Abstract

Deep learning methods for enhancing dark images learn a mapping from input images to output images with pre-determined discrete exposure levels. Often, at inference time the input and optimal output exposure levels of the given image are different from the seen ones during training. As a result the enhanced image might suffer from visual distortions, such as low contrast or dark areas. We address this issue by introducing a deep learning model that can continuously generalize at inference time to unseen exposure levels without the need to retrain the model. To this end, we introduce a dataset of 1500 raw images captured in both outdoor and indoor scenes, with five different exposure levels and various camera parameters. Using the dataset, we develop a model for extreme low-light imaging that can continuously tune the input or output exposure level of the image to an unseen one. We investigate the properties of our model and validate its performance, showing promising results.

1. Introduction

Images captured in low-light are characterized by low photon counts which result in a low signal-to-noise ratio (SNR). Setting the exposure level while capturing an image can be done by the user in manual mode, or automatically by the camera in auto exposure (AE) mode. In manual mode, the user can adjust the ISO, f-number and exposure time. In auto exposure (AE) mode, the camera measures the incoming light based on through-the-lens (TTL) metering and adjusts the exposure values (EVs), which refers to configurations of the above parameters. In a dark environment, adjusting the parameters to increase the SNR has its own limitations. For example, high ISO increases the noise as well and lengthening the exposure time might introduce blur.

We investigate the problem of continuous exposure for extremely low-light imaging based on a single image[7]. Various approaches have been proposed as post processing enhancements in low-light image processing [32, 15, 6, 18, 33, 14]. Such methods often fail to produce satisfactory results in the dark environment which we consider, with as much as 0.02lux. Recent works[22, 16, 7, 25, 29] introduce data-driven approaches to replace the traditional image signal processing pipeline and learn a direct mapping from low-exposure input images to well-lit output images. They are trained to enhance an image with fixed input and output exposure levels, without the ability to generalize to unseen level. However, at inference time, the optimal exposure level for the model of the dark input image and the required output image are continuous and unknown, and might depend on the user’s preference as well.

We present an approach that can resolve this, allowing the flexibility to continuously set the input and output exposure levels at inference time, without the need for retraining. First, we train a network that learns to restore low-light raw images to an initial exposure level. Next, we insert additional blocks into the network and fine-tune it to restore the raw images to a final exposure level. At inference time, the user is able to control the desired exposure levels between the initial and final levels by interpolating the weights of the additional blocks based on an input parameter. In our model, the initial and final exposure levels are for the output image given the exposure of the input image, or vice versa. Our tunable network requires only a small amount of additional parameters with respect to the original one.

Training a tunable model for continuous exposure of extreme low-light images requires images of the same scene.
For a given input, we can continuously select the desired exposure time of the input or the output at runtime. With multiple short and long exposure times over a wide variety of environments. To the best of our knowledge, there is no such public dataset. We have collected 1500 raw images captured with five different exposure levels in extreme low-light conditions, in both indoor and outdoor environments. Using the dataset, we developed and trained our model, showing that it can successfully control the desired exposure level at runtime.

This paper therefore contributes as follows: (a) We introduce a model for enhancing extreme low-light images by continuously adjusting the input or output exposure at inference time. To the best of our knowledge, tunable exposure for extreme low-light imaging has not been researched so far. (b) To support further research in continuous exposure models we introduce a new dataset of 1,500 raw extreme low-light images, with five different exposure levels from both indoor and outdoor scenes, and (c) We validated our approach on both a wide variety of images, showing its ability to dynamically restore the exposure and successfully enhance images captured in dark environments.

### 2. Related Work

**Low-light Image Enhancement** Widely used enhancement methods are histogram equalization, which globally balances the histogram of the image and gamma correction, which increases the brightness of dark pixels. More advanced methods include illumination map estimation [10], semantic map enhancement [30], bilateral learning [9], multi-exposure [5, 31] and Retinex model [28, 34, 8, 4]. In contrast to these methods, we consider extreme low-light environment with very low SNR where the scene is barely visible to the human eye. Chen [7] has introduced an approach for extreme low-light imaging by replacing the traditional image processing pipeline with a deep learning model which is based on raw sensor data. Wang [25] introduced a neural network for enhancing underexposed photos by incorporating illumination map into their model, while Xu [29] presented a model for low-light image enhancement based on frequency-based decomposition.

