Specularity Factorization for Low-Light Enhancement

Supplementary Material

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

Mathematical Derivation: The equations in the main paper (Eq. (2)) are derived directly following relevant sections from Boyd and Vandenberghe [10], Boyd et al. [11], Parikh and Boyd [65].

Initial sparsity objective Eq. (1) can be re-written in augmented Lagrangian form [11] with Y, μ as auxiliary variables and $'$ implying matrix transpose as:

$$\underset{\mathbf{E},\mathbf{A}}{\operatorname{argmin}} \|\mathbf{A}\|_* + \lambda\|\mathbf{E}\|_1 + \mathbf{Y}^{'}(\mathbf{A}+\mathbf{E}-\mathbf{X}) + \frac{\mu}{2}\|A+E-X\|_2^2.$$

After using dual function and variable separation [10, 11], we get the ADMM updates for $\mathbf{E}$, $\mathbf{A}$ and $\mathbf{Y}$:

$$\mathbf{E} \leftarrow \underset{\mathbf{E}}{\operatorname{argmin}} \left(\lambda\|E\|_1 + \mathbf{Y}^{'}(\mathbf{A}+\mathbf{E}-\mathbf{X}) + \frac{\mu}{2}\|\mathbf{A}+\mathbf{E}-\mathbf{X}\|_2^2\right),$$
$$\mathbf{A} \leftarrow \underset{\mathbf{A}}{\operatorname{argmin}} \left(\|A\|_* + \mathbf{Y}^{'}(\mathbf{A}+\mathbf{E}-\mathbf{X}) + \frac{\mu}{2}\|\mathbf{A}+\mathbf{E}-\mathbf{X}\|_2^2\right),$$
$$\mathbf{Y} \leftarrow \mathbf{Y} + \mu(\mathbf{A}+\mathbf{E}-\mathbf{X}).$$

We can eliminate the second term by collecting $\mathbf{Y}$ inside the third square term:

$$\mathbf{E} \leftarrow \underset{\mathbf{E}}{\operatorname{argmin}} \left(\lambda\|E\|_1 + \frac{\mu}{2}\|\mathbf{A}+\mathbf{E}-\mathbf{X}+Y/\mu\|_2^2\right),$$
$$\mathbf{A} \leftarrow \underset{\mathbf{A}}{\operatorname{argmin}} \left(\|A\|_* + \frac{\mu}{2}\|\mathbf{A}+\mathbf{E}-\mathbf{X}+Y/\mu\|_2^2\right),$$
$$\mathbf{Y} \leftarrow \mathbf{Y} + \mu(\mathbf{A}+\mathbf{E}-\mathbf{X})$$

Note that additional $\mathbf{Y}$ terms in each update have no effect on the respective argmin solution. Now using the definition of the p-norm proximal operator [65] given by:

$$\delta_\mu^p(v) := \underset{x}{\operatorname{argmin}}(\|x\|_p + \frac{\mu}{2}\|x-v\|_2^2),$$

we can obtain the Eq. (2) in the main paper using $*$ and L_1 soft-thresholding.

Unsupervised *vs*. Zero-reference LLE: Although similar, there is a crucial difference between the unsupervised and zero-reference LLE paradigms [46]. As mentioned previously, unsupervised LLE solutions like [24, 36, 49, 91, 92, 96] 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 [20, 27, 47, 57, 63, 72, 98] 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 [20], 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 [92] and [96]. [92] 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 [92] on 4 out of 6 and [96] 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 [34]. [34] 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 [14] 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

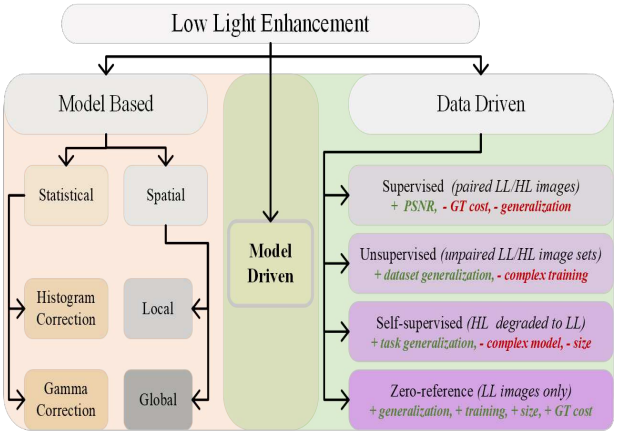

Figure 8. **LLE solutions categorization:** Data-driven methods are of mainly 4 types based on the type of input supervision available with each type having its pros and cons as listed above.

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*.

