SFU SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

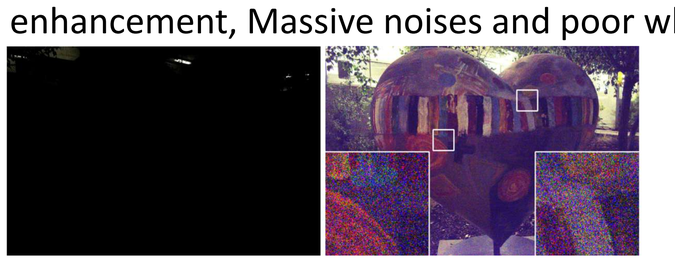
Beyond Vision in the Dark

Chengxi Li, Honghui Wang, Hoppe Wang, Michael Yang, Andy Zhang

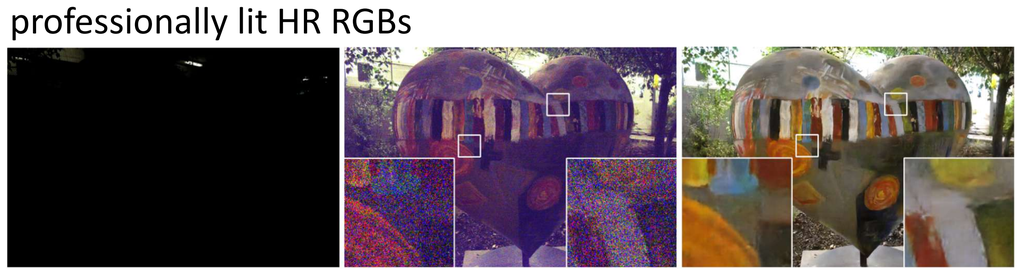
PROBLEMS & LIMITATIONS

LOW LIGHT IMAGE PROCESSING

- **Traditional:** limited effectiveness for denoising, deblurring, and enhancement, Massive noises and poor white balance



- **DarkConvNet:** Use CNN to convert low-light HR RAWs to professionally lit HR RGBs



- **Limitations:** - can only use HR RAWs as input
  - no solution for LR RGBs

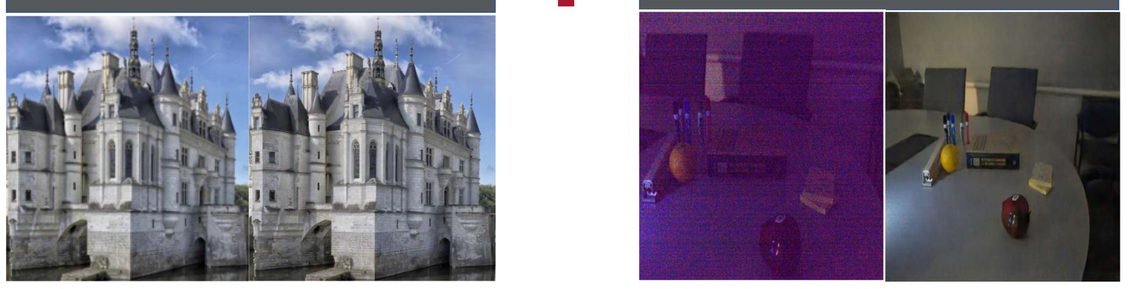
OBJECTIVE

Our Goal:

