Specularity Factorization for Low Light Enhancement

Anonymous CVPR submission

Paper ID 715

Figure 1. **Sepcularity 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 dehazing, deraining and deblurring. On right, our conceptual block diagram.

Abstract

Low Light Enhancement (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 paper, 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 deraining, deblurring and dehazing with negligible overhead as shown in the paper.*

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 [41]. LLE can be a critical pre-processing step before the downstream applications [49, 51]. 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 (Tab. 1). This allows individual manipulation of each factor which simplifies the enhancement operation. A common factorization is based on the Retinex approximation [34, 52], which assumes a multiplicative disentanglement of image I into two intrinsic factors: illumination-invariant, piecewise constant *reflectance* R and color-invariant, smooth *shading* S as $I = R \cdot S$. Other factorization criteria include frequency [31, 80], spatial scale [3, 45], spatio-frequency representation [17, 61], intensity [29], reflectance rank [63, 68], *etc*. Fixed number of fac-

tors [31, 63, 78] and variable number that allow better representation [3, 29, 45] have been used. Some decompose image multiplicatively like Retinex [31, 78], while others split into additive factors which are numerically more stable [19, 28, 61]. Pixel segmentation could be soft or hard based on the membership across factors, with the former introducing fewer artifacts [4]. LLE solutions can be *global* or *local*. Global methods use whole image level statistics like gamma [24], histograms [87], *etc*., to enhance the images. Local methods employ spatially varying features like illumination maps [78], intensity/segmentation masks [29, 57], *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 [21, 26, 91], 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 [3, 78, 81]. *Supervised* methods require annotated input-output pairs of images [79, 81, 93]. *Unsupervised* methods require annotated training data but not necessarily paired [32, 82]. *Zero-reference* methods do not need annotated data and approach the problem by explicitly encoding the domain knowledge from training images [24, 51, 66]. They generalize better and are simpler, lighter, and more interpretable by design.

In this paper, 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. Our major contributions 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 dehazing, deraining and deblurring showing the power of our specularity factorization as a prior.

Criteria	No.	Type	Map	Seg	Example
Retinex	2	$*$	global	soft	[66, 79]
Frequency	2	$+$, low/high	global	hard	[80]
Spectral	2	$*$, fourier	global	soft	[31]
Low Rank	2	$+$	global	soft	[63, 68]
Wavelets	2^n	$+$, pyramid	global	soft	[17, 61]
Multiscale	2^n	$+$, pyramid	global	soft	[3, 45]
Glare/Shadow	3, 4	$*$, $+$	local	hard	[5, 69]
Intensity	var.	$+$, bands	local	hard	[28, 29]
Specularity	var.	$+$	local	soft	RSFNet

Table 1. 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.

2. Background

Low Light Enhancement

Model-based: Early LLE solutions used traditional optimization models using either global statistics [14, 38, 60, 64] or spatially varying illumination maps for local editing [20, 26, 76, 90]. They were more interpretable but required hand-crafted algorithms and heuristics.

Data-driven: 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 [41]. *Supervised* LLE methods require both low-light and well-lit *paired images* like Sharma and Tan [69], Wei et al. [78], Xu et al. [81], Yang et al. [86], Zhang et al. [94]. On the other hand, *unsupervised* methods like Jiang et al. [32], Ni et al. [58], Zhang et al. [87], 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 [67, 85]. *Self-supervised* solutions [43, 57] generate their own annotations using pseudo-labels or synthetic degradations. 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 [24, 42, 51, 66, 89, 98]. These methods, like ours, possess improved generalizability due to explicit induction of domain knowledge and reduced chances of overfitting [24]. Zero-reference insights also provide direct valuable additions to the subsequent solutions in other paradigms.

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 [55] are hybrids which bring the best of both together. Such networks *unroll* optimization

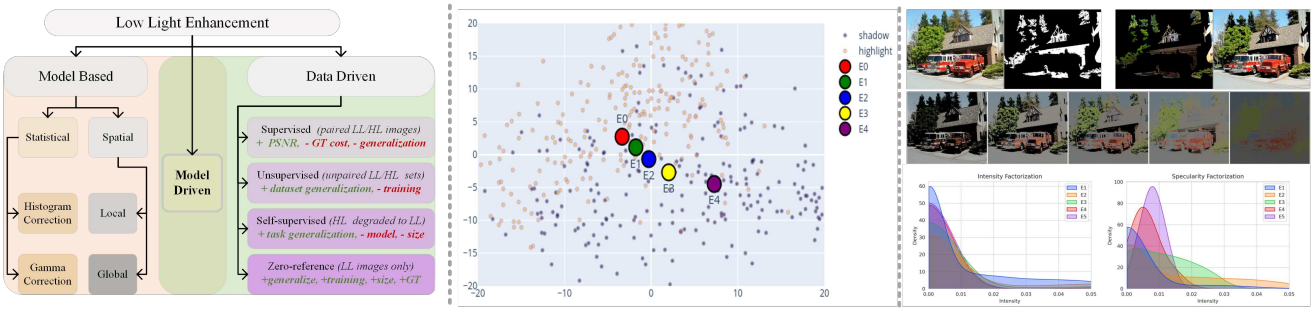

Figure 2. **Categorization and Motivation:** Left shows categorization of various LLE solution types (Sec. 2). Middle plot shows the relationship between five factor cluster centers w.r.t each other and the background comprising of shadow/non-shadow regions estimated using PCA dimensionality reduced DINO features [13]. Gradual progression of feature cluster centers from highlight region to shadow region indicates their capability to capture various illumination regions in an image. Top right shows one data point from CHUK dataset [30] with mask, processed shadow/highlight regions and extracted factors. Bottom right plots distinguish our specular fuzzy factors from intensity thresholding based binary division, with ours allowing more diverse distributions and richer representation.

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 [35, 36], shadow removal [99], dehazing [48], deraining [74], denoising [62] and super-resolution [7, 8]. Such solutions are concise and efficient due to the underlying task specific analytical formulation.

Model-driven LLE solutions are very recent. UretinexNet [79] and UTVNet [95] are both supervised methods which respectively unroll the Retinex and total variational LLE formulations. RUAS [66] and SCI [51] 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.

Image Factorization for LLE

Retinex: Retinex is the most widely used LLE factorization strategy [41]. One major limitation here is due to the Lambertian reflection [33] 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 [72] and shadow segmentation, separate one extra component each in addition to diffuse R and S *e.g*. Sharma and Tan [69] and Baslamisli et al. [5] 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. 2 and Sec. 3 for details).

Others Factorization Strategies: Apart from Retinex, other factorization techniques are listed in Tab. 1. Afifi et al. [3], Lim and Kim [45], Mertens et al. [53], Xu et al. [80] employ spatial or frequency based image decomposition. Recently, Yang et al. [85] used recursively concatenated features from a supervised encoder and Huang et al. [31] proposed a Fourier disentanglement based solution. Apart from these supervised factorizations, Zheng and Gupta [97] proposed semantic classification based ROI identification using a pretrained segmentation network. [24, 57] predict multiple gamma correction maps for enhancement. [28, 29] simulate single image exposure burst using piece-wise thresholded intensity functions whereas [63] 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.

3. Approach

Outline: 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

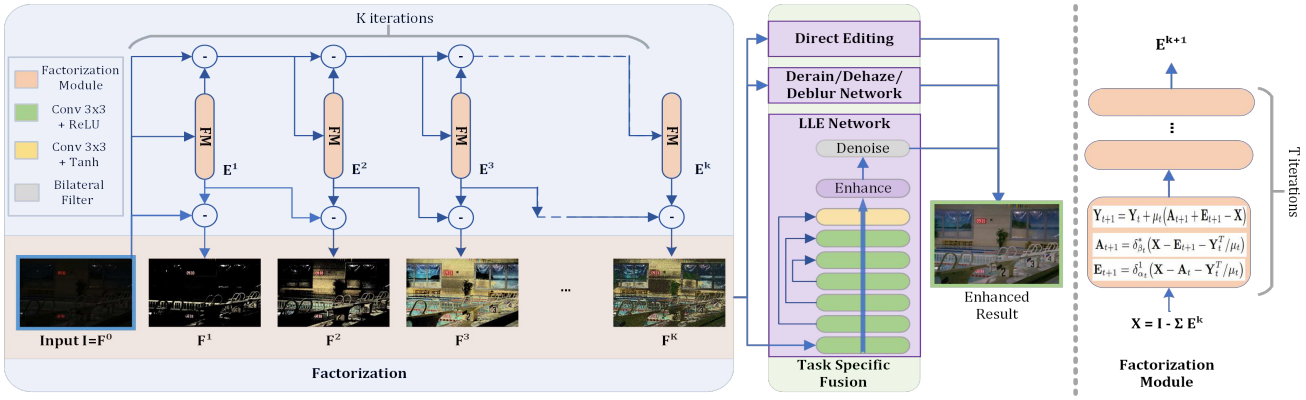

Figure 3. **Block Diagram:** Overview of our proposed RSFNet. Factorization module splits a given image into multiple specular components using model-driven unrolled optimization steps. Fusion module combines all the factors to generate the enhance output.

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 (Sec. 4).