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 [89] 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 [35] 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 to the Retinex division. [69] focuses on hyper-spectral images, whereas [76] uses a complicated quaternion based robust PCA optimization strategy [12] with no unrolled learning or generalization to other applications. *Wavelets* and *Multiscale* decompositions [3, 18] 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, 78] 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 [32, 33] 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

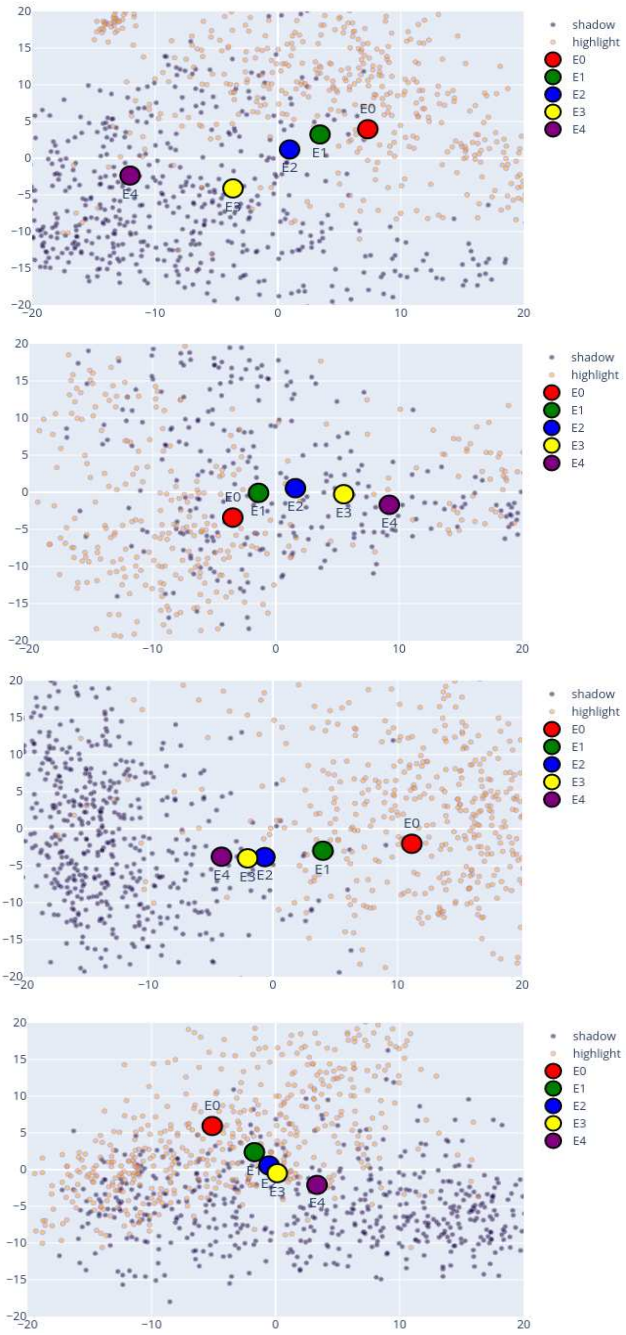

Figure 9. **Factor interpretability and analysis:** We perform factor distribution analysis Fig. 2 on four additional shadow datasets [34] (from top to bottom - shadow_KITTI, shadow_MAP, shadow_USR and shadow_WEB). Each plot represents features of shadow and non-shadow regions which forms the background 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 similar to Fig. 2 which provides more evidence to the validation done in Sec. 3.

any Lol dataset [52, 87, 95], 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 [12, 51]. 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 factors 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 [87]: 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 $\times$ 600 in resolution with mean intensity = 0.05 (*i.e.* very low light).
- Lolv2-real [95]: 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 $\times$ 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 [95]: 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 $\times$ 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 [52]: 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 $\times$ 600 and mean intensity=0.07. Multiple exposure settings here help ascertain model's robustness to input perturbations.

Other five datasets [46] 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, 27, 46], we use the download links provided by Li et al. [46] with the following brief description of each dataset:

- DICM [42]: 69 images, mean=0.32, mixed exposure settings, variable resolutions, real scenes, varying scene including macros, landscapes, indoors, outdoors *etc*.
- LIME [29]: 10 images, mean=0.15, varying resolutions, real scenes, varying scene types.
- MEF [56]: 17 images, mean=0.15, resolution=512 $\times$ 340, relatively darker images, varying scene types.
- NPE [85]: 85 images, mean=0.31, varying resolution, both over and under exposed image regions, mostly outdoor scenes.
- VV [82]: 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 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.

Type	PSNR$_y$ ↑	SSIM$_y$ ↑	PSNR$_c$ ↑	SSIM$_c$ ↑	NIQE ↓	LPIPS ↓
w/o 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 [95] 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.

Configuration	Factorization		Fusion		Experiment
	Trad.	Deep	Trad.	Deep	
C_{11}		✓		✓	RSFNet LLE Fig. 3
C_{10}		✓	✓		Ablation (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

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:

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

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:

$$Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \tag{13}$$

whereas Matlab prefers the digital transformation as:

$$Y \leftarrow 0.2568 \cdot R + 0.5041 \cdot G + 0.0979 \cdot B \tag{14}$$

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.