Low Resolution Dark RGBs → High Resolution Professionally Lit Photos

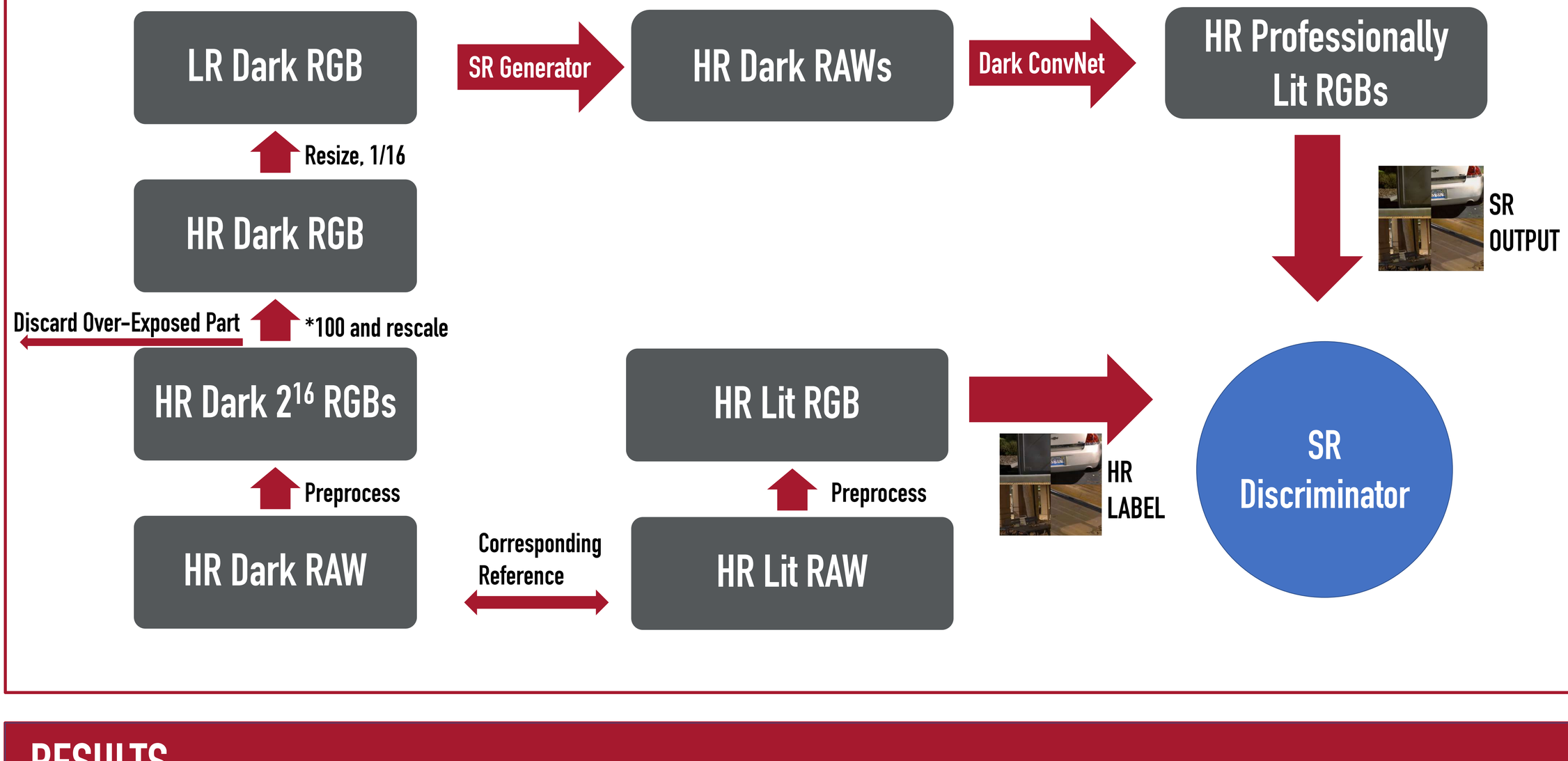
Our Solution: Dark Image Super Resolution:

