

# NoiSER: Noise is All You Need for Low-Light Image Enhancement

Zhao Zhang<sup>1\*</sup>, Suiyi Zhao<sup>1</sup>, Xiaojie Jin<sup>2</sup>, Mingliang Xu<sup>3</sup>, Yi Yang<sup>4</sup>, and Shuicheng Yan<sup>5</sup>

<sup>1</sup> Hefei University of Technology, Hefei, China

<sup>2</sup> Bytedance Research, USA

<sup>3</sup> Zhengzhou University, Zhengzhou, China

<sup>4</sup>Zhejiang University, Hangzhou, China

<sup>5</sup>Sea AI Lab, Singapore

## Abstract

In this paper, we present an embarrassingly simple yet effective solution to a seemingly impossible mission, low-light image enhancement (LLIE) without access to any task-related data. The proposed solution, Noise SElf-Regression (NoiSER), simply learns a convolutional neural network equipped with a instance-normalization layer by taking a random noise image,  $\mathcal{N}(0, \sigma^2)$  for each pixel, as both input and output for each training pair, and then the low-light image is fed to the learned network for predicting the normal-light image. Technically, an intuitive explanation for its effectiveness is as follows: 1) the self-regression reconstructs the contrast between adjacent pixels of the input image, 2) the instance-normalization layers may naturally remediate the overall magnitude/lighting of the input image, and 3) the  $\mathcal{N}(0, \sigma^2)$  assumption for each pixel enforces the output image to follow the well-known gray-world hypothesis [1] when the image size is big enough, namely, the averages of three RGB components of an image converge to the same value. Compared to existing SOTA LLIE methods with access to different task-related data, NoiSER is surprisingly highly competitive in enhancement quality, yet with a much smaller model size, and much lower training and inference cost. With only  $\sim 1K$  parameters, NoiSER realizes about 1 minute for training and 1.2 ms for inference with 600x400 resolution on RTX 2080 Ti. As a bonus, NoiSER possesses automated over-exposure suppression ability and shows excellent performance on over-exposed photos.

## 1. Introduction

Unfavorable illumination is a common occurrence when taking photographs, and the resulted photos are often poorly illuminated, greatly hindering the understanding of their contents. Low-light image enhancement (LLIE) is a task that transforms the low-light images into normal-light ones by refining the illumination of the images. Traditional LLIE methods are usually based on the histogram equalization (HE) [3, 16] or retinex theory [9, 13, 29, 41], and they usu-

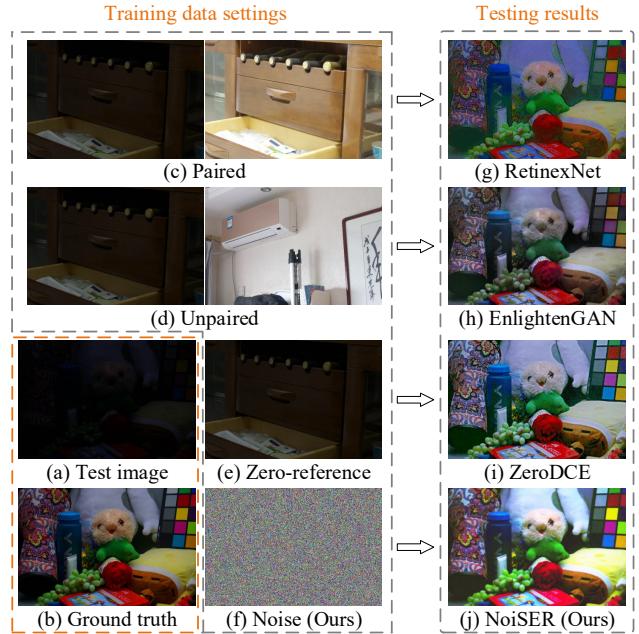


Figure 1. Comparison of different training data used in current LLIE methods, e.g., paired data in RetinexNet [37], unpaired data in EnlightenGAN [12], zero-reference data in Zero-DCE [8], and noise in our NoiSER. Clearly, our NoiSER performs the best.

ally cause unpleasant artifacts and require a long processing time. Recently, deep neural networks (DNNs) [25, 38, 46] have been successfully used for various high-level [27, 30, 33, 48] and low-level [4, 7, 40, 45] vision tasks due to their strong learning abilities. This also gives birth to advanced deep learning-based LLIE methods [8, 12, 17, 22, 39], in which task-related data is a necessary condition.

**Paired data** (see Fig.1(c)) are usually used as a strong LLIE-related constraint for the supervised/semi-supervised models to produce promising results [21, 42–44]. However, paired data are costly to be collected in reality and the overreliance of model training on synthetic paired data limits their practical applications due to the large distribu-

