Comprehensive Review and Implementation of Novel Amplification Factors in Low-Light Denoising

Anonymous Author [1]

Abstract

This paper aims to investigate and review the limits and real-world applications of (Chen et al., 2018) denoising pipeline method. Specifically, I aim to test (Chen et al., 2018) pipeline performances for ultra-fast exposure times and different ISO values. Using Canon EOS 850D and Nikon D600, I built a low-light image dataset with a variety of different scenarios and camera settings to test the cited performance. Many methods, including (Chen et al., 2018), use the ground truth (GT) exposure to calculate their pre-amplification value. This pre-amplification value is an important step in low-light image enhancement and denoising for both training and evaluation of the pipelines. I plan to measure the real-performance difference between GT amplification (GT amp) factors and (Lamba & Mitra, 2021) ground-truth independent amplification factor (Lamba amp). Through my research, I find that (Lamba & Mitra, 2021) pre-amplification factor has a profound impact on ultra-low light images and showcases the diminishing returns of the GT Amplification factor. I also find an interesting inherent difference in the pipeline performance of camera models with different black levels.

1. Introduction

Light is an essential component in image formation. Due to the limited photon count in low-light scenarios, it can be challenging to create an accurate or usable image of the scene. Modeling noise allows us to understand and predict the statistical properties of the noise in an image. Reversing the modeled process removes the noise from an image while preserving the underlying information. Creating efficient and accurate noise models is crucial for nighttime photography on mobile phones, improving surveillance image quality, and improving the performance of autonomous vehicles or drones.

Learning-based low-light image enhancement methods have become the most popular due to their modularity and improved performance over more traditional analytical methods. (Chen et al., 2018) paper is instrumental because it showcased the improved performance of learning-based methods and the importance of having real-world training image datasets, instead of artificial adding in Gaussian or salt & pepper noise to existing clean training data. (Wei et al., 2020) expanded (Chen et al., 2018) by adding internal CMOS noise to the existing real-world training dataset using the Photon Transfer Method to obtain the empirical CMOS noise parameters. In real-world scenarios, GT exposure is unknown to the user and renders GT pre-amplification (GT amp) pipelines unusable in real-world scenarios. (Lamba & Mitra, 2021) approaches a more practical implementation of (Chen et al., 2018) learning-based methods, by simplifying the learning model features and devising a GT-independent pre-amplification factor.

All of these papers used state-of-the-art low-light cameras such as the Sony α7S II and Nikon D850. This means their cited performance is supported by the best-case camera hardware. I plan to reimplement (Chen et al., 2018) pipeline and (Lamba & Mitra, 2021) amplification factor to test their performance on my own cameras and measure the difference from their reported tested results.

Low photon count scenarios require a longer shutter speed to create a properly exposed image; a longer time span allows more photons to interact with the image sensor and create a cleaner image. The longer shutter time induces a higher likelihood of blur created by the movement of the camera or other objects in the image. To reduce the dependence on longer shutter speeds, I want to test (Chen et al., 2018) pipeline performance on a wider range of shutter speeds.

ISO factors control how much amplification is added to an original light intensity value stored in the pixel, meaning a higher ISO can allow for a faster shutter speed. ISO and shutter speeds are designed to be linearly dependent: if you increase the ISO from 100 to 200, the same image exposure would appear in half the shutter speed time. Since ISO is an artificial amplification of the light intensity value, it also amplifies the existing noise in the image. I also aim to explore the impact ISO value has on both (Chen et al., 2018) and (Lamba & Mitra, 2021) amplification value.

2. Review of Paper 1

2.1. Storyline

High-Level Motivation Learning-based approaches to image denoising have shown their ability to adapt to different sensors, cameras, and environments. Their major drawback was their poor performance on tested real-world low-light images. (Chen et al., 2018) showed that by using real-world training data, learning methods could outperform existing analytical solutions and paired data methods. This paper provided a new low-light image dataset (SID Dataset) to train and test future learning-based low-light image denoising methods.

Prior Research A variety of mathematical approaches have been used to perform low-light image enhancement, such as histogram equalization or gamma correction. Histogram Equalization simply balances the histogram of an entire image and gamma correction increases the brightness of the dark region while compressing bright pixels. More advanced models include dark channel prior, the wavelet transform, the Retinex model, and illumination map estimation. Many of these methods are used in traditional image processing pipelines.

In image denoising there have been a variety of techniques for single-image processing, including total variation, wavelet-domain processing, sparse coding, nuclear norm minimization, and 3D transform-domain filtering (BM3D). The Block-matching and 3D-Filtering method (BM3D), where blocks of the image are clustered together (representing different objects) and noise is filtered out has been a prevalent non-deep network model for image denoising.(Dabov et al., 2007) Recently, deep networks have been applied to denoising due to their nonlinear modeling properties. Methods such as stacked sparse denoising auto-encoders (SSDA), trainable nonlinear reaction diffusion (TNRD), multi-layer perceptrons, deep autoencoders, and convolutional networks have been used to model noise.

Finally, multiple-image denoising methods have been utilized, since they have more contextual scene information.

Research gap The significant gap in previous research is the suboptimal performance in real-world noisy images. The mathematical approaches in low-light image enhancement suffer because they assume that the image is an accurate representation of the scene's context. They do not explicitly model image noise and typically apply postprocess off-the-shelf denoising. Using deep network models has actually helped address this issue, but many of the networks are trained on synthetic data created by adding in Gaussian or salt & pepper noise or synthetic bayer filters. This has led to many deep network models to underperform when compared to mathematical methods, such as BM3D. All of these methods underperform significantly in real-world images, because of the stochastic nature of photons in scenes. The RENOIR dataset (Anaya & Barbu, 2018) was the only real-world low-light noisy image dataset, but some photos were spatially misaligned and burst methods were used to create low-noise reference images. This makes it not ideal for training a deep network model as the processing of the image may exhibit spatial misalignment and other non ideal post-processing features.