3.1. Factorization Network

Specularity Estimation: Specularity removal is a well studied problem. Most specularity removal methods [1, 25, 70] exploit the relative sparsity of specular highlights and use pre-defined fixed sparsity thresholds to isolate the specular component. According to dichromatic reflection model [72] 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:

$$\underset{\mathbf{E},\mathbf{A}}{\operatorname{argmin}} \|\mathbf{A}\|_* + \lambda\|\mathbf{E}\|_1 \quad \text{s.t.} \quad \mathbf{X} = \mathbf{A} + \mathbf{E}, \tag{1}$$

where L_1 is relaxation of L_0, $*$ is Frobenius norm regularizer and λ is the sparsity parameter with higher values encouraging sparser results. Eq. (1) can be restated as augmented Lagrangian [9] using dual form and auxiliary parameters $(\mathbf{Y}, \mu)$, which are then solvable using iterative ADMM updates ($t \in [0, T]$) [10] as given below:

$$\begin{aligned} \mathbf{E}_{t+1} &= \delta^1_{\alpha_t}\left(\mathbf{X} - \mathbf{A}_t - \mathbf{Y}_t^T/\mu_t\right) && \text{where } \alpha : \mathcal{F}(\lambda, \mu), \\ \mathbf{A}_{t+1} &= \delta^*_{\beta_t}\left(\mathbf{X} - \mathbf{E}_{t+1} - \mathbf{Y}_t^T/\mu_t\right) && \text{where } \beta : \mathcal{F}(\mu), \\ \mathbf{Y}_{t+1} &= \mathbf{Y}_t + \mu_t\left(\mathbf{A}_{t+1} + \mathbf{E}_{t+1} - \mathbf{X}\right) && \text{where } \mu : \mathcal{F}(\mathbf{X}). \end{aligned} \tag{2}$$

Here δ^p_α is element-wise soft-thresholding operator [59]:

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

We can back-propagate through updates in Eq. (2) [79, 95] and hence can unroll them as neural network layers with learnable parameters $\boldsymbol{\alpha} : \{\alpha\}_0^T$, $\boldsymbol{\beta} : \{\beta\}_0^T$ and $\boldsymbol{\mu} : \{\mu\}_0^T$.

Relation with ISTA: Analyzing the structure of Eq. (2), we can draw parallels with the ISTA problem [16], which seeks a sparse solution to $\mathbf{E}$ for the condition $\mathbf{X} = \mathcal{G}\mathbf{E} + \epsilon$, 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 [23] showed how $\mathbf{E}$ update step can be represented as a weighted function which can then be approximated as finite network layers *i.e.*:

$$\mathbf{E}_{t+1} = \delta_{\alpha_t}(\mathbf{w}_t^1\mathbf{E}_t + \mathbf{w}_t^2\mathbf{X}), \tag{3}$$

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. [15] simplified Eq. (3) 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. [46] 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. [11] unrolled their ADMM updates into a network for robust principal component analysis.

Recursive Factorization: Drawing parallels from ALISTA [46] and its applications [11], we propose to learn the analytically reduced sparsity thresholds and step sizes via unrolled network layers. After optimizing the above mentioned objective Eq. (1) 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. (1) 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:

$$\begin{aligned} \mathbf{X}^{k+1} &= \mathbf{X}^k - \mathbf{E}^k, \qquad \mathbf{Y}^k = \mathbf{X}^k / \left\|\mathbf{X}^k\right\|_2 \\ \alpha^k &= (1 - \nu^k)\hat{X}^k, \qquad \beta^k = \nu^k\hat{X}^k, \qquad \nu^k = k/K, \end{aligned} \tag{4}$$

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:

$$\mathbf{I} = \mathbf{E}^1 + \mathbf{E}^2 + \ldots + \mathbf{E}^K = \textstyle\sum_{k=1}^{K} \mathbf{E}^k \tag{5}$$

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 (Tab. 2). We propose the following novel factorization loss:

$$L_f = \lambda_f \sum_{k=1}^{K} L_f^k \qquad \text{where} \qquad L_f^k = \left| \hat{E}^k / \hat{X}^k - \nu^k \right|. \tag{6}$$

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. 1 and Fig. 2, each one of these factors highlights specific image regions with similar illumination characteristics which can be individually targeted for enhancement.

Motivation/Validation: 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 [30] which contains binary shadow masks in complex real world images (Fig. 2). We extract semantics-rich DINO image features [13] 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. 2 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.*

We also plot the respective factor distribution densities of intensity factorization [28, 29] and our specularity factorization (Fig. 2, 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.

Algorithm 1: LLE Training

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. 4*
      $E_t, A_t, Y_t \leftarrow$ ADMM updates ; *// Eqn. 2*
    end
    $F^k \leftarrow E^k - E^{k-1}$; *// Eqn. 7*
  end
  Compute L_f ; *// Eqn. 6*

  // 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. 9*
  Backpropagate L;
end

3.2. 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 [24, 57, 97]. 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:

$$\mathbf{F}^k = \mathbf{E}^k - \mathbf{E}^{k-1} \text{ where } \mathbf{F}^1 = \mathbf{E}^1. \tag{7}$$

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 [24] for the fused result:

$$O = \Phi\Big(\sum_{k=0}^{K} I + R^k.\big((I)^2 - I\big)\Big). \tag{8}$$

The fused output is finally passed through a differentiable bilateral filtering layer Φ [65] 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.

Loss: We use two widely employed zero-reference losses

Paradigm	Traditional Model Based			Zero-reference							
Method	LIME [26]	DUAL [91]	SDD [27]	ECNet [89]	ZDCE [24]	ZD++ [42]	RUAS [66]	SCI [51]	PNet [57]	GDP [18]	**RSFNet** (Ours)
Params x10^3	-	-	-	16.5x10^3	79.42	10.56	3.43	**0.26**	15.25	552x10^3	2.11
	Lolv1 [78] (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 [86] (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**
	GENERALIZED PERFORMANCE **Mean Scores** (**Lolv1** [78], **Lolv2-real** [86], **Lolv2-syn** [86] and **VE-Lol** [47])										
PSNR$_y$ ↑	18.50	17.83	17.50	18.45	19.26	18.73	17.09	18.07	19.65	15.88	**21.16**
SSIM$_y$ ↑	0.737	0.728	0.781	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	17.35	14.15	**18.45**
SSIM$_c$ ↑	0.596	0.583	0.679	0.538	0.634	0.548	0.532	0.587	0.605	0.504	**0.758**
NIQE ↓	7.855	7.478	4.077	7.543	4.270	7.468	5.841	7.626	7.791	6.726	**3.763**
LPIPS↓	0.291	0.297	**0.266**	0.329	0.273	0.296	0.346	0.295	0.302	0.379	0.276

Table 2. Quantitative comparison of our method RSFNet with other traditional and zero-reference solutions on multiple lowlight benchmarks and six evaluation metrics. Shown here are scores for two datasets LOLv1 and LOLv2-real with mean value across all datasets in the last sub-table (key: ↑ higher better; ↓ lower better; **bold**: best; underline: second best).

for enhancement [24, 57, 87] and one image smoothing loss for denoising. First *color loss* L_c [24, 87] is based on the gray-world assumption which tries to minimize the mean value difference between each color channel pair:

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

Second is the *exposure loss* L_e [24, 29, 53], which penalizes grayscale intensity deviation from the mid-tone value:

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

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:

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

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:

$$L = \lambda_f L_f + \lambda_c L_c + \lambda_e L_e + \lambda_s L_s. \quad (9)$$

4. Experiments and Results

We now report our implementation details, results and extensions. Please see the supplementary document for additional details and results.

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.

Datasets: We evaluate our method using multiple LLE benchmark datasets (Lolv1 [78], Lolv2-real [86], Lolv2-synthetic [86] and VE-Lol [47]) with standard train/test splits (Tab. 2). 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 Tab. 2 and in Fig. 5. Furthermore, we report generalization results (Tab. 4) on five additional no-reference datasets which have significant domain shift: DICM [37], LIME [26], MEF [50], NPE [76] and VV [73].

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) [77] and Learned Perceptual Image Patch Similarity (LPIPS) [92]. For no-reference assessment (without ground truth), we report Naturalness Image Qual-

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

Table 3. Ablation analysis on five variants of our RSFNet (Sec. 4).

NIQE↓ & *LOE*↓	ECNet [89]	ZDCE [24]	ZD++ [42]	RUAS [66]	PNet [57]	SCI [51]	**RSFNet** (Ours)
DICM [37]	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 [26]	**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 [50]	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 [76]	**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 [73]	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 4. 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 4. **Results:** Qualitative comparison of our method (green box) with other solutions (from top left 3 per row: SDD [27], ECNet [89], ZDCE [24]; ZD++ [42], RUAS [66], SCI [51]; PNet [57], GDP [18] and our RSFNet respectively). Our method generate natural looking images by handling noisy over and under exposed regions equally well, without over-saturating color or losing geometric details.

ity Evaluator (NIQE) [54] and Lightness Order Error [75]. Note while PSNR and SSIM gauge performance quantitatively, other three metrics estimate perceptual quality.

Comparisons: We compare against three model-based traditional optimization methods: LIME [26], DUAL [91] and SDD [27] (others ignored due to low performance). For data-driven methods we use seven recent zero-reference methods (chronologically ordered): ECNet [89], zeroDCE [24], zeroDCE++ [42], RUAS [66], SCI [51], PNet [57] and GDP [18]. We use the official code releases with pre-trained weights and default parameters for results generation. Quantitative and qualitative performance comparison is shown in Tab. 2 and Fig. 4 respectively. Qualitatively, our method is cleaner with fewer artifacts and natural illumination (Fig. 4). This is validated by perceptual metrics like NIQE, LPIPS and LOE scores (Tabs. 2 and 4). 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. Each polygon represents a separate LLE method with higher area inside indicating better performance.