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

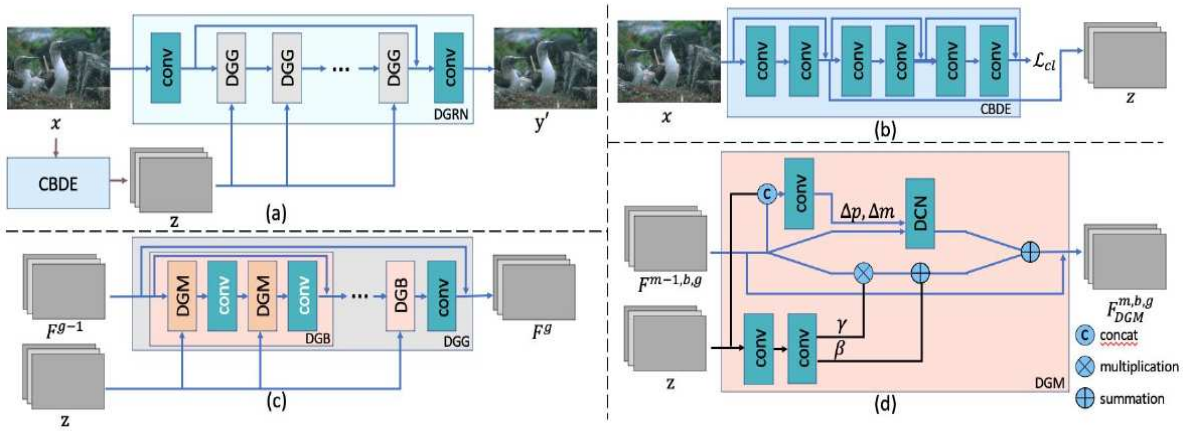

Figure 10. **AirNet:** (a) Block diagram from [45]. CBDE (b) refers to Contrastive-Based Degradation Encoder, DGG (c) means Degradation Guided Groups and DGM (d) is Degradation Guided Module. For complete details refers to [45]. For our usage, we alter first conv layer (first deep blue block on top-left (a)) and the first conv layer in CBDE (first deep blue block on top-right (b)).

TASK →	**DEHAZE** [44]		**DERAIN** [93]		**DEBLUR** [62]		**Mean**	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DL [19]	20.54	0.826	21.96	0.762	19.86	0.672	20.78	0.753
TransWeather [37]	21.32	0.885	29.43	0.905	25.12	0.757	25.29	0.849
TAPE [53]	22.16	0.861	29.67	0.904	24.47	0.763	25.43	0.843
AirNet [45] (multi-task)	21.04	0.884	32.98	0.951	24.35	0.781	26.12	0.872
AirNet [45] (uni-task)	23.18	0.900	34.90	0.966	26.42	0.801	28.17	0.889
AirNet [45] + 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 [45] 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 [90]	RFormer [13]	HEP [96]	NeRCo [92]	RSFNet (Ours)
DICM [42]	3.622	3.076	4.064	3.553	3.230
LIME [29]	3.752	3.910	3.981	3.422	3.800
MEF [56]	3.917	3.135	3.648	3.152	3.000
NPE [85]	3.535	3.63*	2.986	3.241	3.310
VV [82]	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 [90], RFormer [13]) and two unsupervised LLE solutions (last two columns: HEP [96], NeRCo [92]) *vs.* our zero-reference RSFNet method on five no-reference benchmarks namely: DICM [42], LIME [29], MEF [56], NPE [85] and VV [82]. Our method is able to generalize better to unseen data compared to others as observed from the overall lowest NIQE scores [60] 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).

networks. Specifically, we use AirNet [45] (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 sep-

arately (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 [97] 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 [97] 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.

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 [34] in Fig. 11.
- Visualization of our five extracted specular factors for the IIW dataset [6] in Fig. 12.
- Visualization of our five extracted specular factors for extension applications using deraining [93], dehazing [44] and deblurring datasets [62] in Fig. 13.
- Our qualitative results on low light image benchmarks in Fig. 14.
- Qualitative comparison of our results with other zero-reference LLE solutions in Fig. 15.
- Our results for the deraining application on the Rain100L dataset [93] in Fig. 16.
- Our results for the dehazing application on the RESIDE SOTS outdoor dataset [44] in Fig. 17.
- Our results for the deblurring application on the GoPro dataset [62] in Fig. 18.
- High resolution versions of the user controlled edited images (Fig. 7) in GIMP [80] in Fig. 19.
- 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 [60] 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.