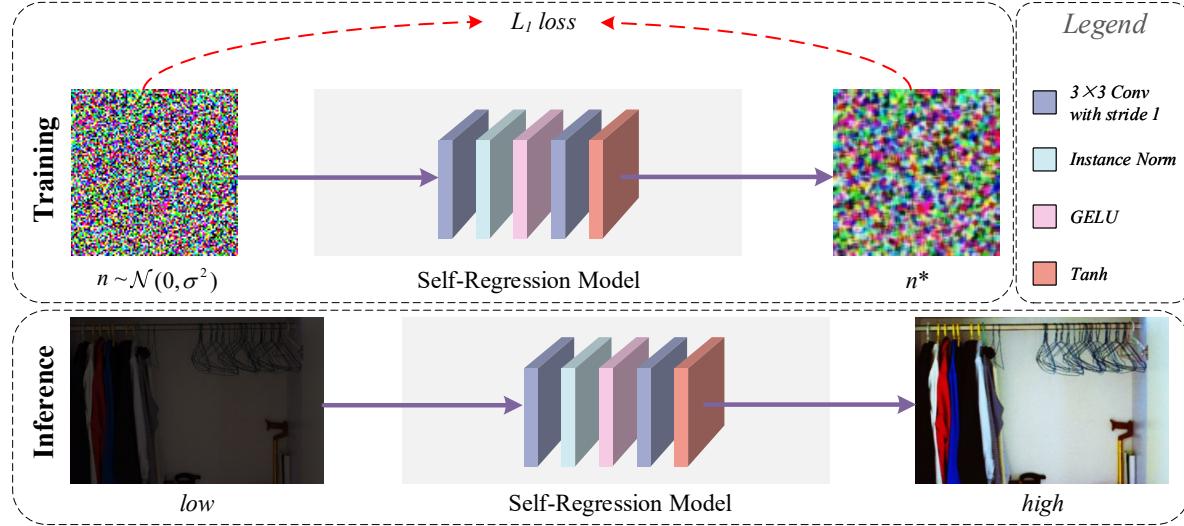


Figure 2. The training and inference pipeline of NoiSER. During training, NoiSER just iteratively samples noise  $n \sim \mathcal{N}(0, \sigma^2)$  as both model input and supervised signal to train a self-regression model (SRM), i.e.,  $n^* = SRM(n) \approx n$ . During inference, the trained SRM can directly enhance the low-light images and obtain impressive results. Note that the designed SRM is equipped with the instance normalization and two non-linear activations layers without shortcuts, which avoids learning an identity mapping.

tion difference between synthetic and real data. **Unpaired data** (see Fig.1(d)) are often used as an ordinarily LLIE-related constraint to train unsupervised models, such as EnlightenGAN [12]. However, these unsupervised methods usually rely on a huge number of parameters to compensate for the weak constraint caused by the unpaired data, which also hinders their practical applications for LLIE. **Zero-reference data** (see Fig.1(e)) facilitate a simple LLIE model that trains a DNN by an elaborate unsupervised weak constraint [8, 17, 20, 22], which can yield impressive results using only zero-reference data, and meanwhile are far more lightweight and efficient than those methods using paired/unpaired data. However, the performance of all of these methods is highly dependent on the training data, but it is hard to find the most suitable data.

It is worth noting that all the training data used in current LLIE methods are always task-related, no matter whether they are strongly, ordinarily or weakly related. If we do not use any task-related data for training, can we still enhance the low-light images by deep learning? Obviously, current methods cannot do this, since deep neural networks require to train their scads of parameters using copious amounts of task-related training data. However, in this paper we will show that just using noise for training can also enhance the low-light images with surprising visual effects. Specifically, by sampling the noise (see Fig.1(f)) from a Gaussian distribution as both model input and supervised signal to fit the parameters, low-light images can be effectively enhanced. We summarize the main contributions as follows:

- To the best of our knowledge, this is the first attempt to learn a general “generalization” directly from task-irrelevant data, instead of the common “fit  $\rightarrow$  general-

ize” procedure in current deep learning paradigms. Experiments show that an effective solution in line with this new learning paradigm is closer to the essence of the task and possesses stronger generalization ability.

- Technically, we propose an embarrassingly simple and effective LLIE method, termed Noise SELf-Regression (NoiSER) via simply learning a convolutional neural network (CNN) equipped with instance-normalization layers. Different from all the existing LLIE methods, NoiSER is a magical method, for it does not need any task-related data for training, and just uses the randomly sampled noise to enable the training of the model, i.e., noise is all NoiSER needs for LLIE.
- On several widely-used datasets, our NoiSER is highly competitive to other related competitors that use different types of task-related data. Without any task-related data, NoiSER can recover the low-light images accurately in detail and naturalness, and most importantly, it has an automated ability to turn extreme light/dark into moderates by the learned CNN. In addition, NoiSER is extremely lightweight and efficient in both training and inference phases. Specifically, with only about 1K parameters, NoiSER achieves about 1 minute for training and 1.2 milliseconds for inference on 600x400 resolution image by a single RTX 2080 Ti.

## 2. Preliminaries

### 2.1. Image Self-Regression Principle

Image self-regression [11, 15, 31] utilizes the input data itself as the supervised signal to reconstruct the output. By assuming  $x$  follows a certain distribution, this process can be represented by minimizing the following empirical risk:

$$\arg \min_{\theta} \mathbb{E}_x\{L(f_{\theta}(x), x)\}, \quad (1)$$

where  $f_{\theta}$  denotes a parametric family of mapping and  $L$  is a loss function. After optimization, given an arbitrary input  $x_0$ ,  $f_{\theta}$  can yield a texture-similar output  $x_0^*$ . We design a self-regression model ( $f_{\theta}$ ) (see Fig.2) to include instance normalization and non-linear activation layers yet without shortcut, which avoids learning an identity mapping.