SRGAN + DarkConvNet

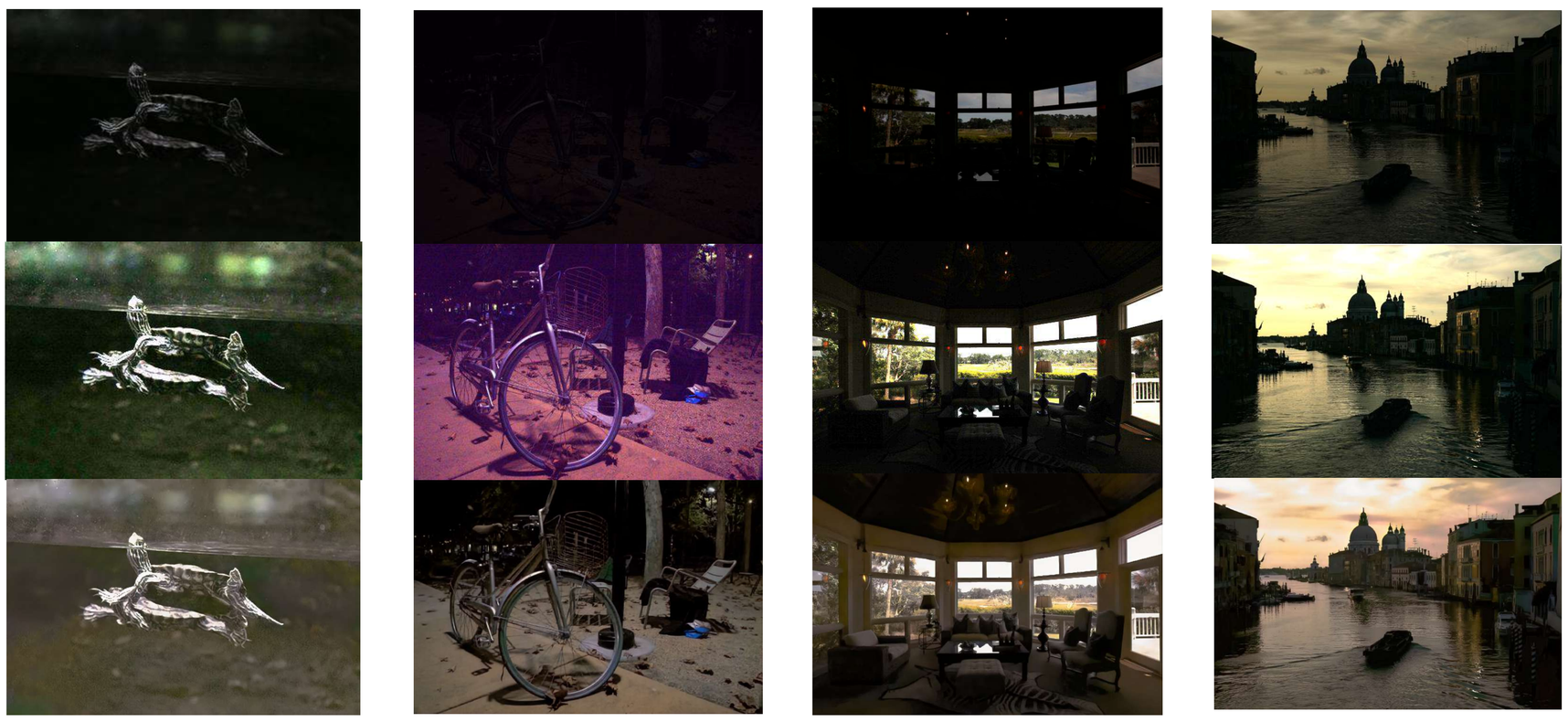


LR Dark RGBs → SR Generator → HR Dark RAWs → DarkConvNet → HR Professionally Lit RGBs

PIPELINE



RESULTS



Beyond Vision in the Dark

Chengxi Li
MSc. in Computer Science
Simon Fraser University
chengxil@sfu.ca

Haopeng Wang
MSc. in Computer Science
Simon Fraser University
hwa125@sfu.ca

Honghui Wang
MSc. in Computer Science
Simon Fraser University
honghuiw@sfu.ca

Chengxi Yang
MSc. in Computer Science
Simon Fraser University
cya91@sfu.ca

Shi Heng Zhang
MSc. in Computer Science
Simon Fraser University
shz1@sfu.ca

Abstract

Low-light image processing is challenging due to limited effectiveness for denoising, deblurring and white balance adjustment in extreme conditions. Despite its ability to produce bright images from dark high-resolution images, current state-of-the-art pipeline DarkConvNet still has its limitations. It heavily depends on the high-resolution RAW image from certain camera sensors, and has no way to produce satisfying outcomes directly from low quality sRGB images. To solve this limitation, we modify the original model and combine it with the Super Resolution using Generative Adversarial Network (SRGAN), the state-of-the-art super resolution model, to create a new pipeline for low-resolution low-light image processing. We use the same dataset provided by DarkConvNet to train our network to provide comparable results. Our model is able to turn low-resolution dim sRGB images to high-resolution professionally lit images.

1 Introduction

Methods for portraying low-resolution images to high-resolution images is referred as super-resolution, which is studied by many in the field of computer vision [1]. Previous work in denoising and adjusting low light images was focused on high-resolution images with certain details preserved, and cannot address images with low-resolution [2]. In order to compensate the loss of details in low-resolution dark image, we utilize the concept from the SRGAN model developed by C. Ledig et al. [3] and further improved on it.

