Supplementary material for

DeepFuse: A Deep Unsupervised Approach for Exposure Fusion with Extreme Exposure Image Pairs

K. Ram Prabhakar, V. Sai Srikar, and R. Venkatesh Babu
Video Analytics Lab, Department of Computational and Data Sciences,
Indian Institute of Science, Bangalore, India

Contents

1 Analysis of Loss functions for DeepFuse-Baseline

As explained in section 4.1 of the main paper, in this experiment the CNN is trained in the presence of a ground truth. We have considered one of the results by Mertens [8] and GFF [2] for ground truth. The results of two methods are evaluated using MEF SSIM [6], the one with maximum MEF SSIM score is selected as ground truth. The choice of loss function to calculate error between ground truth and estimated output is very crucial for training a CNN in supervised fashion. The Mean Square Error or ℓ_2 loss function is generally chosen as default cost function for training CNN. ℓ_2 cost function is desired for its smooth optimization properties. While ℓ_2 loss function is better suited for classification tasks, they may not be a correct choice for image processing tasks. It is well known phenomena that MSE does not correlate well with human perception of image quality [12]. In order to obtain visually pleasing result, the loss function should be well correlated with HVS, like Structural Similarity Index (SSIM) [12]. We have trained proposed CNN model using various loss functions. We have compared the results among ℓ_1, ℓ_2 and SSIM. In this section we shall denote the ground truth image and the network output as G_{fused} and O_{fused} respectively.

ℓ_1 **loss**: ℓ_1 loss is defined as the absolute difference between the two quantities that are compared. In our experiment the two quantities compared are the network output and the ground truth. ℓ_1 loss for a patch P is expressed as,

$$\mathcal{L}^{\ell_1} = \frac{1}{N} \sum_{p \in P} |G_{fused}(p) - O_{fused}(p)| \tag{1}$$

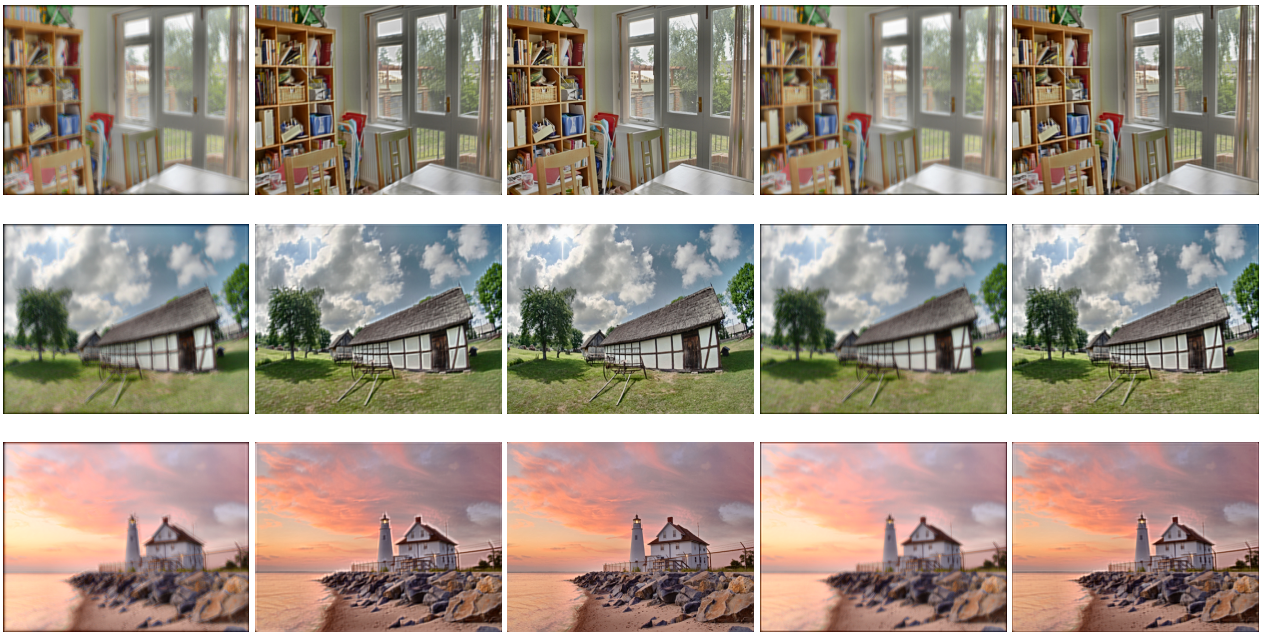
ℓ_2 **loss**: ℓ_2 loss is the squared error loss between the two quantities.

$$\mathcal{L}^{\ell_2} = \frac{1}{N} \sum_{p \in P} \|G_{fused}(p) - O_{fused}(p)\|^2 \tag{2}$$