These methods are optimized to output an enhanced image with a fixed exposure. In cases where the user requires a change in the exposure of the output image, these methods require retraining the models, typically on additional set of images. On the contrary, we introduce an approach that allows continuously setting the exposure at inference time.

**Modulating Networks** Recently, there has been a growing interest in constructing networks which can be continuously tuned at inference time. These can broadly be categorized as models which allow tuning different objectives at runtime [13] or different restoration levels of the same objective. A typical approach is to train a network on different restoration levels and apply interpolation between the resulting weights. Dynamic-Net [24] adds specialized blocks directly after convolution layers, which are optimized during the training to an additional objective. CFSNet [26] uses branches, each is targeted for a different...
Figure 3: The multi exposure dataset for extreme low light. The top two rows are images from outdoor scenes and the bottom two rows are images from indoor scenes. Left to right are exposure times of 0.1s, 0.5s, 1s, 5s, 10s.

Objective. AdaFM [12] adds modulation filters after each convolution layer. Deep Network Interpolation (DNI) [27] trains the same network architecture on the different objective and interpolates all parameters. Son [19] extends the approach of AdaFM with Filter Transition Network (FTN) allowing better non-linear interpolation. These methods are optimized for denoising, super-resolution, style transfer and compression artifacts reduction. We focus on the complementary task of continuous exposure for extreme low-light images.

Datasets Darmstadt Noise Dataset (DND) [21] contains pairs of real images with low and high-ISO to address noise and light effects. The images were mostly captured under normal lighting conditions and cannot be used in our setting. The RENOIR dataset [3] aims to propose a benchmark for noisy images but includes spatial misalignment. The Smartphone Image Denoising Dataset (SIDD) [2] introduces a large collection of ground truth for noisy real images. Their dataset does not offer outdoor scenes with extremely low-light and corresponding ground truth and mostly includes a fixed combination of parameters suited for denoising. The Google HDR+ dataset [11] uses bursts of images to increase dynamic range and reduce noise but has not been captured in low-light settings. [28] introduced low-light paired dataset for underexposure enhancement. The most relevant dataset was introduced by Chen [7], which includes raw sensor data to address extreme low-light imaging. [29] prepared real noisy low-light and ground truth dataset based on [7] for sRGB images. Unlike existing datasets, we introduce long-exposure reference image with multiple exposure times for each extreme low-light scene, in both indoor and outdoor scenes, and directly operate on the raw sensor data.

3. Our Approach

3.1. Multi-Exposure Extreme Low-Light Dataset

We collected a total of 1500 images. In order to capture a variety of realistic low-light conditions and cover a broad range of scenes with extreme low-light conditions, the images were captured in both indoor and outdoor scenes. The images were captured over different days in multiple locations. We captured five different exposures for each of the scenes - 0.1s, 0.5s, 1s, 5s and 10s. The outdoor images were captured late at night under moonlight or street lighting. The indoor images were captured in closed rooms with indirect illumination. Generally, the lowest exposure image in both indoor and outdoor scenes is completely dark and no details of the scene can be observed.
Figure 4: The input to the baseline network is Bayer sensor packed into four channels, each half the height and width of the original color filter array. The preprocessing includes dark level subtraction and multiplication by the ratio between the exposure time of the input image and the reference output image.

All the scenes in the dataset are static to accommodate the long exposure. For each scene, the settings of the camera such as ISO and f-number were adjusted to optimize the longest-exposure image. We used a tripod and mirrorless camera in order to capture the exact same scene without any misalignment. At each scene, after the long exposure image was optimally captured, we used a smartphone application to decrease the exposure and capture the images without touching the camera or changing the camera’s parameters. After capturing the images, we manually verified that the images are aligned and the long-exposure reference images are with high perceptual quality.