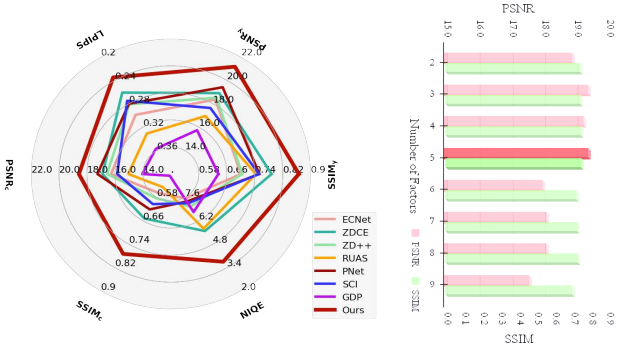

Figure 5. **Analysis:** On left, our average score on all datasets *vs*. other methods (more area implies better). On right, ablation analysis with varying number of factors.

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. 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 Tab. 3. 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:

$$O^{k+1} = (1-w^k)O^k + w^k F^k, \text{ where } w^k = \hat{F}^k / \textstyle\sum_k \hat{F}^k. \quad (10)$$

Figure 6. Image enhancement applications using our specular factors as inputs on the AirNet [40] base model. Shown here left-to-right are our results for Dehazing [39], Deraining [84] and Deblurring [56] tasks respectively using AirNet [40] as base model.

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 Tab. 2).

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 [71], Photoshop [2], *etc*. We show an image relighting example by varying the color and blending modes of factors in (Fig. 1 bottom left, Fig. 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 [40] 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: *(i)* Dehazing on RESIDE dataset [39] *(ii)* Deraining on Rain100L dataset [84] and *(iii)* Debluring on GoPro dataset [56]. As seen in Fig. 6 and Tab. 5, our results are perceptually more pleasing and improve the previously reported scores from multi-task methods consistently [40, 88]. 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. See the supplementary for more results.

Figure 7. Controlable relighting applications using our factors as layers [71] (top:inputs; bottom:results; from left:edited light specularity, indoor color and outdoor intensity respectively).

TASK →	DEHAZE [39]		DERAIN [84]		DEBLUR [56]	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
AirNet (multi-task)	21.04	0.884	32.98	0.951	18.18	0.781
AirNet (uni-task)	23.18	0.900	34.90	0.9657	26.42	0.801
AirNet + Ours	**24.96**	**0.9292**	**36.19**	**0.9718**	**27.29**	**0.827**

Table 5. Our factors can induce structure prior in an existing base model and improve performance for multiple enhancement tasks.

Limitations: Our method is sensitive to initialization conditions like the underlying algorithms [11, 46]. 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. Conclusions

In this paper, we presented a zero-reference LLE method that uses a novel image factorization strategy based on specularity. We learn optimization hyperparameters 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.

Ethical Concerns: Our work attempts to enhance captured images to improve interpretation and poses no special ethical issues.

Specularity Factorization for Low Light Enhancement

Supplementary Material

6. Discussion

Here we provide an elaborate discussion and additional evidences on various points mentioned in the main paper.

Unsupervised *vs*. Zero-reference LLE: Although similar, there is a crucial difference between the unsupervised and zero-reference LLE paradigms [41]. As mentioned previously, unsupervised LLE solutions like [22, 32, 44, 82, 83, 87] 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 [18, 24, 42, 51, 57, 66, 89] 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 as shown in the main paper. Although fair comparison is possible only between the methods of the same paradigm [18], still we report our comparison with various unsupervised solutions in Tab. 11. Note that our method beats several unsupervised LLE solutions and is competitive against the best two unsupervised solutions [83] and [87]. [83] 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 [83] on 4 out of 6 and [87] on 5 out of the 6 reported metrics individually.

Interpretability: 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. 2. Here we repeat the same analysis with other parts of the shadow dataset [30]. [30] 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 was shown in the main paper. 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 [13] and factors were computed using direct optimization using Eq. (1), Eq. (2) and Sec. 3.1. These plots are shown in Fig. 9. 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*.

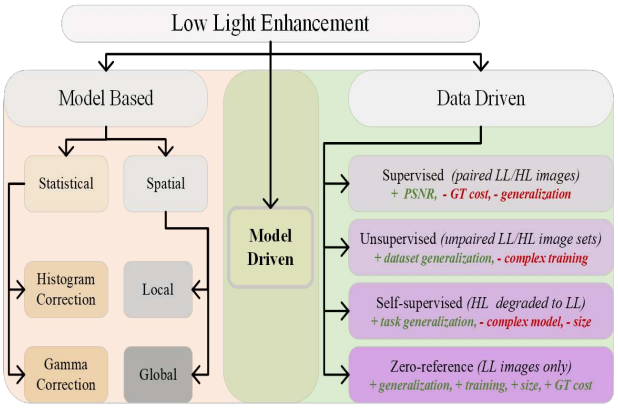

Figure 8. **LLE solutions categorization:** Data-driven methods are of mainly 4 types based on the type of input supervision available with each type having its pros and cons as listed above.

Factorization Strategies: As mentioned in Secs. 1 and 2 and shown in Tab. 1, various LLE solutions adopt different factorization strategies. We have provided a non-exhaustive list in the Tab. 1 but still others are possible. The *Frequency* strategy [80] 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 [31] refers to decomposition into phase and ampli-

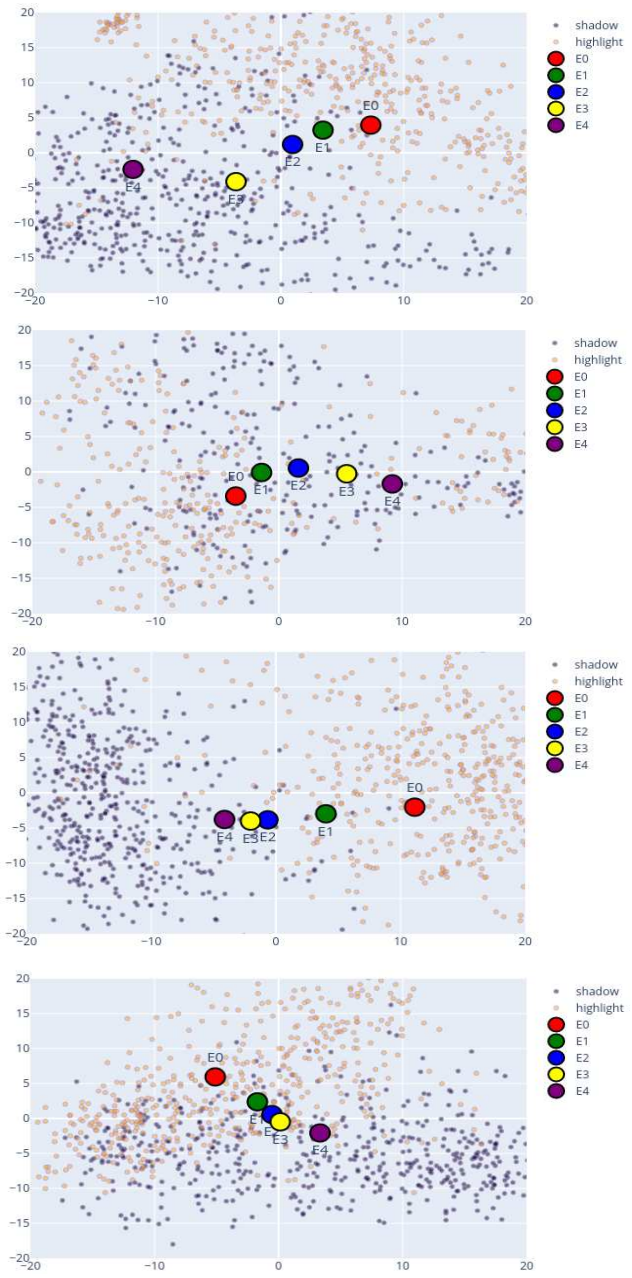

Figure 9. **Factor interpretability and analysis:** We perform factor distribution analysis Fig. 2 on four additional shadow datasets (from top to bottom - shadow_KITTI, shadow_MAP, shadow_USR and shadow_WEB). Each plot represents features of shadow and non-shadow regions which forms the background and cluster centers of the five factors feature distributions are plotted in the foreground. Note how in each case the series of factors is sequentially from bright to the dark region.

tude 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 to the Retinex division. [63] focuses on hyper-spectral images, whereas [68] uses a complicated quaternion based robust PCA optimization strategy [11] with no unrolled learning or generalization to other applications. *Wavelets* and *Multiscale* decompositions [3, 17] 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 [5, 69] 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 [28, 29] has already been discussed in the main paper. Note that the global/local categorization here refers to whether the factors and the subsequent processing is limited to local image regions.

Training: Training time of our RSFNet is quite fast. For any Lol dataset [47, 78, 86], 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. (6) and then freeze the factorization parameters for next 25 epochs to train the fusion module with Eq. (9). 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.

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 the main paper or supplementary are with this setting only.

Several optimization methods are sensitive to initialization conditions and when they are unrolled into model layers [11, 46]. 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 will be provided with the final code in a config file.

Testing: For inference, we can edit the weights of the fac-

tors before concatenation and input into the fusion module to allow varying results. Although all results in the main paper 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 Tab. 10 can be updated to Tab. 6 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 Tab. 10

Datasets: The details of five no-reference (Tab. 4) and four Lol datasets Tab. 10 are given below:

- Lolv1 [78]: It contains 500 low light and well lit image pairs of real worl 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 [86]: 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 [86]: 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 [47]: 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 [41] 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 [3, 24, 41], we use the download links provided by Li et al. [41] with the following breif description of each dataset:

$Type$	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 6. **Factor Weights:** Our updated 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 Tab. 2 and Tab. 10.

