Thermal Image Processing via Physics-Inspired Deep Networks (Supplementary)

Vishwanath Saragadam, Akshat Dave, Ashok Veeraraghavan, and Richard G. Baraniuk
Rice University, Houston TX
{vishwanath.saragadam, akshat.dave, vashok, richb}@rice.edu

1. Introduction

We introduced a novel thermal imaging pipeline called DeepIR that combined the physics of an uncooled microbolometer camera and regularization via deep networks. We motivated our problem and demonstrated the efficacy of DeepIR with several results on simulated and real data. The supplementary document provides details about the learning procedure including the specific neural network architecture, and a more detailed study of some of the real experiments. We also hosted our code online[1] for further research.

2. Learning Details

All the results in the paper were regularized with a deep image prior based regularize. Our goal was to demonstrate the advantages of combining physics and deep networks, and hence our network architecture was an unmodified version of the architecture utilized in the original paper [1]. Specifically, we used a convolutional network with skip connections shown in Fig. 1. We note that alternate networks are possibly and potentially capable of giving better results but was not the focus of our paper.

Optimization details. As mentioned in the paper, we jointly optimized the parameters of the neural network, 6 parameters for each of the N affine matrices, $H \times W$ dimensional gain and offset terms. The input to the neural network was a $H \times W \times 8$ shaped noise that was *not* optimized along with other parameters. We found that random initialization for affine matrices sufficed – however to accelerate convergence we first registered the images to the first image using a pyramidal registration algorithm [2].

Details about super resolution. The image formation model relating the low resolution image $\mathbf{x}_k$ and high resolution image $\mathbf{x}_{\text{HR}}$ is,

$$\mathbf{x}_k = \mathbf{g} \odot (DM_k\mathbf{x}_{\text{HR}}) + \mathbf{o}, \tag{1}$$

where D is the downsampling operator, and M_k is the transformation matrix. To prevent aliasing artifacts endemic to downsampling, we chose D as the following operation,

$$X_{\text{LR}}(u,v) = \frac{1}{Q^2}\sum_{p=1}^{Q}\sum_{q=1}^{Q} X_{\text{HR}}(u+p, v+q) \tag{2}$$

for downsampling by a factor of Q.

Learning parameters. We set the learning rate to 10^{-3} and trained for a total of $2,000$ iterations. For non-uniform correction, there was no penalty for optimizing beyond $2,000$ iterations. However increasing the number of iterations proved to be detrimental for super resolution by producing checker-like artifacts in the final reconstruction This is expected, as deep image prior tends to overfit to noise if run for too many iterations.

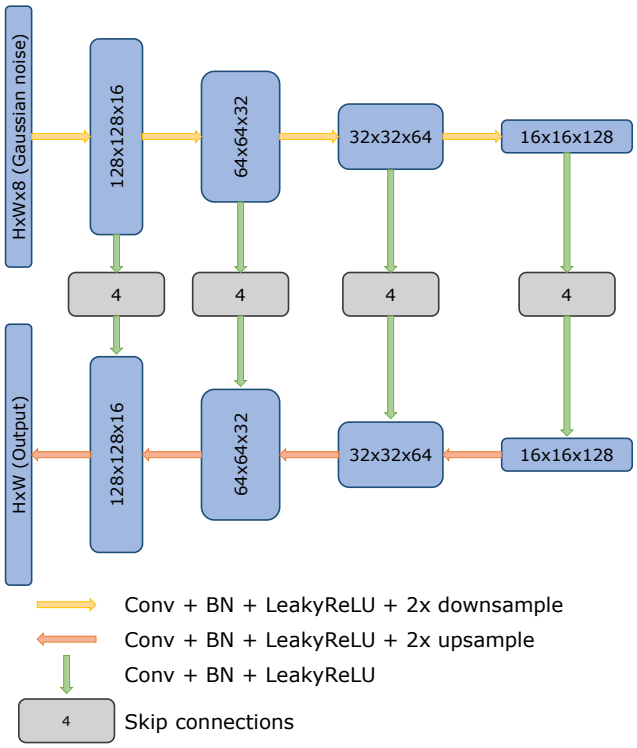

Figure 1: **Network architecture.** We used the default network architecture proposed in [1] for super resolution.