SSIM loss: Structural Similarity Index Metic (SSIM) [12] is a widely used perceptual image quality metric. It factors the local structure and contrast of the images. The SSIM score between input patch x and reference patch y is computed by,

$$SSIM(x, y) = \frac{1}{N} \sum_{p \in P} \left(\frac{2\mu_{x(p)}\mu_{y(p)} + C_1}{\mu_{x(p)}^2 + \mu_{y(p)}^2 + C_1} \cdot \frac{2\sigma_{x(p)y(p)} + C_2}{\sigma_{x(p)}^2 + \sigma_{y(p)}^2 + C_2} \right) \tag{3}$$

where, $\mu_{x(p)}$ and $\mu_{y(p)}$ are the mean value of patch centered around $x(p)$ and $y(p)$, $\sigma_{x(p)}$ is the standard deviation of patch centered around $x(p)$, $\sigma_{x(p)y(p)}$ is the covariance of patches centered at $x(p)$ and $y(p)$, C_1 and C_2 are small positive values added



(a) (b) (c) (d) (e)

Figure 1: Results obtained by CNN trained with different loss functions: (a) ℓ_1, (b) ℓ_2, (c) SSIM, (d) SSIM*ℓ_1 and (e) SSIM*ℓ_2.

on numerator and denominator to avoid numerical instability. Since SSIM is directly proportional to the quality of the image, we compute the SSIM loss as,

$$\mathcal{L}^{SSIM} = 1 - SSIM(G_{fused}, O_{fused}) \tag{4}$$

ℓ_1 **with SSIM**: We try using combination of ℓ_1 and SSIM loss functions. We define the loss as,

$$Error = (\mathcal{L}^{\ell_1})^{\alpha} \cdot (\mathcal{L}^{SSIM})^{\beta}$$

ℓ_2 **with SSIM**: Similar to the previous loss function, we combine ℓ_2 with SSIM as,

$$Error = (\mathcal{L}^{\ell_2})^{\alpha} \cdot (\mathcal{L}^{SSIM})^{\beta}$$

Where, in this combination $SSIM$ is given more priority than L_1 and L_2 by assigning α = 0.25 and β = 0.75. Since SSIM accounts for human perception, we have assigned more weight to SSIM loss. The results after training a CNN with different loss function are shown in Figure 1. The result by ℓ_2 and ℓ_1 has blur effect and halo effect along the edges. Unlike ℓ_1 and ℓ_2, results by CNN trained with SSIM loss function are sharp and without any artifacts.

2 Benchmark Dataset

For the experiments, we captured 50 multi-exposure sequences with varying characteristics. The images were captured with Canon EOS 600D camera mounted in tripod. Each exposure stack has 2 images with $\pm$2 EV difference. The multi-exposure images were captured in Auto-Exposure Bracketing (AEB) mode. The images are captured in RAW format, later converted and resized to 1200 $\times$ 800 TIFF format images. The captured images include many varieties of scenes such as indoor, outdoor, dim background, natural/artificial lighting and many more. A subset of these images are shown in Figure 2 and 3.

(a) *100, f/4, 1/250* (b) *100, f/2.4, 1/115* (c) *100, f/4, 1/250*

(d) *100, f/4, 1/15* (e) *100, f/2.4, 1/50* (f) *100, f/4, 1/15*

Figure 2: Subset of indoor sequences in dataset. Caption below each image denotes the camera settings: ISO, F-stop and exposure time (in seconds), used to capture that image.

(a) *800, f/4, 0.8* (b) *800, f/3.5, 1.3* (c) *1200, f/5.6, 1/6*

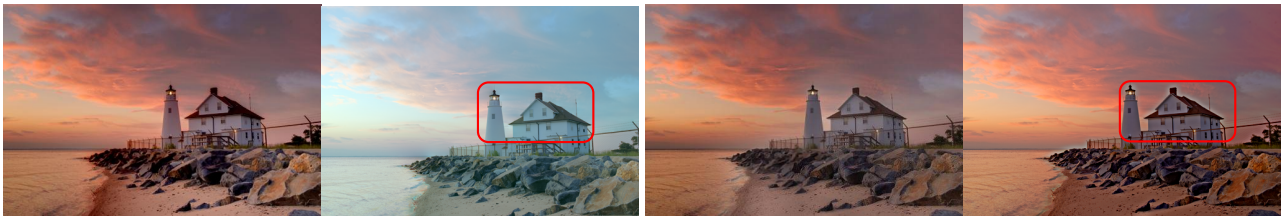
(d) *800, f/4, 6* (e) *800, f/3.5, 10* (f) *1200, f/5.6, 2.5*

Figure 3: Subset of outdoor sequences in dataset. Caption below each image denotes the camera settings: ISO, F-stop and exposure time (in seconds), used to capture that image.

3 Additional Results

3.1 Multi-Exposure Fusion

(a) Underexposed image (b) Overexposed image (c) Li *et al.* [1] (d) Li *et al.* [2]



(e) Mertens *et al.* [7] (f) Raman *et al.* [10] (g) Shen *et al.* [11] (h) Ma *et al.* [5]

(i) Guo *et al.* [3] (j) DeepFuse-Baseline (k) DeepFuse-Unsupervised

Figure 4: Results for Lighthouse image sequence.

