

Supplementary file for

Few-Shot Domain Adaptation for Low Light RAW Image Enhancement

K Ram Prabhakar
 ramprabhakar@iisc.ac.in
 Vishal Vinod*
 vishal114186@gmail.com
 Nihar Ranjan Sahoo*
 niharsahooigiti@gmail.com
 R. Venkatesh Babu
 venky@iisc.ac.in

Video Analytics Lab,
 Department of Computational and Data
 Sciences,
 Indian Institute of Science,
 Bangalore, India

Abstract

This supplementary file contains additional results and more detail on ablation experiments for our submission. We also detail the model architecture and training setup.

1 Network architecture details

In Table 1 of this supplementary file, we provide the details of the camera-specific encoder networks used to extract convolutional features for the source and target domains. In Table 2, we describe the shared \mathbb{N} network architecture along with the sub-pixel layer. The sub-pixel layer is used to perform Bayer conversion of the 12-channel output from the shared \mathbb{N} network from half the spatial resolution to full spatial resolution i.e from $height/2 \times width/2 \times 12$ to $height \times width \times 3$. We find that using just three convolutional layers with filter sizes of 16, 32, 64 and kernel size of 3×3 is sufficient to learn domain-specific denoising features.

Table 1: Encoder architecture.

Layer	Kernel	Filters	Stride	Activation
Conv 2D	3	16	1	LeakyReLU
Conv 2D	3	32	1	LeakyReLU
Conv 2D	3	64	1	LeakyReLU

2 Ablation Study

We discuss relevant ablations for our proposed approach trained with Sony as source and 4-shot Nikon as target (Refer Table 3). We provide more details and build on the select

*Equal contribution

Table 2: Shared N network architecture.

Layer	Kernel	Filters	Stride	Activation
Conv 2D	3	64	1	LeakyReLU
Conv 2D	3	64	1	LeakyReLU
MaxPool	2	-	2	
Conv 2D	3	128	1	LeakyReLU
Conv 2D	3	128	1	LeakyReLU
MaxPool	2	-	2	
Conv 2D	3	256	1	LeakyReLU
Conv 2D	3	256	1	LeakyReLU
MaxPool	2	-	2	
Conv 2D	3	512	1	LeakyReLU
Conv 2D	3	512	1	LeakyReLU
MaxPool	2	-	2	
Conv 2D	3	1024	1	LeakyReLU
Conv 2D	3	1024	1	LeakyReLU
MaxPool	2	-	2	
ConvTranspose 2D	2	512	2	-
Conv 2D	3	512	1	LeakyReLU
Conv 2D	3	512	1	LeakyReLU
ConvTranspose 2D	2	256	2	-
Conv 2D	3	256	1	LeakyReLU
Conv 2D	3	256	1	LeakyReLU
ConvTranspose 2D	2	128	2	-
Conv 2D	3	128	1	LeakyReLU
Conv 2D	3	128	1	LeakyReLU
ConvTranspose 2D	2	64	2	-
Conv 2D	3	64	1	LeakyReLU
Conv 2D	3	64	1	LeakyReLU
Conv 2D	3	32	1	LeakyReLU
Conv 2D	3	16	1	LeakyReLU
Conv 2D	1	12	1	LeakyReLU
Pixel Shuffle		Upscale factor: 2		

ablations presented in the main paper:

1. LSID trained w/ full Sony: An LSID model trained with the full Sony dataset is used as a baseline for further ablation experiments. The LSID model obtained a PSNR of 27.30 and SSIM of 0.774 on the Sony test set.

2. CIE-XYZ Input: In the processing of a RAW image to sRGB, there is an intermediate conversion to the CIE-XYZ common color space before applying non-linear post-processing steps. We formulate this experiment as a CIE-XYZ to sRGB conversion to validate the requirement for domain adaptation for the extreme low-light image enhancement task. An LSID model trained with Sony data as source and fine-tuned on 4 Nikon camera images attains a PSNR of 27.16 and SSIM of 0.894 which is lesser than the proposed approach.

