. We provide benchmarked results of various training methods using this dataset. Through our simple technique, we aim to establish the significance of concept-sensitive training and its potential to improve model interpretability, reduce biases, and induce prior knowledge.

Note that this chapter mainly discusses our concept distillation work [111] and parts of this chapter are shared with other author's thesis. The main experimentation section discussing induction of concepts as priors in the computer vision problem of intrinsic image decomposition is detailed here and for the machine learning application perspective discussing classification debiasing, the reader is encouraged to refer to the main paper or the other author's thesis.

6.1 Introduction

EXplainable Artificial Intelligence (XAI) methods are useful to understand a trained model's behavior [316]. They open the black box of Deep Neural Networks (DNNs) to enable post-training identification of unintended correlations or biases using similarity scores or saliency maps. Humans,

however, think in terms of abstract *concepts*, defined as groupings of similar entities [120]. Recent efforts in XAI have focused on concept-based model explanations to make them more aligned with human cognition. Kim et al. [139] introduce Concept Activation Vectors (CAVs) using a concept classifier hyperplane to quantify the importance given by the model to a particular concept. For instance, CAVs can determine the model's sensitivity on 'striped-ness' or 'dotted-ness' to classify Zebra or Cheetah using user-provided concept samples. They measure the concept sensitivity of the model's final layer prediction with respect to intermediate layer activations (outputs). Such post-hoc analysis can evaluate the transparency, accountability, and reliability of a learned model [56] and can identify biases or unintended correlations acquired by the models via shortcut learning [139, 19].

The question we ask here is: If CAVs can identify and quantify sensitivity to concepts, can they also be used to improve the model? Can we learn less biased and more human-centered models? In this work, we extend CAVs to ante-hoc model improvement through a novel *concept loss* to desensitize/sensitize against concepts. We also leverage the broader conceptual knowledge of a large pre-trained model as a teacher in a *concept distillation* framework for it.

XAI has been used for ante-hoc model improvement during training [145, 136, 18]. They typically make fundamental changes to the model architecture or need significant concept supervision, making extensions to other applications difficult. For example, Koh et al. [145] condition the model by first predicting the underlying concept and then using it for class prediction. Our method can sensitize or desensitize the model to user-defined concepts without modifications to the architecture or direct supervision.

Our approach relies on the sensitivity of a trained model to human-specified concepts. We want the model to be sensitive to relevant concepts and indifferent to others. For instance, a cow classifier might be focusing excessively on the grass associated with cow images. If we can estimate the sensitivities of the classifier to different concepts, we can steer it away from irrelevant concepts. We do that using a *concept loss* term L_C and fine-tuning the trained base model with it. Since the base models could be small and biased in different ways, we use *concept distillation* using a large, pre-trained teacher model that understands common concepts better.

We also extend concepts to work effectively on intermediate layers of the model, where the sensitivity is more pronounced. Kim et al. [139] measure the final layer's sensitivity to any intermediate layer outputs. They ask the question: if any changes in activations are done in the intermediate layer, what is its effect on the final layer prediction? They used the final layer's loss/logit to estimate the model sensitivity, as their interest was to study concept sensitivities for interpretable model prediction. This design choice of calculating only final layer sensitivity is justified by their use case of checking for model sensitivity to certain concepts for classification problems aiming for interpretable final model predictions. We, on the other hand, aim to fine-tune a model by (de)sensitizing it towards a given concept that may be strongest in another layer [16]. Thus, it is crucial for us to measure the sensitivity

in *any* layer by evaluating the effect of the changes in activations in one intermediate layer on another. We employ prototypes or average class representations in that layer for this purpose. Prototypes are estimated by clustering the class sample activations [52, 295, 167, 215, 209, 262, 170]. Our method, thus, allows intervention in any layer.

In this chapter, we present a simple but powerful framework for model improvement using concept loss and concept distillation for a user-given concept defined in any layer of the network. We leverage ideas from post-hoc global explanation techniques and use them in an ante-hoc setting by encoding concepts as CAVs via a teacher model. Our method also admits sample-specific explanations via a local loss [231] along with the global concepts whenever possible. We improve state-of-the-art on classification problems like ColorMNIST and DecoyMNIST [228, 75, 231, 19], resulting in improved accuracies and generalization. We introduce and benchmark on a new and more challenging TextureMNIST dataset with texture bias associated with digits.

We demonstrate concept distillation on two applications:

1. Model debiasing extreme biases on classification problems involving synthetic MNIST datasets [169, 75] and complex and sensitive age-*vs*.-gender bias in the real-world gender classification on BFFHQ dataset [141].

2. Task prior induction by infusing domain knowledge in the reconstruction problem of Intrinsic Image Decomposition (IID) [154] by measuring and improving disentanglement of albedo and shading concepts.

To summarize, our contributions in this work are:

- We extend CAVs from post-hoc explanations to ante-hoc model improvement method to sensitize/desensitize models on specific concepts without changing the base architecture.

- We extend the model CAV sensitivity calculation from only the final layer to *any* layer and enhance it by making it more global using prototypes.

- We introduce concept distillation to exploit the inherent knowledge of large pretrained models as a teacher in concept definition.

- We present benchmark results on standard biased MNIST datasets and on a challenging TextureMNIST dataset that we introduce.

- We show application on a severely biased classification problem involving age bias.

- We show application beyond classification to the challenging multi-branch Intrinsic Image Decomposition problem by inducing human-centered concepts as priors. To the best of our knowledge, this is the first foray of concept-based techniques into non-classification problems.

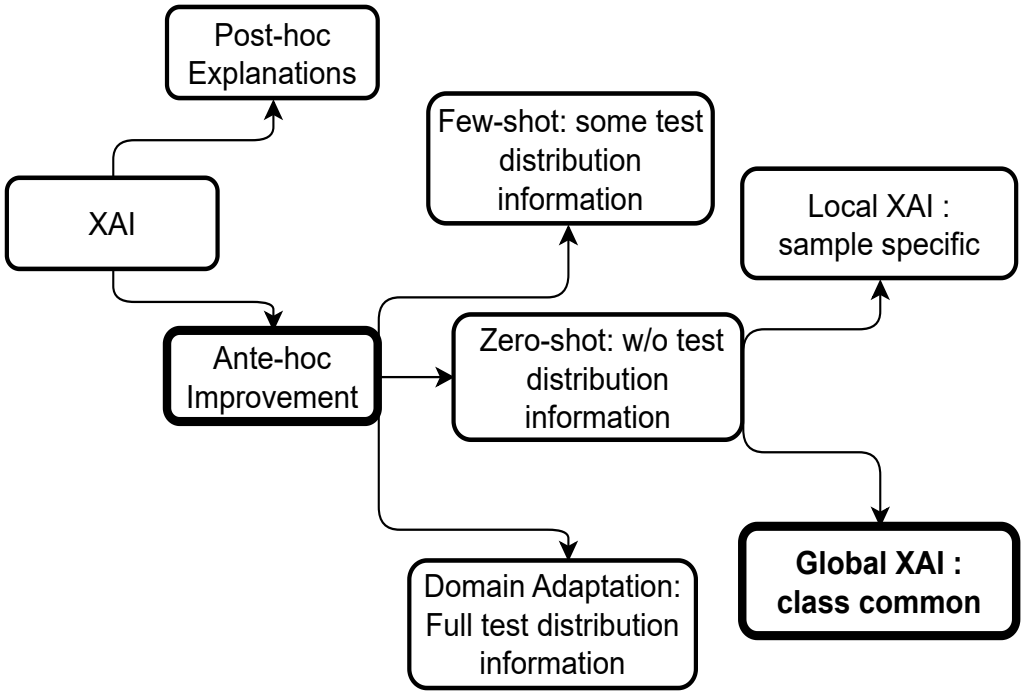


Figure 6.2: XAI Categorization: We can divide XAI techniques into two types: those who do post-hoc explanations *i.e*. inference model analysis after training and those who can do ante-hoc model improvement *i.e*. model updation during training. Ante-hoc methods can be further classified into many-shot, few-shot and zero-shot categories based on whether test domain information is available for explanation generation. These techniques can be either local or global based on whether the explanation is sample specific or applicable to the entire class. Our proposed Concept Distillation method is an ante-hoc, zero-shot global XAI technique.

6.2 Background

6.2.1 XAI Methods

Various methods relevant to the background literature are categorized in Fig. 6.2 in which we show taxonomy of XAI techniques.. Post-hoc (after training) explainability methods include Activation Maps Visualization [247], Saliency Estimation [237], Model Simplification [285], Model Perturbation [81], Adversarial Exemplar Analysis [99], *etc*. Several recent surveys provide a comprehensive discussion on such post-hoc XAI methods [316, 68, 177, 272, 12]. More specifically, concept-based interpretability methods are surveyed by Hitzler and Sarker [120], Schwalbe [246], Holmberg et al. [122].

The ante-hoc model improvement techniques, based on the amount of test distribution information required, can be divided into three categories:

- Zero-shot ante-hoc XAI *e.g*. [287]
- Few-shot ante-hoc XAI *e.g*. [279]
- Multi-shot ante-hoc XAI (domain adaptation) *e.g*. [275]

In zero-shot training, the model never sees the distribution of test samples [287], while in few-shot, the model has access to some examples from the distribution of the test set [279]. Our method is an ante-hoc zero-shot model improvement method and works with abstract concept sets (different than any test sample). Being essentially a zero-shot method, our methods does not require any test data but can can be easily configured to take advantage of few-shot examples if available.

6.2.2 Ante-hoc Model Improvement

Although there is a plethora of post-hoc explanation techniques, only a handful of explainability methods exists which allow model improvement during training. Ross et al. [231] penalize a model if it does not prefer the Right answers for Right Reasons (RRR) by using explanations as constraints. EG Erion et al. [75] augment gradients with a penalizing term to match with user-provided binary annotations. Both of these methods require user annotations or constraints, restricting their wider applicability. More details are available in relevant surveys [88, 282, 113, 31].

Based on the scope of explanation, the existing XAI methods can be divided into global and local methods [316]. Global methods [139, 92] provide explanations that are true for all samples of a class (*e.g*. 'stripiness' concept is focused for the zebra class or 'red color' is focused on the prediction of zero) while local methods [4, 247, 256, 233] explain each of samples individually often indicating regions or patches in the image which lead the model to prediction (example, this particular patch (red) in this image was focused the most for prediction of zero). Kim et al. [139] quantify concept sensitivity using CAVs followed by subsequent works [258, 312, 245]. Kim et al. [139] sample sensitivity is local (sample specific) while they aggregate class samples for class sample sensitivity estimation, which makes their method global (across classes). We enhance their local sample sensitivity to the global level via using prototypes that capture class-wide characteristics by definition [52].

6.2.3 Concept Based XAI

Only a few concept-oriented deep learning methods train neural networks with the help of concepts [243, 242]. Concept Bottleneck Models (CBMs) [145], Concept-based Model Extraction (CME) [136], and Self-Explaining Neural Networks (SENNs) [18] predict concepts from inputs and use them

to infer the output. CBMs [145] require concept supervision, but CME [136] can train in a partially supervised manner to extract concepts combining multiple layers. SENN [18] learns interpretable basis concepts by approximating a model with a linear classifier. These methods require architectural modifications of the base model to predict concepts. Their primary goal is interpretability, which differs from our model improvement goal using specific concepts *without* architectural changes. Closest to our work is ClArC [19], which leverages CAVs to manipulate model activations using a linear transformation to remove artifacts from the final results. Unlike them, our method trains the DNN to be sensitive to *any* specific concept, focusing on improving generalizability for multiple applications.

