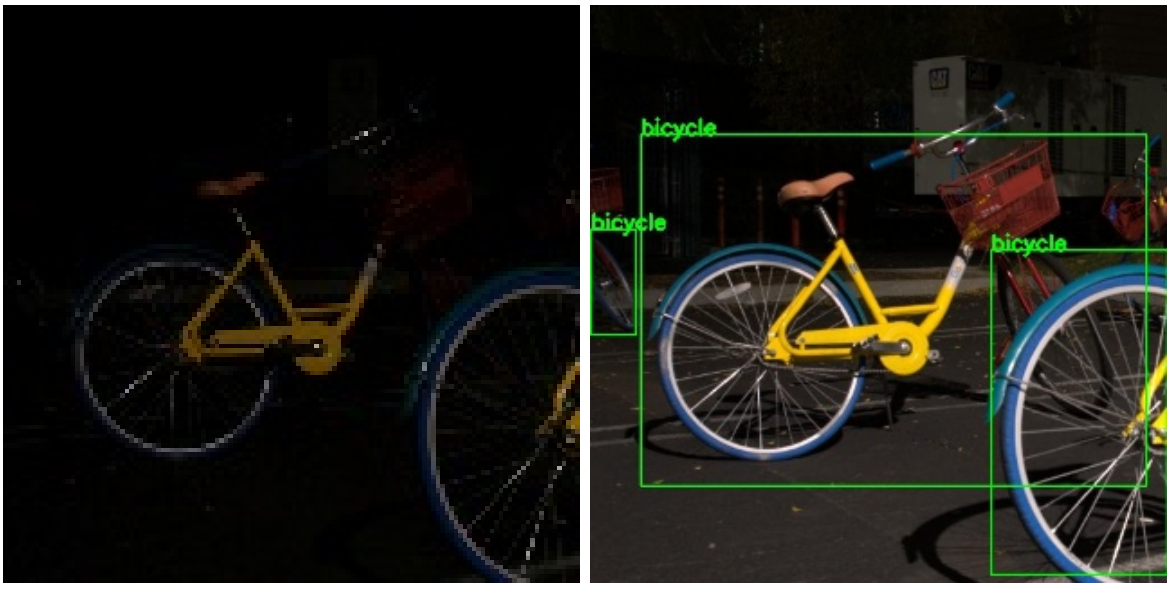
Restoring Extremely Dark Images in Real Time
(Supplementary Material)



Dark input: 20× amplified for visualization

Restored video + Object Detection

Figure 1: *(For animation view in Adobe Reader)* We can restore ultra-high-definition $4K$ resolution night-time images at 32 fps on a GPU. This enables real-time visualization and subsequent inference such as object detection.

Input | No Amp. + Retrain | [2] Amp. + Retrain | Our Amp. + **NO** Retrain | GT

Figure 2: Here we give more visual results pertaining to *Section 4.3 'Amplifier module'* of the main paper. The figure shows restorations obtained using the SID model as the backbone network and augmented with the existing MLP based amplifier [2] and our proposed amplifier module. The MLP based amplifier, even if jointly trained with the backbone network for pre-amplification, causes several artifacts. In comparison, our amplifier module helps achieve much superior restoration without any retraining. More results are shown in Fig. 3.