Config	Factorization		Fusion		Experiment
	Trad.	Deep	Trad.	Deep	
C_{11}		✓		✓	RSFNet LLE Fig. 3
C_{10}		✓	✓		Abla. (w/o Fusion) Tab. 3
C_{01}	✓			✓	Extension Apps. Fig. 6
C_{00}	✓		✓		User Apps. Fig. 7

Table 7. **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

- DICM [37]: 69 images, mean=0.32, mixed exposure settings, variable resolutions, real scenes, varying scene including macros, landscapes, indoors, outdoors *etc*.
- LIME [26]: 10 images, mean=0.15, varying resolutions, real scenes, varying scene types.
- MEF [50]: 17 images, mean=0.15, resolution=512 × 340, relatively darker images, varying scene types.
- NPE [76]: 85 images, mean=0.31, varying resolution, both over and under exposed image regions, mostly outdoor scenes.
- VV [73]: 24 images, mean=0.26, resolution=2304 × 1728, large images, both over and under exposed image regions, both indoor/outdoor scene types.

These results are listed in Tab. 2 Tab. 4 and Tab. 10. 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.

Metrics: 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:

$$p = 10 \log \left[\frac{\frac{1}{N} \sum_{i}^{N} (\hat{y}_i - y_i)^2}{M^2} \right], \tag{11}$$

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. (11) 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:

$$p = 10 \log \left[\frac{\frac{1}{N*C} \sum_{c} \sum_{i}^{N} (\hat{y}_{i,c} - y_{i,c})^2}{M^2} \right]. \quad (12)$$

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:

$$Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \quad (13)$$

whereas Matlab prefers the digital transformation as:

$$Y \leftarrow 0.2568 \cdot R + 0.5041 \cdot G + 0.0979 \cdot B \quad (14)$$

This leads to variability in results (approximately 1 PSNR difference) depending upon the conversion library chosen. In our opinion Eq. (14) should be chosen and the PSNR tables should clearly highlight that it is a single Y channel evaluation.

Configurations: Our proposed method can be used for various applications in one of four possible configurations as shown in Tab. 7. 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. In our main paper we have used all four configurations in one or the other experiment as listed in the Tab. 7. For traditional factorization we use solution to the direct specularity estimation optimization equation Eq. (1) using Eq. (2), whereas for deep solution we use the unrolled layers Fig. 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. (10). 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.

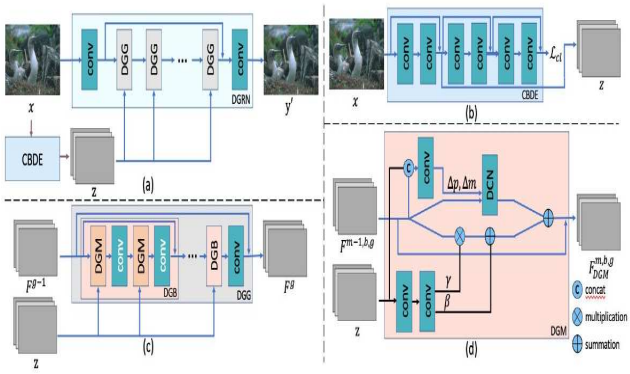

Figure 10. **AirNet:** (a) block diagram from [40]. CBDE (b) refers to Contrastive-Based Degradation Encoder, DGG (c) means Degradation Guided Groups and DGM (d) is Degradation Guided Module. For complete details refers to [40]. For our usage, we alter first conv layer (first deep blue block on top-left (a)) and the first conv layer in CBDE (first deep blue block on top-right (b)).

Extensions: In order to show the utility of our factors beyond the LLE task, we have shown the advantage of using them along with the pre-existing multi-task enhancement networks. Specifically, we use AirNet [40] (Fig. 10) 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 warmup) 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 the extension of Tab. 5 in Tab. 8 as the full comparison table using the values as provided by [88] 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 [88] was not used as the public code is not available at the time of writing of this paper. 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. Note that there is copying error in one of the values in the Tab. 5. Deblurring PSNR value for the first row *i.e.* AirNet (multi-task) configuration, is reported as 18.18 whereas it should be 24.35 as corrected in the Tab. 8. Still there is 4.7 % PSNR and 2.3 % SSIM performance boost of the mean scores with our prior induction.

TASK →	DEHAZE [39]		DERAIN [84]		DEBLUR [56]		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 8. **Prior Induction:** Our factors can induce structure prior in an existing base model [40] and improve performance for multiple enhancement tasks. Here we show extension of Tab. 5 in the main paper in the wider context of similar methods.

NIQE ↓	SNR [81]	RFormer [12]	HEP [87]	NeRCo [83]	RSFNet (Ours)
DICM [37]	3.622	3.076	4.064	3.553	3.230
LIME [26]	3.752	3.910	3.981	3.422	3.800
MEF [50]	3.917	3.135	3.648	3.152	3.000
NPE [76]	3.535	3.63*	2.986	3.241	3.310
VV [73]	2.887	2.183	3.596	3.169	1.960
Mean	3.543	3.187	3.655	3.307	**3.060**

Table 9. **Generalized Performance:** Performance generalization comparison (Tab. 4 extension) of best ranking (Tab. 11) two supervised LLE solutions (first two columns: SNR [81], RFormer [12]) and two unsupervised LLE solutions (last two columns: HEP [87], NeRCo [83]) *vs.* our zero-reference RSFNet method on five no-reference benchmarks namely: DICM [37], LIME [26], MEF [50], NPE [76] and VV [73]. Our method is able to generalize better to unseen data compared to others as observed from the overall lowest NIQE scores [54] 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).

Visualizations: We provide several visualizations of our results mentioned in the main paper. Specifically, we provide the following:

- Visualization of our five extracted specular factors for the shadow_ADE dataset [30] in Fig. 11
- Visualization of our five extracted specular factors for the IIW dataset [6] in Fig. 12
- Our qualitative results on low light image benchmarks in Fig. 13.
- Qualitative comparison of our results with other zero-reference LLE solutions in Fig. 14
- Our results for the deraining application on the Rain100L dataset [84] in Fig. 15.
- Our results for the dehazing application on the RESIDE SOTS outdoor dataset [39] in Fig. 16.
- Our results for the deblurring application on the GoPro dataset [56] in Fig. 17.
- High resolution versions of the user controlled edited images (Fig. 7) in GIMP [71] in Fig. 18.
- Extended quantitative comparison scores with contemporary traditional and zero-reference solutions (extension of Tab. 2) in Tab. 10.
- Quantitative comparison of our method with contemporary unsupervised LLE solutions on three Lol benchmarks in Tab. 11.

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 Tab. 9 . We report NIQE scores [54] to assess the overall perceptual quality and the naturalness of the generated results. As can be seen from the Tab. 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 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.

CVPR
#715

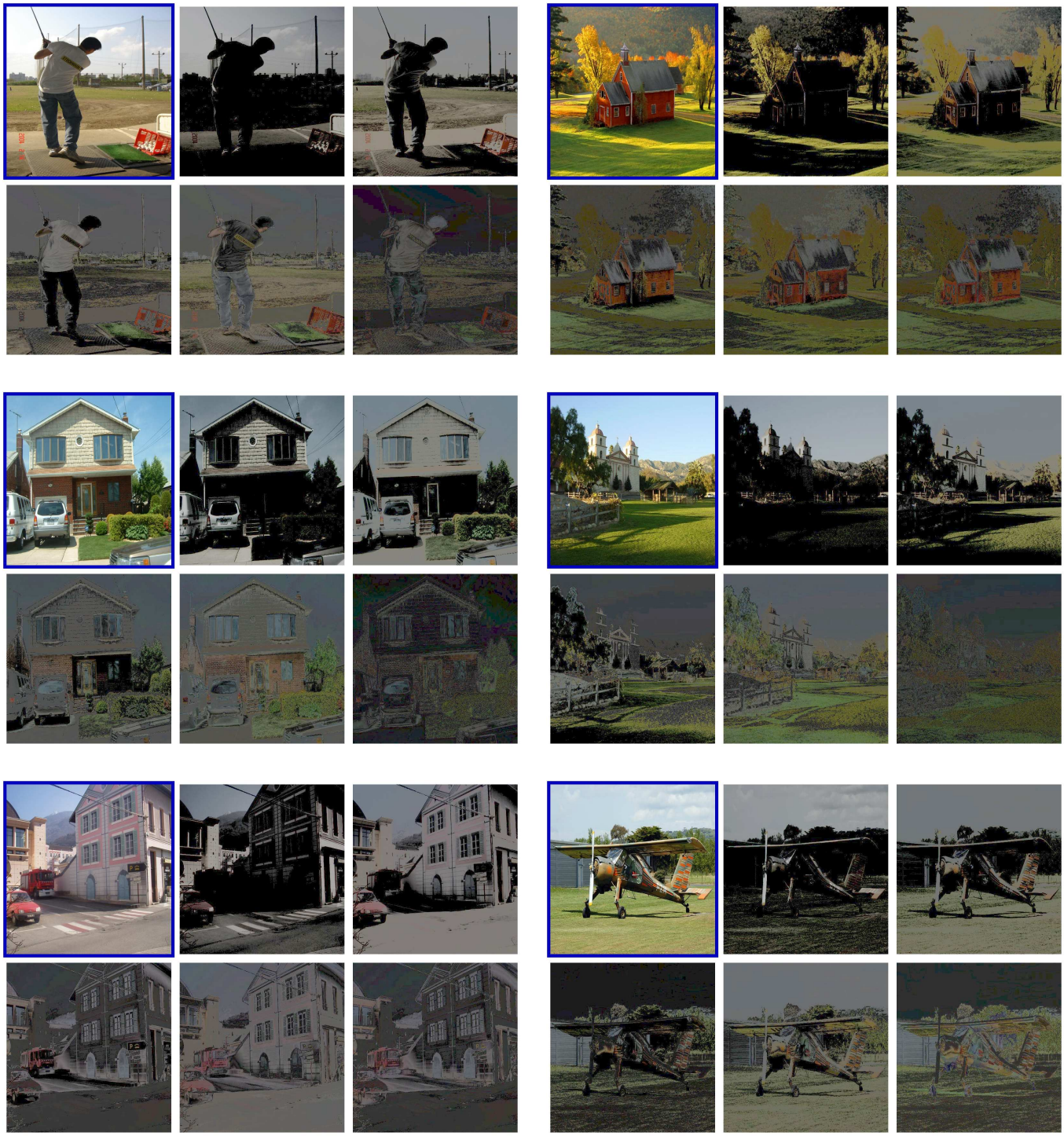

Figure 11. **Factor Visualizations:** We show visualizations of our extracted five specular factors for various scenes. Input images (blue box) are taken from [30] dataset and factors are rescaled for visualization. Note how various regions are captured in the respective factors depending upon whether they are illuminated by directly, indirectly or in shadows.