Contributions (Chen et al., 2018) addresses the issue of creating a real-world dataset to enable further support of data-driven approaches to image processing. Creating a dataset of 5094 raw short-exposure images (high noise, short shutter speed) and corresponding long-exposure reference images (low noise, long shutter speed). Improving upon traditional data pipelines, they used a fully-convolutional network to suppress noise and correctly transform colors in noisy low-light images. The paper showcased the promising results from utilizing deep network modeling on denoising low-light images (.03 - 5 lux).

2.2. Proposed Solution

The SID Dataset was fundamental to proving the point that the underlying data when training denoising models was important to real-world performance. They first started by using tripods, mirrorless cameras, and remote shutter triggers to mitigate the spatial dis-alignment. Next they used low ISO values in the dataset to maximize the impact of the noise from photon arrival and not from the camera. They also used both a Bayer Filter Sensor camera (Sony α7S II) and a X-Trans Sensor camera (Fujifilm X-T2) to collect the dataset. Finally, they used a combination of long shutter speed times for capturing 424 low-noise (reference) images and 5094 short shutter speed times for noisy images.

They first deconstructed the image into its 4 RGGB color channels. Next they trained two different deep network models, a multi-scale context aggregation network (CAN) for fast image processing (Chen et al., 2017) and a U-net (Ronneberger et al., 2015) (Figure 1). Finally the processed sections are joined together to create the final image matrix (Figure 2).

2.3. Claims and Evidence

Claim 1 Most traditional image processing pipelines rely on mathematical approaches and suffer from denoising images represented by inaccurate environments.

Evidence 1 While the traditional image processing pipelines can enhance the intensity of the pixel colors, it fails to accurately model the colors of the image, as shown in Figure 3. This shows how traditional pipelines can cre-

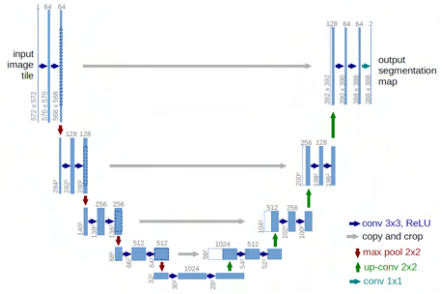

Figure 1. U-Net CNN Model

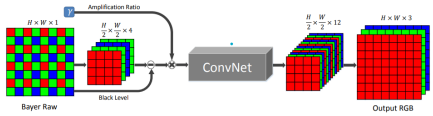

Figure 2. Learning to See in the Dark Pipeline

ate Peak-Signal-To-Noise Ratio (PSNR), but struggle to accurately map the noise pertinent to the image structural information (Luminance, Contrast, Structural).

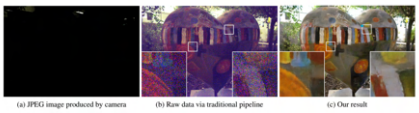

Figure 3. (a) An image captured at night by the Fujifilm X-T2 camera with ISO 800, aperture f/7.1, and exposure of 1/30 second. The illuminance at the camera is approximately 1 lux. (b) Processing the raw data by a traditional pipeline does not effectively handle the noise and color bias in the data. (c) (Chen et al., 2018) result obtained from the same raw data

Claim 2 The Burst Methods still suffer from the drawback of mathematical approaches to noise modeling.

Evidence 2 While the Burst Method is able to accurately model the noise better than BM3D and traditional pipelines as shown by its smoother coloring, the purple hue is still clearly present in the Burst Method image as shown in Figure 4.

Claim 3 Learning to see in the Dark U-net network models suffers to denoise post-processed images.

Evidence 3 Since histogram stretching and other image manipulations are applied in post-processing, it affects the overall accuracy of the model. We can see that in image (b) the white paint has a different color pattern to the rest of the images in Figure 5. This shows how the histogram stretching impacts the similarity of the image.

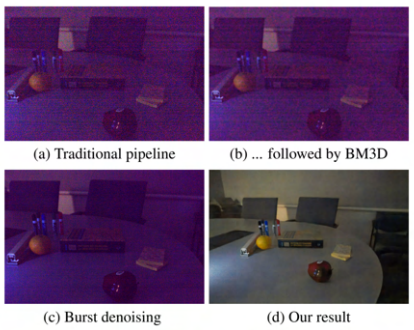

Figure 4. Limited signal recovery in extreme low-light conditions (indoor, dark room, 0.2 lux). (a) An input image in the Sony x300 set, processed by the traditional pipeline and amplified to match the reference. (b) BM3D denoising applied to (a). (c) Burst denoising with 8 images: the result is still bad due to the severe artifacts in all images in the burst. (d) The result of our network; loss of detail is apparent upon close examination.

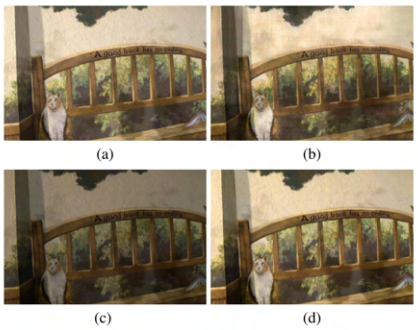

Figure 5. Effect of histogram stretching. (a) A reference image in the Sony x100 set, produced with histogram stretching. (b) Output if trained on histogram-stretched images. The result suffers from artifacts on the wall. (c) Output if trained on images without histogram stretching. The result is darker but cleaner. (d) The image (c) after histogram stretching applied in postprocessing.

2.4. Critique and Discussion

(Chen et al., 2018) clearly showed the importance of using real-world images when training a denoising model. It clearly achieves the claims that data-driven DNN models can far outperform analytical and mathematical methods. The paper leaves many questions about the practicality of this denoising method. First, the performance of the model on different camera devices. Next, is the performance of the model on higher ISO captured images as this model does not take into account different types of noise parameters. Finally, this model still relies on the Ground Truth value, meaning that the noisy image pre-amplification value can only be assessed if you know the original corresponding clean-image exposure value.