Inspired by the above ideas, investigating the possibility of turning low-resolution pictures shot in dark environments into high-resolution images with proper lighting conditions became the interest of this project. The approach is essentially to adjust the brightness to a degree that details are clearly visible, but not so overexposed through one of the two stages. Then the processed picture will be fed into another stage to increase the resolution and give the output. However, the order of the stages might be changed depending on the input and output of the base models during our experiments.

While investigating the input and output format of our base models, the DarkConvNet is limited to work only with RAW image data taken from Sony or Fuji DSLR cameras.[2]. Therefore, there is currently no direct solution for most common picture formats such as JPG and PNG.

In this paper, we propose a solution for approximating dark images with low-resolution to high-resolution with proper lighting conditions and better visibilities built on top of the DarkConvNet model and SRGAN model. Details of the changes and adjustments made toward the mechanisms of these two models will be discussed later in this paper.

1.1 Related work

As mentioned above, our paper is based on 2 state-of-the-art models that have delivered promising results for low-light image processing and image super-resolution respectively. A brief review of these two existing models is provided below.

1.1.1 DarkConvNet

For low-light image denoising, traditional methods are often based on specific image standards such as smoothness, sparsity, low rank, or self-similarity [2]. Researchers have also explored applications of deep networks for denoising; however, these deep networks were evaluated on synthetic Bayer patterns and synthetic noise, rather than real images collected in extreme low-light conditions.

Due to those limitations, in their paper, they collected a new dataset for training and benchmarking single-image processing of RAW low-light images. Each RAW short-exposure low-light image has its corresponding long-exposure bright reference image used as labels. The pipeline can operate directly on raw data (after amplification) collected from camera sensor and give excellent result. Rather than traditional pipeline with a sequence of modules such as white balance, demosaic, denoising, sharpening, color space conversion, gamma correction and etc, they trained a fully-convolutional network (FCN) end to end to perform the entire image processing pipeline. Figure 1 shows the entire pipeline [2].

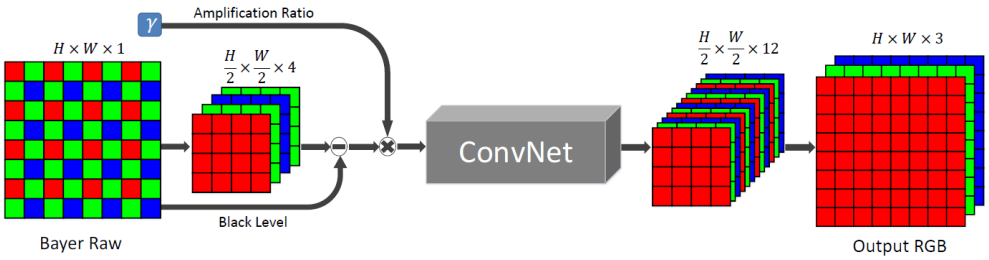


Figure 1: DarkConvNet pipeline

The architecture choice for their FCN is U-net due to GPU memory consideration. They trained the network using L1 loss and the Adam Optimizer. The Amplification Ratio determines the brightness of the output. We used their pretrained model as part of our pipeline.

1.1.2 SRGAN

Many previous state-of-the-art approaches for single image super resolution (SISR) relied on MSE loss as the standard for optimization of their models. However, C. Ledig et al. [3] found that pixel-wise loss functions such as MSE struggle to handle the uncertainty inherent in recovering lost high-frequency details, thus resulting in perceptually unsatisfying solutions with overly smooth textures. In order to achieve a high upscaling factor and improve the details generated by the model, the author built a very deep ResNet architecture (to facilitate the training of deep CNNs) while utilizing the GAN, and replaced MSE-based content loss with a loss calculated on feature maps of the VGG network [4]. By doing that, the author has achieved a new state-of-the-art method for the estimation of photo-realistic SR images with high upscaling factors (4). Figure 2 shows their pipeline [3].