(a) Underexposed image (b) Overexposed image (c) Li *et al.* [1] (d) Li *et al.* [2]

(e) Mertens *et al.* [7] (f) Raman *et al.* [10] (g) Shen *et al.* [11] (h) Ma *et al.* [5]

(i) Guo *et al.* [3] (j) DeepFuse-Baseline (k) DeepFuse-Unsupervised

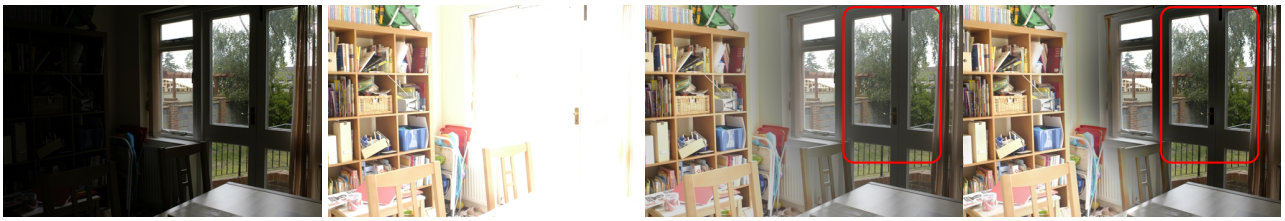
Figure 5: Results for Agia Galini image sequence.

(a) Underexposed image (b) Overexposed image (c) Li *et al.* [1] (d) Li *et al.* [2]

(e) Mertens *et al.* [7] (f) Raman *et al.* [10] (g) Shen *et al.* [11] (h) Ma *et al.* [5]

(i) Guo *et al.* [3] (j) DeepFuse-Baseline (k) DeepFuse-Unsupervised

Figure 6: Results for Balloons image sequence.



(a) Underexposed image (b) Overexposed image (c) Li *et al.* [1] (d) Li *et al.* [2]

(e) Mertens *et al.* [7] (f) Raman *et al.* [10] (g) Shen *et al.* [11] (h) Ma *et al.* [5]

(i) Guo *et al.* [3] (j) DeepFuse-Baseline (k) DeepFuse-Unsupervised

Figure 7: Results for House image sequence.

3.2 Multi-Focus Fusion (MFF)

To test the generalizability of CNN, we have used the already trained DeepFuse CNN to fuse multi-focus images without any fine-tuning with MFF data. From figure [9], the DeepFuse results on publicly available multi-focus dataset show that the filters of CNN have learnt to identify proper regions in each input image and successfully fuse them together. It can also be seen that the learnt CNN filters are generic and could be applied for general image fusion.

(a) Near focused image (b) Far focused image (c) All-in-focus DeepFuse result

Figure 8: Application of DeepFuse CNN to multi-focus fusion. The first two column images are input varying focus images. The result by DeepFuse is shown in third column. Images courtesy of Nejati *et al.* [9].

(a) Near focused image (b) Far focused image (c) All-in-focus DeepFuse result

Figure 9: Application of DeepFuse CNN to multi-focus fusion. The first two column images are input varying focus images. The result by DeepFuse is shown in third column. Images courtesy of Nejati *et al.* [9] and Liu *et al.* [4].

References

[1] S. Li and X. Kang. Fast multi-exposure image fusion with median filter and recursive filter. *IEEE Transactions on Consumer Electronics*, 58(2):626–632, May 2012.

[2] S. Li, X. Kang, and J. Hu. Image fusion with guided filtering. *IEEE Transactions on Image Processing*, 22(7):2864–2875, July 2013.

[3] Z. Li, Z. Wei, C. Wen, and J. Zheng. Detail-enhanced multi-scale exposure fusion. *IEEE Transactions on Image Processing*, 26(3):1243–1252, 2017.

[4] Y. Liu, S. Liu, and Z. Wang. Multi-focus image fusion with dense sift. *Information Fusion*, 23:139–155, 2015.

[5] K. Ma and Z. Wang. Multi-exposure image fusion: A patch-wise approach. In *IEEE International Conference on Image Processing*, 2015.

[6] K. Ma, K. Zeng, and Z. Wang. Perceptual quality assessment for multi-exposure image fusion. *IEEE Transactions on Image Processing*, 24(11):3345–3356, 2015.

[7] T. Mertens, J. Kautz, and F. Van Reeth. Exposure fusion. In *15th Pacific Conference on Computer Graphics and Applications*, 2007.

[8] T. Mertens, J. Kautz, and F. Van Reeth. Exposure fusion: A simple and practical alternative to high dynamic range photography. In *Computer Graphics Forum*, volume 28, pages 161–171. Wiley Online Library, 2009.

[9] M. Nejati, S. Samavi, and S. Shirani. Multi-focus image fusion using dictionary-based sparse representation. *Information Fusion*, 25:72–84, 2015.

[10] S. Raman and S. Chaudhuri. Reconstruction of high contrast images for dynamic scenes. *The Visual Computer*, 27:1099–1114, 2011. 10.1007/s00371-011-0653-0.

[11] R. Shen, I. Cheng, J. Shi, and A. Basu. Generalized random walks for fusion of multi-exposure images. *IEEE Transactions on Image Processing*, 20(12):3634–3646, 2011.

[12] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.