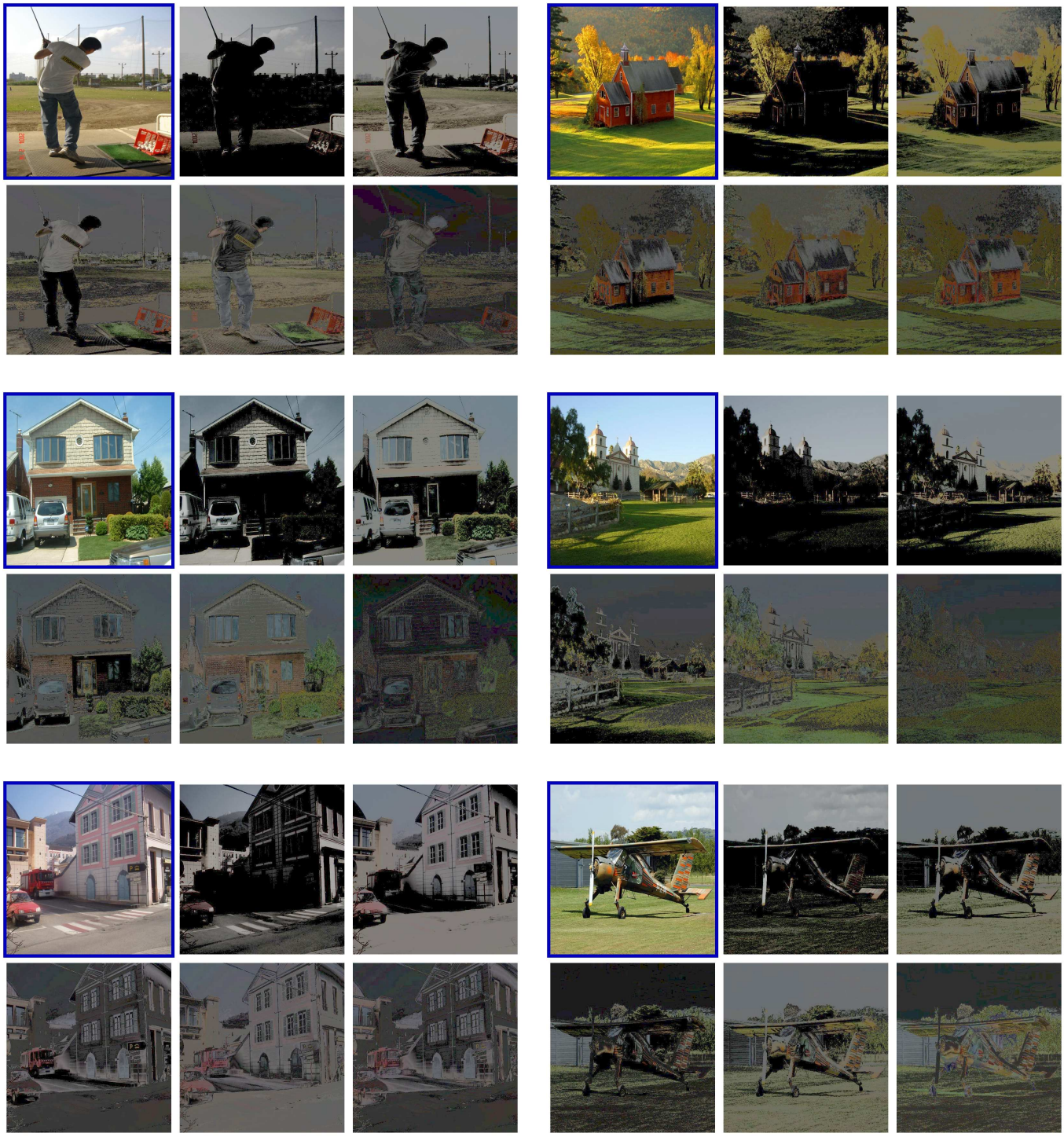

Figure 11. **Factor Visualizations (outdoors):** We show visualizations of our extracted five specular factors for various scenes. Input images (blue box) are taken from [34] dataset and factors are rescaled for visualization. Note how various regions are captured in the respective factors depending upon whether they are illuminated by directly, indirectly or in shadows.