## 2.2. Gray-World Color Constancy Hypothesis

The gray-world color constancy hypothesis [1] (shortly, gray-world hypothesis) is useful in image processing, which says, for an image with large color variations, the averages of the three RGB components converge to the same gray value  $K$ . In recent years, some deep learning-based LLIE methods use the gray-world hypothesis as loss function and obtain impressive LLIE performance [8, 17, 18]. Zero-DCE [8] formulates the gray-world hypothesis as a color constancy constraint, which is expressed as

$$\mathcal{L}_{col} = \sqrt{\sum_{(p,q) \in S} (J^p - J^q)^2}, S = \{(R, G), (R, B), (G, B)\}, \quad (2)$$

where  $J^p$  and  $J^q$  represent the average values of the  $p$  and  $q$  channels in the enhanced image, respectively.

## 2.3. Mirror Visual Feedback (MVF) Therapy

The MVF therapy was first introduced to relieve phantom limb pain in amputees [23]. As a therapy, it works by the following mechanisms: during treatment, the “affected side” of the hemiplegic patient is blocked by a flat mirror, while the patient can see the projection of the “healthy side” from the reflective surface of the mirror. In this way, when “rehabilitation training” is carried out on the “healthy side”, due to the visual feedback, the patient will believe that the “affected side” is able-bodied, and indeed, the “affected side” will also miraculously get better. It is worth noting that the patient’s “affected side” does not perform any “rehabilitation training” during this process, but even so, the “affected side” can be treated to some extent. This gives us such an insight: in a similar way to treating the “affected side” but without directly applying the “rehabilitation training” to the “affected side”, we may try to perform LLIE without directly using LLIE-related data for training.

## 3. Proposed Method

### 3.1. Motivation and Problem Statement

Consider a situation that we need to enhance a low-light image by deep learning, but we have no LLIE task-related data for training, including paired/unpaired/zero-reference data. In such special case, how can we train a deep model to complete the task? Or do we really have hope to enhance a low-light image by deep learning without task-related data?

A *simple effective* solution to this problem will likely produce a major impact on the whole field of deep learning

for computer vision (CV). Next, we discuss the potential impact of such a solution from two aspects.

**For the LLIE task itself, this solution can avoid the inherent pitfalls of existing methods using task-related data.** For all task-related data-based methods, the quality of the data determines the performance of the method. In other words, poor data quality leads to poor performance, which means we need to collect good enough data, whereas we usually have no idea about what kind of data is “good” for the proposed method. Thus, if this simple effective solution does exist, the above pitfalls will no longer exist.

**For the other CV tasks, this simple solution may allow for joint task processing in an elegant way.** Recently, some works jointing LLIE with other tasks have emerged, e.g., joint tasks with deraining [32], denoising [28, 47], object/face detection [5, 19, 34, 35] or semantic segmentation [6]. The methods for all of these joint tasks often choose to design a complex architecture to meet the requirements. As such, if the solution discussed in this paper exists, considering the simplicity of which, there may be an elegant way to embed it into an arbitrary CV task for joint processing, since no LLIE-related data is needed to enhance an image.

## 3.2. Our Message

At first glance, solving such an enhancement problem seems impossible, since we do not have any available information to train a DNN. One conceivable solution for implementing deep training without task-related data is to borrow data from other similar tasks, which is exactly the idea of transfer learning [49]. However, this is not a real-sense solution, since data are still needed to fine-tune the model finally. Is there really no way out?

In this paper, we will show that this problem not only can be solved, but also can be solved in an extremely elegant way. All we need to address this problem is to make good use of the three important theories in Section 2, i.e., image self-regression principle, gray-world hypothesis and mirror visual feedback therapy. Based on the three theories, we can show that random noise can also exhibit great energy. A surprising message is that we will be able to train a DNN model by randomly sampling noise to directly enhance a low-light image. That is, noise is all you need to solve the problem. The following is the specific solution.

### 3.3. Solution

#### 3.3.1 Abstracting and Meeting Task Requirements

Let us recall the problem, i.e., “can we enhance low-light image by deep learning without any task-related data?” We first decompose and abstract it into three requirements: 1) any task-related data cannot be used for training; 2) given an image, the output of the model should have similar adjacent pixel contrast (texture) to the input; 3) given a low-light image, the output of the model should be of normal light.

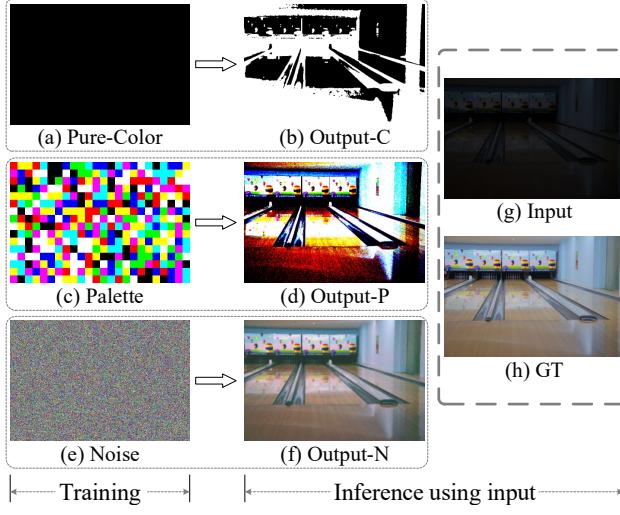


Figure 3. Comparison of the fully-converged enhancement results using different training data. Clearly, the noise self-regression approach can yield visually better results.

For the requirement 1), the easiest solution is to use the random noise to replace the task-related data. The requirement 2) gives us a message that we need to train a model with a reconstruction ability. Considering the image self-regression principle, we can use the noise itself as supervisory signals to train the model. Now, we can satisfy the requirements 1) and 2) in a self-regression manner by using the random noise. For the requirement 3), we can force the output to satisfy the  $K$ -value gray-world hypothesis, where  $K$  should be in a normal gray range, such as 80-140. The proof of why this manner could convert light to normal is given in *Proposition 3* of Section 3.3.3.