The images were captured using Sony α5100 with Bayer sensor. The resolution of the images is 6000 × 4000. Tables 1,2 summarize the dataset properties.

3.2. The Model

Given an input image $I_{in}$ with exposure level $in$, our goal is to develop a deep learning model that can output $I_{out}$ with exposure level between initial and final levels $out_1 \leq out \leq out_2$. A typical approach [12, 24] is to train a basic model to restore an initial corruption level, then add modulation modules to create an adaptive model and fine-tune it to restore the final level. At inference time, the weights of filters in the modulation modules are continuously interpolated based on an input parameter in order to set the desired restoration level.

3.3. Base Network

The basic model is U-Net [23, 7], which replaces the entire image signal processing (ISP) pipeline. The input is a short exposure raw image from Bayer sensor data and the output is an sRGB image. The raw Bayer sensor data is packed into four channels, the spatial resolution is reduced by a factor of two in each dimension and the black level is subtracted. The output is a 12-channel image processed to recover the original resolution of the input image. Fig. 4 shows the input to the network.

The input arrays’ values are multiplied by amplification ratio which represents the ratio between the input image’s exposure time and the required output image’s exposure time, effectively determining the brightness of the output. In practice, the maximum exposure ratio is truncated by 300 and the output is clipped to [0,1]. Thus, the base network can also be regarded as a tunable network, with a direct adaptation of the exposure.

3.4. Modulation Module

Training our base model with two different output exposure levels, $out_1, out_2$, yields different weights for the learned filters in each network. Denote the weights of a given learned filter of the network trained on exposure level $out_2$ as $f_{out_2}$, and correspondingly $f_{out_1}$ for the same network trained for exposure level $out_1$. We employ a modulation block to approximate $f_{out_2}$ based on $f_{out_1}$ and allowing modulation across these two. For each feature map $X$, our objective can be written as:

$$\min_G ||f_{out_2} \ast X - G(f_{out_1}) \ast X||^2$$

where $\ast$ is the convolution operation and $G$ is the modulation block. In other image restoration tasks [12], the filter $f_{out_2}$ is approximated by convolving a kernel $f_g$ with $f_{out_1}$. Following the associativity of convolution $f_g \ast (f_{out_1} \ast X) = (f_g \ast f_{out_1}) \ast X$, $G$ is represented by $f_g$. We use a convolutional kernel of size $3 \times 3$ as our modulation module in our implementation. The modulation block is formulated as $f_l, b_l$

$$f_l \ast X_l + b_l$$

added for each existing feature map $X_l$ in our model. Fig. 5 shows the modulation module. Other alternatives are using non-linear convolutional block [19] and batch normalization (BN) [17], but we did not find these more beneficial in our setting. Fig. 6 shows the U-Net blocks with and without the modulation modules.

3.5. Adaptive Network Training

The training procedure of the adaptive network follows two steps. First, the base model is trained to fit the initial exposure level $e_i$, without any additional modification to the existing architecture. Then we fix the weights of the base model, and the modulation blocks are inserted after each existing convolutional kernel to form the adaptive network. The adpative network is then fine-tuned to fit the final exposure level $e_f$ by learning the weights of the additional convolutional kernels.

3.6. Adaptive Network Testing

Our goal is to set the exposure level of the adaptive network to any given exposure $e$, $e_i \leq e \leq e_f$. Let $f_{e_f}, b_{e_f}$ denote the weights of the filter and bias in the modulation module after the fine-tuning phase. Note that as the base network weights are fixed, by setting the filters in the modulation modules to identity and the biases to zero the network
Table 3: Results for tuning the output exposure. The models are trained on input images with exposure of 0.1s, output images with various outputs, and are tested on target images with exposure of 1s, 5s, and 10s. ⇒ denotes full training and → denotes fine-tuning based on the fixed weights of the initial training using the modulation block. The bold are the two best results. For unseen exposure levels at inference time, our model’s interpolation achieves better accuracy than extrapolation, and both outperform other alternatives.

