CS 7180 Milestone 1

Nalin Gupta
NUID: 001726854

Christopher Botica
NUID: 001726854

Tyler Brown
NUID: 001684955

1 Introduction and Related Work

Images taken in low-light conditions are often too dark, noisy, and distorted to be used in industrial purposes. We propose a deep-learning model that processes low-light images to improve image brightness and increase their overall quality. The problem with imaging in low-light conditions is challenging due to low-photon count and low Signal-to-Noise (SNR) ratio. These yield very dark and noisy images. The most common technique to overcome this problem is long exposure shot. However, this method yields blurry images with the slightest camera shake or object motion[1]. Common post-processing techniques brighten the image at the expense of image quality. Being able to "see in the dark" provides a number of real-world benefits such as photography, computer vision, and social networking.

In the past, the problem of enhancing low light images has been tackled via noise reduction. This noise becomes dominant especially in low-light images due to low SNR. Remez et. al. proposed a deep CNN for noise reduction under the assumption that this low-light noise belongs to a Poisson distribution [2]. They used images from ImageNet [3] as their ground truth data and added synthetic Poisson noise to simulate corrupted images. Even though their model outperform the state-of-the art de-noiser "BM3D", it does not scale well to real world images, due to their underlying assumptions. Furthermore, their model only denoises images but does not brighten them. Motivated by these downfalls, Chen et. al., proposed an end-to-end CNN, "See-in-the-Dark" (SID), which brightens extremely low light images and removes noise without making any underlying assumptions [1]. However these advances come with the added expense of collecting large amounts of low light and bright light images. In the absence of a true vs noisy image dataset, the team captured scenes using various exposure times to generate true (bright light) and corrupted (low light) image pairs called "See-in-the-Dark Dataset" (SID Dataset [1]). Furthermore, their model is camera specific and not easily generalizable.

We propose a transferable CNN for image brightening and denoising. Instead of training our model on actual true (bright light) and corrupted (low light) image pairs, we use images from the publicly available "MIT-Adobe FiveK Dataset" dataset as our baseline and corrupt these by simulating low-light conditions. We train our CNN on the synthetic data to obtain our initial model parameters. Then, using these, and a small fraction of the real image pairs from the SID Dataset, we adopt a transfer learning [4] approach to update our model parameters. We then use this model to test on our SID Dataset. In addition, we aim to test various transfer learning approaches, such as the traditional transfer learning and zero shot learning [5, 6, 7].

The novelty of our approach stems from the idea of "more for less". Our model drastically reduces the overhead costs of data collection by synthesizing readily available training data (MIT-Adobe FiveK). This is particularly beneficial in domains where collecting images pairs is expensive/time consuming.

[1]https://github.com/cchen156/Learning-to-See-in-the-Dark

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

2 Model

We first collect 5,000 images in raw format from the MIT-Adobe dataset, taken with SLR camers by a set of different photographers in different scenarios. We made sure that these photographs cover a broad range of scenes, subjects, and lighting conditions.

Considering these images as the ground truth (i.e., images taken in normal light conditions), we corrupt them to simulate images taken in low-light conditions. These distorted images and original images consist of our training inputs to our first model. In order to simulate the distortion created by taking pictures in poor light conditions, we emulate the traditional image processing pipeline for correcting such images, only in reverse.

As illustrated in Figure 1, the traditional pipeline takes a corrupted image, and applies the following sequence of modules: Reduce Black Level, Denoising, White Balance, and Gamma Correction. The Black Level refers to refers to the level of brightness at the darkest parts of the image, and is reduced by subtracting the minimum pixel value. Denoising is reduced using common algorithms such as BM3D. White Balance refers to the color balance in the image (i.e., white should be true white) and is corrected by re-balancing the intensities of each color RGB. Finally, Gamma Correction controls the overall brightness of the image. We synthetically generate corrupted images by applying the reverse of this pipeline. Gamma Distortion: decrease the brightness of the image during our Gamma Distortion step, White Imbalance: skew the color-space by multiplying each level of RGB by a random weight, Poisson Noise: add Poisson noise to the image, Black Level: add a positive bias to the pixel values (i.e., random black level).