DFA by Lee et al. [161] and EnD by Tartaglione et al. [264] use bias-conflicting or out-of-distribution (OOD) samples for debiasing using few-shot generic debiasing. DFA uses a small set of adversarial samples and separate encoders for learning disentangled features for intrinsic and biased attributes. EnD uses a regularization strategy to prevent the learning of unwanted biases by inserting an information bottleneck using pre-known bias types. It is to be noted that both DFA and EnD are neither zero-shot nor interpretability-based methods. In comparison, our method works in both known bias settings using abstract user-provided concept sets and unknown bias settings using the bias-conflicting samples. Our Concept Distillation method is a *concept sensitive training* method for induction of concepts into the model, with debiasing being an application. Intrinsic Image Decomposition (IID) [39, 90] involves decomposing an image into its constituent Reflectance (R) and Shading (S) components [153, 192] which are supposed to be disentangled [192, 110].

We use CAVs for inducing R and S priors in a pre-trained SOTA IID framework [172] with improved performance. To the best of our knowledge, we are the first to introduce prototypes for CAV sensitivity enhancement. However, prototypes have been used in the interpretability literature before to capture class-level characteristics [140, 293, 137] and also have been used as pseudo-class labels before [52, 295, 167, 215, 209, 262, 170]. Some recent approaches [128, 257] use generative models to generate bias-conflicting samples (*e.g.* other colored zeros in ColorMNIST) and train the model on them to remove bias. Specifically, Jain et al. [128] use SVMs to find directions of bias in a shared image and language space of CLIP [222] and use Dall-E on discovered keywords of bias to generate bias-conflicting samples and train the model on them. Song et al. [257] use StyleGAN to generate bias-conflicting samples. Training on bias-conflicting samples might not always be feasible due to higher annotation and computation costs.

One significant difference between such methods [128, 257] is that they propose data augmentation as a debiasing strategy, whereas we directly manipulate the gradient vectors, which is more interpretable. Moayeri et al. [203] map the activation spaces of two models using the CLIP latent space similarity. Due to the generality of the CLIP latent space, this approach is helpful to encode certain concepts like 'cat, dog, man,' but it is not clear how it will work on abstract concepts with ambiguous definitions like 'shading' and 'reflectance' as seen in the IID problem described above.

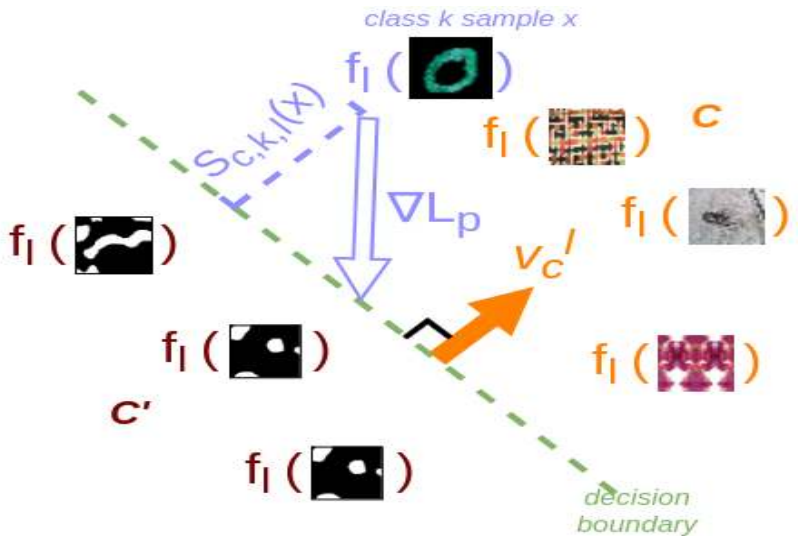


Figure 6.3: Concept Distillation Intuition: For concept distillation we leverage the similarity between concept vector direction and the loss gradient. CAV for a given concept v_c^l is calculated as normal to the separating hyperplane of concept set activations (textures C *vs*. random set C' here). A model biased towards C will have its class samples' loss gradient ∇L_p along v_c^l (measured by sensitivity $S_{C,k,l}(x)$). To desensitize the model for C, we perturb ∇L_p to be parallel to the decision boundary by minimizing the cosine of the projection angle.

6.3 Concept Distillation

Concepts have been used to explain model behavior in a post-hoc manner in the past. Response to abstract concepts can also demonstrate the model's intrinsic preferences, biases, *etc*. Can we use concepts to guide the behavior of a trained base model in desirable ways in an ante-hoc manner? We describe a method to add a *concept loss* to achieve this. We also present concept distillation as a way to take advantage of large foundational models with more exposure to a wide variety of images.

6.3.1 Concept Loss

Building on Kim et al. [139], we represent a concept C using a Concept Activation Vector (CAV) as the normal v_C^l to a linear decision boundary between concept samples C from others C' in a layer l of the model's activation space (Fig. Fig. 6.3). The model's sensitivity is the directional derivative of final layer loss L_o for samples $\boldsymbol{x}$ along v_C^l [139] and given as:

$$S_{C,l}(\boldsymbol{x}) = \nabla L_o \left(f_l(\boldsymbol{x})\right) \cdot v_C^l \tag{6.1}$$

The sensitivity score quantifies the concept's influence on the model's prediction. A high sensitivity for color concepts may indicate a color bias in the model.

These scores were used for post-hoc analysis before by Kim et al. [139]. We use them ante-hoc to desensitize or sensitize the base model to concepts by perturbing it away from or towards the CAV direction (Fig. 6.3). The gradient of loss indicates the direction of maximum change. Nudging the gradients away from the CAV direction encourages the model to be less sensitive to the concept and vice versa. For this, we define a concept loss L_C as the absolute cosine of the angle between the loss gradient and the CAV direction

$$L_C(\boldsymbol{x}) = |\cos\Big(\nabla L_o\left(f_l(\boldsymbol{x})\right), \boldsymbol{v}_C^l\Big)|, \tag{6.2}$$

which is minimized when the CAV lies on the classifier hyperplane (Fig. 6.3). We use the absolute value to avoid introducing the opposite bias by pushing the loss gradient in the opposite direction. A loss of $(1 - L_C(x))$ will sensitize the model to C. We fine-tune the trained base model to desensitize it to concept C for a few epochs using a total loss:

$$L = L_o + \lambda L_C, \tag{6.3}$$

where L_o is the original base model loss used without any modifications.

6.3.2 Concepts Prototypes

Concepts can be present in any layer l, though the above discussion focuses on the sensitivity calculation of the final layer using model loss L_o. The final convolutional layer is proven to learn concepts better than other layers [16]. We can estimate the concept sensitivity of any layer using a loss for that layer. How do we get a loss for an intermediate layer, as no ground truth is available for it?

Class prototypes have been used as pseudo-labels in intermediate layers before [52, 295, 167]. We adapt prototypes to define a loss in intermediate layers. Let $f_l(x)$ be the activation of layer l for sample x. We group the $f_l(x)$ values of the samples from each class into K clusters. The cluster centers P_i together form the prototype for that class. We then define prototype loss for each training sample x using the prototype corresponding to its class as:

$$L_p(x) = \frac{1}{K}\sum_{k=1}^{K} \|f_l(x) - P_k\|^2. \tag{6.4}$$

We use L_p in place of L_o in Eq. (6.2) to define the concept loss in layer l. The prototype loss facilitates the use of intermediate layers for concept (de)sensitization. Prototypes also capture sample sensitivity at a global level using all samples of a class beyond the sample-specific levels. We update the prototypes after a few iterations as the activation space evolves. If P^n is the prototype at Step n and P^c the cluster centres using the current $f_l(x)$ values, the next prototype update for each cluster center k is:

$$P_k^{n+1} = (1 - \alpha)P_k^n + \alpha P_k^c. \tag{6.5}$$

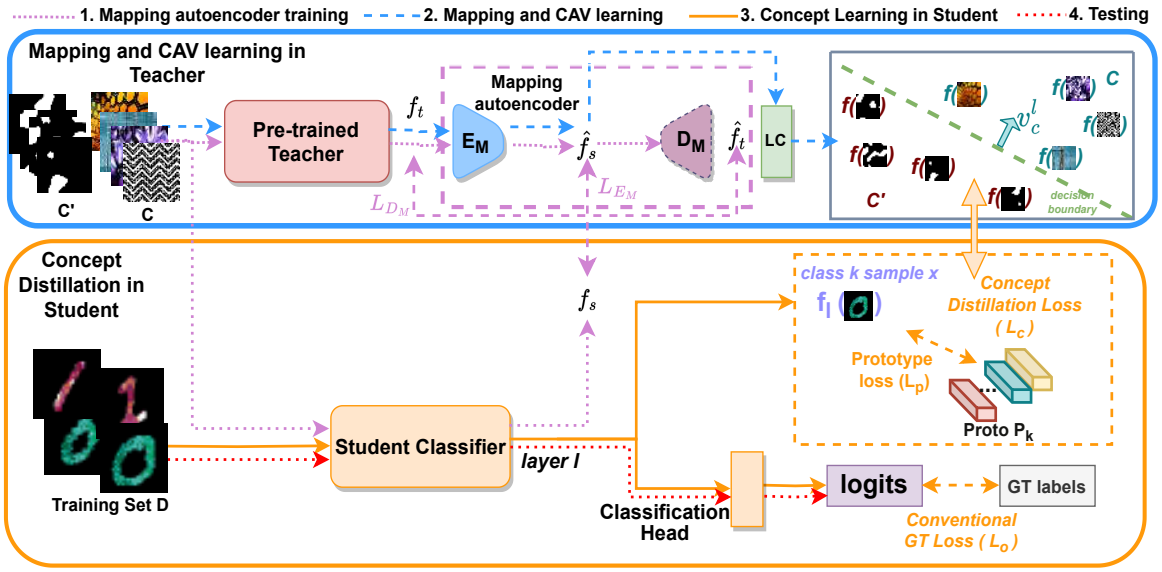


Figure 6.4: Block Diagram: Our framework comprises a concept teacher and a student classifier and has the following four steps: 1) Mapping teacher space to student space for concepts C and C' by training an autoencoder E_M and D_M (dotted purple lines); 2) CAV ($\boldsymbol{v}_c^l$) learning in mapped teacher space via a linear classifier LC (dashed blue lines); 3) Training the student model with Concept Distillation (solid orange lines): We use $\boldsymbol{v}_c^l$ and class prototypes loss L_p to define our concept distillation loss L_c and use it with the original training loss L_o to (de)sensitize the model for concept C; 4) Testing where the trained model is applied (dotted red lines)

6.3.3 Concept Teacher

Concepts are learned from the base model in the above formulation. Base models may have wrong concept associations due to their training bias or limited exposure to concepts. Can we alleviate this problem using a larger model that has seen vast amounts of data as a teacher in a distillation framework? We propose to use a pre-trained teacher model for concept extraction.

6.3.3.1 Teacher Model

For teacher selection, we experimented with various model architectures and chose a pre-trained DINO transformer [53] due to its easy scalability. DINO features are known to be quite generalized and have been proven to work well for a variety of tasks [269, 281]. It is a self-supervised model trained on a large number of images, as the teacher and the base model as the student for concept distillation. A large model knows a variety of concepts due to large dataset supervision and can be

used as a teacher for various tasks. We use the same DINO teacher on very different classification datasets like biased MNIST (ColorMNIST, DecoyMNIST, and TextureMNIST) and BFFHQ, as well as over a completely different problem of IID.

Among the DINO variants, we found ViT-B/8 (85M parameters) to perform the best, aiding student to get 50.93% accuracy on ColorMNIST while ViT-S/8 (21M parameters) aced 39% student accuracy. We thus picked DINO ViT-B/8 for all our experiments. We used the code implementation of the DINO feature extractor by Tschernezki et al. [269] and loaded the checkpoints for DINO variants from Caron et al. [53]. The DINO ViT-B/8 gives 768-dimensional feature images, further reduced to 64 using PCA.