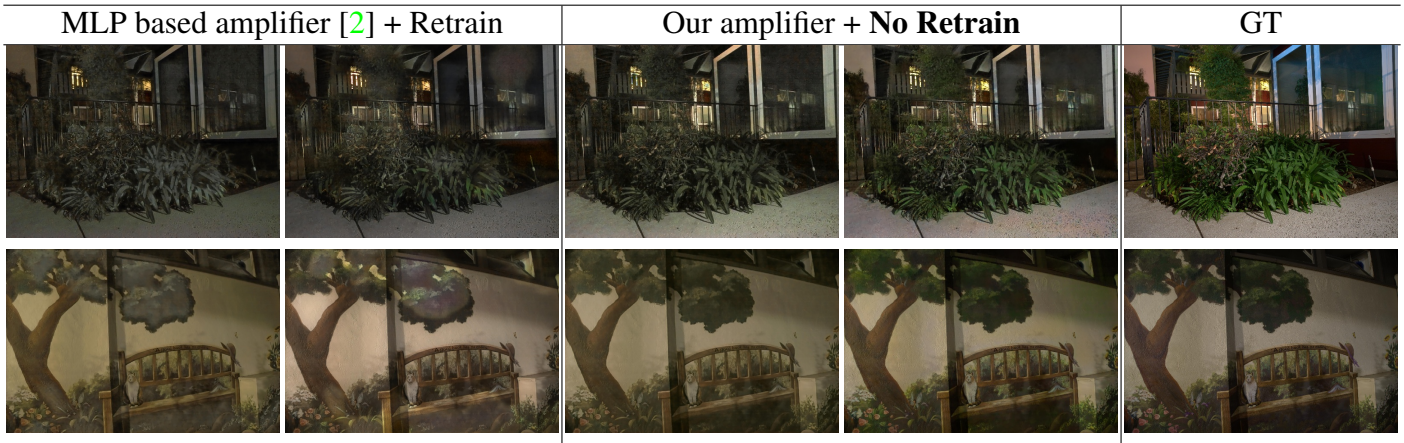


Figure 3: In *Section 4.3 'Amplifier module'* of the main paper, we compared the restoration quality using our amplifier and with the recently proposed MLP based amplifier [2]. Here we give more visual comparisons. We augmented the MLP based amplifier and our amplifier to the extreme low-light restoration methods LLPackNet [2] and our proposed model. In both cases, the MLP based amplifier had to be jointly re-trained with the restoration model while our amplifier required no fine-tuning. For both the restoration models, we observe that our amplifier aids better restoration with significantly less artifacts.

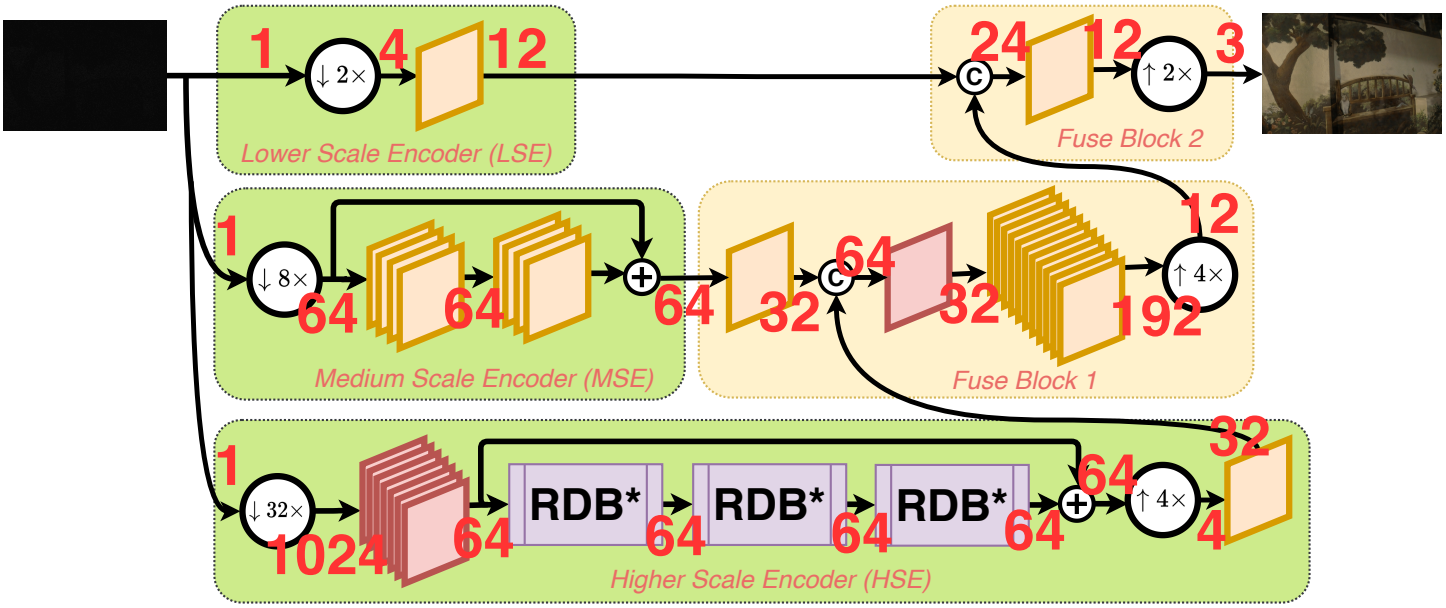


Figure 4: Our model architecture shown in Fig. 3 of the main paper. In this figure we additionally show the number of channels in feature maps processed at various stages of our network.