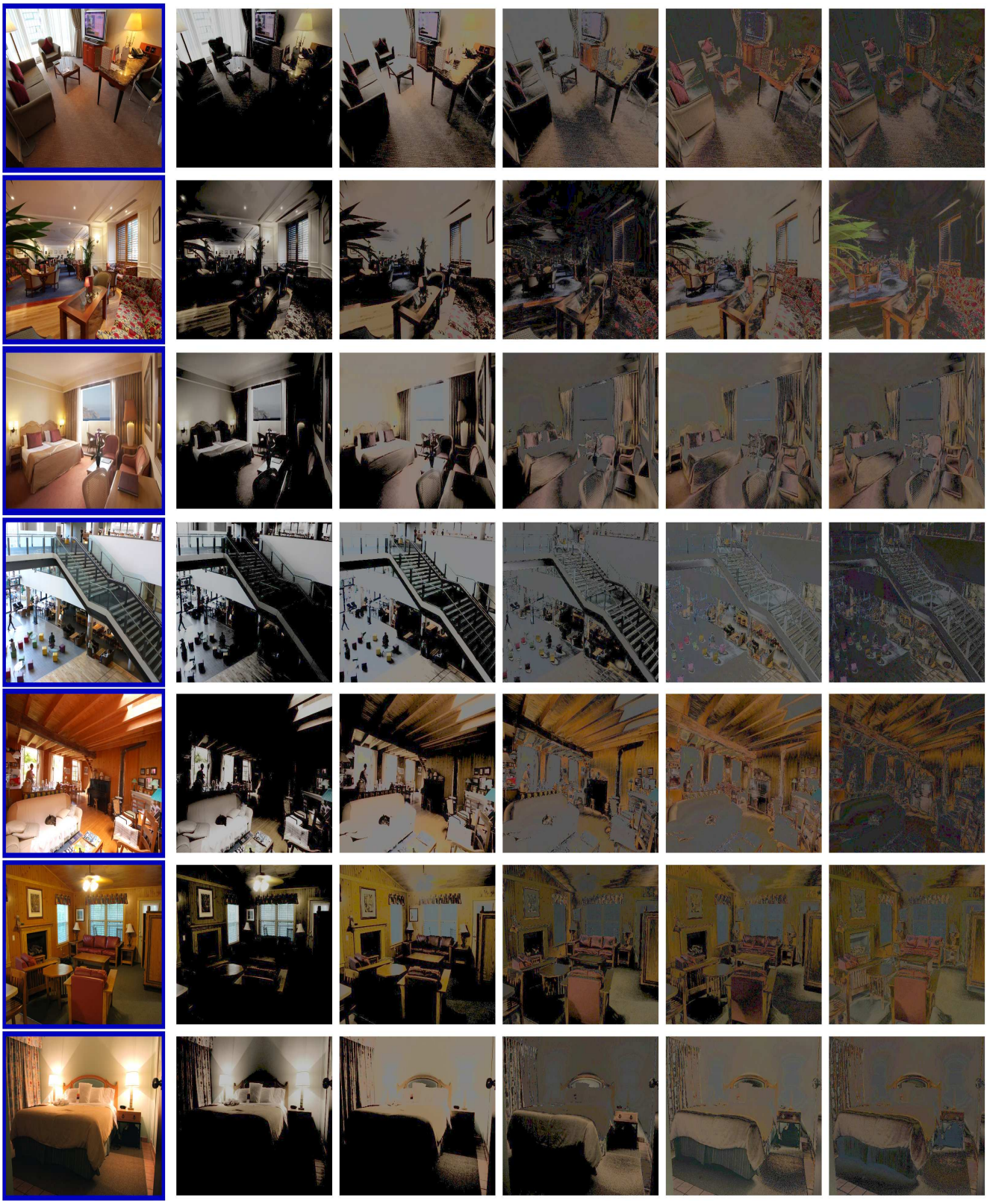

Figure 12. **Factor Visualizations (indoors):** We show visualizations of our extracted five specular factors for various scenes. Input images (blue box) are taken from [6] dataset and factors are rescaled for visualization. Note how various regions are captured in the respective factors depending upon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