6.3.3.2 Mapping Module

The teacher and student models typically have different activation spaces. We map the teacher space to the student space before concept learning. The mapping uses an autoencoder [119] consisting of a simple encoder E_M and a decoder D_M architecture Fig. 6.4. For the mapping module, we choose a pair of one down-convolutional and up-convolutional layers as Encoder and decoder (depending on students and teacher's dimensions, it is determined whether the Encoder is up or down-convolutional and vice versa). In our experiments, we train the autoencoder for a maximum of five epochs and select the Encoder from the best of the first five epochs as our activation space mapping module M. We also tried with other deeper autoencoder architectures in our initial experiments but found the above simple one to give good results while being computationally cheapest. Due to the simple architecture (logistic regression or single up-down convolutions), both our CAV learning and Mapping module training are very lightweight ($<$120K parameters, $<$10MB and $<$1MB) and train within a minute for 10-15 concept sets having a number of images between 50-200 on a single 12GB Nvidia 1080 Ti GPU. The autoencoder is trained to minimize the following loss reconstruction loss:

$$L_R = L_{D_M} + L_{E_M}. \tag{6.6}$$

L_{D_M} is the pixel-wise L_2 loss between the original (f_t) and decoded ($\hat{f}_t$) teacher activations and L_{E_M} is the pixel-wise L_2 loss between the mapped teacher ($\hat{f}_s$) and the student (f_s) activations. The mapping is learned over the concept set of images C and C' (dashed purple lines in Fig. 6.4).

6.3.3.3 CAV Estimation

Next, we learn the CAVs in the distilled teacher space $\hat{f}_s$, keeping the teacher, student, and mapping modules fixed. This is computationally light as only a few (50-150) concept set images are involved. The learned CAV is used in concept loss given in Eq. (6.2). Please note that E_M is used only to align

the two spaces and can be a small capacity encoder, even a single layer trained in a few epochs. Specifically, we use a logistic regression implemented by a single perceptron layer with sigmoid activation. We train it to distinguish between model activations of concept set (C) and its negative counterpart (C') in layer l using a cross-entropy loss for binary classification. This is theoretically the same with a slight implementation difference from Kim et al. [139], who directly use logistic regression from sklearn library.

Kim et al. [139] use t-testing with concept *vs*. multiple random samples to filter robust CAVs for TCAV score estimation. Similar to them, we employed t-testing initially by taking concept vs multiple random samples and selecting only the significant CAVs. This proved to be too expensive computationally during training, especially during frequent CAV updates. Currently, we have a simple filter on CAV classification accuracy > 0.7 to select only the good CAVs *i.e*., CAVs that can differentiate concept vs random. The concept loss corresponding to all such valid CAVs is then averaged before back propagation. This design simplification was empirically verified and found to work equivalent to T-testing heuristic from Kim et al. [139].

Note that CAVs can be calculated for any model layer. The main question then is: Which student layers should be used? We show results only for the last convolution layer of the student model in this work, but theoretically, our framework can be extended to any number of layers at any depth. Our design choice is based on the fact that the deeper layers of the model encode complex higher-order class-level features while the shallower layers encode low-level features. The last convolution layer represents more abstract features, which are easily represent-able for humans in the form of concepts rather than low-level features in other layers. For the same reasons, Akula et al. [16] too use the last convolution layers for conceptual explanation generation.

6.4 Experiments

Till now we have discussed how we learn concept representation as CAVs and distill them from a learned teacher model into a smaller student model. In this section, we explore the utility of our concept distillation technique for two problem categories. In the first category, we focus on classification improvement for scenarios where the test data distribution is unknown or the train data distribution is specifically poisoned with class irrelevant features *e.g*. constant color for each digit in colorMNIST training set which is not present or is randomized in the testing set. A standard classifier will learn the shortcut with various colors mapped to the digit shapes leading to incorrect results during inference. This requires debiasing the learned classifier model away from the incorrectly learned shortcut. In the second category, we extend our concept distillation strategy to the reconstruction problems Section 1.2.1. We show how we can use abstract and ill-defined terms to fine-tune a learned model to induce specific priors in the system, thereby improving the overall performance of the model. Specif-

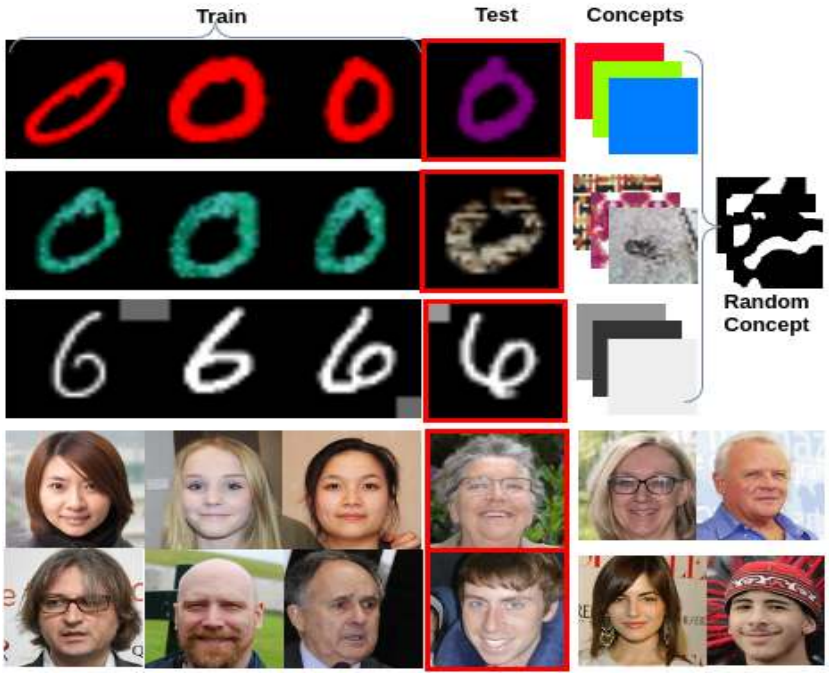


Figure 6.5: Debiasing Datasets: ColorMNIST (top row), TextureMNIST (next row), DecoyMNIST (third row), and BFFHQ (bottom rows). Concepts used include color, textured and gray patches, and bias-conflicting samples shown on the right.

ically, we improve pre-trained IID models by infusing reflectance and shading specific priors into the model learning appropriate concepts based on the definition of the two concepts. Our concept distillation strategy is more general than this and apart from these two categories mentioned here, it can be used in various other computer vision applications and machine learning problems.

6.4.1 Model Debiasing

In a biased training setting, the classifier has access to only poisoned training dataset. Training data is poisoned by associating spurious redundant information in each class *e.g.* cow *vs.* cat classifier learning green color background for cow detection.

Although presented here as an exaggerated scenario, such an effect is possible in various classifiers due to data scarcity, lack of class variety, limited model capacity or structured class specific noise. One naive way to rectify this issue is to retrain the student classifier model with more diverse data but this simple solution is not feasible in all cases. We present concept distillation as a solution to this issue by identifying the necessary concepts needed for model debiasing *e.g.* in case of poisoned MNIST datasets, it could be concepts of color and random shapes. After identification, we represent

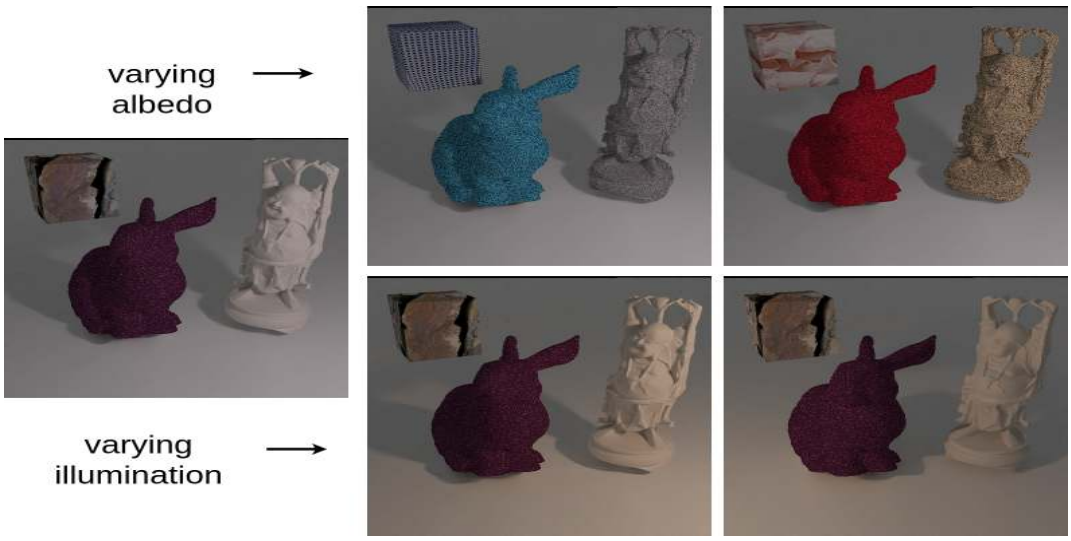


Figure 6.6: Concept Sets: Synthetically rendered concept sets for IID prior induction. We generate two different type of images for each scene. One is for albedo-invariance concept and the other one is for shading-invariance concept. The former includes images of a scene with randomly changing material color for various objects but constant illumination. The latter on the other hand includes images of a scene with varying illumination in intensity and direction but constant object color.

these concepts as small image sets (< 100) which could be synthetically created as random solid color patches and binary shape blobs Fig. 6.5). These are then used to learn the Concept Activation Vectors (CAVs) in the activation space in a large pre-trained teacher model. This teacher CAV is expected to be more generalized and have a better understanding of concepts of color and shape compared to the smaller biased student model. Teacher CAV is then mapped to the chosen student activation space representing the last convolution layer of the student model before the softmax classification layer. Now our designed concept loss can be used in junction with the original student model loss for a fine-tuning iterations to infuse the student with the required knowledge to rectify its bias. This improves the student performance on the final testing set thereby increasing its generalizability.

Note that this distillation can be carried out in zero-shot setting without any access to testing set samples or even in the case of few-shot setting when some test samples are available which can be used to create the concept sets per class by assembling two sets comprising of class specific samples from the training *vs*. testing sets.

All these experiments are detailed extensively in the other student author's thesis and the associated paper [111], which focuses specifically on the classification debiasing application. Here we focus on the second application category and explore its utility in the IID reconstruction problem as detailed in the next section.

6.4.2 Prior Induction

Model	MSE R ↓	MSE S ↓	SSIM R ↑	SSIM S ↑	Synthetic		Real-World
					CSM R ↑	CSM S ↑	CSM R ↑
CGIID [172]	0.066	**0.027**	0.536	0.581	1.790	0.930	5.431
CGIID++	0.080	0.032	0.520	0.552	0.860	0.401	3.421
Ours (R only)	**0.052**	**0.027**	**0.54**	0.581	1.889	**1.260**	4.89
Ours (R & S)	0.059	0.028	0.538	**0.586**	**6.040**	1.023	**78.46**

Table 6.1: **Prior induction in IID:** Inducing human-centered concepts like albedo-invariance of S and illumination invariance of R results in improved IID performance.

To recap, Intrinsic Image Decomposition (IID) is an inverse rendering problem [29] based on the retinex theory [153], which suggests that an image I can be divided into Reflectance R and Shading S components such that $I = R \cdot S$ at each pixel. Here R represents the material color or albedo, and S represents the scene illumination at a point. As per definition, R and S are *disentangled*, with R invariant to illumination and S invariant to albedo. As good ground truth for IID is hard to create, IID algorithms are evaluated on synthetic data or using some sparse manual annotations [33, 147]. Gupta et al. [110] use CAV-based sensitivity scores to measure the disentanglement of R and S. They create concept sets of *(i) varying albedo*: wherein the albedo (material color) of the scene is varied*(ii) varying illumination*: where illumination is varied.