1 More details on Our network architecture

1.1 Calculating floating-point operations for RDB*

In Eq. (2) of the main paper, we estimated the floating-point operations due to the proposed RDB*. In this section, we give additional explanations on how we arrived at that equation. An RDB*, shown in Fig. 3 c) of the main paper, has η convolutional layers. Generally, $4 \leq \eta \leq 6$ and so we set $\eta = 5$. Each convolutional layer accepts depth-wise concatenated outputs of all the preceding layers and outputs a fixed number of channels C_{gr}, called the growth factor. Recent methods [3, 4] have set $C_{gr} = 32$ and we retain this. Thus the operations for this much portion of RDB* is shown below,

$$\frac{H}{r_{HSE}}\,\frac{W}{r_{HSE}}\,n_{HSE}\,k_{HSE}^2\,C_{gr}\sum_{j=1}^{\eta}\left[C_{HSE}^i+(j-1)C_{gr}\right]. \tag{1}$$

Finally, the output of all η convolutional layers and the input are once again depth-wise concatenated. Using pointwise convolution, the resulting channel dimension is reduced to the input's channel dimension, which in our case is $C_{HSE}^i = 64$. The operation count for this part is shown below,

$$\frac{H}{r_{HSE}}\,\frac{W}{r_{HSE}}\,n_{HSE}\,[C_{HSE}^i+\eta\cdot C_{gr}]\,C_{HSE}^i. \tag{2}$$

The large downsampling factor for HSE converts the input raw image with a lot of channels. To reduce the computational complexity for RDB*, we reduced the channel dimension using grouped convolution, which takes about $\frac{H}{r_{HSE}}\,\frac{W}{r_{HSE}}\,k_{HSE}^2\,r_{HSE}^2$ operations. If we now add the operations count given by Eq.(1), Eq.(2) and the grouped convolution, we arrive at Eq. (2) of the main paper, as shown below:

$$\begin{aligned}\frac{H}{r_{HSE}}\,\frac{W}{r_{HSE}}\Bigg(&n_{HSE}\,k_{HSE}^2\,C_{gr}\sum_{j=1}^{\eta}\left[C_{HSE}^i+(j-1)C_{gr}\right]\\&+n_{HSE}[C_{HSE}^i+\eta\cdot C_{gr}]C_{HSE}^i\;+\;k_{HSE}^2\,r_{HSE}^2\Bigg).\end{aligned} \tag{3}$$

Once the HSE has the output of n_{HSE} RDB*s, HSE needs to upsample and process it before concatenating it to the output of MSE. For this additional processing, HSE uses a simple convolutional layer whose operations count can be obtained using Eq. (1) of the main paper. Thus, equating the operations count of HSE and MSE, we find that $n_{HSE} = 4$. But this calculation does not account for channel concatenation operation. Channel concatenation is comparatively a much faster operation as no convolution is involved, but the RDB/RDB* use this operation very frequently, roughly about $n_{HSE}\,\frac{(\eta)(\eta+1)}{2}$ number of times. Thus, we take a more conservative estimation and fix $n_{HSE} = 3$ in our model architecture.

1.2 Amplifier module

In this section, we give more explanations on how we arrived at Eq. (6) of the main paper, which describes the binning pattern used in our amplifier. An extreme low-light image will have very small intensity values, and so we found that having finer binning for smaller intensity values is preferable. We thus used exponential binning such that the bins are equispaced in the logarithmic scale. If we consider the most widely used case of quantizing intensity into $2^8 = 256$ levels, the bin width in the logarithmic scale is given by,

$$\frac{log_2(2^8)-log_2(1)}{n}=\frac{8}{n}\implies log_2(b_k)=\frac{k\cdot 8}{n}\;\forall\, k\in[1,n]. \tag{4}$$

In the linear scale, the bin edges are thus $2^{\frac{k\cdot 8}{n}}$. But as we work with images normalized in the range $0-1$,

$$b_k = 2^{\frac{k\cdot 8}{n}}/2^8\;\forall\, k\in[1,n]\text{ (same as Eq.(6) of the main paper).} \tag{5}$$

Likewise, in estimating the appropriate amplification for the extreme low-light image, we wanted to give more importance to lower intensities, $x_{i,j}$. We thus chose exponentially decaying weights. Please note that the bins denoted in Eq. (5) vary exponentially. Thus we use that same equation for the weights $w_{i,j}$ but with index k replaced with $n-(k-1)$. This then results in Eq. (7) of the main paper:

$$w_{i,j} = 2^{\frac{(n-k+1)\cdot 8}{n}}/2^8 \text{ if } b_{k-1} < x_{i,j} \leq b_k. \tag{6}$$

Input | w/o spurious highlight masking (Eq. (4) of main paper) | Proposed Spurious highlights masking Eq. (5) of the main paper | GT

Figure 5: Ablation study on the proposed amplifier module. In *Section 3.2 'Amplifier module'* of the main paper we mentioned that captured extreme low-light images usually have small regions of saturated pixels. Sometimes called *highlights*, these very bright pixels (pointed by red arrows) tend to introduce several distortions in the restored images, especially the *halo artifact*, and recovering information in their vicinity is quite challenging. Such spurious light sources do not aid better visual perception of the scene but cause Eq. (4) of the main paper to wrongly estimate a much lower amplification.