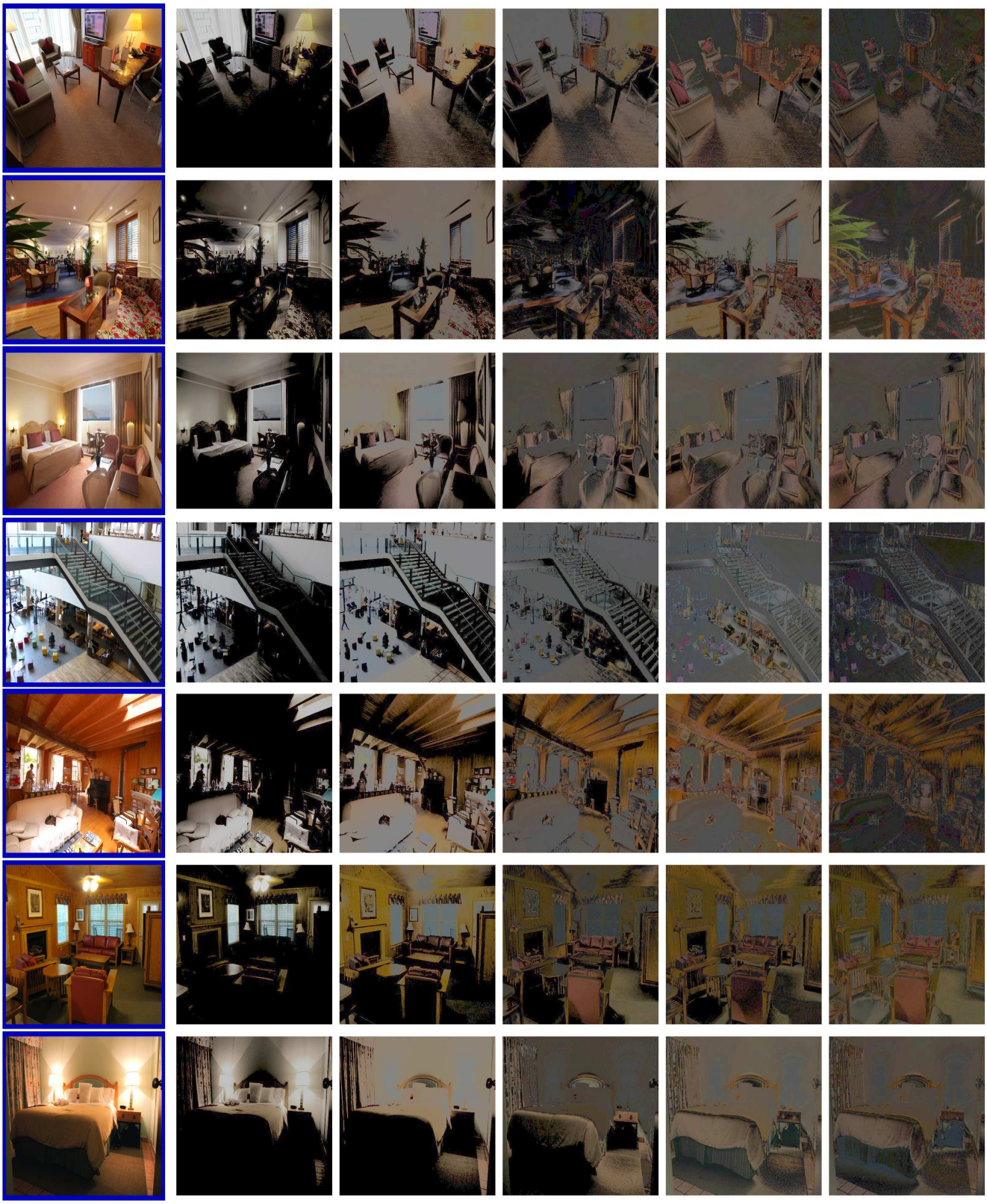

Figure 12. **More Factor Visualizations:** We show visualizations of our extracted five specular factors for various scenes. Input images (blue box) are taken from [6] dataset and factors are rescaled for visualization. Note how various regions are captured in the respective factors depending upon whether they are illuminated by directly, indirectly or in shadows.

Figure 13. **Our LLE Results**: Additional low light enhancement results from multiple LOl-x datasets [78, 86]. Each set contains input image (blue box), ground truth (red box) and our result (green box).

CVPR
#715

Figure 14. **Qualitative Comparisons**: Additional low light enhancement comparisons (Fig. 4 extension). Each set row in the grid contains results from: [SDD[27], ECNet[89], ZDCE[24]]; [ZD++[42], RUAS[66], SCI[51]]; [PNet[57], GDP[18], 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 Tab. 4 and Fig. 5.

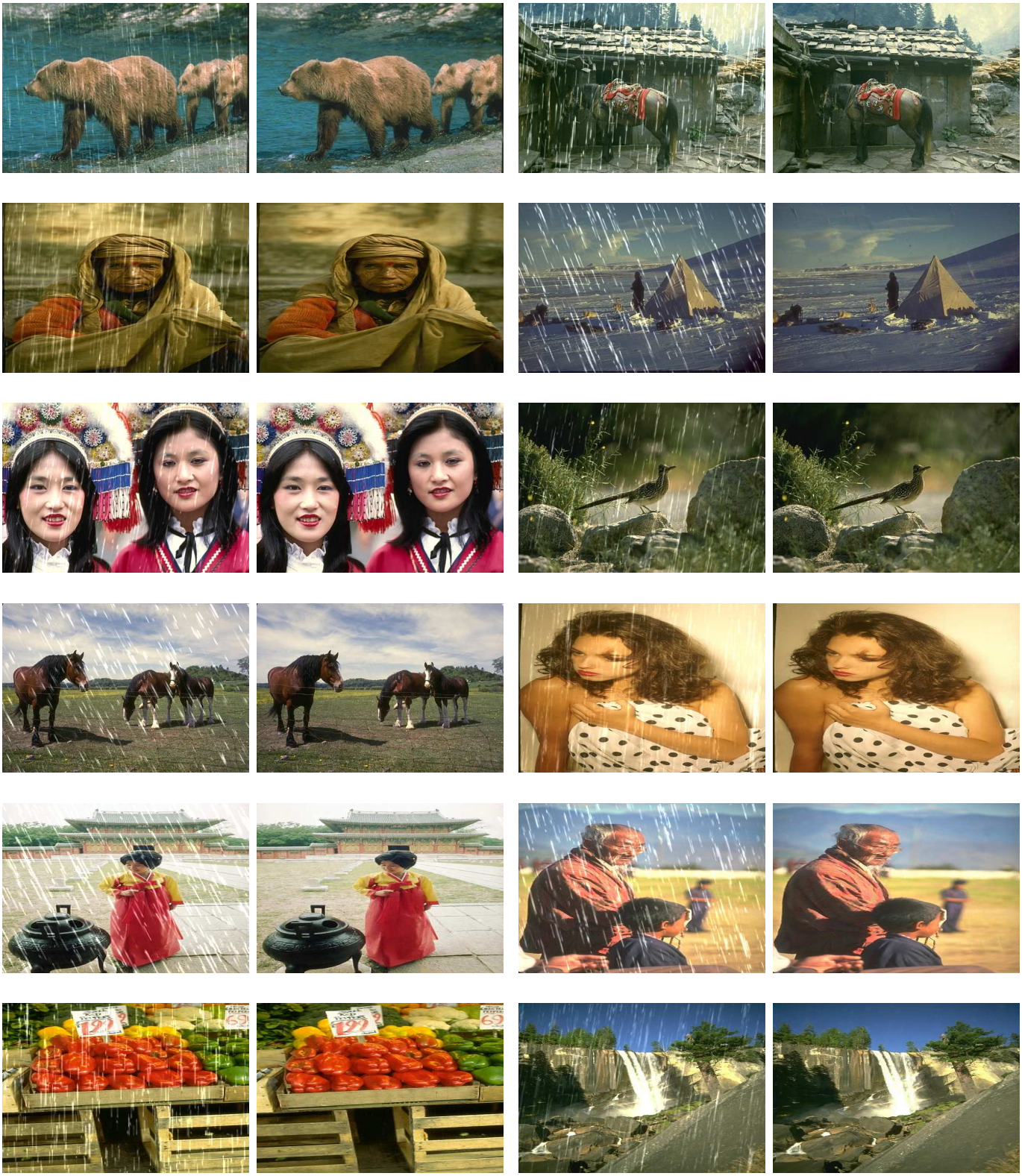

Figure 15. **Our Deraining Results**: Additional results (Fig. 6 extension) for the deraining application on the Rain100L dataset [84].