From the definition of IID, Reflectance shall only be affected by albedo variations and not by illumination variations and vice-versa for Shading. They defined *Concept Sensitivity Metric (CSM)* to measure R-S disentanglement and evaluate IID methods post-hoc. Using our concept distillation framework, we extend their post-hoc quality evaluation method to ante-hoc training of the IID network to increase disentanglement between R and S. We train in different experimental settings wherein we only train the R branch (R only) and both R and S branches together ($R\&S$) with our concept loss in addition to the original loss [172] and report results in Table 6.1.

We choose the state-of-the-art CGIID [172] network as the baseline. The last layer of both R and S branches is used for concept distillation, and hence no prototypes are used. Following Li and Snavely [172], we fine-tune over the CGIntrinsics [172], IIW [33], and SAW [147] datasets while we report

results over ARAP dataset [39], which consists of realistic synthetic images. For L_o, we adopt the same losses as Li and Snavely [172], which consist of an IIW loss, GT Reconstruction loss, and a SAW loss. For concept sets, we follow [110]. They create the concepts of *varying albedo* and *varying illumination* on different objects and scenes keeping the viewpoint fixed. They create scenes consisting of single object and multiple objects and render them in blender. Following them, we use 100 images per scene for *varying albedo* concept and 44 for *varying illumination* concept. We train the model for 5-10 epochs, saving the best checkpoint (on validation). Our method converges in 16-20 hours on two 12GB Nvidia 1080 Ti GPUs.

We also train the baseline model for additional epochs to get CGIID++ for a fair comparison. Table 6.1 shows different measures to compare two variations of our method with CGIID and CGIID++ on the ARAP dataset [39]. We see improvements in MSE and SSIM scores which compare dense-pixel wise correspondences CSM scores [110] which evaluate R-S disentanglement. Specifically, CSM_S measures albedo invariance of S and CSM_R measures illumination invariance of R predictions. We report CSM scores in two concept set settings of Synthetic vs Real-World from Gupta et al. [110]. The improvement in MSE and SSIM scores appear minor quantitatively, but our method performs significantly better in terms of CSM scores. It also observes superior performance qualitatively, as seen in (Fig. 6.7). Our R outputs are less sensitive to illumination, with most of the illumination effects captured in S.

6.4.3 Limitations:

Our concept distillation framework can work on different classification and reconstruction problems, as we demonstrated. Our method can work well in both zero-shot (with concept sets) and few-shot (with bias-conflicting samples) scenarios. Bias-conflicting samples may not be easy to obtain for many real-world applications. Our required user-provided concept samples can incur annotation costs, though concept samples are usually easier to obtain than bias-conflicting samples. When neither bias-conflicting samples nor user-provided concept sets are available, concept discovery methods like ACE [94] could be used. ACE discovers concepts used by the model by image super-pixel clustering. Automatic bias detection methods like Bahadori and Heckerman [22] can be used to discover or synthesize bias-conflicting samples for our method. Our method can also be used to induce prior knowledge into complex reconstruction/generation problems, as we demonstrated with IID. The dependence on the teacher for conceptual knowledge could be another drawback of our method, as with all distillation frameworks [118].

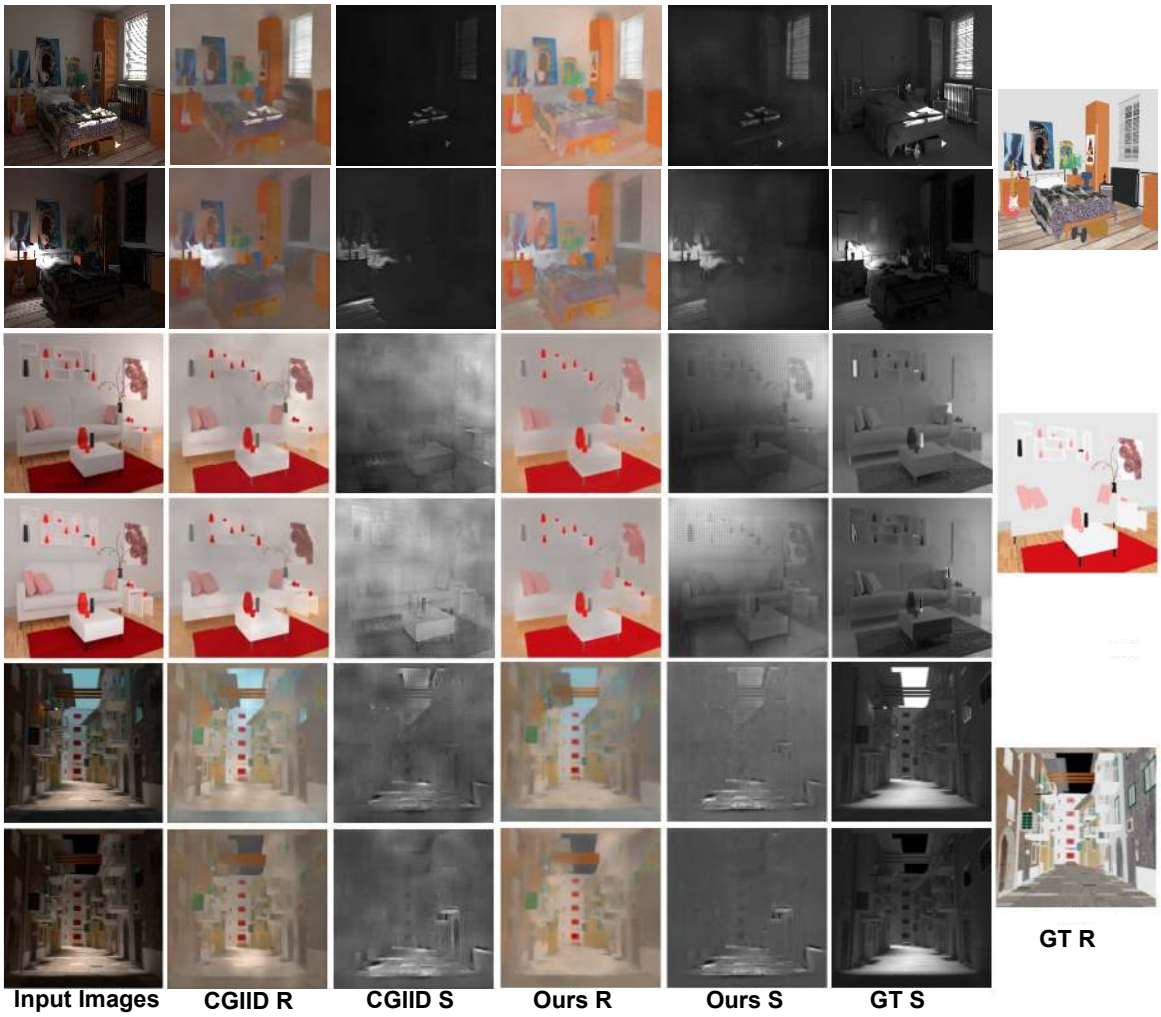


Figure 6.7: Qualitative IID results: Our method uses complex concepts like albedo and illumination to enhance $\hat{R}$ and $\hat{S}$ predictions that are illumination and albedo invariant respectively. First column shows two input images of the same scene under varying illumination. Next two columns are Reflectance and Shading results from baseline method, followed by ours. Last two columns show the two ground truth shading components and the common reflectance image. Our method is able to make the $\hat{R}$ less sensitive to illumination (thereby removing the concept of illumination from $\hat{R}$ and during this $\hat{R}$ predictions become flatter without specifically introducing the flatness prior suggesting disentanglement of R-S is a better way to improve IID. Also, the illumination information removed from $\hat{R}$ is introduced in $\hat{S}$.

6.5 Conclusion

We presented a concept distillation framework that can leverage human-centered explanations and the conceptual knowledge of a pre-trained teacher to distill explanations into a student model. Our method can desensitize ML models to selected concepts by perturbing the activations away from the CAV direction without modifying its underlying architecture. We presented results on multiple classification problems. We also showed how prior knowledge can be induced into the real-world IID problem. In future, we would like to extend our work to exploit automatic bias detection and concept-set definition. Our approach also has potential to be applied to domain generalization and multitask learning problems which we plan to explore in the future.

Chapter 7

Summary

To summarize, in this thesis we have discussed several image factorization solutions. We start from diffuse only, simpler Lambertian reflection assumption and with multiple input images for intrinsic image decomposition. We then provide the solution for more realistic single but complex single input image setting. Later we dive deeper into the illumination component analysis by discarding the diffuse only assumption and present a novel illumination decomposition model for the low light enhancement task. To this end, we provide two types of solutions starting from complex quaternion based optimization based solution first which we then unroll into a learnable network with data-driven parameter estimation for multiple enhancement tasks. Although several limitations still exist for proposed solutions but we hope that with the discussion and analysis provided here we have helped further the research in this field by raising relevant questions and illuminating future directions. Although most of the factorization discussed here focus on image editing applications but it would be interesting to see their applicability in other problems. It would also be interesting to explore the utility of these factorization methods in the context of other signals apart from the normal RGB images like hyperspectral content, LiDAR data, medical image domain, underwater imagery *etc*. Sample applications discussed here can also be extended to other related applications like material modification, object insertion, image matting, video enhancement *etc*.

Apart from the input domain and application variations, one could also dive deeper into more fundamental questions regarding the nature and quality of disentanglement, recommended type and order of decompositions, mathematical properties of underlying components, relationship of proposed components with pre-understood terms of the literature and certainly myriad other algorithms and networks for optimization.

We hope our work inspires the reader in exploring the image factorization problem both from the fundamental and application perspectives and the insights presented here, even if slightly, illuminate the path ahead for the future researchers.

Chapter 8

List of Publications

Related Publications

- Saurabh Saini, P. J. Narayanan. *Specularity Factorization for Low Light Enhancement.* CVPR (2024).
- Avani Gupta, Saurabh Saini, P. J. Narayanan. *Concept Distillation: Leveraging Human-Centered Explanations for Model Improvement.* NeurIPS (2023).
- Saurabh Saini, P. J. Narayanan. *Quaternion Factroized Simulated Exposure Fusion.* ICVGIP (2022).
- Saurabh Saini, P. J. Narayanan. *Semantic Hierarchical Priors for Intrinsic Image Decomposition.* Arxiv CoRR abs/1908.09530 (2019).
- Saurabh Saini, P. J. Narayanan. *Semantic Priors for Intrinsic Image Decomposition.* BMVC (2018).
- Saurabh Saini, Parikshit Sakurikar, P. J. Narayanan. *Intrinsic image decomposition using focal stacks.* ICVGIP (2016).

Other Publications

- Venkat Adithya Amula, Saurabh Saini, Avani Gupta, Sunayana Samavedam, P. J. Narayanan. *Prototype Guided Backdoor Defense.* (2024) [under review]
- Rahul Goel, Dhawal Sirikonda, Saurabh Saini, P. J. Narayanan. *Interactive Segmentation of Radiance Fields.* CVPR (2023).
- Rahul Goel, Dhawal Sirikonda, Saurabh Saini, P. J. Narayanan. *StyleTRF: Stylizing Tensorial Radiance Fields.* ICVGIP (2022).

- Avani Gupta, Saurabh Saini, P. J. Narayanan. *Interpreting Intrinsic Image Decomposition using Concept Activations.* ICVGIP (2022).
- Hsin-Ping Huang, Hung-Yu Tseng, Saurabh Saini, Maneesh Singh, Ming-Hsuan Yang. *Learning to Stylize Novel Views.* ICCV (2021).
- Aakash K. T., Parikshit Sakurikar, Saurabh Saini, P. J. Narayanan. *A Flexible Neural Renderer for Material Visualization.* Siggraph Asia Technical Briefs (2019).
- Gaurav Mishra, Saurabh Saini, Kiran Varanasi, P. J. Narayanan. *Human Shape Capture and Tracking at Home.* WACV (2018).
- Aditya Singh, Saurabh Saini, Rajvi Shah, P. J. Narayanan. *From Traditional to Modern: Domain Adaptation for Action Classification in Short Social Video Clips.* GCPR (2016).
- Aditya Singh, Saurabh Saini, Rajvi Shah, P. J. Narayanan. *Learning to hash-tag videos with Tag2Vec.* ICVGIP (2016).