\[
\begin{array}{|c|c|c||c|c|c||c|c|c|}
\hline
\text{Baseline - Single Exposure} & \text{Train} & \text{Test Output Exposure} & \text{1s PSNR} & \text{SSIM} & \text{5s PSNR} & \text{SSIM} & \text{10s PSNR} & \text{SSIM} \\
\hline
0.1s ⇒ 1s & \text{Baseline - Single Exposure} & 0.3817 & 0.95 & 30.7 & 0.87 & 27.7 & 0.84 \\
0.1s ⇒ 5s & & 36.82 & 0.94 & 33.35 & 0.91 & 28 & 0.86 \\
0.1s ⇒ 10s & & 34.88 & 0.9 & 30.52 & 0.88 & 30 & 0.88 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c||c|c|c||c|c|c|}
\hline
\text{Baseline - Multiple Exposures} & \text{Train} & \text{Test Output Exposure} & \text{1s PSNR} & \text{SSIM} & \text{5s PSNR} & \text{SSIM} & \text{10s PSNR} & \text{SSIM} \\
\hline
0.1s ⇒ 1s,10s & \text{Baseline - Multiple Exposures} & 30.1 & 0.82 & 28.56 & 0.83 & 27.71 & 0.85 \\
0.1s,0.5s,1s ⇒ 1s,5s,10s & & 35.77 & 0.92 & 29.55 & 0.86 & 26.25 & 0.82 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c||c|c|c||c|c|c|}
\hline
\text{CE-Net (Ours), Extrapolation} & \text{Train} & \text{Test Output Exposure} & \text{1s PSNR} & \text{SSIM} & \text{5s PSNR} & \text{SSIM} & \text{10s PSNR} & \text{SSIM} \\
\hline
0.1s ⇒ 1s,5s → 5s & CE-Net (Ours), Extrapolation & 38.17 & 0.95 & 31.78 & 0.89 & 28.65 & 0.86 \\
0.1s ⇒ 1s → 10s & & 38.17 & 0.95 & 32.35 & 0.89 & 29.67 & 0.87 \\
\hline
\end{array}
\]

4. Implementation Details

4.1. Our Setting

We implemented our network using Pytorch [20]. For training, validation and test sets we used 70%, 10% and 20% of the images respectively, with uniform sampling and equal representation for indoor and outdoor scenes in each set. The models were trained on multiple NVidia 2080Ti GPUs. The base model is trained with L1 and the Adam optimizer. The ground truth images are the corresponding long-exposure images processed by LibRaw[1] to sRGB format. During training, the input are random 512x512 patches which are cropped, rotated and flipped. The learning rate is $10^{-4}$ for 1000 epochs and then $10^{-5}$ for additional 1000 epochs, a total of 2000 epochs for the training phase. Fine tuning the model for the final exposure level is by additional 1000 epochs.

4.2. Ablation Study

\textbf{Filter Size.} We compared different filter sizes in the modulation module. We considered various filter sizes –
Figure 7: The results of tuning the output exposure of an image for 5s. The input is a dark image where no details can be observed. Top left: the input image. Top right: the output of the baseline, trained for all possible input and output exposures (0.1s,0.5s,1s to 1s,5s,10s). Bottom left: the output of our model trained from 0.1s to 1s and fine tuned to 10s. Bottom right: the ground truth image.

| 1 × 1 | 3 × 3 | 5 × 5 | 7 × 7 |
|-------|-------|-------|-------|
| 31.87 | 32.35 | 32.39 | 32.48 |

Table 4: Filter size comparisons. Model is trained from 0.1s to 1s and fine tuned to 10s, and tested for unseen exposure level of 5s.

Tuning Direction. We tested the optimal direction for the tuning. We compared two models. The first one is trained from 0.1s to 1s and is fine tuned for 10s. The second one is trained from 0.1s to 10s and fine tuned for 1s. We compare the results of the fine tuning process with respect to an unseen output image with exposure time of 5s. Forward direction from 0.1s to 10s achieved better results than the backward one, with PSNR of 32.17 vs. 28.2.