3. Review of Paper 2

3.1. Storyline

High-Level Motivation Denoisers are used to enhance the image quality of low-light images (.5 - 5 lux settings). Most practical applications of denoiser models have significantly higher computing constraints than theoretical research applications. (Lamba & Mitra, 2021) explores a variety of optimizations to make the denoiser solution computationally faster and more memory-efficient while trying to preserve restoration quality. Their goal is a denoiser solution that can be used in practical real-world applications.

Prior Research Previous research has focused on three main areas: low-light enhancement, low-light restoration, and lightweight models. A variety of low-light enhancements initially involved using histogram equalization and its variants to increase the dynamic range. These methods were then improved to incorporate illumination and reflectance components using Retinex theory. (Chen et al., 2018) proposed the SID Dataset with poor colors and high noise components using real-life images. His U-Net style encoder-decoder has been used by many for improved low-light restoration. (Wei et al., 2020) used a physics-based CMOS noise formation simulator and synthesized images to increase the amount of training data. Burst shot methods have also been used in low-light image restoration. (Guo et al., 2020) limits the amount of convolutional layers to limit the MAC operations to create a lighter model. (Lamba et al., 2020) uses only a scale-space operating at 16xlower resolution.

Research gap Most existing low-light restorations use (Chen et al., 2018) U-Net style method and thus suffer from huge computational overhead. (Wei et al., 2020) increases the complexity of any existing models by requiring a large calibration dataset to find the specific physic model parameters to create synthetic images. Burst methods require extra treatment to ensure proper alignment and are susceptible to ghosting artifacts. (Guo et al., 2020) still has high latency because most of the processing occurs at lower scales. (Lamba et al., 2020) methods significantly reduce the scale-space but suffer from blurry results due to their lower-resolution inputs. Also, all the learning methods use a Ground Truth (GT) exposure value to calculate the pre-amplification value, and impractical to use as we need to know the ideal exposure value before the image is inputted into the denoisers model. While (Lamba et al., 2020) MLP-based amplifier does not need the GT value, it needs to be retrained with the model and also introduces artifacts in the image.

Contributions To increase the practical application (Lamba & Mitra, 2021) creates a new amplification factor from the inputted raw image without any fine-tuning and does not require the GT exposure value (Formula 1). Next, they created a new learning approach that jumps over intermediate scales to achieve faster and more memory-efficacy denoising solutions. Finally, they used Residual Dense Block (RDB) (Zhang et al., 2018) to improve the resolution.

3.2. Proposed Solution

(Lamba & Mitra, 2021) deep learning approach uses three scale encoders: Lower Scale Encoder (LSE) operates at ½ resolution of the inputted image, Medium Scale Encoder (MSE) at 1/8 resolution, and Higher Scale Encoder (HSE) at 1/32 resolution. Since the HSE requires the most computation, it uses the RDB to improve resolution and restoration while not impacting the time or memory complexity. The execution of all scales improves the computation and uses Fuse Blocks (FB) to fuse the details from all the scales. In order to properly mosaic the single channel intensity values, they use 2,8,32 downsampling factors to match the 2x2 Bayer pattern. Finally, it is trained using L1 loss and MS-SSIM loss between the restored image and the ground-truth image

They originally used Formula 2 to create the amplification factor but found that low-light images had saturated highlights (lights) that introduced halo artifacts and would incorrectly estimate a much lower amplification factor. Instead, they assigned an adjustable weight value w_i,j that exponentially reduces from 1 to 0 for large light intensities (Formula 3). It also used b_k to create finer bins for lower intensities and coarser bins for larger intensities (Formula 4).

3.3. Claim-Evidence

$$Lamba\ Amp = m * (\frac{\sum_{i,j} x_{i,j} * w_{i,j}}{\sum_{i,j} w_{i,j}})^{-1} \tag{1}$$

$$Amp = m * (\frac{\sum_{i,j} x_{i,j}}{H * W})^{-1} \tag{2}$$

$$w_{i,j} = \frac{2^{\frac{(n-k+1)*8}{n}}}{2^8}\ if\ b_{k-1} < x_{i,j} \leq b_k \tag{3}$$

$$b_k = \frac{2^{\frac{k*8}{n}}}{2^8}\ \forall k \in [1, n]\ and\ b_0 = 0 \tag{4}$$

3.4. Claims and Evidence

Claim 1 Their denoiser method is faster and more memory-efficient than other state-of-the-art methods while maintaining competitive restoration quality.

Evidence 1 Figure 6 shows that the proposed method has fewer parameters, lower MAC operations, lower memory usage, and faster inference speed than six recent methods on both Bayer and X-Trans sensors. The proposed method also has similar or better perceptual scores (Ma and NIQE) than the competing methods.

		SID [10]	DID [44]	SGN [18]	LLPackNet [34]	DCE [19]	LDC [65]	Ours
Parameters *(in million* ↓)		7.7	2.5	3.5	1.16	0.79	8.6	**0.78**
GMACs	**Bayer**	562.06	>2000	>2000	83.46	361	>2000	**59.8**
(↓)	**X-Trans**	1118.8	>2000	>2000	166.12	719.26	>2000	**119.03**
Memory	**Bayer**	6.184	10.72	9.3	2.23	3.58	24.40	**1.16**
***(in GB* ↓)**	**X-Trans**	11.27	19.12	18.21	4.94	8.13	>30	**2.48**
GPU inference	**Bayer**	156.85	>1000	867.50	40.17	116.21	>1000	**30.37**
time *(in ms* ↓)	**X-Trans**	303.64	>1000	>1000	78.94	225.13	>1000	**59.42**
CPU inference	**Bayer**	9.35	110.29	35.20	2.75	6.20	>200	**1.17**
time *(in sec.* ↓)	**X-Trans**	18.59	>200	68.94	4.79	12.32	>200	**2.38**
PSNR / SSIM	**Bayer**	28.88/0.787	28.41/0.780	28.91/0.789	27.83/0.75	26.53/0.73	**29.56/0.799**	28.66/0.790
***(in dB* ↑/↑)**	**X-Trans**	26.61/0.680	–	**26.90/0.683**	24.13/0.59	–	26.70/0.681	26.60/0.682
NIQE / Ma	**Bayer**	**4.39 / 6.93**	4.52/ 6.85	4.40 / 6.92	5.12/ 5.98	4.64 /6.09	4.40 / 6.91	4.41/6.92
(↓ / ↑)	**X-Trans**	**4.45 / 6.78**	–	4.46 / **6.78**	5.20 / 5.97	–	4.47 / 6.77	4.46 / 6.77