Bibliography

[1] Android camera 2 api. https://developer.android.com/guide/topics/media/camera. Accessed: 2023-12-12.

[2] Magic lantern. http://magiclantern.fm/.

[3] Wikimedia commons. https://commons.wikimedia.org/wiki/Main_Page. Accessed: 2023-12-12.

[4] C. Aditya, S. Anirban, D. Abhishek, and H. Prantik. Grad-cam++: improved visual explanations for deep convolutional networks. arxiv 2018. *arXiv preprint arXiv:1710.11063*, 2018.

[5] A. Adler, M. Elad, Y. Hel-Or, and E. Rivlin. Sparse coding with anomaly detection. In *2013 IEEE MLSP*, 2013.

[6] A. Adler, M. Elad, Y. Hel-Or, and E. Rivlin. Sparse coding with anomaly detection. In *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2013.

[7] Adobe Inc. Adobe photoshop, 2023. URL https://www.adobe.com/products/photoshop.html.

[8] M. Afifi, K. Derpanis, B. Ommer, and M. Brown. Learning multi-scale photo exposure correction. In *CVPR*, 2021.

[9] M. Afifi, K. G. Derpanis, B. Ommer, and M. S. Brown. Learning multi-scale photo exposure correction. *CVPR*, 2021.

[10] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *ACM Transactions on Graphics*, 2004.

[11] Y. Akashi and T. Okatani. Separation of reflection components by sparse non-negative matrix factorization. *Computer Vision and Image Understanding*, 146(C):77–85, 2016.

[12] N. Akhtar. A survey of explainable ai in deep visual modeling: Methods and metrics. *ArXiv*, abs/2301.13445, 2023.

[13] Y. Aksoy, T. O. Aydin, A. Smolic, and M. Pollefeys. Unmixing-based soft color segmentation for image manipulation. *ACM Transactions on Graphics*, 36:1–19, 2017.

[14] Y. Aksoy, T.-H. Oh, S. Paris, M. Pollefeys, and W. Matusik. Semantic soft segmentation. *ACM ToG (SIGGRAPH)*, 37(4), 2018.

[15] Y. Aksoy, T.-H. Oh, S. Paris, M. Pollefeys, and W. Matusik. Semantic soft segmentation. *ACM Transactions on Graphics (TOG)*, 37:1 – 13, 2018.

[16] A. Akula, S. Wang, and S.-C. Zhu. Cocox: Generating conceptual and counterfactual explanations via fault-lines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2594–2601, 2020.

[17] D. S. Alexiadis and G. D. Sergiadis. Estimation of motions in color image sequences using hypercomplex fourier transforms. *IEEE Transactions on Image Processing*, 18:168–187, 2009.

[18] D. Alvarez Melis and T. Jaakkola. Towards robust interpretability with self-explaining neural networks. *NeurIPS*, 2018.

[19] C. J. Anders, L. Weber, D. Neumann, W. Samek, K.-R. Müller, and S. Lapuschkin. Finding and removing clever hans: Using explanation methods to debug and improve deep models. *Information Fusion*, 77:261–295, 2022.

[20] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. *CVPR*, 2014.

[21] D. G. Assefa, L. Mansinha, K. F. Tiampo, H. Rasmussen, and K. Abdella. Local quaternion fourier transform and color image texture analysis. *Signal Process.*, 90:1825–1835, 2010.

[22] M. T. Bahadori and D. E. Heckerman. Debiasing concept-based explanations with causal analysis. *arXiv preprint arXiv:2007.11500*, 2020.

[23] J. T. Barron. Shape, albedo, and illumination from a single image of an unknown object. *CVPR*, 2012.

[24] J. T. Barron and J. Malik. Color constancy, intrinsic images, and shape estimation. *ECCV*, 2012.

[25] J. T. Barron and J. Malik. Intrinsic scene properties from a single rgb-d image. *CVPR*, 2013.

[26] J. T. Barron and J. Malik. Intrinsic scene properties from a single rgb-d image. *TPAMI*, 2015.

[27] J. T. Barron, A. Adams, Y. Shih, and C. Hernández. Fast bilateral-space stereo for synthetic defocus. In *CVPR*, 2015.

[28] H. G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. 1978.

[29] H. G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. 1978.

[30] A. S. Baslamisli, P. Das, H.-A. Le, S. Karaoglu, and T. Gevers. Shadingnet: Image intrinsics by fine-grained shading decomposition. *IJCV*, 129(8), 2021.

[31] K. Beckh, S. Müller, M. Jakobs, V. Toborek, H. Tan, R. Fischer, P. Welke, S. Houben, and L. von Rueden. Sok: Harnessing prior knowledge for explainable machine learning: An overview. In *First IEEE Conference on Secure and Trustworthy Machine Learning*, 2023.

[32] S. Bell, K. Bala, and N. Snavely. Intrinsic images in the wild. *ACM Trans. on Graphics (SIGGRAPH)*, 33(4), 2014.

[33] S. Bell, K. Bala, and N. Snavely. Intrinsic images in the wild. *ACM Transactions on Graphics (SIGGRAPH)*, 33(4), 2014.

[34] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte. Deep burst super-resolution. *CVPR*, 2021.

[35] G. Bhat, M. Danelljan, F. Yu, L. V. Gool, and R. Timofte. Deep reparametrization of multi-frame super-resolution and denoising. *ICCV*, 2021.

[36] S. Bi, X. Han, and Y. Yu. An L_1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. *ACM Transactions on Graphics*, 34(4), 2015.

[37] S. Bi, N. K. Kalantari, and R. Ramamoorthi. Deep Hybrid Real and Synthetic Training for Intrinsic Decomposition. *EGSR*, 2018.

[38] N. L. Bihan and S. J. Sangwine. Quaternion principal component analysis of color images. *Proceedings 2003 International Conference on Image Processing*, 1:I–809, 2003.

[39] N. Bonneel, B. Kovacs, S. Paris, and K. Bala. Intrinsic Decompositions for Image Editing. *Computer Graphics Forum (Eurographics State of The Art Report)*, 2017.

[40] A. Bousseau, S. Paris, and F. Durand. User assisted intrinsic images. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 28(5), 2009.

[41] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E.-H. Zahzah. Decomposition into low-rank plus additive matrices for background/foreground separation. *Comput. Sci. Rev.*, 23(C), 2017.

[42] T. Bouwmans, S. Javed, H. Zhang, Z. Lin, and R. Otazo. On the applications of robust pca in image and video processing. *Proceedings of the IEEE*, 106:1427–1457, 2018.

[43] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

[44] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1), 2011.

[45] P. J. Burt. Pyramid-based computer graphics the pyramid representation and computer graphics. 1985.

[46] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. *ECCV*, 2012.

[47] V. Bychkovsky, S. Paris, E. Chan, and F. Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *CVPR*, 2011.

[48] H. Cai, J. Liu, and W. Yin. Learned robust pca: A scalable deep unfolding approach for high-dimensional outlier detection. *NeurIPS*, 34, 2021.

[49] J. Cai, S. Gu, and L. Zhang. Learning a deep single image contrast enhancer from multi-exposure images. *IEEE Transactions on Image Processing*, 27(4), 2018.

[50] Y. Cai, H. Bian, J. Lin, H. Wang, R. Timofte, and Y. Zhang. Retinexformer: One-stage retinex-based transformer for low-light image enhancement. In *ICCV*, 2023.

[51] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58 (3), 2011.

[52] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[53] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[54] T. Celik and T. Tjahjadi. Contextual and variational contrast enhancement. *IEEE TIP*, 20(12), 2011.

[55] T.-S. T. Chan and Y.-H. Yang. Complex and quaternionic principal component pursuit and its application to audio separation. *IEEE Signal Processing Letters*, 23, 2016.

[56] D. T. Chang. Concept-oriented deep learning: Generative concept representations. *arXiv preprint arXiv:1811.06622*, 2018.

[57] J. Chang, R. Cabezas, and J. W. Fisher. Bayesian nonparametric intrinsic image decomposition. *ECCV*, 2014.

[58] Y. Chang, L. Yan, T. Wu, and S. Zhong. Remote sensing image stripe noise removal: From image decomposition perspective. *IEEE Transactions on Geoscience and Remote Sensing*, 54: 7018–7031, 2016.

[59] S. Chaudhuri and A. N. Rajagopalan. *Depth from defocus: a real aperture imaging approach*. Springer Science & Business Media, 2012.

[60] Q. Chen and V. Koltun. A simple model for intrinsic image decomposition with depth cues. In *2013 IEEE International Conference on Computer Vision*, 2013.

[61] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud. Isolating sources of disentanglement in vaes. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NeurIPS. Curran Associates Inc., 2018.

[62] X. Chen, J. Liu, Z. Wang, and W. Yin. Theoretical linear convergence of unfolded ista and its practical weights and thresholds. In *NeurIPS*, 2018.

[63] E. Creager, D. Madras, J.-H. Jacobsen, M. Weis, K. Swersky, T. Pitassi, and R. Zemel. Flexibly fair representation learning by disentanglement. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*. PMLR, 2019.

[64] I. Daubechies, M. Defrise, and C. D. Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57, 2003.

[65] J. Deng, W. Dong, R. Socher, L. jia Li, K. Li, and L. Fei-fei. Imagenet: A large-scale hierarchical image database. *CVPR*, 2009.

[66] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *ICML*, 2014.

[67] P. G. Doyle and J. L. Snell. *Random Walks and Electric Networks*. Number Book 22 in Carus Mathematical Monographs. Mathematical Association of America, 1984.

[68] M. Du, N. Liu, and X. Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77, 2019.

[69] S. Duchêne, C. Riant, G. Chaurasia, J. L. Moreno, P.-Y. Laffont, S. Popov, A. Bousseau, and G. Drettakis. Multiview intrinsic images of outdoors scenes with an application to relighting. *ACM Transactions on Graphics*, 34, 2015.

[70] S. Duchêne, C. Riant, G. Chaurasia, J. L. Moreno, P.-Y. Laffont, S. Popov, A. Bousseau, and G. Drettakis. Multiview intrinsic images of outdoors scenes with an application to relighting. *ACM Transactions on Graphics*, 34(5), 2015.

[71] C. Eastwood and C. K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=By-7dz-AZ`.

[72] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics*, 2004.

[73] T. A. Ell and S. J. Sangwine. Hypercomplex fourier transforms of color images. *IEEE Transactions on Image Processing*, 16:22–35, 2007.

[74] T. A. Ell and S. J. Sangwine. Hypercomplex fourier transforms of color images. *IEEE Transactions on Image Processing*, 16:22–35, 2007.

[75] G. Erion, J. D. Janizek, P. Sturmfels, S. M. Lundberg, and S.-I. Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature machine intelligence*, 3(7):620–631, 2021.

[76] C.-M. Fan, T.-J. Liu, and K.-H. Liu. Half wavelet attention on m-net+ for low-light image enhancement. In *IEEE ICIP*, 2022.

[77] Q. Fan, J. Yang, G. Hua, B. Chen, and D. Wipf. Revisiting deep intrinsic image decompositions. *CVPR*, 2018.

[78] P. Favaro and S. Soatto. A geometric approach to shape from defocus. *IEEE TPAMI*, 27(3): 406–417, 2005.

[79] B. Fei, Z. Lyu, L. Pan, J. Zhang, W. Yang, T. Luo, B. Zhang, and B. Dai. Generative diffusion prior for unified image restoration and enhancement. In *CVPR*, 2023.

[80] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6), 1981.