5. Experiments

We evaluate two cases. The first one is tuning the model to adjust the exposure of the output image to an unseen one. The second one is tuning the model to adjust the exposure of the input image to an unseen one. For each case we consider both interpolation and extrapolation tests. Our model is trained on initial and final exposure levels. For interpolation test, the model generalizes to an unseen exposure level which is between the initial and final levels. For extrapolation test, the model generalizes to an unseen exposure level which is either more or less than the initial and final levels.

In all experiments, the baseline is the SID [7] model. The SID model enhances raw image data as ours and can tune the brightness of the output image using an input parameter.
5.1. Tunable Output Exposure

Interpolation Evaluation. The goal is to test the tuning of the exposure level of the output image by our model for a level between the initial and final exposure levels seen during training. Our tunable model is trained on input images with exposure of 0.1s and ground truth images with exposure of 1s, and is fine tuned to ground truth images with exposure of 10s. The test is the ability to generalize to an output image with unseen exposure time of 5s.

We compare the following baselines:

- The SID model trained for output exposure of 1s and 10s. This setting is comparable to ours, where both models have not been trained using the target exposure of 5s. Recall that an amplification ratio is fed into the SID model thus such a comparison can be regarded as evaluation of our tuning model vs. a simple tuning parameter.

- The SID model trained for output exposure of 1s, 5s and 10s. In such a setting the baseline is optimized using the target exposure whereas our tunable model has not seen such an exposure level.

- SID trained from 0.1s to 5s. This tests the ability of our model to approximate an optimal fitting of 5s.

Figure 8: The output of our model for unseen output exposure levels. The model is trained based on input images with exposure time 0.1s and ground truth images with exposure time of 1s. It is fine tuned for ground truth images with exposure time of 10s. Tuning factors are: top left: 0, top right: 0.2, bottom left: 0.6 and bottom right: 1. The input is a dark image.
Table 5: Results for tuning the input exposure. The models are trained on input images with various exposures and are tested on input images with exposure of 0.1s, 0.5s and 1s and output images with exposure of 10s. ⇒ denotes full training and → denotes fine-tuning based on the fixed weights of the initial training using the modulation block. The bold are the two best results.

| Baseline - Single Exposure | Train / Test Input Exposure | 0.1s PSNR SSIM | 0.5s PSNR SSIM | 1s PSNR SSIM |
|----------------------------|-----------------------------|---------------|---------------|--------|
| 0.1s ⇒ 10s                | 30 0.88                     | 24.76 0.85    | 25 0.86       |
| 0.5s ⇒ 10s                | 25.54 0.8                    | **31.37 0.91** | **31.74 0.91** |
| 1s ⇒ 10s                  | 25.8 0.81                    | 31.27 0.9     | **31.36 0.91** |
| Baseline - Multiple Exposures | 0.1s, 0.5s, 1s ⇒ 1s, 5s, 10s | **35.77 0.92** | 29.55 0.86    | 26.25 0.82   |
| CE-Net (Ours), Extrapolation | 1s ⇒ 10s → 0.5s          | **31.11 0.89** | **31.34 0.9**  | **31.36 0.91** |
| CE-Net (Ours), Interpolation | 0.1s ⇒ 10s → 1s           | 30 0.88       | 27.75 0.88    | 29.77 0.89   |

- SID trained from 0.1s to 1s, and from 0.1s to 10s. This tests the ability to tune the SID model based on a single exposure level.

**Extrapolation Evaluation.** The goal is to test the tuning of our model for an exposure level above the initial and final exposure levels seen during training. Our tunable model is trained on input images with exposure of 0.1s and ground truth images with exposure of 1s, and is fine tuned to ground truth images with exposure of 5s. The test is the ability to generalize to an unseen exposure time of 10s. Similar to above, we tested against the baseline model trained for a single output exposure, two output exposures and trained on all possible input and output exposures.