Figure 6. The best result is in bold and the second best is underlined. GT exposure has been used for pre-amplification.

Claim 2 Their denoiser method can be used with different cameras and tasks without any fine-tuning camera-specific parameters.

Evidence 2 Fig 7 shows that the proposed method achieves better PSNR/SSIM/Ma/NIQE scores than three state-of-the-art methods [(Chen et al., 2018), (Xu et al., 2020), (Gu et al., 2019)] when trained on a Sony α7S II camera and tested on images captured by Canon EOS 70D, Canon EOS 700D, and Nikon D850 cameras. Table 4 shows that the proposed method achieves better mean Average Precision (mAP) than two state-of-the-art methods [(Chen et al., 2018), (Gu et al., 2019)] when used with (Shih et al., 2019) for object detection on the SID Sony dataset.

		SID [10]	SGN [18]	LDC [65]	Ours
Canon	**PSNR/SSIM**	24.80/0.51	23.75/0.45	22.65/0.40	**24.90/0.53**
EOS 70D	**Ma/NIQE**	6.84/4.32	6.70/4.35	6.07/4.93	**6.85/4.31**
	GMACs	925.64	>3000	>3000	**98.48**
Canon	**PSNR/SSIM**	25.32/0.51	25.09/0.48	24.21/0.41	**25.33/0.52**
EOS 700D	**Ma/NIQE**	6.85/4.25	6.72/4.28	6.74/4.31	**6.87/4.23**
	GMACs	830.77	>3000	>3000	**88.38**
Nikon	**PSNR/SSIM**	26.01/0.60	25.88/0.57	25.18/0.51	**26.27/0.63**
D850	**Ma/NIQE**	6.88/4.61	6.85/4.63	6.79/4.68	**6.90/4.60**
	GMACs	912.16	>3000	>3000	**97.04**

Figure 7. Training on Sony α7S II and testing on images captured using different cameras.

Claim 3 Their amplification denoiser method does not introduce artifacts in the image.

Evidence 3 GT-GC is an image extraction feature that uses shift phases, correct baselines, and arithmetic operations. GT-GC changes the structure of the photo and is it a good test to see if the amplifier introduces artifacts? Looking at Figure 8 we can see that the GT-GC column has a 2-3dB improvement from GT. This shows the amplifier only induces a different global brightness and doesn't introduce artifacts.

Methods	**PSNR ↑ / SSIM ↑**		**Ma ↑ / NIQE ↓**
NR: No Retrain, R: Retrain	**GT**	**GT-GC**	**No reference**
LLPackNet + [34] amp. + R	23.27/0.69	23.43/0.71	5.83 / 5.50
LLPackNet + Our amp. + NR	**23.35/0.71**	**24.34/0.74**	**5.97 / 5.12**
SID + [34] amp. + R	22.98/0.71	23.17/0.72	6.21 / 4.90
SID + Our amp. + NR	**23.84/0.74**	**26.13/0.78**	**6.94 / 4.39**
Ours + [34] amp. + R	23.30/0.71	23.80/0.72	6.11 / 4.62
Ours + Our amp. + NR	**23.85/0.75**	**26.08/0.79**	**6.93 / 4.40**
DID + Our amp. + NR	23.29/0.71	25.64/0.74	6.84 / 4.49
SGN + Our amp. + NR	23.90/0.73	26.12/0.77	6.94 / 4.40
DCE + Our amp. + NR	22.72/0.70	25.13/0.72	6.08 / 4.65
LDC + Our amp. + NR	23.86/0.75	26.10/0.79	6.92 / 4.40

Figure 8. Quantitative superiority of our amplifier module over the MLP based amplifier (Lamba et al., 2020)

3.5. Critique and Discussion

It is clear from the testing results that the new deep learning architecture is computationally fast and more memory efficient. It also demonstrates that the model is able to be used on different camera devices better than the (Chen et al., 2018) method on different cameras. The paper was unclear on how the model is able to generalize so well on different camera devices and I would have liked to see more reasons for how the model is able to achieve such high results without any calibration techniques. They should set up their model against physic-based learning methods which specifically tune their noise simulators to different camera noise. Their new amplification formula experiment setup makes sense, I would like to see further testing on the image on uniformly lit images to reduce the impact of the exponential weight adjuster.

4. Review of Paper 3

4.1. Storyline

High-Level Motivation Low-light places have very demanding constraints on photography due to limited photon count and inescapable noise. The paper is trying to create a denoise learning base method to enhance low-light raw images. It aims to improve the performance of learning-based methods by creating realistic synthetic data to train the deep neural network. Its purpose is to create a physics-based noise model that covers both internal CMOS characteristic-induced noise and also photon-shot noise. The goal is to achieve a better denoiser model by increasing the number of image pairs in the dataset.