[81] R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pages 3429–3437, 2017.

[82] G. Fu, Q. Zhang, C. Song, Q. Lin, and C. Xiao. Specular highlight removal for real-world images. *Computer Graphics Forum*, 38(7), 2019.

[83] X. Fu, D. Zeng, Y. Huang, Y. Liao, X. Ding, and J. Paisley. A fusion-based enhancing method for weakly illuminated images. *Signal Processing*, 129, 2016.

[84] X. Fu, D. Zeng, Y. Huang, Y. Liao, X. Ding, and J. W. Paisley. A fusion-based enhancing method for weakly illuminated images. *Signal Process.*, 129:82–96, 2016.

[85] X. Fu, D. Zeng, Y. Huang, X.-P. Zhang, and X. Ding. A weighted variational model for simultaneous reflectance and illumination estimation. In *CVPR*, 2016.

[86] Z. Fu, Y. Yang, X. Tu, Y. Huang, X. Ding, and K.-K. Ma. Learning a simple low-light image enhancer from paired low-light instances. In *CVPR*, 2023.

[87] D. GAO, X. Li, Y. Dong, P. Peers, K. Xu, and X. Tong. Deep inverse rendering for high-resolution svbrdf estimation from an arbitrary number of images. *ACM Trans. Graph.*, 38(4), 2019.

[88] Y. Gao, S. Gu, J. Jiang, S. R. Hong, D. Yu, and L. Zhao. Going beyond xai: A systematic survey for explanation-guided learning. *arXiv preprint arXiv:2212.03954*, 2022.

[89] E. Garces, A. Munoz, J. Lopez-Moreno, and D. Gutierrez. Intrinsic images by clustering. *Computer Graphics Forum (Proc. EGSR)*, 31(4), 2012.

[90] E. Garces, C. Rodriguez-Pardo, D. Casas, and J. Lopez-Moreno. A survey on intrinsic images: Delving deep into lambert and beyond. *Int. J. Comput. Vision*, 130(3):836–868, 2022.

[91] P. V. Gehler, C. Rother, M. Kiefel, L. Zhang, and B. Schölkopf. Recovering intrinsic images with a global sparsity prior on reflectance. *NeurIPS*, 2011.

[92] A. Ghandeharioun, B. Kim, C.-L. Li, B. Jou, B. Eoff, and R. W. Picard. Dissect: Disentangled simultaneous explanations via concept traversals. *arXiv preprint arXiv:2105.15164*, 2021.

[93] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36:1 – 12, 2017.

[94] A. Ghorbani, J. Wexler, J. Zou, and B. Kim. Towards automatic concept-based explanations. *arXiv preprint arXiv:1902.03129*, 2019.

[95] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[96] I. Gkioulekas, S. Zhao, K. Bala, T. Zickler, and A. Levin. Inverse volume rendering with material dictionaries. *ACM Trans. Graph.*, 32(6), 2013.

[97] T. Goldstein and S. Osher. The split bregman method for l1-regularized problems. *SIAM J. Img. Sci.*, 2(2), 2009.

[98] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. URL `http://www.deeplearningbook.org`.

[99] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[100] L. Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11), 2006.

[101] K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *ICML*, 2010.

[102] R. Grosse, M. K. Johnson, E. H. Adelson, and W. T. Freeman. Ground-truth dataset and baseline evaluations for intrinsic image algorithms. In *ICCV*, 2009.

[103] R. Grosse, M. K. Johnson, E. H. Adelson, and W. T. Freeman. Ground-truth dataset and baseline evaluations for intrinsic image algorithms. *ICCV*, 2009.

[104] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. *CVPR*, pages 2862–2869, 2014.

[105] C. Guo, C. Li, J. Guo, C. C. Loy, J. Hou, S. Kwong, and C. Runmin. Zero-reference deep curve estimation for low-light image enhancement. *CVPR*, 2020.

[106] J. Guo, Z. Zhou, and L. Wang. Single image highlight removal with a sparse and low-rank reflection model. In *ECCV*, 2018.

[107] J. Guo, Z. Zhou, and L. Wang. Single image highlight removal with a sparse and low-rank reflection model. In *ECCV*, 2018.

[108] X. Guo, X. Cao, and Y. Ma. Robust separation of reflection from multiple images. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2195–2202, 2014. doi: 10.1109/CVPR.2014.281.

[109] X. Guo, Y. Li, and H. Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE TIP*, 26(2), 2016.

[110] A. Gupta, S. Saini, and P. J. Narayanan. Interpreting intrinsic image decomposition using concept activations. In *Proceedings of the Thirteenth Indian Conference on Computer Vision, Graphics and Image Processing*, ICVGIP '22. Association for Computing Machinery, 2023.

[111] A. Gupta, S. Saini, and P. J. Narayanan. Concept distillation: Leveraging human-centered explanations for model improvement. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[112] S. Hao, X. Han, Y. Guo, X. Xu, and M. Wang. Low-light image enhancement with semi-decoupled decomposition. *IEEE TMM*, 22(12), 2020.

[113] P. Hase and M. Bansal. When can models learn from explanations? a formal framework for understanding the roles of explanation data. *arXiv preprint arXiv:2102.02201*, 2021.

[114] S. W. Hasinoff and K. N. Kutulakos. Light-efficient photography. *IEEE TPAMI*, 33(11), 2011.

[115] C. Hessel. Simulated exposure fusion. *Image Process. Line*, 9:469–482, 2019.

[116] C. Hessel. Simulated Exposure Fusion. *Image Processing On Line*, 9, 2019.

[117] C. Hessel and J.-M. Morel. An extended exposure fusion and its application to single image contrast enhancement. In *WACV*, 2020.

[118] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[119] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[120] P. Hitzler and M. Sarker. Human-centered concept explanations for neural networks. *Neuro-Symbolic Artificial Intelligence: The State of the Art*, 342(337):2, 2022.

[121] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005.

[122] L. Holmberg, P. Davidsson, and P. Linde. Mapping knowledge representations to concepts: A review and new perspectives. *ArXiv*, abs/2301.00189, 2022.

[123] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? *BMVC*, 2014.

[124] X. Hou, J. Harel, and C. Koch. Image signature: Highlighting sparse salient regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:194–201, 2012.

[125] X. Hu, T. Wang, C.-W. Fu, Y. Jiang, Q. Wang, and P.-A. Heng. Revisiting shadow detection: A new benchmark dataset for complex world. *IEEE TIP*, 30, 2021.

[126] J. Huang, Y. Liu, F. Zhao, K. Yan, J. Zhang, Y. Huang, M. Zhou, and Z. Xiong. Deep fourier-based exposure correction network with spatial-frequency interaction. In *ECCV*, 2022.

[127] D. E. Jacobs, J. Baek, and M. Levoy. Focal stack compositing for depth of field control. *Stanford Computer Graphics Laboratory Technical Report*, 1(1):2012, 2012.

[128] S. Jain, H. Lawrence, A. Moitra, and A. Madry. Distilling model failures as directions in latent space. *arXiv preprint arXiv:2206.14754*, 2022.

[129] J. Jeon, S. Cho, X. Tong, and S. Lee. Intrinsic image decomposition using structure-texture separation and surface normals. *ECCV*, 2014.

[130] Z. Jia, M. K. Ng, and G.-J. Song. Robust quaternion matrix completion with applications to image inpainting. *Numerical Linear Algebra with Applications*, 26, 2019.

[131] Y. Jiang, X. Gong, D. Liu, Y. Cheng, C. Fang, X. Shen, J. Yang, P. Zhou, and Z. Wang. Enlightengan: Deep light enhancement without paired supervision. *IEEE TIP*, 30, 2021.

[132] N. K. Kalantari and R. Ramamoorthi. Deep high dynamic range imaging of dynamic scenes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2017)*, 36(4), 2017.

[133] A. T. Kamal and M. T. El-Melegy. Color image processing using reduced biquaternions with application to face recognition in a pca framework. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 3039–3046, 2017.

[134] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36 (11), 2014.

[135] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *TPAMI*, 36(11), 2014.

[136] D. Kazhdan, B. Dimanov, M. Jamnik, P. Liò, and A. Weller. Now you see me (cme): concept-based model extraction. *arXiv preprint arXiv:2010.13233*, 2020.

[137] M. Keswani, S. Ramakrishnan, N. Reddy, and V. N. Balasubramanian. Proto2proto: Can you recognize the car, the way i do? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10233–10243, 2022.

[138] A. Khosla, T. Zhou, T. Malisiewicz, A. Efros, and A. Torralba. Undoing the damage of dataset bias. *ECCV*, 2012.

[139] B. Kim, M. Wattenberg, J. Gilmer, C. J. Cai, J. Wexler, F. B. Viégas, and R. Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *ICML*, 2018.

[140] E. Kim, S. Kim, M. Seo, and S. Yoon. Xprotonet: Diagnosis in chest radiography with global and local explanations. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15714–15723, 2021.

[141] E. Kim, J. Lee, and J. Choo. Biaswap: Removing dataset bias with bias-tailored swapping augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14992–15001, 2021.

[142] S. Kim, K. Park, K. Sohn, and S. Lin. Unified depth prediction and intrinsic image decomposition from a single image via joint convolutional neural fields. *ECCV*, 2016.

[143] R. Kindermann and L. Snell. *Markov random fields and their applications*, volume 1. American Mathematical Society, 1980.

[144] N. Kodali, J. Hays, J. Abernethy, and Z. Kira. On convergence and stability of gans. *arXiv: Artificial Intelligence*, 2018.

[145] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020.

[146] N. Kong, P. V. Gehler, and M. J. Black. Intrinsic video. *ECCV*, 2014.

[147] B. Kovacs, S. Bell, N. Snavely, and K. Bala. Shading annotations in the wild. *CVPR*, 2017.

[148] V. Kwatra, M. Han, and S. Dai. Shadow removal for aerial imagery by information theoretic intrinsic image analysis. *International Conference on Computational Photography (ICCP)*, 2012.

[149] I. Kyrchei. Explicit representation formulas for the minimum norm least squares solutions of some quaternion matrix equations. *Linear Algebra and its Applications*, 438:136–152, 2013.

[150] P.-Y. Laffont, A. Bousseau, S. Paris, F. Durand, and G. Drettakis. Coherent intrinsic images from photo collections. *ACM Transactions on Graphics*, 31(6), 2012.

[151] P.-Y. Laffont, A. Bousseau, and G. Drettakis. Rich intrinsic image decomposition of outdoor scenes from multiple views. *IEEE Transactions on Visualization and Computer Graphics*, 19 (2), 2013.

[152] J. H. Lambert. *Photometria sive de mensura et gradibus luminis, colorum et umbrae*. Klett, 1760.

[153] E. H. Land. The retinex theory of color vision. *Scientific American*, 237 6, 1977.

[154] E. H. Land. The retinex theory of color vision. *Scientific American*, 237 6, 1977.

[155] E. H. Land and J. J. McCann. Lightness and retinex theory. *J. Opt. Soc. Am.*, 61(1), 1971.

[156] B. Lecouat, J. Ponce, and J. Mairal. Designing and learning trainable priors with non-cooperative games. *NeurIPS*, 2020.

[157] B. Lecouat, J. Ponce, and J. Mairal. Fully trainable and interpretable non-local sparse models for image restoration. *ECCV*, 2020.

[158] C. Lee, C. Lee, and C.-S. Kim. Contrast enhancement based on layered difference representation of 2d histograms. *IEEE TIP*, 22(12), 2013.

[159] C. Lee, C. Lee, and C.-S. Kim. Contrast enhancement based on layered difference representation of 2d histograms. *IEEE TIP*, 22(12), 2013.

[160] C. Lee, C. Lee, and C.-S. Kim. Contrast enhancement based on layered difference representation of 2d histograms. *IEEE Transactions on Image Processing*, 22(12):5372–5384, 2013.