Table 3 presents the results of our output exposure tuning. For baseline model trained on a single exposure level, our model, in both interpolation and extrapolation tests outperformed the baseline. This result might be expected, as the baseline model has not been trained for various output exposure levels. For baseline trained on multiple output and multiple input exposure levels, including the tested exposure level, our model still achieves better accuracy.

Fig. 7 shows the results of tuning the output exposure of an image for 5s. Fig. 8 presents the output of our model for unseen output exposure levels.

### 5.2. Tunable Input Exposure

We test the ability to tune the exposure of the input image for various levels. Note that the metrics are evaluated on the output image which is always with an exposure of 10s.

**Interpolation Evaluation.** The goal is to test the tuning of the exposure of the input image by our model for a level between the initial and final exposure levels of the input images seen during training. Our tunable model is trained on input images with exposure of 0.1s and ground truth images with exposure of 10s, and is fine tuned to ground truth images with exposure of 1s. The test is the ability to generalize to an input image with unseen exposure time of 0.5s.

**Extrapolation Evaluation.** We train our base model for input exposure of 1s and ground truth images with exposure of 10s, and fine tune it to 0.5s. The test is the ability of our model to generalize to unseen input exposure of 0.1s. Baselines are as above.

**Baselines.** The base model is trained on input images with exposures of 0.1s, 0.5s or 1s and ground truth images with exposure of 10s. For baseline with multiple exposures, it is trained with input images of 0.1s, 0.5s and 1s and output images of 1s, 5s, and 10s.

Table 5 shows the results of tuning the input exposure.

### 6. Conclusion

Extreme low light imaging is challenging and has recently gained a growing interest. Current methods allow enhancement of dark images, assuming the input exposure and the optimal output exposure are known at inference time, which prevents its adaptation in practical scenarios. We collected a dataset of 1500 images with multiple exposure levels for extreme low light imaging. Based on our dataset, we presented an approach that enables continuously controlling the exposure levels for both the input and output of the images at runtime, without the need to retrain the model. We showed that our model presents promising results on a wide range of both indoor and outdoor images. We believe that our dataset as well as our tunable model will support further research in the field of tunable low-light imaging, making a step forward towards its widespread adoption.
A. Supplementary Material