[1] https://github.com/vishwa91/DeepIR

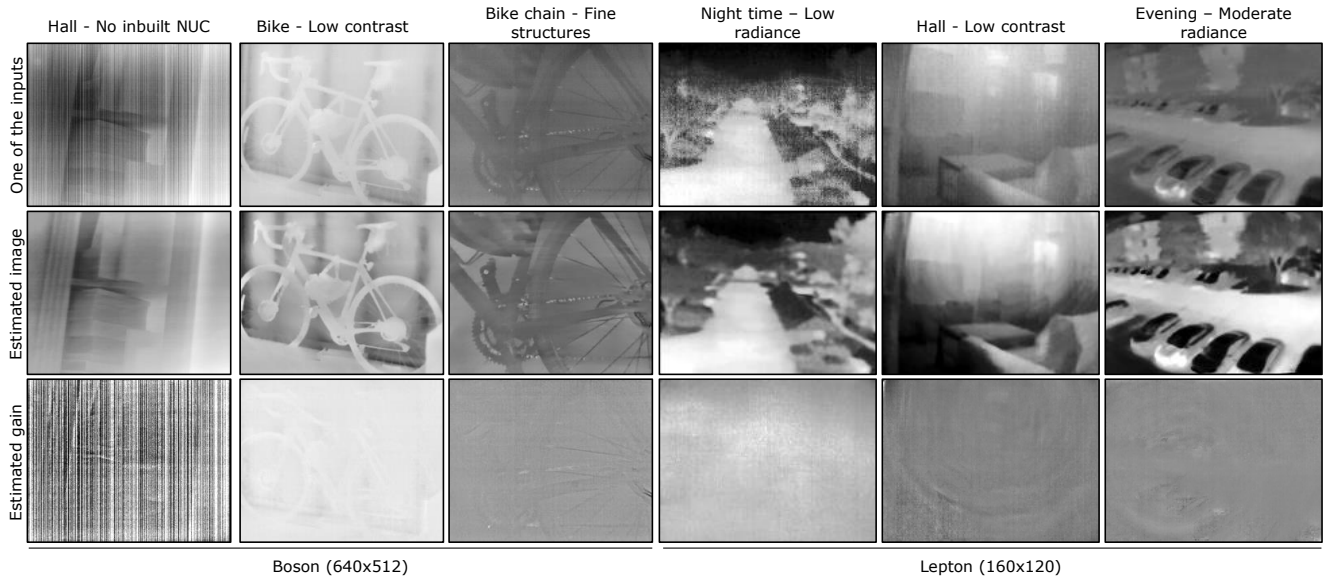

Figure 2: **NUC on diverse scenes.** Our approach is capable of non-uniformity correction for a wide variety of noise levels and scene complexities.