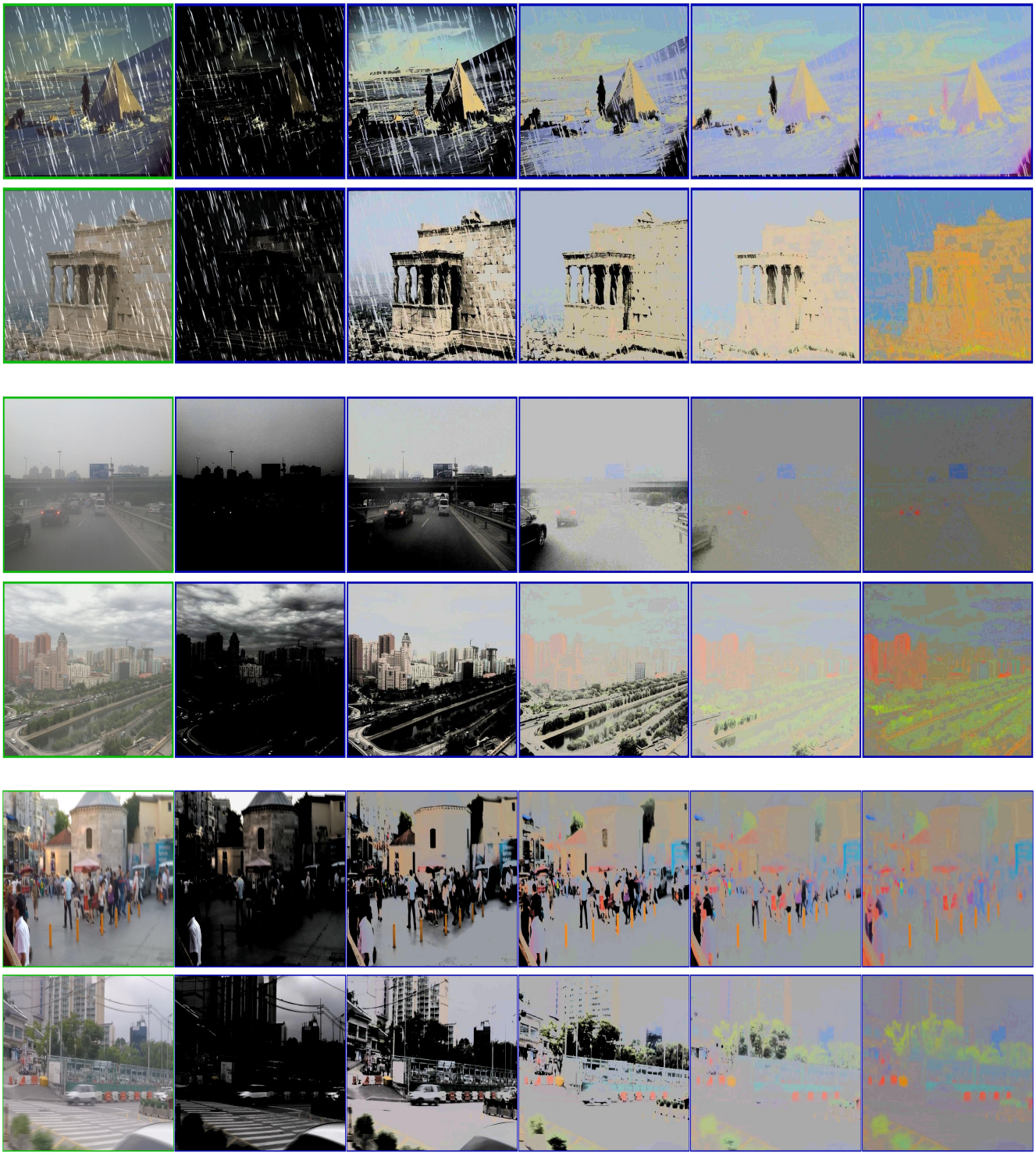

Figure 13. **Factor Visualizations (extensions):** We show visualizations of our extracted five specular factors for various scenes with different degradations. Input images (green box) are taken from 3 degraded images datasets [44, 62, 93] and factors (blue boxes) are rescaled for visualization. Note how specific degradation gets highlighted in different factors depending on the scene and the type of degradation.

Figure 14. **Our LLE Results**: Additional low light enhancement results from multiple Lol-x datasets [87, 95]. Each set contains input image (blue box), ground truth (red box) and our result (green box).

Figure 15. **Qualitative Comparisons**: Additional low light enhancement comparisons (Fig. 4 extension). Each set row in the grid contains results from: [SDD[31], ECNet[98], ZDCE[27]]; [ZD++[47], RUAS[72], SCI[57]]; [PNet[63], GDP[20], 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.