Prior Research In the past, most single-image low-light denoisers were based on the assumption that the signal & noise exhibit statistical regularities and could be separated from a single observation. Most modern single-image denoisers are data-driven and use a DNN to learn the statistical regularities to create clean images from their noisy counterparts. Many of these data-driven methods use synthetic image data generated by additive Gaussian and white noise models. Other methods have collected large datasets of paired real-world data to train the model. Finally, a large number of new methods have tried to integrate physics-based models to generate data by trying to model the arrival statistics (shot noise) and sensor read-out statistics (read noise). [(Mildenhall et al., 2018),(Brooks et al., 2019)] have used signal-dependent heteroscedastic Gaussian models to characterize the noise. While others have used generative models to produce synthetic data.

Research gap Synthetic image datasets using additive white or Gaussian noise fail to create realistic evaluation scenarios and fail when testing on real data. Real-paired datasets with ground-truth labels to stop overfitting are extremely time-consuming and expensive to create. Physics-based synthetic image dataset methods achieve their purpose of creating more images to train, but most don't incorporate noise inherent to the image sensor. Heteroscedastic Gaussian noise is only effective in modeling noise in daylight or moderate low-light settings. Generative models are not effective in modeling CMOS image sensor characteristics and struggle to create accurate synthetic noisy images.

This leads to noise models that struggle to simulate realistic extreme low-light or low-light images. Since these models are trained on a single camera device, the prior methods are not adaptable to a different camera device and limit their applicability.

Contributions The paper's main contribution is creating a physic-based noise model for extreme low-light image denoising. They specifically incorporate the CMOS image sensor noise characteristic into the noise model which becomes more relevant in low-light scenarios. They also created a noise calibration method that can adapt the physic model to a given camera device. They also use a variety of camera devices to create the ELD dataset and verify the effectiveness of their method for generality and adaptability.

4.2. Proposed Solution

Their proposed solution first consists of their physics-based noise model (Figure 9). At every stage of the CMOS image sensor analog image pipeline, the noise is modeled. The Photon Shot Noise uses the Photon Transfer method as a Poisson and models the noise from the photon in pixel well to electrons. The Read Noise uses a Tukey-Lambda distribution to model the noise from the capacitors inside the image sensor. Finally, the Quantization Noise is the error created by converting the analog intensity to a discrete digital value.

Using the metadata from a camera the physics model variance and mean values can be estimated using empirical values that are unique to the camera.

After the synthetic image data is created it is mixed in with the real-world noisy data and the model is trained on U-net architecture. The image is then cropped into 512x512 image patches to be trained. The model uses L_1 loss with an Adam optimizer with a batch size of 1.

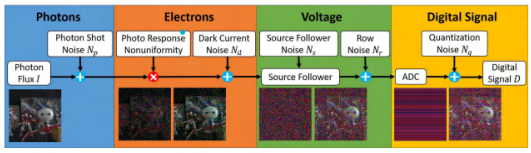

Figure 9. Overview of electronic imaging pipeline and visualization of noise sources and the resulting image at each stage.

4.3. Claims and Evidence

Claim 1 The proposed noise model can generate realistic synthetic data that resembles real data under extreme low-light conditions.

Evidence 1 This visual shows a comparison between real noise, gaussian model noise generator, and (Wei et al., 2020) physics-based simulator synthetic noise. The (Wei et al., 2020) method clearly achieves a better model of the noise which provides better training data for the denoiser model.

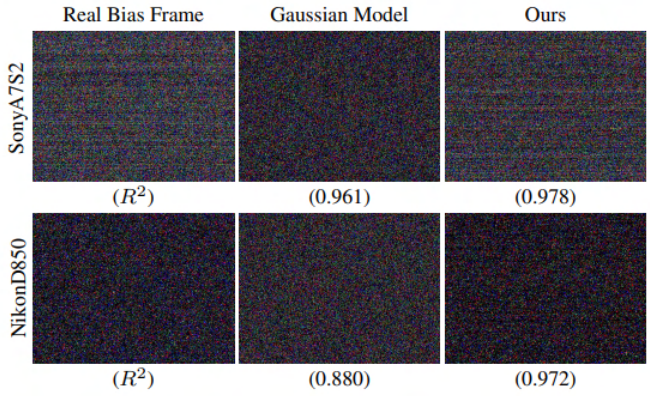

Figure 10. Overview of electronic imaging pipeline and visualization of noise sources and the resulting image at each stage.

Claim 2 The proposed noise model can improve the performance and generalization of neural networks for low-

light denoising.

Evidence 2 Figure 11 shows the quantitative comparison between different training schemes on the SID Dataset. The results show that the network trained with synthetic data from the proposed model outperforms or matches the networks trained with real data or other synthetic data.

Model	×100 PSNR / SSIM	×250 PSNR / SSIM	×300 PSNR / SSIM
BM3D	32.92 / 0.758	29.56 / 0.686	28.88 / 0.674
A-BM3D	33.79 / 0.743	27.24 / 0.518	26.52 / 0.558
Paired real data	38.60 / 0.912	37.08 / 0.886	36.29 / 0.874
Noise2Noise	37.42 / 0.853	33.48 / 0.725	32.37 / 0.686
G	36.10 / 0.800	31.87 / 0.640	30.99 / 0.624
G+P	37.08 / 0.839	32.85 / 0.697	31.87 / 0.665
G+P^*	38.31 / 0.884	34.39 / 0.765	33.37 / 0.730
G^*+P^*	39.10 / 0.911	36.46 / 0.869	35.69 / 0.855
G^*+P^*+R	39.23 / 0.912	36.89 / 0.877	36.01 / 0.864
G^*+P^*+R+U	39.27 / 0.914	37.13 / 0.883	36.30 / 0.872

Figure 11. Quantitative Results on Sony set of the SID Dataset. The noise models are indicated as follows. G: the Gaussian model for read noise N_{read}; G: the tukey lambda model for N_{read}; P: the Gaussian approximation for photon shot noise N_p; P : the true Poisson model for N_p; R: the Gaussian model for row noise N_r; U: the uniform distribution model for quantization noise N_q. The best results are indicated by red color and the second best results are denoted by blue color.