3. Fine-tuning only the last layer of LSID: We train an LSID model with the full Sony camera dataset as source and then fine-tune only the last layer with 4 Nikon camera images as target to investigate the transfer learning performance across camera sensor domains. We

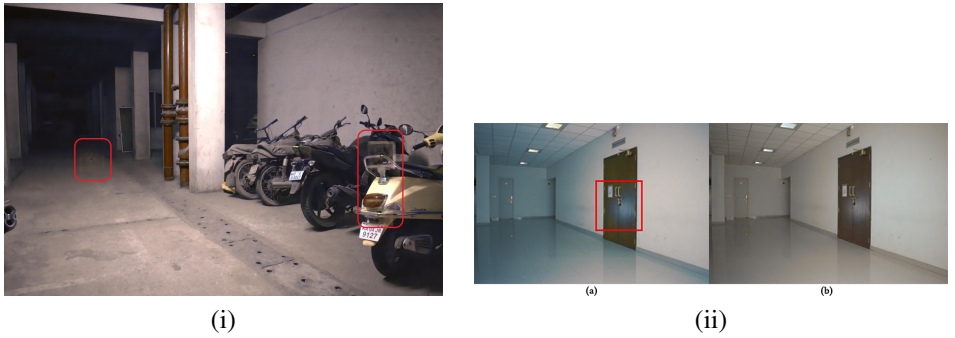


Figure 1: (i) Output of LSID trained on Sony dataset and last-layer fine-tuned on 4 Nikon camera images. The artifacts are encircled in red. (ii) An qualitative example from (a) model trained with combined encoder and decoder ablation, (b) Proposed approach.

Table 3: Ablation study on different baselines for Sony source and 4-shot Nikon as target training. The table reports mean PSNR over three different runs. See Section 2 for details.

No.	Details	PSNR	SSIM
1	LSID trained w/ full Sony	28.50	0.774
2	CIE-XYZ Input	27.16	0.894
3	Fine-tune last layer of LSID	27.30	0.886
4	Fine-tune all layers of LSID	27.93	0.899
5	Separate Encoder and Decoder	28.62	0.867
6	Combined Encoders	29.20	0.890
7	LAB Input	27.90	0.827
8	Proposed w/ Source L_1 loss	27.14	0.807
9	Proposed w/o Source SSIM loss	29.38	0.902
10	Proposed	30.30	0.913

repeat this experiment with 3 different sets to obtain the average performance. We obtain a PSNR value of 27.30dB and SSIM score of 0.886 on the Nikon camera test set, and also observe several artifacts in the outputs (as shown in Fig. 1).

4. Fine-tuning all layers of LSID: We first train an LSID model on the full Sony source dataset and then fine-tune all the layers using 4 Nikon target camera images. We observed that the outputs from the fine-tuned model were unable to capture the color distribution of the target Nikon camera and was affected by the color distribution of the source Sony camera data. The transfer learning methods have performed better than the Sony only baseline whereas fine-tuning all layers attains a PSNR of 27.93dB and SSIM of 0.899. We have used the L_1 loss for both the fine-tuning experiments.

5. Separate encoder and decoder: As discussed, the transfer learning methods performed sub-optimally primarily due to the color bias. Hence, we modified our proposed model to include domain-specific decoders in addition to the domain-specific encoders and \mathbb{N} network to learn the different camera color spaces individually for each domain. Since the domain-specific encoders improved the denoising performance in terms of increase in the SSIM metric, for this experiment, we hypothesised that using domain-specific encoders may allow the architecture to better learn the color space and finer corrective details after the processing from the shared \mathbb{N} network. However, this approach only obtained a PSNR of 28.62dB and SSIM of 0.867, and the results had visible color gaps between the model's

output and ground truth, showing that using separate decoders leads to a decrease in performance. Along with the color bias, we also note the increased computation time.

6. Combined encoder: To verify the domain-specific denoising performed by using separate encoders, we trained a single encoder and shared \mathbb{N} network. We obtained a PSNR of 29.20dB and SSIM of 0.890, which is better than fine-tuning the LSID model, but we also noticed the colors of the target domain’s output validation images to be dominated by a large number of source training images (Refer Fig. 1 (ii)). From this ablation and the separate encoder and decoder experiment, we find that using only domain-specific encoders and the shared \mathbb{N} network performs better few-shot domain adaptation for camera sensor domains in extreme low-light conditions.