Figure 9: The architecture of our network. There are two input parameters, $\alpha_1$ is the amplification ratio and $\alpha_2$ is the tuning factor. $\alpha_1$ sets the brightness of the raw input data based on the ratio between the exposure time of the input and output images. $\alpha_2$ modulates the weights of the filters and tunes the network which operates as an Image Signal Processing (ISP) unit. During training, for each value of $\alpha_1$ there is a single value of $\alpha_2$. For initial and final exposure levels, $\alpha_1$ changes according to the exposure ratios and $\alpha_2$ changes between zero and one accordingly. At inference time each parameter can be set independently of the other. See next Figure.
Figure 10: The effect of the amplification ratio and the tuning factor. The input is a dark image with exposure of 0.1s. The model is trained based on input images with exposure time 0.1s and ground truth images with exposure time of 1s. It is fine-tuned for ground truth images with exposure time of 10s. The amplification and the tuning factor presented here are with respect to the input image. For an amplification ratio not presented in the training set, such as 50, the ability to tune the network improves the enhancement of the dark image.
| Layer (type)  | Output Shape       | Param #  |
|--------------|--------------------|----------|
| Conv2d-1     | [-1, 32, 512, 512] | 1,184    |
| Modulation-1 | [-1, 32, 512, 512] | 320      |
| Conv2d-2     | [-1, 32, 512, 512] | 9,248    |
| Modulation-2 | [-1, 32, 512, 512] | 320      |
| Conv2d-3     | [-1, 64, 256, 256] | 18,496   |
| Modulation-3 | [-1, 64, 256, 256] | 640      |
| Conv2d-4     | [-1, 64, 256, 256] | 36,928   |
| Modulation-4 | [-1, 64, 256, 256] | 640      |
| Conv2d-5     | [-1, 128, 128, 128]| 73,856   |
| Modulation-5 | [-1, 128, 128, 128]| 1,280    |
| Conv2d-6     | [-1, 128, 128, 128]| 147,584  |
| Modulation-6 | [-1, 128, 128, 128]| 1,280    |
| Conv2d-7     | [-1, 256, 64, 64]  | 295,168  |
| Modulation-7 | [-1, 256, 64, 64]  | 2,560    |
| Conv2d-8     | [-1, 256, 64, 64]  | 590,080  |
| Modulation-8 | [-1, 256, 64, 64]  | 2,560    |
| Conv2d-9     | [-1, 512, 32, 32]  | 1,180,160|
| Modulation-9 | [-1, 512, 32, 32]  | 5,120    |
| Conv2d-10    | [-1, 512, 32, 32]  | 2,359,808|
| Modulation-10| [-1, 512, 32, 32]  | 5,120    |
| ConvTranspose2d-1 | [-1, 256, 64, 64] | 524,544  |
| Conv2d-11    | [-1, 256, 64, 64]  | 1,179,904|
| Modulation-11| [-1, 256, 64, 64]  | 2,560    |
| Conv2d-12    | [-1, 256, 64, 64]  | 590,080  |
| Modulation-12| [-1, 256, 64, 64]  | 2,560    |
| ConvTranspose2d-2 | [-1, 128, 128, 128] | 131,200 |
| Conv2d-13    | [-1, 128, 128, 128]| 295,040  |
| Modulation-13| [-1, 128, 128, 128]| 1,280    |
| Conv2d-14    | [-1, 128, 128, 128]| 147,584  |
| Modulation-14| [-1, 128, 128, 128]| 1,280    |
| ConvTranspose2d-3 | [-1, 64, 256, 256] | 32,832  |
| Conv2d-15    | [-1, 64, 256, 256]  | 73,792   |
| Modulation-15| [-1, 64, 256, 256]  | 640      |
| Conv2d-16    | [-1, 64, 256, 256]  | 36,928   |
| Modulation-16| [-1, 64, 256, 256]  | 640      |
| ConvTranspose2d-4 | [-1, 32, 512, 512] | 8,224    |
| Conv2d-17    | [-1, 32, 512, 512]  | 18,464   |
| Modulation-17| [-1, 32, 512, 512]  | 320      |
| Conv2d-18    | [-1, 32, 512, 512]  | 9,248    |
| Modulation-18| [-1, 32, 512, 512]  | 320      |
| Conv2d-19    | [-1, 12, 512, 512]  | 396      |
| Modulation-19| [-1, 12, 512, 512]  | 120      |

**Figure 11:** The architecture of our network. There is a total of 7,790,308 parameters. Modulation blocks’ parameters are 29,560, only 0.0037 of the total parameters.
References

