Towards Fast and Light-Weight Restoration of Dark Images – Supplementary Material

Mohit Lamba*
ee18d009@smail.iitm.ac.in
Atul Balaji
ee16b002@smail.iitm.ac.in
Kaushik Mitra
kmitra@ee.iitm.ac.in

Computational Imaging Lab
Dept. of Electrical Engineering
Indian Institute of Technology Madras
Chennai, India

1 More qualitative results

We show more qualitative results comparing the performance of the proposed method with existing methods. Fig. 1 shows results for enhancing extremely dark images for the practical scenario when the ratio of GT to input image exposure is not available. Refer to Fig. 4 (B) and Table 1 in main paper for more information about this setting.

Fig. 2 shows qualitative results for the LOL dataset corresponding to Table 2 in the main paper.

2 Worked out example of Pack/UnPack operation

Pack/UnPack operators perform intermixing of pixels for better color correlation. This intermixing is shown in Fig. 2 of the main paper. To further facilitate how the Pack2×/UnPack2× do the shuffling in LR we display a worked-out example below. Consider an input tensor of 2×2 spatial resolution with 12 channels as shown below.

Channel Count	1^{st} Channel		2^{nd} Channel		3^{rd} Channel		···	12^{th} Channel	
Channel	1	2	5	6	9	10	···	45	46
Values	3	4	7	8	11	12	···	47	48

Then, applying the UnPack 2× operation we get a tensor of 4×4 spatial resolution with 3 channels as shown below.

```
Red Channel or the first channel
          [ 1, 13,  2, 14]
          [25, 37, 26, 38]
          [ 3, 15,  4, 16]
          [27, 39, 28, 40],

Green Channel or the second channel
```