7. LAB input: We trained the proposed model using input images in the LAB color space to investigate color quality performance. We used separate models to train for the L-channel and the a,b-channels. The L_1 and \mathcal{L}_{SSIM} loss are used for the L-channel; only L_1 loss was used for the a,b-channels. We obtain L and ab outputs separately from the model during inference time and convert them to the corresponding RGB image. The outputs showed higher color loss and an increase in noise obtaining a PSNR of 27.90dB and SSIM of 0.827.

8. Proposed approach with Source \mathcal{L}_1 : Training with a strong supervision loss for the source domain strongly influences the colors of the target domain’s output. Hence, this ablation prompted us to use a weak supervision loss such as the Cosine similarity loss for the source domain in order to ensure that the appropriate color spaces were preserved. For the ablation with L_1 loss for the source, we obtained a PSNR of 27.14dB and SSIM of 0.807.

9. Proposed approach without \mathcal{L}_{SSIM} : As discussed in the main paper, we observe the source domain influencing the target output image color. For our proposed model architecture, we experimented with various loss functions including combinations of L2 loss, grayscale SSIM, gradient loss, L1 loss, and cosine similarity loss. We found that using a combination of cosine similarity loss and SSIM loss for the source domain and the L_1 loss for the target led to better preservation of color and structural information. In the ablation without the SSIM loss for the source, we obtained a PSNR of 29.38 and SSIM of 0.902 whereas our proposed approach attains a PSNR of 30.30 and SSIM of 0.913.

10. Proposed: We found that a combination of 16-bit Cosine similarity loss and 8-bit SSIM loss for source camera, and L_1 loss for target pipeline led to better preservation of color and structure. In terms of performance (PSNR/SSIM), this approach outperforms other baseline experiments. We also noticed the outputs to have increased noise reduction and correct color transformations during inference.

The \mathcal{D} network for SSIM loss: As discussed in section 3, type-casting the source output from 16-bit to 8-bit space will still possess domain specific details. The SSIM loss is used to compare the brightness and structural details but not the color quality (cosine similarity is for color). Thus, following the camera ISP processing steps, we train a 16-to-8-bit conversion U-net model (\mathcal{D} in Fig. 2(b) in the main paper) to convert the 16-bit data to 8-bit data. The \mathcal{D} network is trained to perform the following non-linear operations: White balancing, Gamma correction, Quantization and JPEG compression. From experiments, we observe that a U-net is necessary to learn all the above mentioned non-linear operations.

3 Additional Results

In Figures 2 and 3, we present additional results for the Nikon camera dataset. In Figures 4 and 5, we present additional results on the Canon dataset. In addition, we also provide qual-

itative results for the two smartphone cameras trained with our proposed few-shot domain adaptation approach: OnePlus (Fig. 6, 7) and Google Pixel (Fig. 8, 9).

In addition, we also present qualitative results for the ablation experiments in Table 3 with Sony dataset as source and 4 Nikon images as target in Figures 10 and 11.

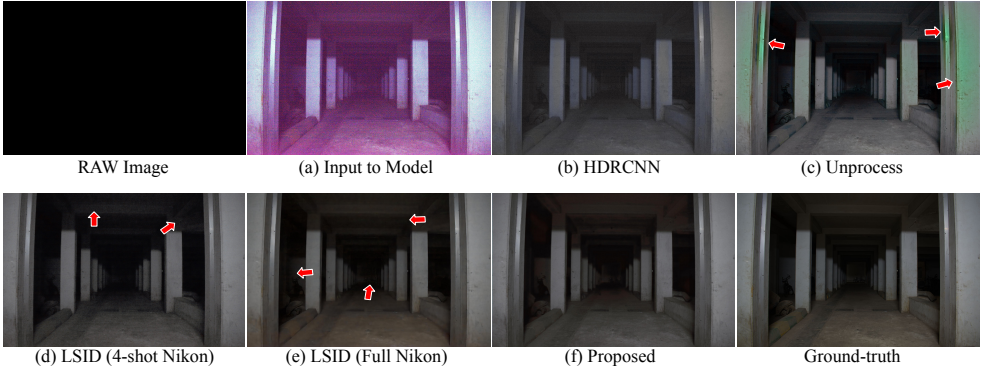


Figure 2: Qualitative comparison between different methods tested on Nikon target images. The models (b) HDRCNN and (c) Unprocess are trained on Sony source and fine-tuned on 4-Nikon target images, (d) only with 4-Nikon target images, (e) full ($k=53$) Nikon training dataset, and (f) 4-Nikon target images and 161 Sony source images with our proposed few-shot domain adaptation approach.