Claim 3 The proposed noise model is adaptable to different camera devices by using the proposed calibration method.

Evidence 3 Figure 12 shows the visual comparison of the increase in performance when incorporating the synthetic image dataset created by the physics model. This clearing shows the PSNR gains possible new physics-based implementation.

4.4. Critique and Discussion

This physics-based noise simulator proved that synthetic data can be generated to help the denoiser improve performance on real-world images. This innovative method builds on data-driven methods that don't need costly real-paired datasets to create an accurate model. In fact, the physics-based method helps achieve better results than paired data, probably, because of the simulator's smoothing effect on the data-driven model. The claims logically match the data and make sense. The one problem is their unclear methods for how they calibrate the noise-simulator and get their crucial probabilistic noise-model parameters (variance & mean). A lot of the parameters are based on empirical values, most camera manufacturers don't publicly release the

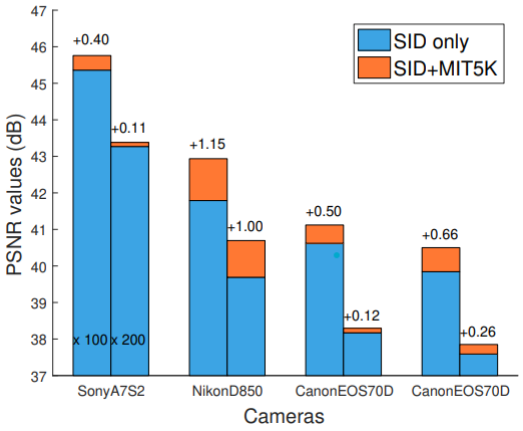

Figure 12. The performance boost when training with more synthesized data.

values needed to calculate the variance & mean parameters, such as the capacitance of the photodiode in the camera sensor. While there are calibration methods, their calibration procedure is not explicitly laid out.

5. Implementation

5.1. Implementation Motivation

As someone who is researching low-light noise modeling, I have been building my own dataset of raw images taken on a Nikon D600. There are 2 things I want to accomplish with my implementation.

1. Implement a new Amplification Factor
   (a) When utilizing the (Chen et al., 2018) denoising model the "amplification ratio must be chosen externally". The amplification factor value outlined in (Chen et al., 2018) involved having prerequisite knowledge of the Ground Truth (GT) exposure. The GT Amplifier value is calculated by dividing the GT low-noise (reference) image exposure and dividing it by the inputted noisy image exposure (Equation 1). When dealing with mobile phone photography or autonomous vehicle footage the lighting environments may dramatically change and it is important to find alternate methods to calculate an amplification factor since the proper exposure value is unlikely to be available in real-world settings. I hope to learn the effectiveness of (**?**) new amplification factor method that doesn't need the reference image's exposure value.
2. Test the PSNR and SSIM of (Chen et al., 2018) pipeline on new dataset images

(a) (Chen et al., 2018) has only tested the dataset with GT-Ratio values below 300, but I want to see the performance of the pipeline with more demand amplification ratio such as 500 or 1000. Currently, the (Chen et al., 2018) pipeline relies on only changing the black level of the camera stored in the raw photo metadata, to implement the model. Since ISO value can alter a photo before it goes through an image processing pipeline, I want to see how effective the model is at denoising photos with high ISO values and fast shutter speed in my own dataset.

Through my testing I hope to gain insight into the effectiveness of (Chen et al., 2018) method on different cameras, ISO values, and shutter speeds. I expect there to be a considerable drop in testing performance for the U-net model since it has not factored in internal camera noise. (Wei et al., 2020) underscores the significance of internal camera noise in tandem with diminished lighting conditions, thereby accentuating the nuanced impact on the overall final processed image quality.

$$GT\ AmplificationRatio = \frac{GT\ Exposure}{Input\ Exposure} \quad (5)$$

5.2. Implementation Setup and Plan

I used the SSIM and PSNR metrics to compare the processed image outputs to the ground truth image captured and gauge the performance of (Chen et al., 2018) amplification factor (GT-Ratio),(Chen et al., 2018) U-net, and (Lamba & Mitra, 2021) amplification factor (Lamba). I used the SID test images as a control variable while building my own low-light image dataset using a Canon EOS 850D and Nikon D600 to measure the impact of the pipeline and the different amplification factors. Since the Nikon D600 has more granularity in the ISO settings, I used it to measure the performance impact with different ISO settings. I used the Cannon to measure the impact of faster shutter speeds on the performance.

Using the official Github implementation, I implemented the image processing U-net in tensorflow using the pre-train model weights from the (Chen et al., 2018) github.

In order to test the impact that the new amplification value had on denoising the image, I coded out the SSIM and PSNR metrics using the Scipy library. This allowed me to have a quantifiable metric and compare the inputted high-noise image and the outputted processed image with the clean low-noise (reference) image.

Next, I implemented the (Lamba & Mitra, 2021) amplification factor value, with two functions. The weights function calculated the bin array and helped make sure that spurious light sources didn't impact the rest of the image and created a weighted intercity average. The amplification_value function computed the final amplification factor value (Equation 1).

After collecting my independent datasets, I used the Rawpy and Exiftool libraries to read my raw images and also scrape the metadata from the images to get critical values, such as image ISO, shutter speed, and image intensities.

Finally, each dataset's tested outputs were saved to a CSV file for further analysis in pandas.

5.3. Implementation Details

The (Chen et al., 2018) Github implementation is coded in Tensorflow 1 and an outdated version of Scipy. I first wrote updates to the downloaded GitHub code to ensure it could run on Tensorflow 2. Next, I downloaded the Sony SID Dataset and used the download_models.py script to install the model's weights.