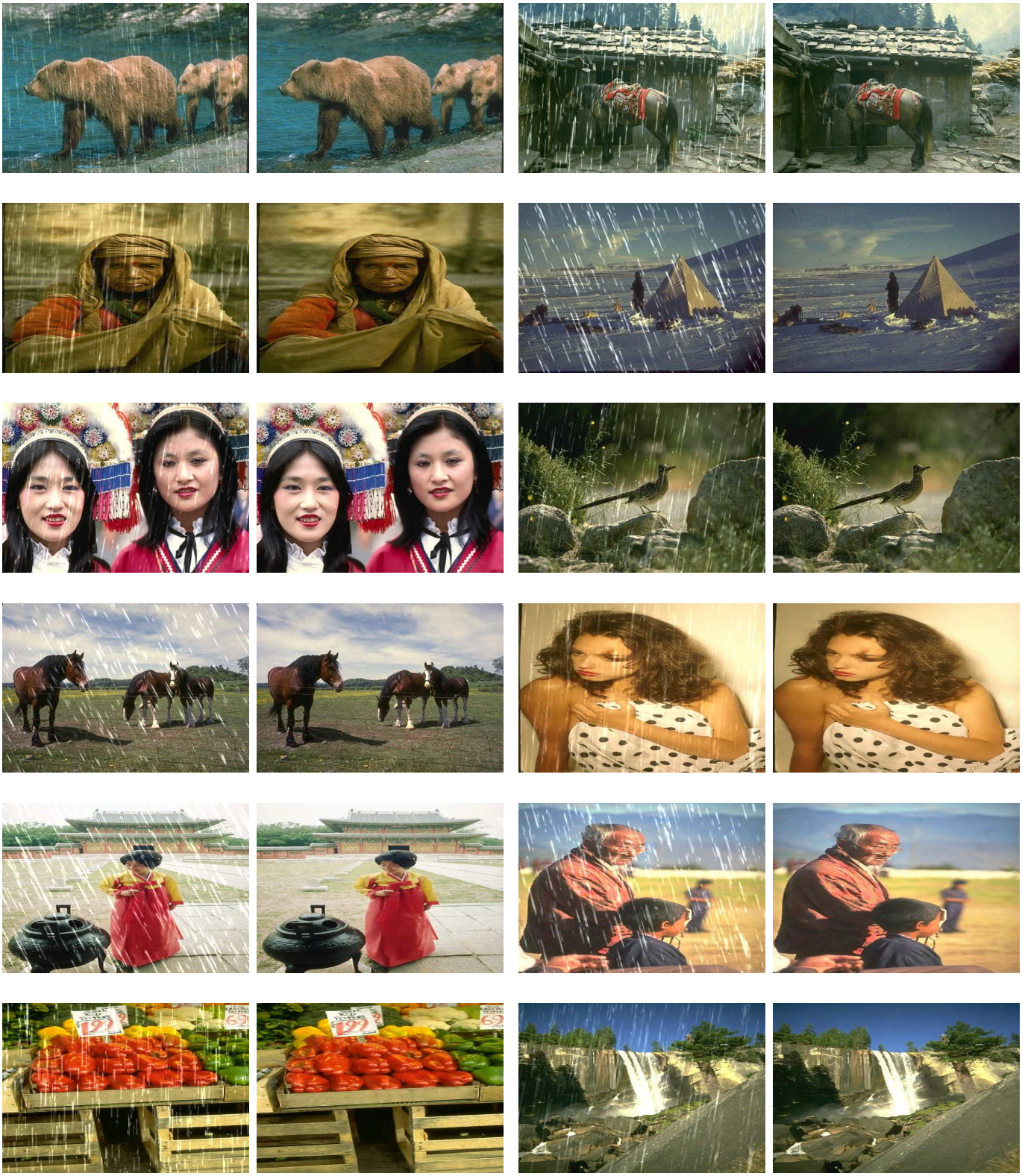

Figure 16. **Our Deraining Results**: Additional results (Fig. 6 extension) for the deraining application on the Rain100L dataset [93].

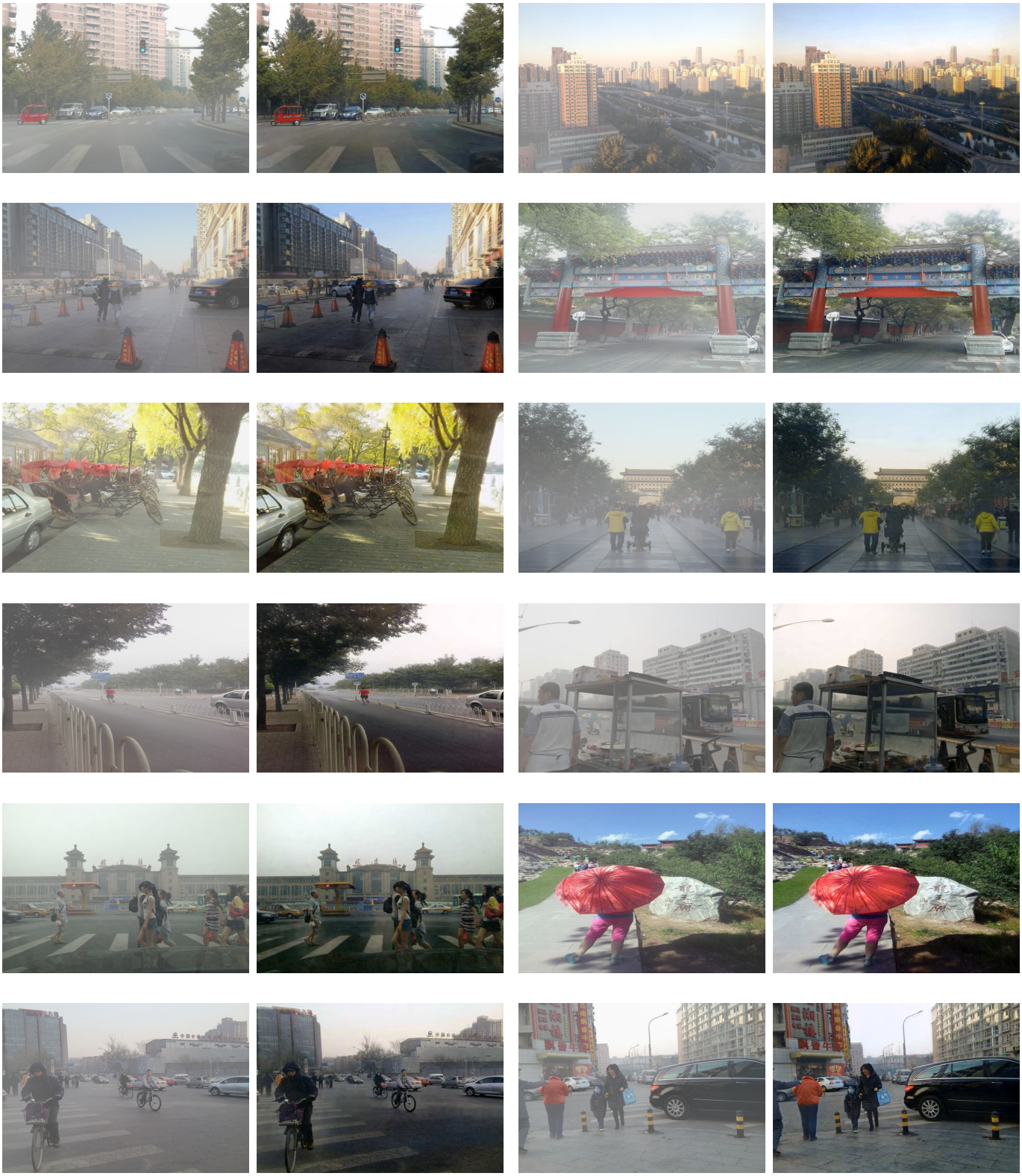

Figure 17. **Our Dehazing Results**: Additional results (Fig. 6 extension) for the dehazing application on the RESIDE dataset [44].