[161] J. Lee, E. Kim, J. Lee, J. Lee, and J. Choo. Learning debiased representation via disentangled feature augmentation. *Advances in Neural Information Processing Systems*, 34:25123–25133, 2021.

[162] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE TPAMI*, 2008.

[163] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang. Benchmarking single-image dehazing and beyond. *IEEE TIP*, 28(1), 2019.

[164] B. Li, X. Liu, P. Hu, Z. Wu, J. Lv, and X. Peng. All-In-One Image Restoration for Unknown Corruption. In *CVPR*, 2022.

[165] C. Li, C. Guo, L. Han, J. Jiang, M.-M. Cheng, J. Gu, and C. C. Loy. Low-light image and video enhancement using deep learning: A survey. *IEEE TPAMI*, 2021.

[166] C. Li, C. Guo, and C. C. Loy. Learning to enhance low-light image via zero-reference deep curve estimation. *IEEE TPAMI*, 2021.

[167] J. Li, P. Zhou, C. Xiong, and S. C. Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020.

[168] Y. Li and M. S. Brown. Single image layer separation using relative smoothness. *CVPR*, 2014.

[169] Y. Li and N. Vasconcelos. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9572–9581, 2019.

[170] Y. Li, L. Guo, and Y. Ge. Pseudo labels for unsupervised domain adaptation: A review. *Electronics*, 12(15):3325, 2023.

[171] Z. Li and N. Snavely. Learning intrinsic image decomposition from watching the world. *CVPR*, 2018.

[172] Z. Li and N. Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. *ECCV*, 2018.

[173] J. Liang, Y. Xu, Y. Quan, B. Shi, and H. Ji. Self-supervised low-light image enhancement using discrepant untrained network priors. *IEEE TCSVT*, 32(11), 2022.

[174] Z. Liang, C. Li, S. Zhou, R. Feng, and C. C. Loy. Iterative prompt learning for unsupervised backlit image enhancement. In *ICCV*, 2023.

[175] S. Lim and W. Kim. Dslr: Deep stacked laplacian restorer for low-light image enhancement. *IEEE TMM*, 23, 2021.

[176] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *ArXiv*, abs/1009.5055, 2010.

[177] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020.

[178] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *TPAMI*, 33(5), 2011.

[179] J. Liu, X. Chen, Z. Wang, and W. Yin. ALISTA: Analytic weights are as good as learned weights in LISTA. In *ICLR*, 2019.

[180] J. Liu, X. Dejia, W. Yang, M. Fan, and H. Huang. Benchmarking low-light image enhancement and beyond. *IJCV*, 129, 2021.

[181] X. Liu, L. Wan, Y. Qu, T.-T. Wong, S. Lin, C.-S. Leung, and P.-A. Heng. Intrinsic colorization. *ACM Transactions on Graphics*, 27(5), 2008.

[182] Y. Liu, J. Pan, J. Ren, and Z. Su. Learning deep priors for image dehazing. In *ICCV*, 2019.

[183] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, 2019.

[184] Y. P. Loh and C. S. Chan. Getting to know low-light images with the exclusively dark dataset. *CVIU*, 178, 2019.

[185] J. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? *NeurIPS*, 2014.

[186] C. Loscos, K. Jacobs, G. Patow, and X. Pueyo. Inverse Rendering: From Concept to Applications. In N. Magnenat-Thalmann and K. Bühler, editors, *Eurographics 2006: Tutorials*. The Eurographics Association, 2006.

[187] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, 1999.

[188] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. Are gans created equal? a large-scale study. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*. 2018.

[189] F. Lv, Y. Li, and F. Lu. Attention guided low-light image enhancement with a large scale low-light simulation dataset. *IJCV*, 129(7), 2021.

[190] K. Ma, K. Zeng, and Z. Wang. Perceptual quality assessment for multi-exposure image fusion. *IEEE TIP*, 24(11), 2015.

[191] L. Ma, T. Ma, R. Liu, X. Fan, and Z. Luo. Toward fast, flexible, and robust low-light image enhancement. In *CVPR*, 2022.

[192] Y. Ma, X. Feng, X. Jiang, Z. Xia, and J. Peng. Intrinsic image decomposition: A comprehensive review. In *Image and Graphics: 9th International Conference, ICIG 2017, Shanghai, China, September 13-15, 2017, Revised Selected Papers, Part I 9*, pages 626–638. Springer, 2017.

[193] J. Malik, P. Arbeláez, J. Carreira, K. Fragkiadaki, R. Girshick, G. Gkioxari, S. Gupta, B. Hariharan, A. Kar, and S. Tulsiani. The three r's of computer vision: Recognition, reconstruction and reorganization. *Pattern Recognition Letters*, 72, 2016. Special Issue on ICPR 2014 Awarded Papers.

[194] S. Manen, M. Guillaumin, and L. V. Gool. Prime object proposals with randomized prim's algorithm. In *ICCV*, 2013.

[195] S. R. Marschner. *Inverse Rendering for Computer Graphics*. PhD thesis, USA, 1998. AAI9839924.

[196] E. Mathieu, T. Rainforth, N. Siddharth, and Y. W. Teh. Disentangling disentanglement in variational autoencoders. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*. PMLR, 2019.

[197] MATLAB. *version 9.4.0 (R2018a)*. The MathWorks Inc., Natick, Massachusetts, 2018.

[198] J. J. McCann. Retinex at 50: color theory and spatial algorithms, a review. *Journal of Electronic Imaging*, 26, 2017.

[199] T. Mertens, J. Kautz, and F. V. Reeth. Exposure fusion: A simple and practical alternative to high dynamic range photography. *Computer Graphics Forum*, 28, 2009.

[200] T. Mertens, J. Kautz, and F. V. Reeth. Exposure fusion: A simple and practical alternative to high dynamic range photography. *Computer Graphics Forum*, 28, 2009.

[201] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019.

[202] A. Mittal, R. Soundararajan, and A. C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3), 2013.

[203] M. Moayeri, K. Rezaei, M. Sanjabi, and S. Feizi. Text-to-concept (and back) via cross-model alignment. *arXiv preprint arXiv:2305.06386*, 2023.

[204] V. Monga, Y. Li, and Y. C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2), 2021.

[205] Y. Mäkinen, L. Azzari, and A. Foi. Collaborative filtering of correlated noise: Exact transform-domain variance for improved shrinkage and patch matching. *IEEE Transactions on Image Processing*, 29, 2020.

[206] H. Nagahara, S. Kuthirummal, C. Zhou, and S. K. Nayar. Flexible depth of field photography. In *ECCV*. 2008.

[207] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017.

[208] T. Narihira, M. Maire, and S. X. Yu. Direct intrinsics: Learning albedo-shading decomposition by convolutional regression. *ICCV*, 2015.

[209] I. Nassar, M. Hayat, E. Abbasnejad, H. Rezatofighi, and G. Haffari. Protocon: Pseudo-label refinement via online clustering and prototypical consistency for efficient semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11641–11650, 2023.

[210] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[211] S. K. Nayar and Y. Nakagawa. Shape from focus. *IEEE TPAMI*, 16(8):824–831, 1994.

[212] T. Nestmeyer and P. V. Gehler. Reflectance adaptive filtering improves intrinsic image estimation. *CVPR*, 2017.

[213] H. Nguyen, D. Tran, K. Nguyen, and R. Nguyen. Psenet: Progressive self-enhancement network for unsupervised extreme-light image enhancement. In *WACV*, 2023.

[214] Z. Ni, W. Yang, S. Wang, L. Ma, and S. Kwong. Towards unsupervised deep image enhancement with generative adversarial network. *IEEE TIP*, 29, 2020.

[215] C. Niu, H. Shan, and G. Wang. Spice: Semantic pseudo-labeling for image clustering. *IEEE Transactions on Image Processing*, 31:7264–7278, 2022.

[216] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3), 2014.

[217] G. Patow and X. Pueyo. A survey of inverse rendering problems. *Comput. Graph. Forum*, 22: 663–688, 12 2003.

[218] S. Pizer, R. Johnston, J. Ericksen, B. Yankaskas, and K. Muller. Contrast-limited adaptive histogram equalization: speed and effectiveness. In *[1990] Proceedings of the First Conference on Visualization in Biomedical Computing*, pages 337–345, 1990.

[219] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld. Adaptive histogram equalization and its variations. *CVGIP*, 39(3), 1987.

[220] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. M. ter Haar Romeny, and J. B. Zimmerman. Adaptive histogram equalization and its variations. *Graphical Models graphical Models and Image Processing computer Vision, Graphics, and Image Processing*, 39:355–368, 1987.

[221] D. Puthussery, H. Panikkasseril Sethumadhavan, M. Kuriakose, and J. Charangatt Victor. Wdrn: A wavelet decomposed relightnet for image relighting. In *ECCV workshop*, 2020.

[222] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[223] R. Ramamoorthi and P. Hanrahan. A signal-processing framework for inverse rendering. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, 2001.

[224] C. Ren, Y. Pan, and J. Huang. Enhanced latent space blind model for real image denoising via alternative optimization. In *NeurIPS*, 2022.

[225] X. Ren, W. Yang, W.-H. Cheng, and J. Liu. Lr3m: Robust low-light enhancement via low-rank regularized retinex model. *IEEE TIP*, 29, 2020.

[226] A. M. Reza. Realization of the contrast limited adaptive histogram equalization (clahe) for real-time image enhancement. *J. VLSI Signal Process. Syst.*, 38(1), 2004.

[227] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *WACV*, 2020.

[228] L. Rieger, C. Singh, W. Murdoch, and B. Yu. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *International conference on machine learning*, pages 8116–8126. PMLR, 2020.

[229] L. Risheng, M. Long, Z. Jiaao, F. Xin, and L. Zhongxuan. Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement. In *CVPR*, 2021.

[230] T. Robert, N. Thome, and M. Cord. Hybridnet: Classification and reconstruction cooperation for semi-supervised learning. In *ECCV*, 2018.

[231] A. S. Ross, M. C. Hughes, and F. Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017.

[232] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500), 2000.

[233] G. Saha and K. Roy. Saliency guided experience packing for replay in continual learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5273–5283, 2023.

[234] S. Saini and P. J. Narayanan. Quaternion factorized simulated exposure fusion. In *ACM ICVGIP*, 2023.

[235] S. Saini, P. Sakurikar, and P. J. Narayanan. Intrinsic image decomposition using focal stacks. *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing*, 2016.

[236] P. Sakurikar and P. J. Narayanan. Dense view interpolation on mobile devices using focal stacks. *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 138–143, 2014.

[237] W. Samek, T. Wiegand, and K.-R. Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.

[238] S. Sangwine and N. Le Bihan. Quaternion toolbox for matlab. version 3., 2022. URL `http://qtfm.sourceforge.net/`.

[239] S. J. Sangwine. Colour image edge detector based on quaternion convolution. *Electronics Letters*, 34, 1998.

[240] S. J. Sangwine and N. L. Bihan. Quaternion singular value decomposition based on bidiagonalization to a real or complex matrix using quaternion householder transformations. *Appl. Math. Comput.*, 182:727–738, 2006.

[241] S. J. Sangwine and T. A. Ell. Colour image filters based on hypercomplex convolution. 2000.

[242] Y. Sawada and K. Nakamura. C-senn: Contrastive self-explaining neural network. *ArXiv*, abs/2206.09575, 2022.

[243] Y. Sawada and K. Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022.

[244] B. Schauerte and R. Stiefelhagen. Quaternion-based spectral saliency detection for eye fixation prediction. In *ECCV*, 2012.

[245] J. Schrouff, S. Baur, S. Hou, D. Mincu, E. Loreaux, R. Blanes, J. Wexler, A. Karthikesalingam, and B. Kim. Best of both worlds: local and global explanations with human-understandable concepts. *arXiv preprint arXiv:2106.08641*, 2021.