Note that just meeting the above three requirements is still not enough to solve the problem, as there is one more important challenge, i.e., there is a huge distribution gap between the training data (i.e., noise) and inference data (i.e., low-light image). Instead of narrowing the intractable distribution gap, the MVF therapy brings a fresh perspective, i.e., *maybe we can roughly regard the forward propagation of non-task-related data as “healthy side” and the forward propagation of low-light images as “affected side”, and see the self-regression as “rehabilitation training”*. Therefore, we no longer need to bother with the gap in distribution, but only need to adjust the self-regression training to find out what is really helpful in enhancing a low-light image. In this way, if we can eventually find a proper solution in the form of the MVF therapy, all we need to do is to clarify the mechanism behind the solution.

We show the ultimate pipeline of training and inference and detailed structure of our self-regression model for an intuitive observation in Fig.2. It is noteworthy that the noise itself is still relatively complex data. Thus, prior to training with noise, we introduce two types of data with much simpler structures, i.e., pure-color (Fig.3(a)) and palette

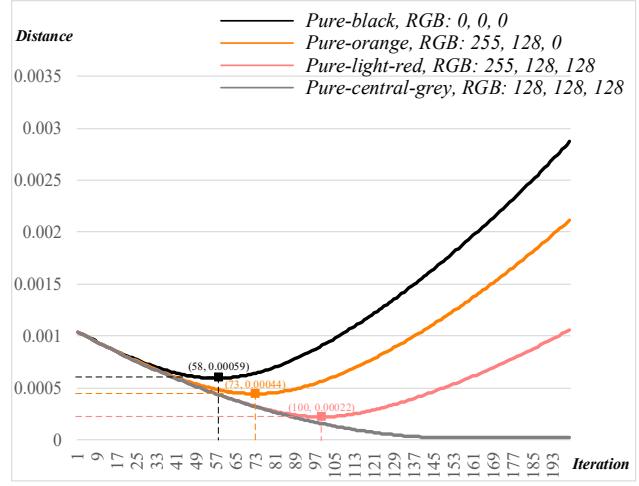


Figure 4. Iteration-Distance curves using different pure-colors for C-regression training. The distance is the  $L_2$  norm between the output and the pure-central-grey during training.

(Fig.3(c)). In what follows, we present and introduce three self-regressions asymptotically.

### 3.3.2 Pure-Color Self-Regression (C-Regression)

We first use pure-color for self-regression training, where “pure-” denotes an image rather than a color, e.g., a pure-black or pure-light-red image. Following the general image self-regression principle, C-regression minimizes the deviation between model output and the pure-color  $I_c$  itself according to certain loss function:

$$\arg \min_{\theta} \mathbb{E}_{I_c} \{ L(f_{\theta}(I_c), I_c) \}, \quad (3)$$

where  $f_{\theta}$  is a parametric family of mappings and  $L_1$  is used for minimizing the reconstruction error of the pure-color.

We first use the pure-black for training and directly apply the trained model to enhance a low-light image as an example, as shown in Fig.3(g) and Fig.3(b). We can easily see that the low-light input is crudely broken down into binary colors, i.e., black and white, which implies that pure-black cannot enhance the image although it is close to a dark image. However, we still gain some inspiration, e.g., C-regression has roughly reconstructed the texture of the input image. Thus, it is crucial to reveal the mechanism on how the input is divided into two colors.

Deep neural networks often work in black box manner and we cannot explain the intrinsic mechanism at the micro level. But at the macro level, we can still obtain a correct conclusion based on induction reasoning. Specifically, we follow the steps of incomplete induction reasoning, i.e., *individual phenomenon  $\rightarrow$  individual conclusion  $\rightarrow$  universal conclusion  $\rightarrow$  intra-domain validation*. Specifically, based on the phenomenon of pure-black self-regression (i.e., dividing low-light input into black and white), we can induce

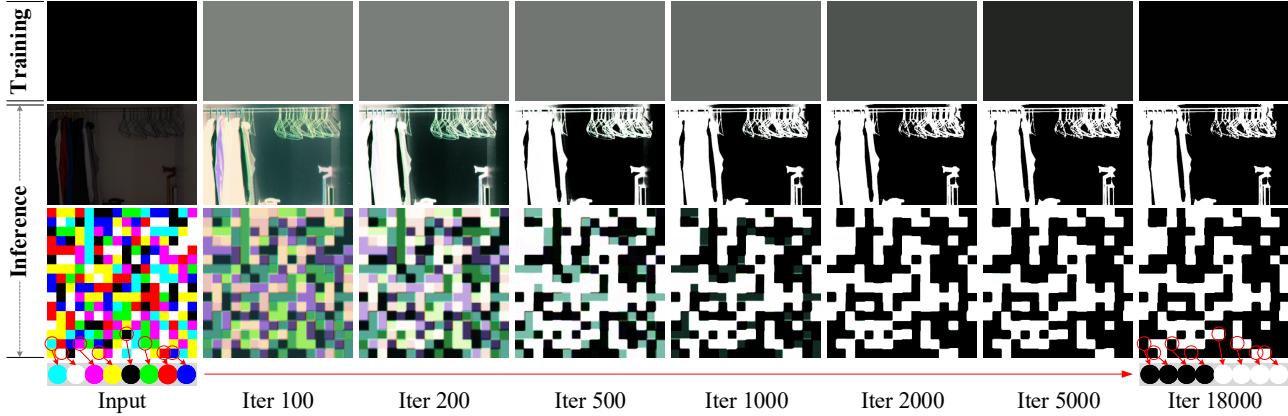


Figure 5. The phenomenon obtained from the pure-black self-regression. For an arbitrary input with a degree of contrast, the model finally tend to divide the input to be either black or white.