5.3.1. $test\_Sony.py$

The lrelu, $upsample\_and\_concat$, network, and $pack\_raw$ functions remained the same from the GitHub $test\_Sony.py$, but many of the functions in the main part of the file needed to be updated in order to work on Tensorflow 2.9.0. Next, I implemented PSNR and SSIM metrics from the Skimage library to compare the performance of the processed images and the ground truth image. From (Lamba & Mitra, 2021) section 3.2, I wrote two functions: $weights\_arr$ and amp_val. When first testing the Lamba amplification factor, the outputs were overexposed and had a high error from the expected PSNR and SSIM values from the paper. After checking their GitHub, I realized two things: the hyperparameter (m) was different in their implementation than the paper, and I had not reversed the weights to prioritize the lower weight values (np.flipud(weights)). I changed my weight function and set my hyperparameter (m) to .05 instead of .5 from the paper.

5.3.2. $test\_Cannon.py$

$test\_Cannon.py$ used the same improved $test\_Sony.py$ based code but added new functionality to process the raw image metadata and perform analysis on the testing results. Using the os and Exiftools library, I first wrote code to parse the Canon dataset folder and segment the dataset to the different scenes for analysis. Next, I used the Exiftools library to extract the metadata from all the raw photos for analysis later in Pandas. Finally, I wrote pandas code to analyze the effectiveness of (Chen et al., 2018) pipeline and the different amplification methods.

5.3.3. *test_Nikon.py*

test_Nikon.py is built on the *test_Cannon.py* code but has added code contributions to factor in the different ISO values taken in the dataset. It also has specific pandas code to analyze the effectiveness of (Chen et al., 2018) pipeline and the different amplification methods.

test_Nikon.py also has slight changes in the *pack_raw* function and *weighted_image*. Despite no mention in (Chen et al., 2018) or (Lamba & Mitra, 2021), the 512 in those values are for the camera's black level. Canon's and Sony's black level are the same at 512, but the Nikon D600's black level is around 8. I had to empirically find this value by taking 2 back-to-back dark frames and averaging the results of the two images' pixel values.

5.3.4. *analysis.py*

This script was made at the end to perform analysis on all my result image metric tests. It reads in the stored metric data from a generated csv file after running a test script and performs some simple analysis using the pandas library. After running the Nikon Dataset metric test, it was extended to include a test to see if the image sensor array layouts were different.

5.4. Results and Interpretation

	PSNR	SSIM
Input	14.20	.109
GT Amp Factor	28.62	.766
Lamba Amp Factor	23.86	.731

Table 1. SID Dataset Test Average PSNR & SSIM Results

Table 1 shows my reimplementation's control SID Dataset test results odiffer by less than .02 for the SSIM and .2 for the PSNR from the expected results in the respective papers. This demonstraites that my testing is done on the same on methods from (Chen et al., 2018) and (Lamba & Mitra, 2021).

Both Canon and Nikon Dataset GT-Ratios range from 1.333 to 1300, but the minimum GT-Ratio in the SID Dataset was 100. This means that lower GT-Ratio inputted Nikon and Canon images (1.33 - 100) are too similar to the GT image and can not be fairly evaluated to the results of the SID Dataset, because they will skew the data results. Table 2 and Table _ show the actual average results for Canon and Nikon Datasets respectively, but for a fair comparison I will eliminate the images with GT-Ratios less than 100 to compare results between the SID Dataset and my own datasets.

5.4.1. Canon Results

When controlling for the same GT-Ratio conditions from the SID Dataset, the performance of the pipeline and amplification values are slightly lower than the expected values from the SID Dataset (Table 4). This is to be expected as the Canon is a completely different camera and probably has different internal noise characteristics as shown by a lower PSNR, but a similar SSIM score.

Lamba Amp surprisingly improved its performance relative to the GT Amp in the ultra-low light scenarios, with an average SSIM score better than the GT Amp and PSNR within .2 (Table 5). While this is an impressive result, looking at Figure 12 we can see that while the metric scores are similar the quality of the outputted photo is still bad. Without a tripod the shutter speed of $\frac{1}{3}$ sec is still too slow to take by hand and there would be some slight motion blur in the photo.

There does appear to be a drop in performance capabilities when testing on the Canon. This is to be expected as there is no CMOS specific internal noise model integrated into the training data, which becomes the dominant noise in low-light scenarios as discussed in (Wei et al., 2020).

	PSNR	SSIM
Input	21.48	0.336
GT Amp Factor	24.27	0.770
Lamba Amp Factor	16.32	0.610

Table 2. Canon Total Test Average PSNR & SSIM Results

	PSNR	SSIM
Input	15.61	0.077
GT Amp Factor	22.02	0.679
Lamba Amp Factor	19.80	0.651

Table 3. Canon Adjusted Test Average PSNR & SSIM Results

	PSNR	SSIM
Input	14.84	0.066
GT Amp Factor	19.69	0.608
Lamba Amp Factor	19.43	0.610

Table 4. Canon $GT - Ratio > 300$ Adjusted Test Average PSNR & SSIM Results

	PSNR	SSIM
Input	16.54	0.091
GT Amp Factor	24.82	0.764
Lamba Amp Factor	20.24	0.701

Table 5. Canon $GT - Ratio \leq 300$ AdjustedTest Average PSNR & SSIM Results

5.4.2. NIKON RESULTS

The Nikon test results showcase marginal to no performance gains in the PSNR. While there is some improvement in the SSIM, it appears much of the noise is still present in the processed image with marginal improvements in the SSIM metric and coloring. This is in stark contrast to the Canon dataset which showed some improvement with the pipeline. Table shows this is not a result of the higher ISO values. There seems to be an issue in the (Chen et al., 2018) pipeline that causes the Nikon images to not enhance well. This is significant because the end goal of learning-based image denoisers is to be able to enhance images from different camera types and measure the impact of ISO.