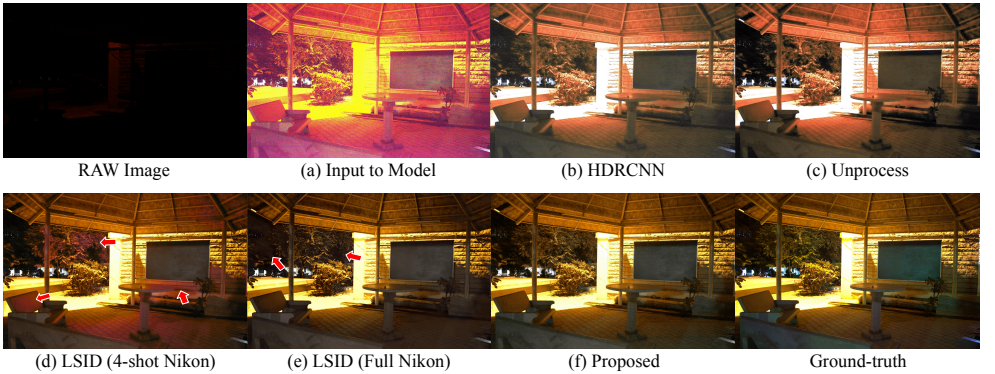


Figure 3: Qualitative comparison between different methods tested on Nikon target images. The models (b) HDRCNN and (c) Unprocess are trained on Sony source and fine-tuned on 4-Nikon target images, (d) only with 4-Nikon target images, (e) full ($k=53$) Nikon training dataset, and (f) 4-Nikon target images and 161 Sony source images with our proposed few-shot domain adaptation approach.

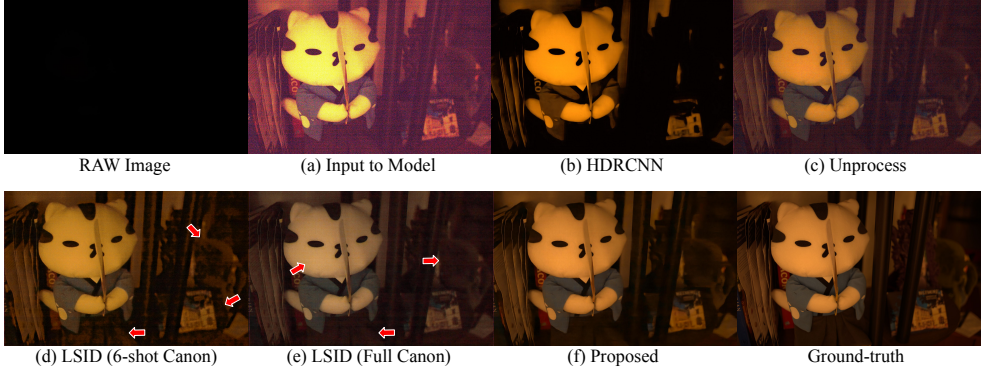


Figure 4: Qualitative comparison between different methods tested on Canon target images. The models (b) HDCNN and (c) Unprocess are trained on Sony source and fine-tuned on 6-Canon target images, (d) only with 6-Canon target images, (e) full ($k=44$) Canon training dataset, and (f) 6-Canon target images and 161 Sony source images with our proposed few-shot domain adaptation approach. Note that the color distribution in (f) is closer to the ground-truth as compared with all other baseline results.