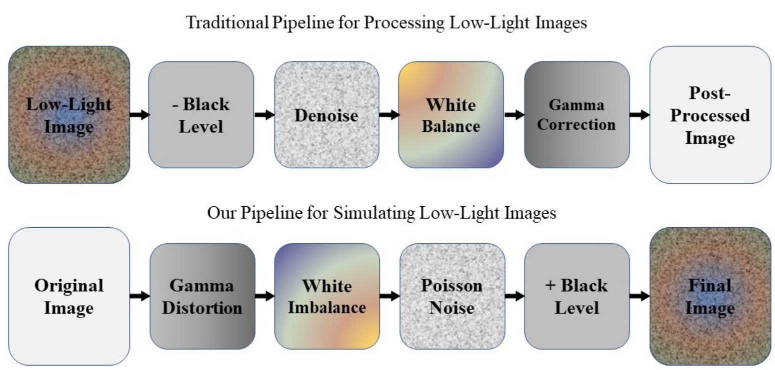


Figure 1: Top: Traditional Pipeline for processing low-light images. Bottom: Our pipeline to simulate low-light images based on the traditional, only in reverse.

Our model is based on the one developed for SID, with the addition of 2 fully-connected layers. Note that we may increase this if training runtime permits. Using the transfer learning approach, we will first train our model on the MIT dataset, then erase the learned weights from the last two fully-connected layers, and retrain on the SID dataset. This is highlighted in our model framework in Figure 2.

3 Experiment

We first collect 5,000 images in raw format from the MIT-Adobe dataset, taken with SLR cameras by a set of different photographers in different scenarios. We then run these images through our pipeline to generate our simulated low-light images. An example of two of such images is represented in Figure 3. We use the SID Model as our baseline and our performance measure will be achieving a Peak Signal-to-Noise Ratio (PSNR) greater or equal to the baseline.

Using the GitHub repository provided by Chen et. al. [1], we were gather and run to run their code but ran into limitations with computational resources. Chen et. al. [1] used a separate model for each of the two cameras. They specified a minimum of 64GB of GPU RAM for the Sony model and 128 GB of GPU RAM for the Fuji model. We decided to try replicating the less resource intensive Sony model. The hardware requirements for the Sony model are nontrivial so we have explored several

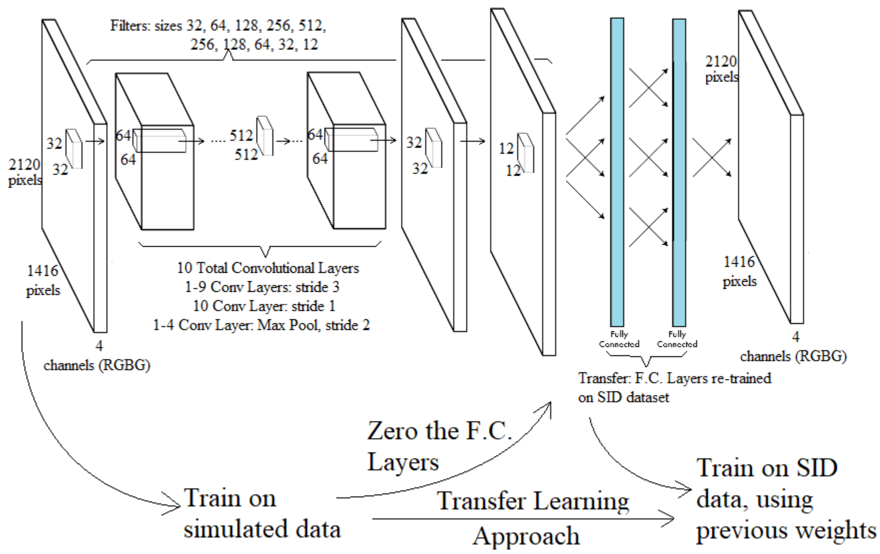
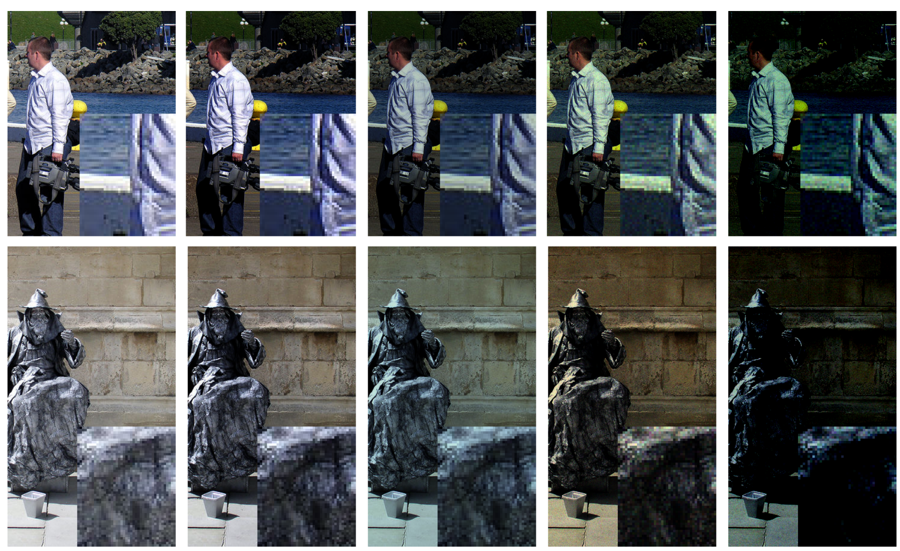