an individual conclusion and then generalize it from individual to universal, and as a result, this universal conclusion can be applied to other pure-color (e.g., pure-red). Then, we need to perform self-regression training experiments using other colors, and if the experimental results also satisfy the universal conclusion, this conclusion can be proven to be correct. Prior to perform incomplete induction reasoning, we introduce the required definitions and propositions.

*Definition 1 (central grey):* For a specific range of image values  $[a,b]$ , the central grey is a color  $C_g \in \mathbb{R}^3$  if it satisfies

$$C_g^i = \frac{a+b}{2}, \quad i \in \{R, G, B\}, \quad (4)$$

where  $C_g = (127.5, 127.5, 127.5)$  in the general 256 grey level (range:  $[0, 255]$ ), while actually the channel value is replaced by 128, since only integers are allowed.

*Definition 2 (opposite color):* For a specific range of image values  $[a,b]$ , a color  $C_1 \in \mathbb{R}^3$  is the opposite color of the color  $C_2 \in \mathbb{R}^3$  if they satisfy the following condition:

$$|C_1^i - C_g^i| = |C_2^i - C_g^i|, \quad i \in \{R, G, B\}, \quad (5)$$

where  $C_g$  denotes the central grey, and  $C_1 \neq C_2$ .

*Proposition 1:* Based on a model with random initialization, arbitrary C-regression tends to construct the pure-central-grey in the initial iterations.

*Validation for Proposition 1:* Given an image  $I$ , we use Eqn.2 to measure the distance between this image and the pure-central-grey with small modification as follows:

$$\mathcal{D}(I) = \sqrt{\sum_{i \in S} (I^i - C_g^i)^2}, \quad S = \{R, G, B\}, \quad (6)$$

where  $I^i$  is the average value of the  $i$  channel in image  $I$ .

In this study, we choose four pure-colors with different distances from the pure-central-grey for validation, i.e., pure-black ( $\text{RGB}(0,0,0)$ ), pure-orange ( $\text{RGB}(255,128,0)$ ), pure-light-red ( $\text{RGB}(255,128,128)$ ) and the pure-central-grey itself ( $\text{RGB}(128,128,128)$ ). Clearly, the relation of the

distances satisfies:  $\mathcal{D}(\text{pure-black}) > \mathcal{D}(\text{pure-orange}) > \mathcal{D}(\text{pure-light-red}) > \mathcal{D}(\text{pure-central-grey}) = 0$ . In Fig.4, we show the iteration-distance curves based on these pure-colors during C-regression training. For non-pure-grey self-regression training, no supervised signals are pure-grey; nevertheless, the overall trend of the curves is initially down and then up rather than directly up, which indicates *Proposition 1* is True. For the pure-grey self-regression training, the curve falls directly and converges gradually, which additionally proves the correctness of *Proposition 1*.

**Individual phenomenon.** Fig.3(a), Fig.3(g) and Fig.3(b) show two examples of using pure-black self-regression training to infer a low-light image. We see from Fig.3(h) and Fig.3(b) that the trained model tends to map some colors (e.g., white and yellow) to the color used in C-regression training (i.e., black,  $\text{RGB}(0,0,0)$ ), while mapping the other colors (e.g., black, red and green) to the opposite color (i.e., white,  $\text{RGB}(255,255,255)$ ). This inspires us to explore the mapping relationship of the colors before and after inference. To this end, we build a palette containing rich colors as the image to be inferred. Fig.5 shows the process of inferring a low-light image and shows a palette in pure-black self-regression iterative training. We see that there are 8 colors in the palette, of which 4 colors (i.e., cyan, white, purple and yellow) are mapped to black and the other 4 colors (i.e., black, green, red and blue) are mapped to the opposite color of black, which is applicable to each pixel in the palette.

**Individual conclusion.** Table 1(a) illustrates the specific induction process from individual phenomenon to individual conclusion based on pure-black self-regression. Specifically, the individual phenomenon is shown in rows 4 and 12, while the induction process of the individual conclusion is shown in rows 1-3 and 5-11. During training, *Proposition 1* tells us that the initial output tends to the pure-central-grey (row 1); as the iterations increase, the output becomes very close to the pure-black (row 2). Thus, we can find that for all three RGB channels, the pure-black self-regression per-

Table 1. Illustration of the C-regression mechanism.