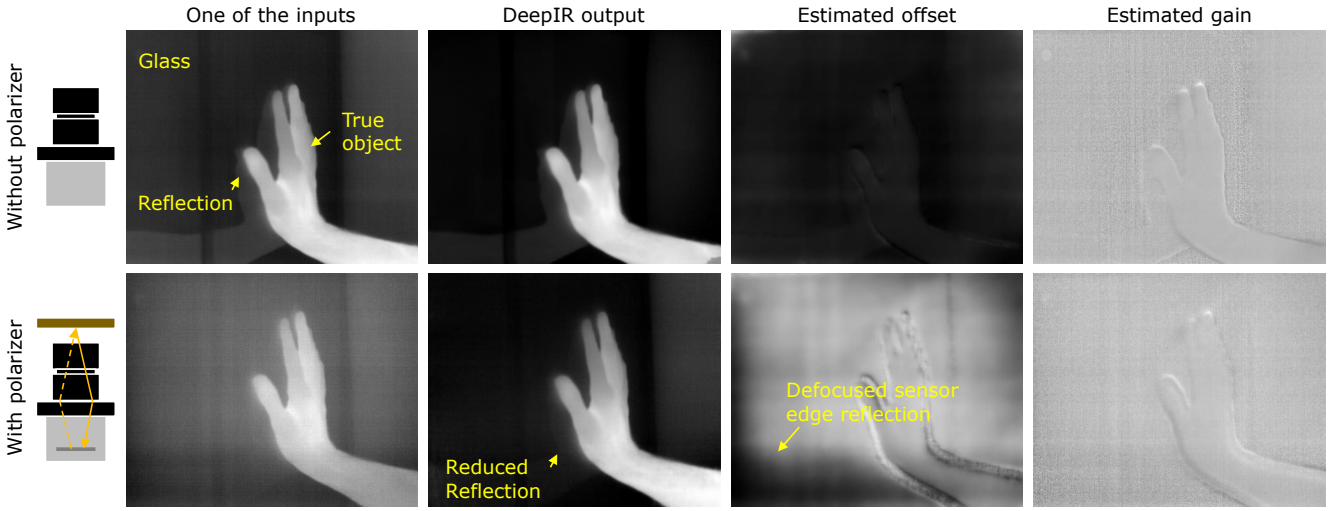

Figure 3: **Suppressing narcissus effect.** Since we model both gain and offset terms, DeepIR is capable of removing narcissus effects due to external optics like polarizers.

Our loss function consisted of MSE loss between predicted image $\mathbf{g} \odot M_k(\mathbf{x}_o + \mathbf{o})$ and the ground truth $\mathbf{x}_k$, a 2D total variation (TV) prior on the latent image, and a TV loss on the offset term. The motivation behind the TV loss for the offset is due to it arising from reflections off of optics which tend to be spatially smooth. We found this to be an effective strategy in separating the gain and offset terms. We set the weight of the TV loss on the latent image to be 10^{-5}, and the weight of the TV loss on the offset term to be 10. We used a batch size equal to the number of input images. The model was trained a system with Nvidia RTX 2080 GPU with 8GB memory along with 48GB RAM. The optimization was implemented with the `pytorch` framework [3]. The code ran for 10 minutes on our computer for five images of size 640×512 for a total of $2,000$ iterations. We will release our optimization code to the public for further research in this direction.

3. Real Results

We demonstrate some more results and provide sensitivity to parameters.

Hardware details. We used the FLIR Boson camera with

640×512 spatial resolution capturing images at 60 frames per second (fps), and the FLIR Lepton camera with 160×120 spatial resolution capturing images at 9 fps. We used the `flirpy` [4] package to control the cameras which allowed us to disable periodic NUC and capture images at full frame rate of the individual cameras. The Boson camera was equipped with inbuilt flat field correction (FFC), supplementary correction for lens reflections, and temporal noise reduction. We showed results with and without FFC in the main paper. In all cases, we disabled temporal noise reduction, as we found that enabling it produced ghosting artifacts.

Non-uniformity correction. We showed NUC results on some scenes with the Boson camera in the main paper. We next demonstrate some more experiments to underline the advantages of DeepIR. Figure 2 shows the non-uniformity correction with the various scenes at varying levels of scene complexity. All experiments included recovery with five images. We found the offset to be nearly zero and hence did not visualize it. DeepIR performs promisingly in low contrast conditions, absence of inbuilt NUC, low and low radiance levels.

Suppressing narcissus effect. Figure 3 shows the images with and without polarizer. Since we model both gain and offset, we were able to suppress the narcissus effect arising out of back reflections from the polarizer. Notice the defocused edge that is visible in the estimated offset in the image captured with a polarizer. The edge artifacts looking like the hard were due to minor motion between frames, and can be corrected with a more accurate model of transformation such as optical flow.

References

[1] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *IEEE Comp. Vision and Pattern Recognition (CVPR)*, 2018. 1

[2] Philippe Thevenaz, Urs E Ruttimann, and Michael Unser. A pyramid approach to subpixel registration based on intensity. *IEEE Trans. Image Processing*, 7(1):27–41, 1998. 1

[3] Adam Paszke et al. Pytorch: An imperative style, high-performance deep learning library. 2019. 2

[4] Flirpy. https://flirpy.readthedocs.io/en/latest/, 2021. 3