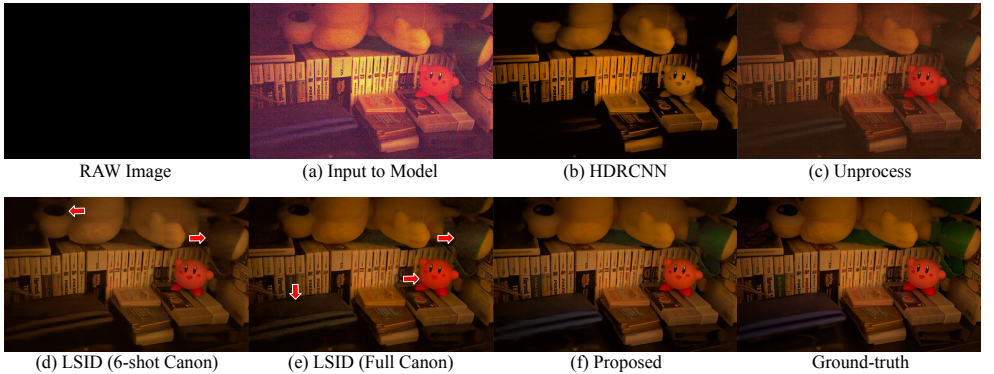


Figure 5: Qualitative comparison between different methods tested on Canon target images. The models (b) HDCNN and (c) Unprocess are trained on Sony source and fine-tuned on 6-Canon target images, (d) only with 6-Canon target images, (e) full ($k=44$) Canon training dataset, and (f) 6-Canon target images and 161 Sony source images with our proposed few-shot domain adaptation approach.

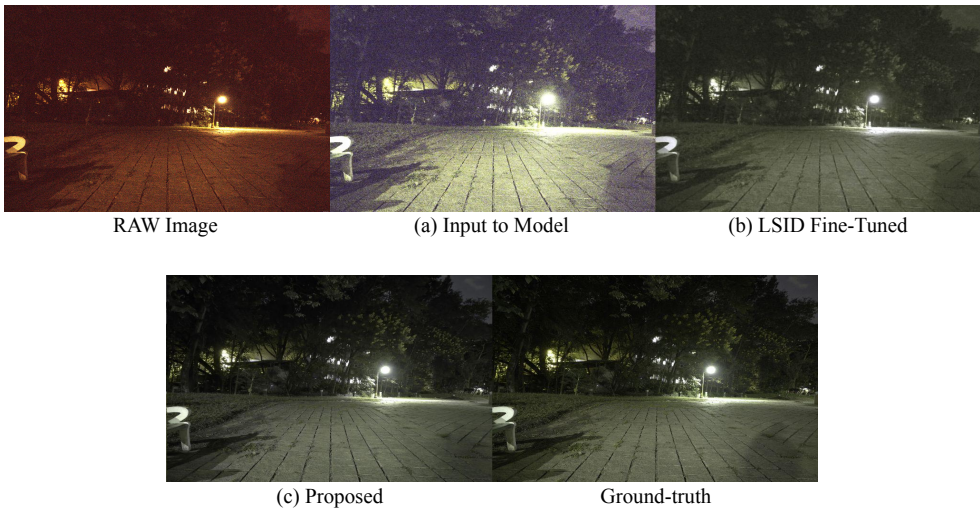


Figure 6: Qualitative results for Sony as source and OnePlus 5 camera images as target. Note that in the LSID Fine-tuned output for the first image, the color distribution is different from the ground-truth whereas our method is able to adapt to the OnePlus camera domain.