[1] LibRaw Project Goals and Objectives, 2018.
[2] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
[3] Josue Anaya and Adrian Barbu. Renoir—a dataset for real low-light image noise reduction. Journal of Visual Communication and Image Representation, 51:144–154, 2018.
[4] Bolun Cai, Xianming Xu, Kailing Guo, Kui Jia, Bin Hu, and Dacheng Tao. A joint intrinsic-extrinsic prior model for retinex. In Proceedings of the IEEE International Conference on Computer Vision, pages 4000–4009, 2017.
[5] Jianrui Cai, Shuhang Gu, and Lei Zhang. Learning a deep single image contrast enhancer from multi-exposure images. IEEE Transactions on Image Processing, 27(4):2049–2062, 2018.
[6] Turgay Celik and Tardi Tjahjadi. Contextual and variational contrast enhancement. IEEE Transactions on Image Processing, 20(12):3431–3441, 2011.
[7] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3291–3300, 2018.
[8] Xueyang Fu, Delu Zeng, Yue Huang, Xiao-Ping Zhang, and Xinghao Ding. A weighted variational model for simultaneous reflectance and illumination estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2782–2790, 2016.
[9] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. ACM Transactions on Graphics (TOG), 36(4):1–12, 2017.
[10] Xiaojie Guo, Yu Li, and Haibin Ling. Lime: Low-light image enhancement via illumination map estimation. IEEE Transactions on imaging processing, 26(2):982–993, 2016.
[11] Samuel W Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. ACM Transactions on Graphics (TOG), 35(6):1–12, 2016.
[12] Jingwen He, Chao Dong, and Yu Qiao. Modulating image restoration with continual levels via adaptive feature modification layers. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 11056–11064, 2019.
[13] Jingwen He, Chao Dong, and Yu Qiao. Multi-dimension modulation for image restoration with dynamic controllable residual learning. arXiv preprint arXiv:1912.05293, 2019.
[14] Zhe Hu, Sunghyun Cho, Jue Wang, and Ming-Hsuan Yang. Deblurring low-light images with light streaks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3382–3389, 2014.
[15] Sung Ju Hwang, Ashish Kapoor, and Sing Bing Kang. Context-based automatic local image enhancement. In European conference on computer vision, pages 569–582. Springer, 2012.
[16] Andrey Ignatov, Nikolay Kobychev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Dslr-quality photos on mobile devices with deep convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, pages 3277–3285, 2017.
[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.
[18] Chulwoo Lee, Chul Lee, and Chang-Su Kim. Contrast enhancement based on layered difference representation of 2d histograms. IEEE transactions on image processing, 22(12):5372–5384, 2013.
[19] Hyeongmin Lee, Taeho Kim, Hanbin Son, Sangwook Baek, Minsu Cheon, and Sangyoun Lee. Smoother network tuning and interpolation for continuous-level image processing. arXiv preprint arXiv:2010.02270, 2020.
[20] Adam Paszke, Sam Gross, Christoph Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc., 2019.
[21] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1586–1595, 2017.
[22] Tal Remez, Or Litany, Raja Giryes, and Alex M Bronstein. Deep convolutional denoising of low-light images. arXiv preprint arXiv:1701.01687, 2017.
[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-aided intervention, pages 234–241. Springer, 2015.
[24] Alon Shoshan, Roey Mechrez, and Lihi Zelnik-Manor. Dynamic-net: Tuning the objective without re-training for synthesis tasks. In Proceedings of the IEEE International Conference on Computer Vision, pages 3215–3223, 2019.
[25] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6849–6857, 2019.
[26] Wei Wang, Ruiming Guo, Yapeng Tian, and Wenming Yang. Cfsnet: Toward a controllable feature space for image restoration. In Proceedings of the IEEE International Conference on Computer Vision, pages 4140–4149, 2019.
[27] Xintao Wang, Ke Yu, Chao Dong, Xiaou Tang, and Chen Change Loy. Deep network interpolation for continuous imagery effect transition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1692–1701, 2019.
[28] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. *arXiv preprint arXiv:1808.04560*, 2018. 2, 3

[29] Ke Xu, Xin Yang, Baocai Yin, and Rynson WH Lau. Learning to restore low-light images via decomposition-and-enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2281–2290, 2020. 1, 2, 3

[30] Zhicheng Yan, Hao Zhang, Baoyuan Wang, Sylvain Paris, and Yizhou Yu. Automatic photo adjustment using deep neural networks. *ACM Transactions on Graphics (TOG)*, 35(2):1–15, 2016. 2

[31] Zhenqiang Ying, Ge Li, and Wen Gao. A bio-inspired multi-exposure fusion framework for low-light image enhancement. *arXiv preprint arXiv:1711.00591*, 2017. 2

[32] Lu Yuan and Jian Sun. Automatic exposure correction of consumer photographs. In *European Conference on Computer Vision*, pages 771–785. Springer, 2012. 1

[33] Xiangdong Zhang, Peiyi Shen, Lingli Luo, Liang Zhang, and Juan Song. Enhancement and noise reduction of very low light level images. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 2034–2037. Ieee, 2012. 1

[34] Yonghua Zhang, Jiawan Zhang, and Xiaojie Guo. Kindling the darkness: A practical low-light image enhancer. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1632–1640, 2019. 2