(a) Individual phenomenon → Individual conclusion							(b) Universal conclusion → Intra-domain validation														
Training (Black)	1	Initial R:128	G:128	B:128	Black or White?				Training (Red)	1	Initial R:128	G:128	B:128	Red or Cyan?							
	2	Final R:0	G:0	B:0						2	Final R:255	G:0	B:0								
	3	Trend R: $\downarrow$	G: $\downarrow$	B: $\downarrow$						3	Trend R: $\uparrow$	G: $\downarrow$	B: $\downarrow$								
Inference	4	Color	White	Cyan	Purple	Yellow	Red	Green	Blue	Black	4	Color	White	Cyan	Purple	Yellow	Red	Green	Blue	Black	
	5	R $\downarrow$	255	0	255	255	255	0	0	0	5	R $\uparrow$	255	0	255	255	255	0	0	0	
	6	G $\downarrow$	255	255	0	255	0	255	0	0	6	G $\downarrow$	255	255	0	255	0	255	0	0	
	7	B $\downarrow$	255	255	255	0	0	0	255	0	7	B $\downarrow$	255	255	255	0	0	0	255	0	
	8	R?	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$				8	R?		$\checkmark$				$\checkmark$	$\checkmark$	$\checkmark$	
	9	G?	$\checkmark$	$\checkmark$		$\checkmark$		$\checkmark$			9	G?	$\checkmark$	$\checkmark$		$\checkmark$		$\checkmark$			
	10	B?	$\checkmark$	$\checkmark$	$\checkmark$			$\checkmark$			10	B?	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$			$\checkmark$		
	11	No.	3	2	2	2	1	1	1	0	11	No.	2	3	1	1	0	2	2	1	
	12	To	Black	Black	Black	Black	White	White	White	White	12	To	Red?	Red?	Cyan?	Cyan?	Cyan?	Red?	Red?	Cyan?	

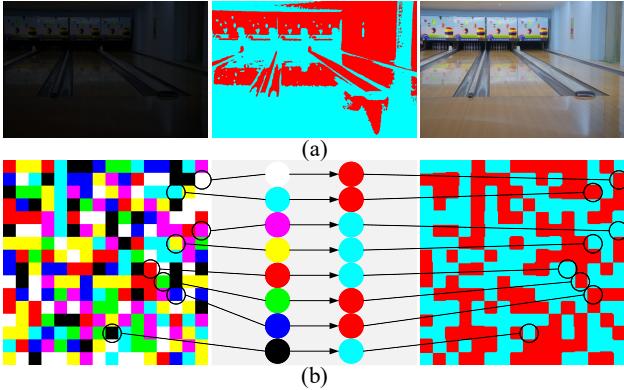


Figure 6. Intra-domain validation using pure-red self-regression.

haps learns a downtrend (row 3). During inference, we see from the phenomenon that the input is divided into black and white (rows 4 and 12), which may be related to the learned trends. To this end, firstly, we list the RGB values (rows 5-7) of the colors in the fourth row; secondly, we check whether the specific channel satisfies the learned trend (rows 8-10). Taking “Cyan” as an example, since all three RGB channels learn to trend down, the G,B channels in “Cyan” satisfies the trend, but R channel does not (0 will overflow if a downward trend is taken); thirdly, we record the number of channels that satisfy the learned trend (row 11); finally, based on the results in row 11 and the observations in row 12, we believe if a color meets at least two channel trends, it will be mapped to black (i.e., training color), and conversely, to white (the opposite of training color).

**Universal conclusion.** According to the individual conclusion, we can directly generalize it to a universal one, i.e., *during training, C-regression attempts to learn the trends of the three channels of RGB. During inference, if a color satisfies over two channel trends, it will be mapped to the training color, otherwise, to the opposite of training color.*

**Intra-domain validation.** To verify the correctness of the universal conclusion, we need to perform C-regression training based on the other colors and see whether the experimental results are consistent with this conclusion. Taking the pure-red self-regression as an example, we perform the reasoning process in Table 1(b). Row 12 displays the rea-

soning results, from which we see that white, cyan, green and blue are mapped to red, while purple, yellow, red and black are mapped to cyan. We further conduct and display the corresponding experiments in Fig.6. Clearly, the results keep consistent with the reasoning results, which proves that the conclusion is correct. Here, there is one point that must be noticed, i.e., the condition for the above conclusion to be valid is that the distribution between the training and inference data is far different.

At this point, we have fully understood the mechanism of C-regression and can assert the following facts:

- C-regression initially implements the reconstruction of the input texture, and if the texture integrity is not considered for the LLIE task, C-Regression already satisfies the requirements 1) and 2) in Section 3.3.1.
- The color binarisation of the C-regression results relies on the color homogeneity in the training samples, and it is conceivable that a more accurate texture might be reconstructed if each training sample is enriched with multiple colors and has a degree of contrast.

### 3.3.3 Palette Self-Regression (P-Regression)

In this section, we illustrate that using the palette as training data for self-regression enables the generation of a normal-light image and finer reconstruction of the image textures. Similar to C-regression, given palettes  $I_p$  as training data, P-regression is the process whereby the palette learns to rebuild itself according to certain loss function:

$$\arg \min_{\theta} \mathbb{E}_{I_p} \{L(f_{\theta}(I_p), I_p)\}, \quad (7)$$

where we also employ the  $L_1$  loss for minimizing the reconstruction error based on the palette.

Fig.3(g) and Fig.3(d) display an example of inferring a low-light image through P-regression. It is clear that P-regression successfully reconstructs the textures of the input low-light image. Next, we introduce the required propositions to figure out the mechanism behind P-regression.

**Proposition 2:** Based on a model with random initialization, P-regression tends to start building the pure-central-grey at a starting point and continues until it converges.

Table 2. List of RGB channel means for different datasets under the general 256 grey level. We can conclude that, for an image of normal light, the RGB channel means usually fall in the range of 80-140. The second column is the mean of the noise sampled by  $\mathcal{N}(0, \sigma^2)$  (0 for [-1,1] is equivalent to 128 for [0,255]).