[246] G. Schwalbe. Concept embedding analysis: A review. *arXiv preprint arXiv:2203.13909*, 2022.

[247] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[248] S. Sengupta, J. Gu, K. Kim, G. Liu, D. W. Jacobs, and J. Kautz. Neural inverse rendering of an indoor scene from a single image. In *International Conference on Computer Vision (ICCV)*, 2019.

[249] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. *CVPR Workshops*, 2014.

[250] A. Sharma and R. T. Tan. Nighttime visibility enhancement by increasing the dynamic range and suppression of light effects. *CVPR*, 2021.

[251] S. Shekhar, M. Reimann, M. Mayer, A. Semmo, S. Pasewaldt, J. Döllner, and M. Trapp. Interactive photo editing on smartphones via intrinsic decomposition. *Computer Graphics Forum*, 40(2), 2021.

[252] E. Shelhamer, J. T. Barron, and T. Darrell. Scene intrinsics and depth from a single image. *ICCV Workshops*, 2015.

[253] L. Shen, C. Yeo, and B.-S. Hua. Intrinsic image decomposition using a sparse representation of reflectance. *TPAMI*, 35(12), 2013.

[254] R. Shen, I. Cheng, J. Shi, and A. Basu. Generalized random walks for fusion of multi-exposure images. *IEEE Transactions on Image Processing*, 20, 2011.

[255] R. Shen, I. Cheng, J. Shi, and A. Basu. Generalized random walks for fusion of multi-exposure images. *IEEE TIP*, 2011.

[256] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[257] Y. Song, S. K. Shyn, and K.-s. Kim. Img2tab: Automatic class relevant concept discovery from stylegan features for explainable image classification. *arXiv preprint arXiv:2301.06324*, 2023.

[258] R. Soni, N. Shah, C. T. Seng, and J. D. Moore. Adversarial tcav - robust and effective interpretation of intermediate layers in neural networks. *ArXiv*, abs/2002.03549, 2020.

[259] P. P. Srinivasan, R. Tucker, J. T. Barron, R. Ramamoorthi, R. Ng, and N. Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[260] Ö. N. Subakan and B. C. Vemuri. A quaternion framework for color image smoothing and segmentation. *International Journal of Computer Vision*, 91:233–250, 2010.

[261] S. Suwajanakorn, C. Hernandez, and S. M. Seitz. Depth from focus with your mobile phone. In *CVPR*, 2015.

[262] K. Tanwisuth, X. Fan, H. Zheng, S. Zhang, H. Zhang, B. Chen, and M. Zhou. A prototype-oriented framework for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 34:17194–17208, 2021.

[263] M. F. Tappen, W. T. Freeman, and E. H. Adelson. Recovering intrinsic images from a single image. *TPAMI*, 27(9), 2005.

[264] E. Tartaglione, C. A. Barbano, and M. Grangetto. End: Entangling and disentangling deep representations for bias correction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13508–13517, 2021.

[265] The GIMP Development Team. Gimp, 2023. URL `https://www.gimp.org`.

[266] S. Tominaga. Dichromatic reflection models for a variety of materials. *Color Research and Application*, 19, 1994.

[267] S. Tominaga. Dichromatic reflection models for a variety of materials. *Color Research and Application*, 19, 1994.

[268] A. Torralba and A. A. Efros. Unbiased look at dataset bias. *CVPR*, 2011.

[269] V. Tschernezki, I. Laina, D. Larlus, and A. Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. *arXiv preprint arXiv:2209.03494*, 2022.

[270] R. Tucker and N. Snavely. Single-view view synthesis with multiplane images. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 548–557, 2020.

[271] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. Subspace learning from image gradient orientations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:2454–2466, 2012.

[272] S. Vojíř and T. Kliegr. Editable machine learning models? a rule-based framework for user studies of explainability. *Advances in Data Analysis and Classification*, 14(4):785–799, 2020.

[273] V. Vonikakis. Busting image enhancement and tonemapping algorithms., 2022. URL `https://sites.google.com/site/vonikakis/datasets`.

[274] H. Wang, Q. Xie, Q. Zhao, and D. Meng. A model-driven deep neural network for single image rain removal. 2020.

[275] M. Wang and W. Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312: 135–153, 2018.

[276] S. Wang, J. Zheng, H.-M. Hu, and B. Li. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE TIP*, 22(9), 2013.

[277] S. Wang, J. Zheng, H.-M. Hu, and B. Li. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE TIP*, 22(9), 2013.

[278] S. Wang, J. Zheng, H.-M. Hu, and B. Li. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE Transactions on Image Processing*, 22:3538–3548, 2013.

[279] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

[280] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TIP*, 13(4), 2004.

[281] X. Wanyan, S. Seneviratne, S. Shen, and M. Kirley. Dino-mc: Self-supervised contrastive learning for remote sensing imagery with multi-sized local crops. *arXiv preprint arXiv:2303.06670*, 2023.

[282] L. Weber, S. Lapuschkin, A. Binder, and W. Samek. Beyond explaining: Opportunities and challenges of xai-based model improvement. *Information Fusion*, 2022.

[283] C. Wei, W. Wang, Y. Wenhan, and J. Liu. Deep retinex decomposition for low-light enhancement. In *BMVC*, 2018.

[284] Y. Weiss. Deriving intrinsic images from image sequences. *ICCV*, 2001.

[285] M. Wu, M. Hughes, S. Parbhoo, M. Zazzi, V. Roth, and F. Doshi-Velez. Beyond sparsity: Tree regularization of deep models for interpretability. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[286] W. Wu, J. Weng, P. Zhang, X. Wang, W. Yang, and J. Jiang. Uretinex-net: Retinex-based deep unfolding network for low-light image enhancement. In *CVPR*, 2022.

[287] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018.

[288] D. Xu, Y. Xia, and D. P. Mandic. Optimization in quaternion dynamic systems: Gradient, hessian, and learning algorithms. *IEEE Transactions on Neural Networks and Learning Systems*, 27:249–261, 2016.

[289] K. Xu, X. Yang, B. Yin, and R. W. Lau. Learning to restore low-light images via decomposition-and-enhancement. In *CVPR*, 2020.

[290] N. Xu, K.-H. Tan, H. Arora, and N. Ahuja. Generating omnifocus images using graph cuts and a new focus measure. In *ICPR*, 2004.

[291] X. Xu, R. Wang, C.-W. Fu, and J. Jia. Snr-aware low-light image enhancement. In *CVPR*, 2022.

[292] Z. Xu, K. Sunkavalli, S. Hadap, and R. Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (TOG)*, 37(4), 2018.

[293] M. Xue, Q. Huang, H. Zhang, L. Cheng, J. Song, M. Wu, and M. Song. Protopformer: Concentrating on prototypical parts in vision transformers for interpretable image recognition. *arXiv preprint arXiv:2208.10431*, 2022.

[294] W. Yan, R. T. Tan, and D. Dai. Nighttime defogging using high-low frequency decomposition and grayscale-color networks. In *ECCV*, 2020.

[295] L. Yang, B. Huang, S. Guo, Y. Lin, and T. Zhao. A small-sample text classification model based on pseudo-label fusion clustering algorithm. *Applied Sciences*, 13(8):4716, 2023.

[296] S. Yang, M. Ding, Y. Wu, Z. Li, and J. Zhang. Implicit neural representation for cooperative low-light image enhancement. In *ICCV*, 2023.

[297] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Deep joint rain detection and removal from a single image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1694, 2017. doi: 10.1109/CVPR.2017.183.

[298] W. Yang, S. Wang, Y. Fang, Y. Wang, and J. Liu. Band representation-based semi-supervised low-light image enhancement: Bridging the gap between signal fidelity and perceptual quality. *IEEE TIP*, 30, 2021.

[299] W. Yang, W. Wang, H. Huang, S. Wang, and J. Liu. Sparse gradient regularized deep retinex network for robust low-light image enhancement. *IEEE Transactions on Image Processing*, 30, 2021.

[300] W. Yang, W. Wang, H. Huang, S. Wang, and J. Liu. Sparse gradient regularized deep retinex network for robust low-light image enhancement. *IEEE TIP*, 30, 2021.

[301] G. Ye, E. Garces, Y. Liu, Q. Dai, and D. Gutierrez. Intrinsic video and applications. *ACM Transactions on Graphics*, 2014.

[302] Q. Yin, J. Wang, X. Luo, J. Zhai, S. K. Jha, and Y. Q. Shi. Quaternion convolutional neural network for color image classification and forensics. *IEEE Access*, 7:20293–20301, 2019.

[303] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? *NeurIPS*, 2014.

[304] Y. Yu and W. A. Smith. Inverserendernet: Learning single image inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[305] L. Yuan and J. Sun. Automatic exposure correction of consumer photographs. In *ECCV*, 2012.

[306] F. Zhang, Y. Shao, Y. Sun, K. Zhu, C. Gao, and N. Sang. Unsupervised low-light image enhancement via histogram equalization prior. *arXiv:2112.01766*, 2021.

[307] J. Zhang, J. Huang, M. Yao, Z. Yang, H. Yu, M. Zhou, and F. Zhao. Ingredient-oriented multi-degradation learning for image restoration. *CVPR*, 2023.

[308] L. Zhang, L. Zhang, X. Liu, Y. Shen, S. Zhang, and S. Zhao. Zero-shot restoration of back-lit images using deep internal learning. *ACM MM*, 2019.

[309] Q. Zhang, G. Yuan, C. Xiao, L. Zhu, and W.-S. Zheng. High-quality exposure correction of underexposed photos. In *ACM MM*, 2018.

[310] Q. Zhang, Y. Nie, and W. Zheng. Dual illumination estimation for robust exposure correction. *Computer Graphics Forum*, 38, 2019.

[311] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[312] R. Zhang, P. Madumal, T. Miller, K. A. Ehinger, and B. I. Rubinstein. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11682–11690, 2021.

[313] W. Zhang, X. Zhao, J.-M. Morvan, and L. Chen. Improving shadow suppression for illumination robust face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(3), 2019.

[314] Y. Zhang, J. Zhang, and X. Guo. Kindling the darkness: A practical low-light image enhancer. *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.

[315] Y. Zhang, X. Guo, J. Ma, W. Liu, and J. Zhang. Beyond brightening low-light images. *IJCV*, 129, 2021.

[316] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742, 2021.

[317] Q. Zhao, P. Tan, Q. Dai, L. Shen, E. Wu, and S. Lin. A closed-form solution to retinex with nonlocal texture constraints. *TPAMI*, 34(7), 2012.

[318] C. Zheng, D. Shi, and W. Shi. Adaptive unfolding total variation network for low-light image enhancement. *ICCV*, 2021.

[319] N. Zheng, M. Zhou, Y. Dong, X. Rui, J. Huang, C. Li, and F. Zhao. Empowering low-light image enhancer through customized learnable priors. 2023.

[320] S. Zheng and G. Gupta. Semantic-guided zero-shot learning for low-light image/video enhancement. In *WACV*, 2022.

[321] T. Zhou, P. Krahenbuhl, and A. A. Efros. Learning data-driven reflectance priors for intrinsic image decomposition. *ICCV*, 2015.

[322] X. Zhou, C. Yang, H. Zhao, and W. Yu. Low-rank modeling and its applications in image analysis. *ACM Computing Surveys (CSUR)*, 47:1 – 33, 2014.

[323] A. Zhu, L. Zhang, Y. Shen, Y. Ma, S. Zhao, and Y. Zhou. Zero-shot restoration of underexposed images via robust retinex decomposition. *ICME*, 2020.

[324] Y. Zhu, Z. Xiao, Y. Fang, X. Fu, Z. Xiong, and Z.-J. Zha. Efficient model-driven network for shadow removal. *AAAI*, 2022.

[325] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman. Learning ordinal relationships for mid-level vision. *ICCV*, 2015.