* Corresponding author

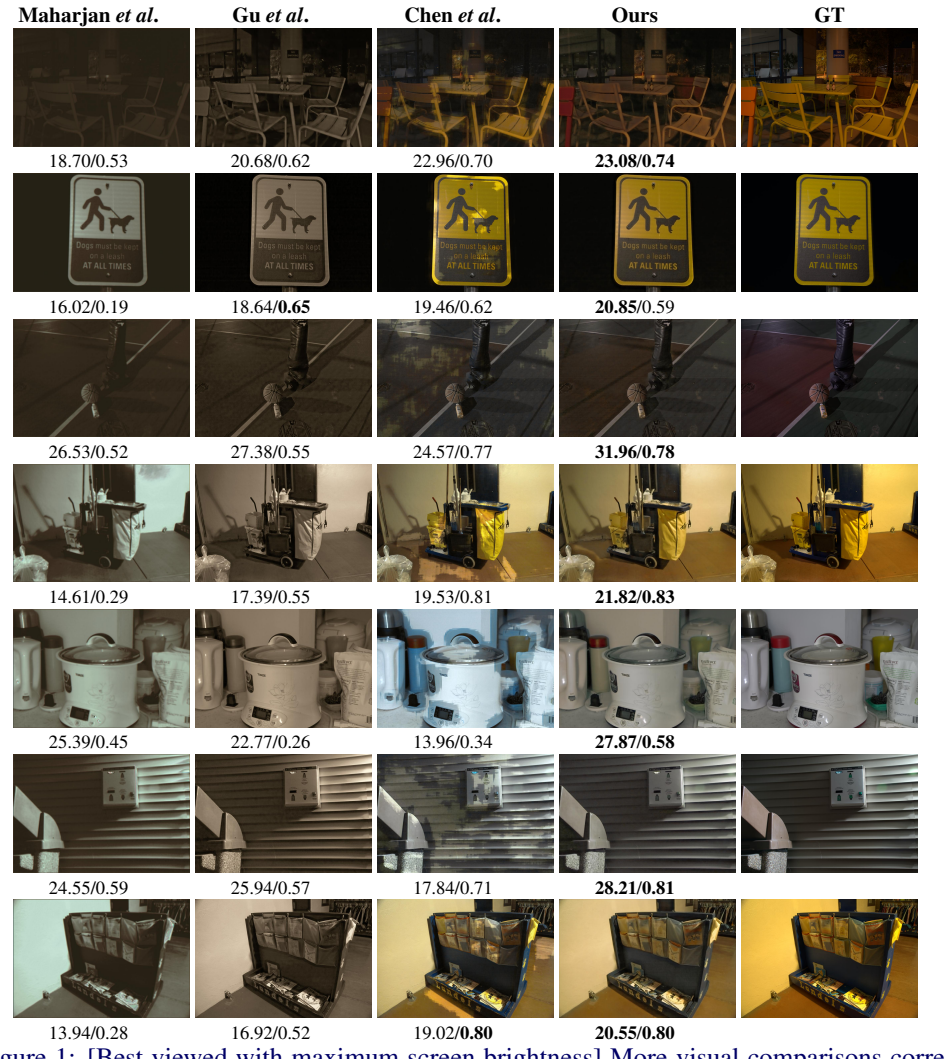

Figure 1: [Best viewed with maximum screen brightness] More visual comparisons corresponding to Table 1 of main paper for the practical case when GT exposure is not available.

```
            [ 5, 17,  6, 18]
            [29, 41, 30, 42]
            [ 7, 19,  8, 20]
            [31, 43, 32, 44],

Blue Channel or the third channel
            [ 9, 21, 10, 22]
            [33, 45, 34, 46]
            [11, 23, 12, 24]
            [35, 47, 36, 48]
```

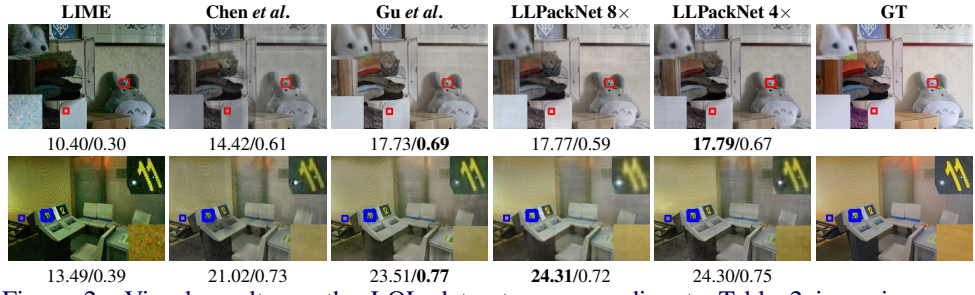

Figure 2: Visual results on the LOL dataset corresponding to Table 2 in main paper. LLPackNet-8× is pretty fast with good color restoration but exhibits slight blurriness. This is rectified by LLPackNet-4× which chooses a lower downsampling factor befitting the low-resolution.

3 Comparing Pack/UnPack with other popular down/up sampling operations

Downsampling: Max-pooling is the most popular technique for downsampling feature maps. This has been used in many deep learning methods, including Chen *et al*.'s network. But for a large downsampling it will cause huge loss of information. For example, when doing a 8× downsampling, max-pooling will choose only a single element from an 8×8 block.

Another popular downsampling technique is strided convolution, usually done with small kernels such as 3×3 or 5×5. But, for a large downsampling factor, say 8, a stride of 8 is required. However, with such small kernels it would lead to loss of information. To alleviate these issues, we used the novel Pack operation for downsampling feature maps without loss of information.

Upsampling: We have already shown the effectiveness of UnPack operation over the PixelShuffle operation in the main paper. Here, we compare with two other popular approaches – Transposed convolution as used by Chen *et al*. and interpolation suggested by Odena *et al*. [1]. The transposed convolution is very slow as compared to the UnPack operation because it has to iterate the convolution kernel over the entire feature map. Moreover it increases the parameter count of the network. On the other hand, the interpolation technique suggested by Odena *et al*. has no learnable parameters but is still a slower operation. This can be seen in Table 1.

H×W; Channels	Execution Time in Seconds			Number of Learnable Parameters		
	TransposeConv2D	UnPack	Interpolation	TransposeConv2D	UnPack	Interpolation
1024×1024; 32 -> 2048×2048; 8	0.18	0.05	0.13	1032	–	–
256×256; 128 -> 512×512; 32	0.04	0.01	0.04	16416	–	–
32×32; 512 -> 64×64; 128	0.0025	0.0006	0.0025	262272	–	–

Table 1: We compare the execution time and learnable parameters required by Transposed Convolution (TransposeConv2D), Interpolation [1] and the novel UnPack operation to perform upsampling by a factor of 2. We use feature maps of different spatial resolutions and channel dimensions. Akin to modern deep methods we use fewer channels for large kernels and more channels for smaller kernels. To report these numbers we use the PyTorch framework on Intel Xeon E5-1620V4 @ 3.50 GHz CPU. The proposed UnPack is 3–4× faster than the popular techniques such as transposed convolution used by Chen *et al*.

References

[1] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. URL http://distill.pub/2016/deconv-checkerboard.