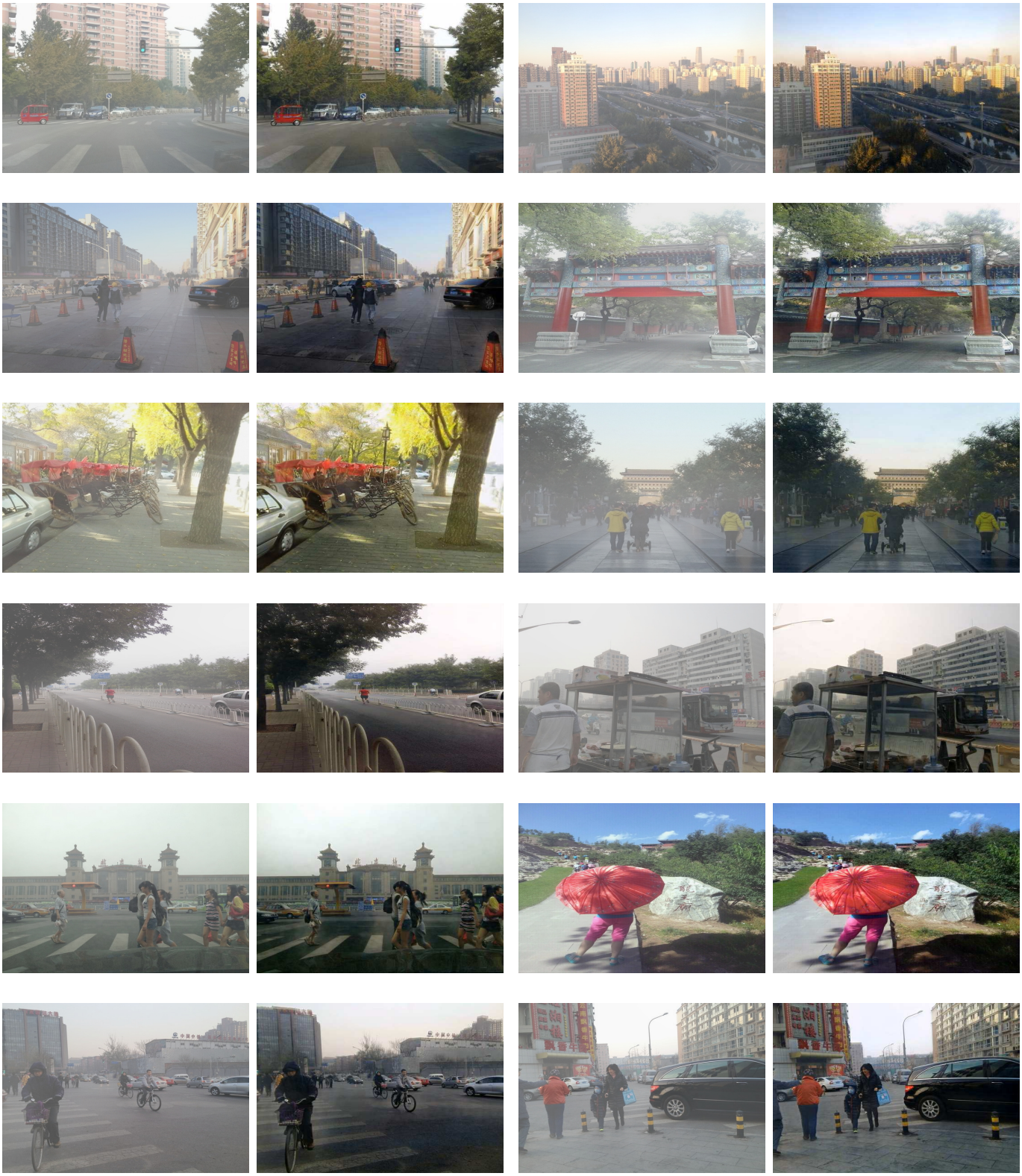

Figure 16. **Our Dehazing Results**: Additional results (Fig. 6 extension) for the dehazing application on the RESIDE dataset [39].

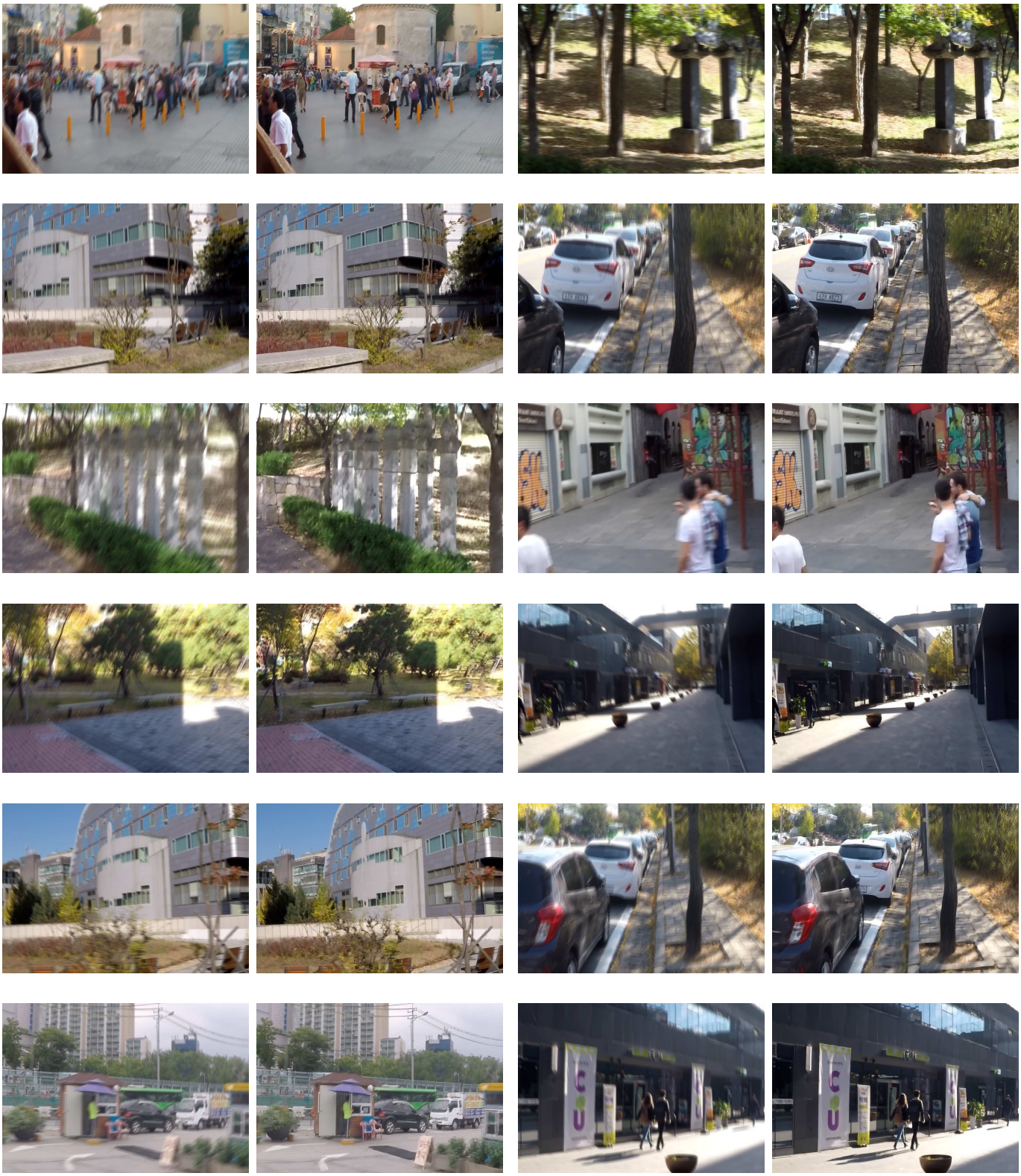

Figure 17. **Our Deblurring Results**: Additional results (Fig. 6 extension) for the deblurring application on the GoPro dataset [56].

Figure 18. **User-controlled Edits:** Here we show high resolution version of our results in Fig. 7. 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 [71] 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.

Paradigm	Traditional Model Based			Zero-reference							
Method	LIME [26]	DUAL [91]	SDD [27]	ECNet [89]	ZDCE [24]	ZD++ [42]	RUAS [66]	SCI [51]	PNet [57]	GDP [18]	**RSFNet** (Ours)
#Params	-	-	-	16.5M	79.42K	10.56K	3.43K	**0.26**K	15.25K	552K	2.11K
Lolv1 [78] (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 [86] (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 [86] (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 [47] (dataset split: 1400/100, mean≈ 0.07, resolution: 400 × 600)											
$PSNR_y$ ↑	19.31	19.10	18.47	18.72	20.31	19.36	17.49	19.37	20.39	16.29	**21.18**
$SSIM_y$ ↑	0.705	0.704	0.792	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	17.64	14.42	**18.06**
$SSIM_c$ ↑	0.537	0.535	0.678	0.457	0.580	0.442	0.493	0.540	0.557	0.498	**0.714**
NIQE ↓	9.076	9.083	4.191	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	0.310	0.340	0.390	0.355	0.368	0.444	0.397
Mean Scores (**Lolv1** [78], **Lolv2-real** [86], **Lolv2-syn** [86] and **VE-Lol** [47])											
$PSNR_y$ ↑	18.50	17.83	17.50	18.45	19.26	18.73	17.09	18.07	19.65	15.88	**21.16**
$SSIM_y$ ↑	0.737	0.728	0.781	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	17.35	14.15	**18.45**
$SSIM_c$ ↑	0.596	0.583	0.679	0.538	0.634	0.548	0.532	0.587	0.605	0.504	**0.758**
NIQE ↓	7.855	7.880	4.106	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	0.273	0.296	0.346	0.295	0.302	0.379	0.276

Table 10. **Quantitative comparison** of our method RSFNet with other **traditional and zero-reference** solutions on multiple lowlight benchmarks and six evaluation metrics. Shown here are scores for two datasets LOLv1 and LOLv2-real with mean value across all datasets in the last sub-table. Our scores here are same as the ones reported in last sub-table in Tab. 2 in the main paper (key: ↑ higher better; ↓ lower better; **bold**: best; underline: second best; '-': NaN error computing value).