Figure 2: Our proposed model framework



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

Figure 3: Simulating two low-light images via our pipeline: (a) Original image, (b) Gamma Distortion, (c) White Imbalance, (d) Poisson Noise addition, (e) Black Level, final low-light simulation

options.

Using AWSEducate did not work for us. We were unable to create roles with IAM authentication so it's really hard or impossible to move data from a S3 Bucket to an EC2 Instance. We tried to create a $p3.8xlarge$ instance but these instances are not allowed even though they are listed. Using regular AWS does work but is costly. Ran a single AWS EC2 $p3.8xlarge$ instance with 32 CPU, 244 GB of Memory, 4 Tesla V100 GPUs, and 64 GPU Memory. This costs $12.24 an hour. This is the amount of GPU Memory requested by the paper authors as a minimum amount.

Given these initial facts, we then tried to estimate run time requirements for the Sony model. We trained the Sony model for 90 minutes using an AWS $p3.8xlarge$ instance. We were able to complete

$\approx$ 12 training epochs. Figure 4 contains an example test image/labelled image pairing. We then tried to use the training parameters to test the model but could not get this to work without modifying the script provided by Chen et. al. [1]. The model parameters were available to download, we used these parameters to run against test data for 30 minutes. Figure 5 contains example output from this test run. Both Figure 4 and Figure 5 are pictures of a similar yellow bike. We can see a qualitative improvement across the three renderings of the same predicted test image in Figure 5.

Additional time was required to extract the Sony dataset which decompressed to about 115GB of image files. We estimate that replicating the Chen et. al. [1] training parameters for *only* the Sony model takes approximately 500 hours or $ 6,120.

4 Appendix

Figure 4: Example Training Data

(a) Input (b) Output

Figure 5: Example Test Data

References

[1] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. *arXiv preprint arXiv:1805.01934*, 2018.

[2] Tal Remez, Or Litany, Raja Giryes, and Alex M Bronstein. Deep convolutional denoising of low-light images. *arXiv preprint arXiv:1701.01687*, 2017.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[5] Hugo Larochelle and Y Bengio. Classification using discriminative restricted boltzmann machines. pages 536–543, 01 2008.

[6] Mark Palatucci, Dean Pomerleau, Geoffrey Hinton, and Tom M. Mitchell. Zero-shot learning with semantic output codes. In *Proceedings of the 22Nd International Conference on Neural Information Processing Systems*, NIPS'09, pages 1410–1418, USA, 2009. Curran Associates Inc.

[7] Richard Socher, Milind Ganjoo, Hamsa Sridhar, Osbert Bastani, Christopher D. Manning, and Andrew Y. Ng. Zero-shot learning through cross-modal transfer, 2013.