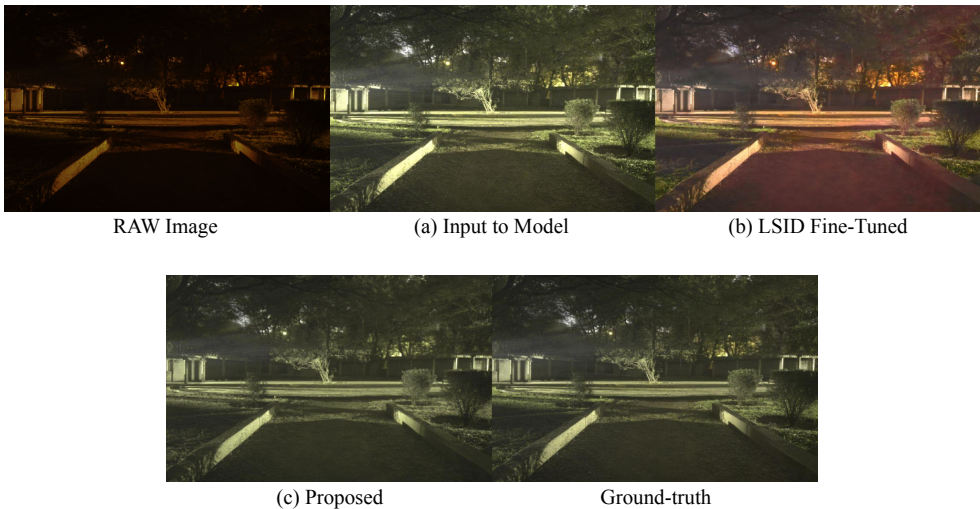


Figure 7: Qualitative results for Sony as source and OnePlus 5 camera images as target. Note that in the LSID Fine-tuned output for the first image, the color distribution is different from the ground-truth whereas our method is able to adapt to the OnePlus camera domain.

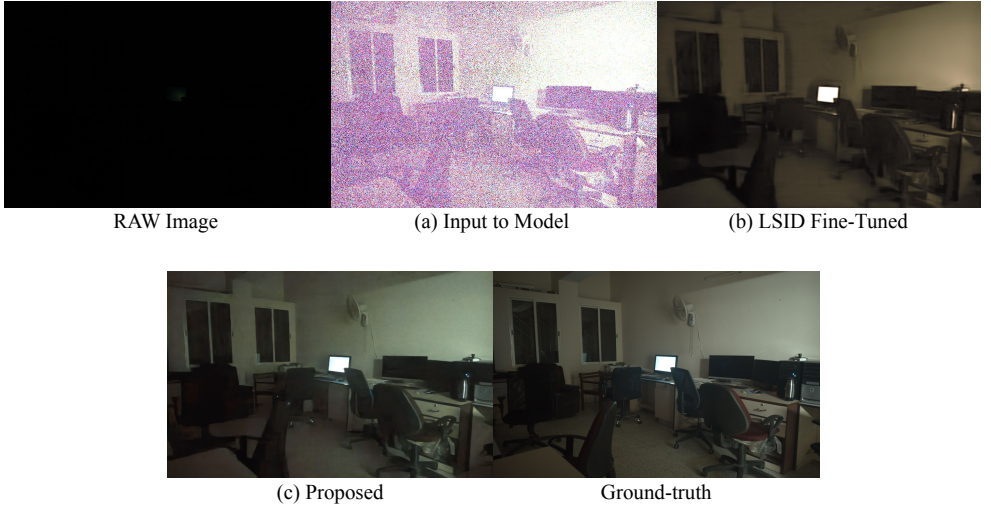


Figure 8: Qualitative results for Sony as source and Google Pixel images as target. Note that the input to model includes severe noise and the the LSID Fine-tuned output includes blurry artifacts and color distortions whereas our method is able to adapt to the Google Pixel camera domain.

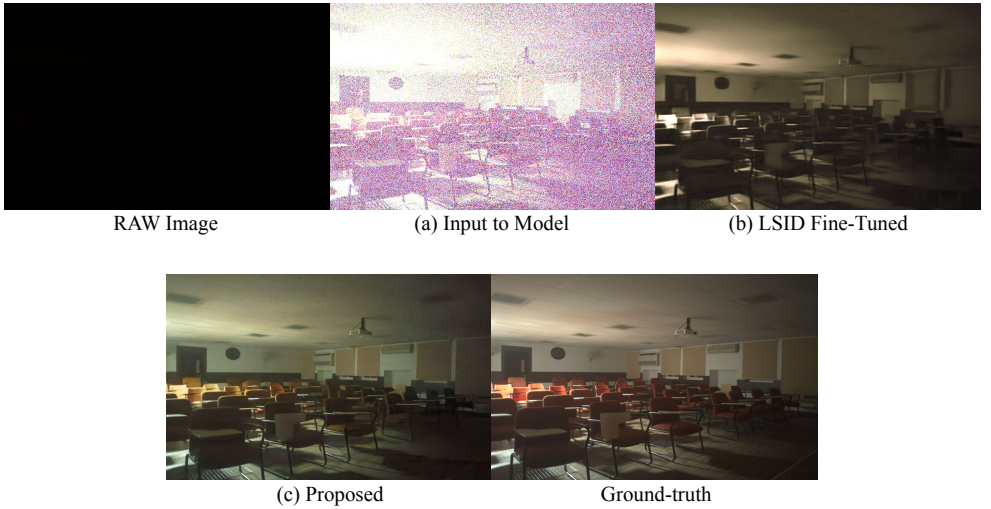


Figure 9: Another qualitative result for Sony as source and Google Pixel images as target. Note that the input to model includes severe noise and the the LSID Fine-tuned output includes blurry artifacts and color distortions whereas our method is able to adapt to the Google Pixel camera domain.