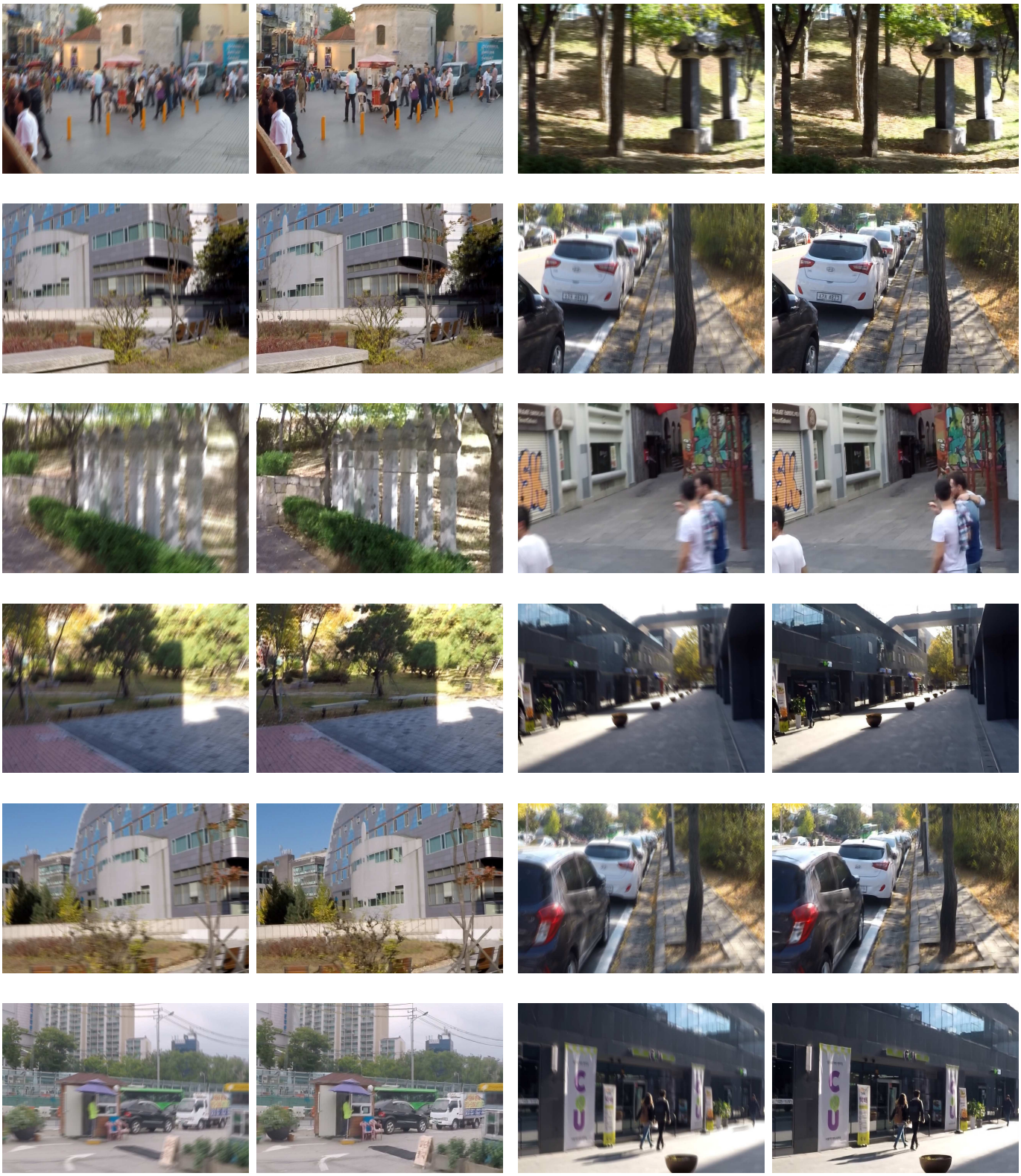

Figure 18. **Our Deblurring Results**: Additional results (Fig. 6 extension) for the deblurring application on the GoPro dataset [62].

Figure 19. **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 [80] 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 [29]	DUAL [100]	SDD [31]	ECNet [98]	ZDCE [27]	ZD++ [47]	RUAS [72]	SCI [57]	PNet [63]	GDP [20]	**RSFNet** (Ours)
#Params	-	-	-	16.5M	79.42K	10.56K	3.43K	**0.26**K	15.25K	552K	2.11K
Lolv1 [87] (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 [95] (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 [95] (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 [52] (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** [87], **Lolv2-real** [95], **Lolv2-syn** [95] and **VE-Lol** [52])											
$PSNR_y$ ↑	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 [87] and Lolv2-real [95] 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	*URetinex* [88]	*CUE* [105]	*SNR* [90]	*RFormer* [13]	EGAN [36]	HEP [96]	PairLIE* [24]	CLIP-LIT [49]	NeRCo* [92]	**RSFNet** (Ours)
Lolv1 [87] (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 [95] (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 [95] (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** [87], **Lolv2-real** [95], **Lolv2-syn** [95])										
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 [24, 36, 49, 92, 96] and four recent Supervised LLE solutions [13, 88, 90, 105] 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 [24] and NeRCo [92], 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 [92] and [96], 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.).