	Default $m = 0.5$			**Best of** $m = \{0.2 \longrightarrow 0.8\}$		
	PSNR ↑ / SSIM ↑		**Ma ↑ / NIQE ↓**	**PSNR ↑ / SSIM ↑**		**Ma ↑ / NIQE ↓**
	GT	**GT-GC**	**No reference**	**GT**	**GT-GC**	**No reference**
LLPackNet + MLP amp. + Retrain	23.27/0.69	23.43/0.71	5.83 / 5.50	–	–	–
LLPackNet + Our amp. + No Retrain	**23.35/0.71**	**24.34/0.74**	**5.97 / 5.12**	27.03/0.73	27.06/0.73	5.97 / 5.14
SID + MLP amp. + Retrain	22.98/0.71	23.17/0.72	6.21 / 4.90	–	–	–
SID + Our amp. + No Retrain	**23.84/0.74**	**26.13/0.78**	**6.94 / 4.39**	28.33/0.78	28.35/0.78	6.93 / 4.40
Ours + MLP amp. + Retrain	23.30/0.71	23.80/0.72	6.11 / 4.62	–	–	–
Ours + Ours amp. + No Retrain	**23.85/0.75**	**26.08/0.79**	**6.93 / 4.40**	28.28/0.79	28.30/0.79	6.92 / 4.41
DID + Our amp. + No Retrain	23.29/0.71	25.64/0.74	6.84 / 4.49	28.05/0.76	28.06/0.76	6.83 / 4.51
SGN + Our amp. + No Retrain	23.90/0.73	26.12/0.77	6.94 / 4.40	28.35/0.78	28.36/0.78	6.93 / 4.41
DCE + Our amp. + No Retrain	22.72/0.70	25.13/0.72	6.08 / 4.65	26.12/0.72	26.12/0.72	6.04 / 4.69
LDC + Our amp. + No Retrain	23.86/0.75	26.10/0.79	6.92 / 4.40	28.34/0.79	28.35/0.79	6.91/4.41

Table 1: An extended version of Table 2 in the main paper. Here we compare the proposed amplifier against the recently proposed MLP based amplifier [2]. We observe that for each restoration model, our amplifier achieves better quantitative scores. For image enhancement, the choice of GT is quite subjective. For example, for an image captured late in the night, one can potentially select any image captured with randomly higher exposure as the GT. Thus, in this situation, the use of PSNR/SSIM is quite inappropriate as they compare only against a single GT and do not correlate well with human visual perception. To overcome this limitation, we take to two approaches. In the first approach, we generate multiple GT images with different brightness levels by gamma correcting the reference images in a small window of 0.7 to 1.3 with a step size of 0.03. The maximum PSNR/SSIM values are reported in the columns 'GT-GC'. We immediately note a $2-3dB$ improvement for our amplifier. This confirms that our amplifier does not introduce any artifacts in the restoration but only induces a different global brightness. However, a similar improvement is not seen for the MLP based amplifier because it actually causes several distortions in the restored image. In the second approach, we also vary the brightness level of the restored image by varying the amplifier hyperparameter m from $0.2-0.8$. We again observe another $2-3dB$ gain, because this time the restoration's brightness matches GT's restoration and consequently GT-GC does not achieve any significant improvement. This large fluctuation in PSNR values only due to a change in overall restoration brightness is not desirable for benchmarking image enhancement. Opposite to this behavior of PSNR/SSIM, the no-reference perceptual metric Ma/NIQE are much more stable and better capture the human visual perception in assessing the restoration quality. This is evident from the fact that Ma/NIQE remain almost the same for $m = 0.5$ column and for the column in which m is varied from $0.2-0.8$. The default setting with $m = 0.5$, however, is generally higher.

1.3 Choosing Pixel-shuffle for down/up sizing the feature maps

We choose Pixel Shuffle (PS) for down and upsampling over max-pooling [1] because for large downsampling operations such as 32, max-pooling causes excessive information loss. Strided convolution [1] is another popular alternative. But the kernel size for strided convolution is proportional to the downsampling factor, which directly impacts the network speed and model size. Since PS relies only on shifting image pixels and not on convolutional operation, it is a much swifter operation. Similar arguments are valid for choosing PS over transposed

convolution [1] which is by far the most popular upsampling technique.

References

[1] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv:1603.07285*, 2016. 4, 5

[2] Mohit Lamba, Atul Balaji, and Kaushik Mitra. Towards fast and light-weight restoration of dark images. In *BMVC*, 2020. 1, 2, 4

[3] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCVW*, 2018. 2

[4] Qingsen Yan, Dong Gong, Qinfeng Shi, Anton van den Hengel, Chunhua Shen, Ian Reid, and Yanning Zhang. Attention-guided network for ghost-free high dynamic range imaging. In *CVPR*, 2019. 2