Paradigm	*Supervised LLE*				**Unsupervised LLE**					**Zero Reference**
Method	*URe-tinex* [79]	*CUE* [96]	*SNR* [81]	*RFormer* [12]	EGAN [32]	HEP [87]	PairLIE* [22]	CLIP-LIT [44]	NeRCo* [83]	**RSFNet** (Ours)
					Lolv1 [78] (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 [86] (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 [86] (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** [78], **Lolv2-real** [86], **Lolv2-syn** [86])					
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**
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 11. **Quantitative comparison** of our method RSFNet with five other **Unsupervised LLE** solutions [22, 32, 44, 83, 87] and four recent Supervised LLE solutions [12, 79, 81, 96] 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 [22] and NeRCo [83], 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 [83] and [87], 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.).

References

[1] Amir Adler, Michael Elad, Yacov Hel-Or, and Ehud Rivlin. Sparse coding with anomaly detection. In *2013 IEEE MLSP*, 2013. 4

[2] Adobe Inc. Adobe photoshop, 2023. 8

[3] Mahmoud Afifi, Konstantinos Derpanis, Bjorn Ommer, and Michael Brown. Learning multi-scale photo exposure correction. In *CVPR*, 2021. 1, 2, 3

[4] Yağız Aksoy, Tae-Hyun Oh, Sylvain Paris, Marc Pollefeys, and Wojciech Matusik. Semantic soft segmentation. *ACM ToG (SIGGRAPH)*, 37(4), 2018. 2

[5] Anil S. Baslamisli, Partha Das, Hoang-An Le, Sezer Karaoglu, and Theo Gevers. Shadingnet: Image intrinsics by fine-grained shading decomposition. *IJCV*, 129(8), 2021. 2, 3

[6] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM Trans. on Graphics (SIGGRAPH)*, 33(4), 2014. 5, 7

[7] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Deep burst super-resolution. *CVPR*, 2021. 3

[8] Goutam Bhat, Martin Danelljan, Fisher Yu, Luc Van Gool, and Radu Timofte. Deep reparametrization of multi-frame super-resolution and denoising. *ICCV*, 2021. 3

[9] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 4

[10] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1), 2011. 4

[11] HanQin Cai, Jialin Liu, and Wotao Yin. Learned robust pca: A scalable deep unfolding approach for high-dimensional outlier detection. *NeurIPS*, 34, 2021. 4, 8, 2

[12] Yuanhao Cai, Hao Bian, Jing Lin, Haoqian Wang, Radu Timofte, and Yulun Zhang. Retinexformer: One-stage retinex-based transformer for low-light image enhancement. In *ICCV*, 2023. 5, 15

[13] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 3, 5, 1

[14] Turgay Celik and Tardi Tjahjadi. Contextual and variational contrast enhancement. *IEEE TIP*, 20(12), 2011. 2

[15] Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Theoretical linear convergence of unfolded ista and its practical weights and thresholds. In *NeurIPS*, 2018. 4

[16] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57, 2003. 4

[17] Chi-Mao Fan, Tsung-Jung Liu, and Kuan-Hsien Liu. Half wavelet attention on m-net+ for low-light image enhancement. In *IEEE ICIP*, 2022. 1, 2

[18] Ben Fei, Zhaoyang Lyu, Liang Pan, Junzhe Zhang, Weidong Yang, Tianyue Luo, Bo Zhang, and Bo Dai. Generative diffusion prior for unified image restoration and enhancement. In *CVPR*, 2023. 6, 7, 1, 9, 14

[19] Gang Fu, Qing Zhang, Chengfang Song, Qifeng Lin, and Chunxia Xiao. Specular highlight removal for real-world images. *Computer Graphics Forum*, 38(7), 2019. 2

[20] Xueyang Fu, Delu Zeng, Yue Huang, Yinghao Liao, Xinghao Ding, and John Paisley. A fusion-based enhancing method for weakly illuminated images. *Signal Processing*, 129, 2016. 2

[21] Xueyang Fu, Delu Zeng, Yue Huang, Xiao-Ping Zhang, and Xinghao Ding. A weighted variational model for simultaneous reflectance and illumination estimation. In *CVPR*, 2016. 2

[22] Zhenqi Fu, Yan Yang, Xiaotong Tu, Yue Huang, Xinghao Ding, and Kai-Kuang Ma. Learning a simple low-light image enhancer from paired low-light instances. In *CVPR*, 2023. 1, 15

[23] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *ICML*, 2010. 4

[24] Chunle Guo, Chongyi Li, Jichang Guo, Chen Change Loy, Junhui Hou, Sam Kwong, and Cong Runmin. Zero-reference deep curve estimation for low-light image enhancement. *CVPR*, 2020. 2, 3, 5, 6, 7, 1, 9, 14

[25] Jie Guo, Zuojian Zhou, and Limin Wang. Single image highlight removal with a sparse and low-rank reflection model. In *ECCV*, 2018. 4

[26] Xiaojie Guo, Yu Li, and Haibin Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE TIP*, 26(2), 2016. 2, 6, 7, 3, 5, 14

[27] Shijie Hao, Xu Han, Yanrong Guo, Xin Xu, and Meng Wang. Low-light image enhancement with semi-decoupled decomposition. *IEEE TMM*, 22(12), 2020. 6, 7, 9, 14

[28] Charles Hessel. Simulated Exposure Fusion. *Image Processing On Line*, 9, 2019. 2, 3, 5

[29] Charles Hessel and Jean-Michel Morel. An extended exposure fusion and its application to single image contrast enhancement. In *WACV*, 2020. 1, 2, 3, 5, 6

[30] Xiaowei Hu, Tianyu Wang, Chi-Wing Fu, Yitong Jiang, Qiong Wang, and Pheng-Ann Heng. Revisit-

ing shadow detection: A new benchmark dataset for complex world. *IEEE TIP*, 30, 2021. 3, 5, 1, 6

[31] Jie Huang, Yajing Liu, Feng Zhao, Keyu Yan, Jinghao Zhang, Yukun Huang, Man Zhou, and Zhiwei Xiong. Deep fourier-based exposure correction network with spatial-frequency interaction. In *ECCV*, 2022. 1, 2, 3

[32] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. Enlightengan: Deep light enhancement without paired supervision. *IEEE TIP*, 30, 2021. 2, 1, 15

[33] Johann Heinrich Lambert. *Photometria sive de mensura et gradibus luminis, colorum et umbrae*. Klett, 1760. 3

[34] Edwin Herbert Land. The retinex theory of color vision. *Scientific American*, 237 6, 1977. 1

[35] Bruno Lecouat, Jean Ponce, and Julien Mairal. Designing and learning trainable priors with non-cooperative games. *NeurIPS*, 2020. 3

[36] Bruno Lecouat, Jean Ponce, and Julien Mairal. Fully trainable and interpretable non-local sparse models for image restoration. *ECCV*, 2020. 3

[37] Chulwoo Lee, Chul Lee, and Chang-Su Kim. Contrast enhancement based on layered difference representation of 2d histograms. *IEEE TIP*, 22(12), 2013. 6, 7, 3, 5

[38] Chulwoo Lee, Chul Lee, and Chang-Su Kim. Contrast enhancement based on layered difference representation of 2d histograms. *IEEE TIP*, 22(12), 2013. 2

[39] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE TIP*, 28(1), 2019. 8, 5, 11

[40] Boyun Li, Xiao Liu, Peng Hu, Zhongqin Wu, Jiancheng Lv, and Xi Peng. All-In-One Image Restoration for Unknown Corruption. In *CVPR*, 2022. 8, 4, 5

[41] Chongyi Li, Chunle Guo, Linghao Han, Jun Jiang, Ming-Ming Cheng, Jinwei Gu, and Chen Change Loy. Low-light image and video enhancement using deep learning: A survey. *IEEE TPAMI*, 2021. 1, 2, 3

[42] Chongyi Li, Chunle Guo, and Chen Change Loy. Learning to enhance low-light image via zero-reference deep curve estimation. *IEEE TPAMI*, 2021. 2, 6, 7, 1, 9, 14

[43] Jinxiu Liang, Yong Xu, Yuhui Quan, Boxin Shi, and Hui Ji. Self-supervised low-light image enhancement using discrepant untrained network priors. *IEEE TCSVT*, 32(11), 2022. 2

[44] Zhexin Liang, Chongyi Li, Shangchen Zhou, Ruicheng Feng, and Chen Change Loy. Iterative prompt learning for unsupervised backlit image enhancement. In *ICCV*, 2023. 1, 15