Sets	Noise	LOL [37]	LSRW (Huawei) [10]	LSRW (Nikon) [10]
Light	-	Normal Low	Normal Low	Normal Low
R	128	120.47 15.49	110.39	18.92
G	128	113.88 15.22	98.22	16.39
B	128	110.38 14.73	86.63	15.20
				122.69 42.26
				124.55 43.19
				124.69 43.48

*Validation for Proposition 2:* Given an image  $I$ , similar to C-regression, we use Eqn.6 to measure the distance between this image and the pure-central-grey. The blue curve in Fig.7(a) shows the distance between the output and pure-central-grey. We see that after a certain starting point (about the 61st iteration), the output gradually approaches the pure-central-grey, although we did not add any constraints during training, which means that *Proposition 2* is True. Besides, we use pure-colors as inference images for validation, and obtain a consistent conclusion with *Proposition 2*, i.e., all outputs appear grey (see Fig.7(b)).

*Proposition 3:* Known ①: The image satisfies the  $K$ -value gray-world hypothesis, where  $K$  is approximately between 80 and 140; Known ②: The image is of normal light. Then, ① is a statistically necessary condition for ②.

*Validation for Proposition 3:* In Table 2, we describe the RGB channel means (i.e., the mean value of each color channel) over several widely-used-datasets, LOL [37] and LSRW [10]. We conclude that: (1) all datasets satisfy the gray-world hypothesis (the channel deviation is also not too large for LSRW (Huawei)-Normal); (2) the channel means of the normal datasets usually lie between 80-140. Note (1) and (2) provide the proof of *Proposition 3*.

For P-regression, the training samples themselves have a degree of contrast (i.e., variation between non-overlapping patches), and based on the self-regression principle, the contrast can be reconstructed after self-regression training, which enables it to generate the texture and structure of the input image more accurately than C-regression. In addition, according to *Propositions 2 and 3*, we know the P-regression has the ability to map any color to approach grey, which means the channel means of the output meet the K range (80-140) of the normal-light images. As such, suppose the P-regression further satisfies the gray-world hypothesis and we can use P-regression to get the normal-light output that satisfies all three requirements mentioned in Section 3.3.1. However, this supposition is valid but not strictly. We use the color constancy loss (Eqn.2) to measure the satisfaction degree towards the gray-world hypothesis (not applying it to back propagation). As the grey curves in Fig.7(a) show, the overall downward trend suffers from sharp fluctuations. Nevertheless, the P-regression can still yield a relatively favorable result.

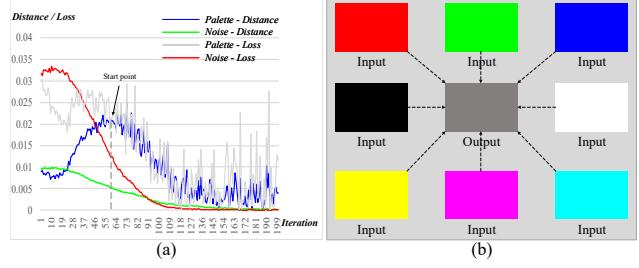


Figure 7. (a) Iteration-Distance/Loss curves for P-regression and NoiSER training. (b) Pure-color mappings after training.

Now, we can clarify the mechanism of P-regression and assert the following facts:

- P-regression initially meets all the three requirements mentioned above, which means it can already enhance a low-light image to some extent.
- The enhancement results of P-regression are unpleasant due to the sharp fluctuations in convergence to gray-world hypothesis. It is conceivable that a more pleasing result might be generated, if we can smooth out the sharp curve fluctuations.

### 3.3.4 Noise Self-Regression (NoiSER)

Instead of defining an additional noise, we directly sample a Gaussian noise  $I_n \sim \mathcal{N}(0, \sigma^2)$  for each pixel, and the specific regression process can be expressed as follows:

$$\arg \min_{\theta} \mathbb{E}_{I_n} \{L(f_{\theta}(I_n), I_n)\}, \quad (8)$$

where we still use the  $L_1$  loss for minimizing the reconstruction error based on the noise.

Fig.3(g) and Fig.3(f) show an example of inferring a low-light image via NoiSER. As can be seen, given a low-light input, NoiSER successfully generates a visually pleasing enhanced result. We first introduce a required proposition to understand the specific mechanisms of NoiSER.

*Proposition 4:* Based on a model with random initialization, NoiSER tends to approach pure-central-grey while satisfying the gray-world hypothesis.

*Validation for Proposition 4:* Given an image  $I$ , similar to P-regression, we use Eqn.6 to measure the distance between  $I$  and the pure-central-grey, and use Eqn.2 to measure the satisfaction degree towards the gray-world hypothesis. The green and red curves in Fig.7(a) demonstrate the specific iterative processes. As can be seen, though we do not apply the equation for back propagation, the curves still converge smoothly, demonstrating the correctness of *Proposition 4*. Similar to P-regression, NoiSER training can also make pure-colors appear grey (see Fig.7(b)). We summarize the *Proposition 4* as the fact that NoiSER has the ability to learn a gray-world mapping.