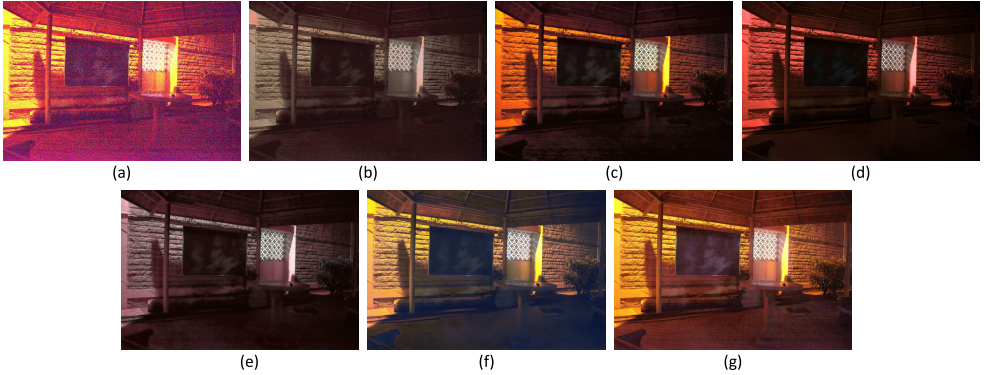


Figure 10: Qualitative example for various ablations results on Sony source and 4-shot Nikon images as target. (a) Input, (b) Separate encoder and decoder (No. 5 in Table 3), (c) Proposed approach w/o source SSIM loss (No. 9 in Table 3), (d) Combined encoders (No. 6 in Table 3), (e) Fine-tuning all layers of LSID model (No. 4 in Table 3), (f) Fine-tuning only last layer of LSID model (No. 3 in Table 3), (g) our proposed approach (No. 10 in Table 3).

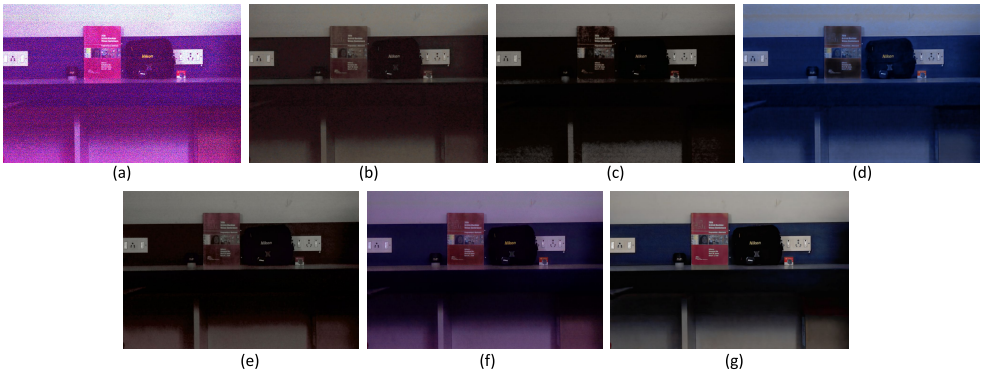


Figure 11: Another qualitative example for various ablations results on Sony source and 4-shot Nikon images as target. (a) Input, (b) Separate encoder and decoder (No. 5 in Table 3), (c) Proposed approach w/o source SSIM loss (No. 9 in Table 3), (d) Combined encoders (No. 6 in Table 3), (e) Fine-tuning all layers of LSID model (No. 4 in Table 3), (f) Fine-tuning only last layers of LSID model (No. 3 in Table 3), (g) our proposed approach (No. 10 in Table 3).