[45] Seokjae Lim and Wonjun Kim. Dslr: Deep stacked laplacian restorer for low-light image enhancement. *IEEE TMM*, 23, 2021. 1, 2, 3

[46] Jialin Liu, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. ALISTA: Analytic weights are as good as learned weights in LISTA. In *ICLR*, 2019. 4, 8, 2

[47] Jiaying Liu, Xu Dejia, Wenhan Yang, Minhao Fan, and Haofeng Huang. Benchmarking low-light image enhancement and beyond. *IJCV*, 129, 2021. 6, 2, 3, 14

[48] Yang Liu, Jinshan Pan, Jimmy Ren, and Zhixun Su. Learning deep priors for image dehazing. In *ICCV*, 2019. 3

[49] Yuen Peng Loh and Chee Seng Chan. Getting to know low-light images with the exclusively dark dataset. *CVIU*, 178, 2019. 1

[50] Kede Ma, Kai Zeng, and Zhou Wang. Perceptual quality assessment for multi-exposure image fusion. *IEEE TIP*, 24(11), 2015. 6, 7, 3, 5

[51] Long Ma, Tengyu Ma, Risheng Liu, Xin Fan, and Zhongxuan Luo. Toward fast, flexible, and robust low-light image enhancement. In *CVPR*, 2022. 1, 2, 3, 6, 7, 9, 14

[52] John J. McCann. Retinex at 50: color theory and spatial algorithms, a review. *Journal of Electronic Imaging*, 26, 2017. 1

[53] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion: A simple and practical alternative to high dynamic range photography. *Computer Graphics Forum*, 28, 2009. 3, 6

[54] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3), 2013. 7, 5

[55] Vishal Monga, Yuelong Li, and Yonina C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2), 2021. 2

[56] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017. 8, 5, 12

[57] Hue Nguyen, Diep Tran, Khoi Nguyen, and Rang Nguyen. Psenet: Progressive self-enhancement network for unsupervised extreme-light image enhancement. In *WACV*, 2023. 2, 3, 5, 6, 7, 1, 9, 14

[58] Zhangkai Ni, Wenhan Yang, Shiqi Wang, Lin Ma, and Sam Kwong. Towards unsupervised deep image enhancement with generative adversarial network. *IEEE TIP*, 29, 2020. 2

[59] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3), 2014. 4

[60] Stephen M Pizer, E Philip Amburn, John D Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *CVGIP*, 39(3), 1987. 2

[61] Densen Puthussery, Hrishikesh Panikkasseril Sethumadhavan, Melvin Kuriakose, and Jiji Charangatt Victor. Wdrn: A wavelet decomposed relightnet for image relighting. In *ECCV workshop*, 2020. 1, 2

[62] Chao Ren, Yizhong Pan, and Jie Huang. Enhanced latent space blind model for real image denoising via alternative optimization. In *NeurIPS*, 2022. 3

[63] Xutong Ren, Wenhan Yang, Wen-Huang Cheng, and Jiaying Liu. Lr3m: Robust low-light enhancement via low-rank regularized retinex model. *IEEE TIP*, 29, 2020. 1, 2, 3

[64] Ali M. Reza. Realization of the contrast limited adaptive histogram equalization (clahe) for real-time image enhancement. *J. VLSI Signal Process. Syst.*, 38 (1), 2004. 2

[65] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *WACV*, 2020. 5

[66] Liu Risheng, Ma Long, Zhang Jiaao, Fan Xin, and Luo Zhongxuan. Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement. In *CVPR*, 2021. 2, 3, 6, 7, 1, 9, 14

[67] Thomas Robert, Nicolas Thome, and Matthieu Cord. Hybridnet: Classification and reconstruction cooperation for semi-supervised learning. In *ECCV*, 2018. 2

[68] Saurabh Saini and P. J. Narayanan. Quaternion factorized simulated exposure fusion. In *ACM ICVGIP*, 2023. 1, 2

[69] Aashish Sharma and Robby T. Tan. Nighttime visibility enhancement by increasing the dynamic range and suppression of light effects. *CVPR*, 2021. 2, 3

[70] Sumit Shekhar, Max Reimann, Maximilian Mayer, Amir Semmo, Sebastian Pasewaldt, Jürgen Döllner, and Matthias Trapp. Interactive photo editing on smartphones via intrinsic decomposition. *Computer Graphics Forum*, 40(2), 2021. 4

[71] The GIMP Development Team. Gimp, 2023. 8, 5, 13

[72] Shoji Tominaga. Dichromatic reflection models for a variety of materials. *Color Research and Application*, 19, 1994. 3, 4

[73] Vassilios Vonikakis. Busting image enhancement and tone-mapping algorithms. https://sites.google.com/site/vonikakis/datasets/, 2007. [Online; accessed 26-Oct-2023]. 6, 7, 3, 5

[74] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. 2020. 3

[75] Shuhang Wang, Jin Zheng, Hai-Miao Hu, and Bo Li. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE TIP*, 22(9), 2013. 7

[76] Shuhang Wang, Jin Zheng, Hai-Miao Hu, and Bo Li. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE TIP*, 22(9), 2013. 2, 6, 7, 3, 5

[77] Zhou 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. 6

[78] Chen Wei, Wenjing Wang, Yang Wenhan, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. In *BMVC*, 2018. 2, 6, 3, 8, 14, 15

[79] Wenhui Wu, Jian Weng, Pingping Zhang, Xu Wang, Wenhan Yang, and Jianmin Jiang. Uretinex-net: Retinex-based deep unfolding network for low-light image enhancement. In *CVPR*, 2022. 2, 3, 4, 15

[80] Ke Xu, Xin Yang, Baocai Yin, and Rynson W.H. Lau. Learning to restore low-light images via decomposition-and-enhancement. In *CVPR*, 2020. 1, 2, 3

[81] Xiaogang Xu, Ruixing Wang, Chi-Wing Fu, and Jiaya Jia. Snr-aware low-light image enhancement. In *CVPR*, 2022. 2, 5, 15

[82] Wending Yan, Robby T Tan, and Dengxin Dai. Nighttime defogging using high-low frequency decomposition and grayscale-color networks. In *ECCV*, 2020. 2, 1

[83] Shuzhou Yang, Moxuan Ding, Yanmin Wu, Zihan Li, and Jian Zhang. Implicit neural representation for cooperative low-light image enhancement. In *ICCV*, 2023. 1, 5, 15

[84] Wenhan Yang, Robby T. Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng 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. 8, 5, 10

[85] Wenhan Yang, Shiqi Wang, Yapplicationsuming Fang, Yue Wang, and Jiaying Liu. Band representation-based semi-supervised low-light image enhancement: Bridging the gap between signal fidelity and perceptual quality. *IEEE TIP*, 30, 2021. 2, 3

[86] Wenhan Yang, Wenjing Wang, Haofeng Huang, Shiqi Wang, and Jiaying Liu. Sparse gradient regularized deep retinex network for robust low-light image enhancement. *IEEE TIP*, 30, 2021. 2, 6, 3, 8, 14, 15

[87] Feng Zhang, Yuanjie Shao, Yishi Sun, Kai Zhu, Changxin Gao, and Nong Sang. Unsupervised low-light image enhancement via histogram equalization prior. *arXiv:2112.01766*, 2021. 2, 6, 1, 5, 15

[88] Jinghao Zhang, Jie Huang, Mingde Yao, Zizheng Yang, Huikang Yu, Man Zhou, and Fengmei Zhao. Ingredient-oriented multi-degradation learning for image restoration. *CVPR*, 2023. 8, 4

[89] Lin Zhang, Lijun Zhang, Xinyu Liu, Ying Shen, Shaoming Zhang, and Shengjie Zhao. Zero-shot restoration of back-lit images using deep internal learning. *ACM MM*, 2019. 2, 6, 7, 1, 9, 14

[90] Qing Zhang, Ganzhao Yuan, Chunxia Xiao, Lei Zhu, and Wei-Shi Zheng. High-quality exposure correction of underexposed photos. In *ACM MM*, 2018. 2

[91] Qing Zhang, Yongwei Nie, and Weishi Zheng. Dual illumination estimation for robust exposure correction. *Computer Graphics Forum*, 38, 2019. 2, 6, 7, 14

[92] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[93] Yonghua Zhang, Jiawan Zhang, and Xiaojie Guo. Kindling the darkness: A practical low-light image enhancer. In *ACM MM*, 2019. 2

[94] Yonghua Zhang, Xiaojie Guo, Jiayi Ma, Wei Liu, and Jiawan Zhang. Beyond brightening low-light images. *IJCV*, 129, 2021. 2

[95] Chuanjun Zheng, Daming Shi, and Wentian Shi. Adaptive unfolding total variation network for low-light image enhancement. *ICCV*, 2021. 3, 4

[96] Naishan Zheng, Man Zhou, Yanmeng Dong, Xiangyu Rui, Jie Huang, Chongyi Li, and Fengmei Zhao. Empowering low-light image enhancer through customized learnable priors. 2023. 15

[97] Shen Zheng and Gaurav Gupta. Semantic-guided zero-shot learning for low-light image/video enhancement. In *WACV*, 2022. 3, 5

[98] Anqi Zhu, Lin Zhang, Ying Shen, Yong Ma, Shengjie Zhao, and Yicong Zhou. Zero-shot restoration of underexposed images via robust retinex decomposition. *ICME*, 2020. 2

[99] Yurui Zhu, Zeyu Xiao, Yanchi Fang, Xueyang Fu, Zhiwei Xiong, and Zheng-Jun Zha. Efficient model-driven network for shadow removal. *AAAI*, 2022. 3