1.2 Our contributions

In this paper, we try to utilize a similar architecture as SRGAN, and combine the SRGenerator and DarkConvNet together to solve the original limitations of DarkConvNet, and extends its application on low-resolution sRGB as input. Our main contributions are:

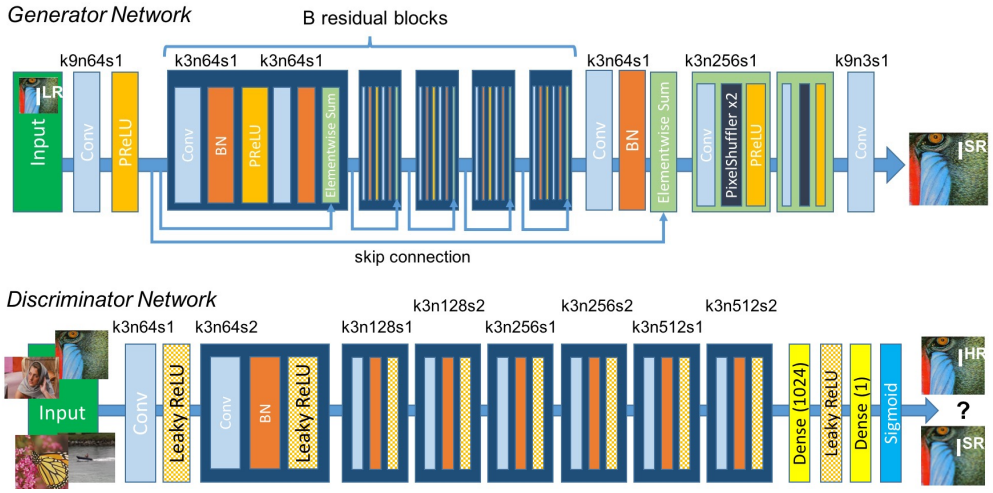


Figure 2: SRGAN model architecture

- Modify the SRGenerators output structure to match the input structure of the DarkConvNet, so that we can combine those two models.

- Propose a new pipeline to convert low-resolution dark RGB images into high-resolution bright RGB images.

We preprocess the original See-in-the-Dark dataset, and convert those RAW images into RGB formats for later work. We use the same dataset for the purpose of comparing our results to original DarkConvNet results. We describe the pipeline architecture and the dataset in detail in Section 2. Details about model training as well as visual illustration are provided in Section 3.

2 Approach

Our approach is essentially to put a low resolution RGB image into the modified SRGenerator to upscale 4 times and output a NumPy array that can be interpreted as a RAW image. Then, we can feed the output into the pretrained DarkConvNet model and get the bright and clear image in RGB format. With this, we can solve the limitation of the DarkConvNet and can handle any image found on the internet.

2.1 Datasets

We use the original See-in-the-Dark dataset [2] in order to provide comparable results. The dataset contains 5094 short exposure dark images in RAW format, each with a corresponding long-exposure bright reference image in RAW format as well. Multiple short-exposure images can correspond to the same long-exposure reference image. The dataset contains both indoor and outdoor images to cover different situations. The author also used 2 cameras with different sensors to capture the image. The Sony camera has a full-frame Bayer sensor and the Fuji camera has an APS-C X-Trans sensor. This supports evaluation of low-light image processing pipelines on images produced by different filter arrays. We only use the Sony dataset to train our model.