Looking at Table 6,7,8,9,10 we can clearly see that the PSNR performance is within .5 compared to the inputted image. Figure 13 also demonstrates the poor denoising and enhancement when compared to similar GT-Ratios in the Canon Dataset.

	PSNR	SSIM
Input	24.49	0.606
GT Amp Factor	23.65	0.635
Lamba Amp Factor	14.64	0.434

Table 6. Nikon Dataset Total Test Average PSNR & SSIM Results

	PSNR	SSIM
Input	21.04	0.393
GT Amp Factor	21.16	0.525
Lamba Amp Factor	19.52	0.521

Table 7. Nikon Adjusted Test Average PSNR & SSIM Results

	PSNR	SSIM
Input	22.61	0.467
GT Amp Factor	22.52	0.565
Lamba Amp Factor	19.58	0.546

Table 8. Nikon Adjusted $GT - Ratio \leq 300$ Test Average PSNR & SSIM Results

	PSNR	SSIM
Input	21.34	0.400
GT Amp Factor	21.20	0.559
Lamba Amp Factor	20.16	0.562

Table 9. Nikon $ISO \geq 800$ Adjusted Test Average PSNR & SSIM Results

	PSNR	SSIM
Input	21.58	0.427
GT Amp Factor	21.63	0.469
Lamba Amp Factor	19.53	0.472

Table 10. Nikon $ISO \leq 200$ Adjusted Test Average PSNR & SSIM Results

Figure 12. (a)

Figure 12. (b)

Figure 12. (c)

Figure 12. (d)

Figure 12. (a) GT Amp Output Image, (b) Lamba Amp Output Image, (c) Input Image, (d) Reference GT Image
Canon EOS 850D Dataset Image 1, ISO 200, Shutter Speed $\frac{1}{3}$ sec, $GT - Ratio$ 449

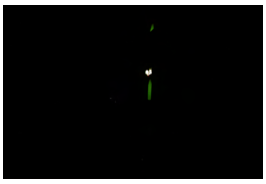

Figure 13. (a)

Figure 13. (b)

Figure 13. (c)

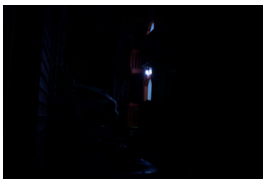

Figure 13. (d)

Figure 13. (a) GT Amp Output Image, (b) Lamba Amp Output Image, (c) Input Image, (d) Reference GT Image
Nikon D600 Dataset Image 2, ISO 100, Shutter Speed $\frac{1}{30}$ sec, $GT - Ratio$ 450

6. Conclusion

Through my research, the Lamba Amplification factor appears to be a suitable alternative to the GT Amplification factor. The performance of the Lamba Amp excels especially in ultra-low lighting conditions showcased by its performance in the Canon dataset when the GT Ratio was greater than 300. The Lamba Amp does tend to over-amplify in brighter lighting conditions causing overexposure in more ideal lighting conditions. This overexposure is not a big issue in the real world as you can lower the hyperparameter (m) to properly expose the images in real time.

The (Chen et al., 2018) pipeline had a lower performance with the Canon Dataset but no measurable image enhancement on my Nikon Dataset. Originally, I thought I had made an error in the pipeline but the Canon Dataset results showed both SSIM and PSNR enhancement improvements. I then changed the analysis script to see the image metadata black channel level and empirically found the black level through dark-frame averages. I then integrated the new black channel value into the $test\_Nikon.py$, which did improve the performance to not have a negative effect on the enhancement of the inputted images. Finally, I tested my internal pixel array configuration to see if the Bayer Array pattern was different and if all the cameras had the same pixel and color configurations. The only other option would be to retrain the pipeline on my Nikon Dataset, but I don't have the computing resources available to me. In general, the pipeline was a great showcase of how real-world data could be used to create and improve low-light image enhancement denoisers but it lacked generality to different cameras.

References

Anaya, J. and Barbu, A. Renoir – a dataset for real low-light image noise reduction. *Journal of Visual Communication and Image Representation*, 51:144–154, 2018. ISSN 1047-3203. doi: https://doi.org/10.1016/j.jvcir.2018.01.012. URL https://www.sciencedirect.com/science/article/pii/S1047320318300208.

Brooks, T., Mildenhall, B., Xue, T., Chen, J., Sharlet, D., and Barron, J. T. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11036–11045, 2019.

Chen, C., Chen, Q., Xu, J., and Koltun, V. Learning to see in the dark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Chen, Q., Xu, J., and Koltun, V. Fast image processing with fully-convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2497–2506, 2017.

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8): 2080–2095, 2007. doi: 10.1109/TIP.2007.901238.

Gu, S., Li, Y., Gool, L. V., and Timofte, R. Self-guided network for fast image denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2511–2520, 2019.

Guo, C., Li, C., Guo, J., Loy, C. C., Hou, J., Kwong, S., and Cong, R. Zero-reference deep curve estimation for low-light image enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1780–1789, 2020.

Lamba, M. and Mitra, K. Restoring extremely dark images in real time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3487–3497, 2021.

Lamba, M., Balaji, A., and Mitra, K. Towards fast and light-weight restoration of dark images. *arXiv preprint arXiv:2011.14133*, 2020.

Mildenhall, B., Barron, J. T., Chen, J., Sharlet, D., Ng, R., and Carroll, R. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2502–2510, 2018.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation, 2015.

Shih, K.-H., Chiu, C.-T., Lin, J.-A., and Bu, Y.-Y. Real-time object detection with reduced region proposal network via multi-feature concatenation. *IEEE transactions on neural networks and learning systems*, 31(6):2164–2173, 2019.

Wei, K., Fu, Y., Yang, J., and Huang, H. A physics-based noise formation model for extreme low-light raw denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Xu, K., Yang, X., Yin, B., and Lau, R. W. Learning to restore low-light images via decomposition-and-enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., and Fu, Y. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2472–2481, 2018.