2.2 Pipeline

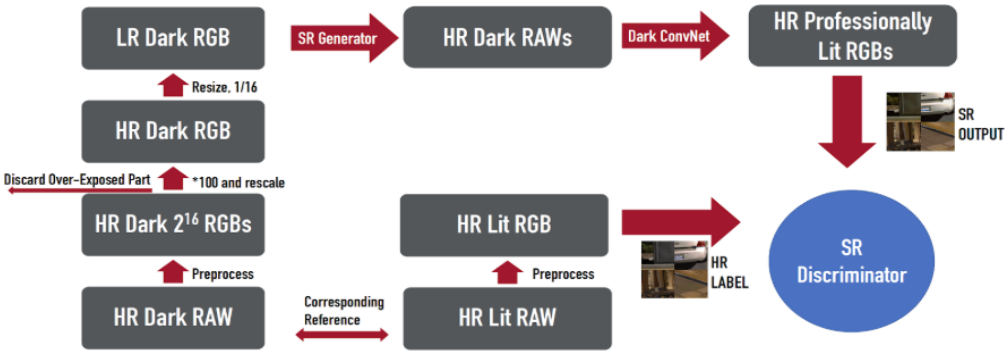


Figure 3: Re-defined architecture of our network

We first convert a high-resolution dark RAW image into a low-resolution amplified RGB image (details for data preprocessing will be discussed in section 3). We feed this data into our modified SR Generator for detail tuning and super resolution. The output of the SR Generator is a NumPy array that can be interpreted as a high-resolution 4-channel RAW, so we can feed that into the pre-trained DarkConvNet (based on the Sony camera sensor) for image processing. The output would be a high-resolution professionally lit RGB image. We use this output as training features for the SR Discriminator. We also convert the corresponding bright reference of the dark RAW image into a RGB image, and use it as the label for the SR Discriminator. As we learned in the course, we optimize the Discriminator network in an alternating manner along with the Generator to solve the adversarial min-max problem.

$$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{I^{HR} \sim p_{train}(I^{HR})}[\log D_{\theta_D}(I^{HR})] + \mathbb{E}_{I^{LR} \sim p_G(I^{LR})}[\log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR}))] \quad (1)$$

Equation 1 shows the min-max formula used by original SRGAN. The idea is that it allows us to train a Generator Network with the goal of fooling a Discriminator Network that is trained to differentiate and distinguish super-resolved images from real images. With this approach the Generator can learn to create solutions that are highly similar to real images and thus difficult to classify by Discriminator. We basically use the same architecture as the SRGenerator and SRDiscriminator, only modify a last few layer of the SRGenerator to match the DarkConvNet's input structure. Specifically, we change the last Convolution layer from 3 kernel to 4 kernel which corresponds to the 4 channel RAW Sony image data, and take out the last upsampling layer to make the output resolution consistent. We also follow the SRGAN paper to use VGG network [4] as the perceptual loss function to achieve better results.

3 Experiments

3.1 Data preprocessing

We first convert the high-resolution dark RAW images into 16-bits RGB channels by using RawPy, and rescale the range of values in each channel to $[0, 1]$. For dark images, most pixels have very low values, so we scale the data by an amplification ratio according to the corresponding label, and discard the high light part (the overexposed pixels with values bigger than 1 after amplification are replaced by value of 1). It works as normalization for the input and it will stabilize the brightness of the overexposed area as well as amplify the inherent noises from the camera sensor. Although this may decrease the quality of the input pictures and hence make the training more difficult, the low quality pictures will also enable the model to learn more about low-resolution images. Namely,

with poorer training sets, our new model can learn to distinguish noises and pixels with poor white balance from regular pixels that preserve the details such as textures of objects and information of different light reflections of objects and environments.

After above steps, we obtain the low-resolution counterparts for input by downsampling the normalized high-resolution dark images using bicubic kernel with downsampling factor equals 4.

For the label, we simply convert the corresponding high resolution lit RAW image into 16-bits RGB and rescale the pixel value to $[0, 1]$.

3.2 Training details and parameters

We use the original pretrained SRGenerator's weights as initial weights for our SRGenerator, but randomly initialize the last 4 layers. We also use the pretrained DarkConvNet's weight and keep it fixed. We only update the weight of our SRGenerator and SRDiscriminator during training.

While training, we closely monitor the generator-loss, discriminator-loss, MSE-loss, and VGG-loss. Since we rework the model and only have a few days to train the model with limited computing power, we use some of the pretrained weights as baseline to accelerate the training process. Among all of the losses, only the discriminator-loss struggles to decrease when it is around 0.83 for each Epoch, while all the other losses are well under 10^{-3}. Therefore, we modify and experiment several learning rates(varied from 0.1 to 0.0001), and finally the discriminator-loss seems to converge around 0.75.

We train the GAN model on a NVIDIA GTX 1070 GPU with 8GB of GPU memory, and use the original See-in-the-Dark dataset [2]. For each batch, we randomly crop the image and input 16 96x96x3 matrix and output 16 384x384x3 matrix. Since the default for Python is single-thread, each Epoch is taking extremely long time to finish. We then move all the training data onto a SSD and change to three-threads for loading the input files; this decreases the training time by about 50%.

3.3 Model performance and quantitative evaluation

The original DarkConvNet uses Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM) as the metrics to evaluate the performance. These metrics are common measures to evaluate the Super Resolution algorithm, and in order to provide a comparable result between our work to the original DarkConvNet, we choose to use the same. Table 1 shows the performance of our model vs. other methods. The metrics are calculated based on the same DarkConvNet test sets.

We notice that the PSNR/SSIM of our model are slightly lower than the original DarkConvNet. However, DarkConvNet uses high-resolution dark RAWs as input; whereas our pipeline uses low-resolution dark sRGB. Therefore, we can conclude that, our model can provide very similar results as the original DarkConvNet model even when it is given much worse inputs. Our model can solve the limitation of the original DarkConvNet model and is able to transform low-resolution dark sRGB to high-resolution professionally lit photos. Due to limited time and computing power, our model was trained only 70 epochs. We believe that with longer training time and better computing power, our model is able to give a better result.

Table 1: mean PSNR/SSIM for each condition

Metric	DarkConvNet (Sony)	DarkConvNet (Fuji)	Traditional Pipeline	Our Pipeline
PSNR	28.88	26.61	21.26	25.73
SSIM	0.88	0.68	0.60	0.74

3.4 Model performance and qualitative comparison

From Figure 4, we can notice that even our model uses low-resolution sRGB as input, we can still achieve similar perceptive results as DarkConvNet. Figure 5 shows our model results given random online images as input.

(a) Amplified input (low-resolution)

(b) Label

(c) DarkConvNet (high-resolution RAW as input)

(d) Our Pipeline (low-resolution RGB as input)

Figure 4: Qualitative performance comparison on original DarkConvNet datasets

4 Conclusion

With the elaborate combination of the two models, we transform pictures that are sRGB with low-resolution and very low brightness into high-resolution images with near-natural lighting conditions. As shown in the previous section, the proposed model is able to generate outputs with enhanced details, finer textures and visible contours of some of the seemingly hidden objects in the original input. Furthermore, given dark inputs, the model provides realistic lighting adjustment results which are close to that of the professionally lit photographs for both indoor and outdoor images. Quantitatively, the metrics (PSNR and SSIM) show that our pipeline achieved exceedingly remarkable scores well past the results one could expect from the traditional pipeline. Our model does not rely on high-resolution RAW images as DarkconvNet does, and can achieve equally good results using low-resolution dim sRGB inputs.

4.1 Discussion and future work

Our proposed model requires to linearly amplify the input image according to a ratio which need to be designed manually. Future work might consider to let the model learn this amplification ratio automatically. Furthermore, due to the deep structure of the model, the program can not operate in real time. One possible extension of our model could be optimizing the structure and make it feasible to use in the field of surveillance. For security industry, estimating low-resolution frames sourced from video footage of security cameras during the night or low light environments to brighter videos with critical details clearly shown will provide game changing conveniences for monitoring or investigation works. For this purpose, we might need to simplify our network structure at a sacrifice of certain performance, and can introduce LSTM RNN to achieve consistency of different time frame in a video.

Figure 5: Results of our model. The picture on the up-left corner of every image encompassed by red rectangular is the input RGB image. The big images are the outputs of our model. The figure shows their true relative size, where the input is only 1/16th of the output.

Acknowledgments

The authors would like to acknowledge Chen Chen, Qifeng Chen, Jia Xu, Vladlen Koltun, C. Ledig, L. Theis, F. Huszr, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani,J. Totz, Z. Wang, and W. Shi. for their amazing works on the DarkConvNet and SRGAN on which our model bases. We would like to also thank the TAs for their constructive comments and Professor Mori for his inspiring and witty guidance.

Contributions of members

Haopeng Wang: worked on modifying the SRGAN model and solved the RGB to RAW problem

Chengxi Yang: designed and created the poster, debugged and tested the modified SRGAN model

Honghui Wang: brought up the idea, researched extensively on the topic and use Tensorboard to visualize the model to help debug.

Chengxi Li: set up Google Colab environment for experiments. Debugged and tested the modified SRGAN model.

Shi Heng Zhang: set up the training environment and worked on modifying the SRGAN model and solved the RGB to RAW problem.

We all contributed our work to the report, and we agreed that the work is evenly distributed.

Reference

[1] Kamal Nasrollahi and Thomas B. Moeslund. Super-resolution: a comprehensive survey. *Machine Vision and Applications*, 25(6):1423–1468, Aug 2014.

[2] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to See in the Dark. *arXiv e-prints*, page arXiv:1805.01934, May 2018.

[3] C. Ledig, L. Theis, F. Huszr, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, July 2017.

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.