It is clear that NoiSER satisfies all the properties of P-regression while smoothing the convergence curve towards

the gray-world hypothesis, which means NoiSER has possessed the ability to enhance a low-light image with pleasing visual effects. The final pipeline of training and inference and the detailed structure of the self-regression model (SRM) are shown in Fig.2. Please note that the designed SRM is equipped with the instance normalization layers. Besides, the inputs/outputs of the model are processed to the range [-1,1], and the median of which is exactly the mean  $\mu$  of the Gaussian distribution, i.e.,  $\mu = 0$ . In such setting, 0 roughly corresponds to the means of RGBs for normal-light image (see Table 2), i.e., zero-mean Gaussian distribution roughly describes the distribution of normal-light image.

However, we still face an essential question, i.e., “noise is totally different from low-light image, how does training with noise ensure low-light images can be mapped into normal light? ” In Section 3.3.1, we did not bother with the large distribution gap between noise and low-light image based on the MVF therapy, and directly perform self-regression training to obtain the result. Then, we will show that the instance normalization (IN) layer in SRM plays a similar role to “visual feedback” in MVF to bridge the training and inference. Specifically, in the training process, we use  $\mathcal{N}(0, \sigma^2)$  for the input. But because of the role of IN, even if the input in the reference process has a distribution shift, e.g.  $\mathcal{N}(\mu, \sigma^2)$ , the IN may potentially help rectify such a difference in feature layers. As a result, we can infer that the instance normalization layers may naturally remediate the overall magnitude/lighting of the input image. In addition, according to the image self-regression principle, the noise self-regression itself can reconstruct the texture and maintain a similar contrast between adjacent pixels as the input image. Combining these two, low-light image can be enhanced properly. However, if the output doesn’t meet the gray-world hypothesis, visual effect will become poor (e.g., P-regression in Section 3.3.3). Fortunately, Fig.7 demonstrates that NoiSER satisfies the gray-world hypothesis, which means that NoiSER may enhance low-light images with visually pleasing effects.

## 4. Experimental Results and Analysis

### 4.1. Experimental Settings

**Evaluated datasets.** We conduct experiments on two real-world image datasets (LOL [37] and LSRW [10]) and a multi-exposure dataset (SICE [2]). We train and test the methods on LOL (NoiSER does not use any task-related data but noise), and directly test them on LSRW using the pretrained model on LOL. Besides, we use SICE to measure the overexposure suppression capability.

**Evaluation metrics.** We use three widely-used performance metrics to evaluate the quantitative performance of each method, i.e., PSNR, SSIM [36] and NIQE [24]. Besides, we also use three application metrics to evaluate the possibility of each method for practical deployment and ap-

plication, i.e., training time (TT, in minutes), inference time (IT, in milliseconds) and number of parameters (No.P). In all experiments, we use  $\uparrow$  to indicate the higher the better, and use  $\downarrow$  to indicate the lower the better.

**Compared methods.** Two optimization-based methods (LIME [9] and Zhang et al. [41]) and six deep learning-based methods are included for comparison. For deep methods, different types of training data are leveraged to fit the model, i.e., paired (RetinexNet [37]), unpaired data (EnlightenGAN [12]) and zero-reference data (Zero-DCE [8], Zero-DCE++ [17], RUAS [20] and SCI [22]). Note that we mainly compare with the four zero-reference methods, as they are closer to our NoiSER in terms of data constraints.

**Our method.** In general, we adopt the standard Gaussian distribution  $\mathcal{N}(0, 1)$  to train the model. Training our NoiSER to full convergence (denoted as NoiSER-FC) will yield better quantitative performance, but we see that the visual effect appears to be obscured by a gray layer, as the goal of NoiSER is to learn a gray-world mapping. As such, we use an early stopping mechanism (denoted as NoiSER-ES), trading part of the quantitative performance for visual effect. In addition, to obtain better visual effect, we use a Gaussian distribution whose standard deviation  $\sigma = 3$  for self-regression (denoted NoiSER-Var3). We conduct experiments to evaluate the three versions of NoiSER.

**Implementation Details of NoiSER.** Based on PyTorch 1.10.1 [26] and Python 3.6.9, we train and evaluate our NoiSER on single NVIDIA RTX 2080 Ti GPU. We train our NoiSER for 2000 iterations (600 for NoiSER-ES) with a batch size of 1 and a fixed learning rate of 2e-4. We sample the noise from standard Gaussian distribution  $\mathcal{N}(0, 1)$  ( $\mathcal{N}(0, 3)$  for NoiSER-Var3) with a shape of  $104 \times 104$  for training. Besides, the Adam optimizer [14] is utilized for training with  $\beta_1 = 0.5$  and  $\beta_2 = 0.999$ . To alleviate the noise in the enhanced results, we add TV regularization during training. The total loss function used in the training of our proposed NoiSER is defined as

$$\mathcal{L}_{total} = \mathcal{L}_{l_1} + \mathcal{L}_{tv}. \quad (9)$$

### 4.2. Quantitative Evaluations

#### 4.2.1 Results on LOL Dataset

To examine the fitting ability of each method, we evaluate them on LOL in Table 3. We do not list the training time of RUAS [20], since it involves a process of neural architecture search. We find that: 1) optimization-based methods take longer inference time for the use of CPU for computation, which is a major impediment to real applications; 2) when training with paired/unpaired data, RetinexNet [37] and EnlightenGAN [12] yield better results due to the relatively strong constraints, but they have a larger number of parameters and slower inference speed; 3) the zero-reference methods require minimal computational resources, which are relatively efficient and have a stronger application potential;

Table 3. Numerical results on LOL dataset [37], with best performance marked in red and the second best marked in blue. Clearly, our NoiSER has significant advantages, not only for performance metrics but also for application metrics.

	Training data	Methods	Performance metrics			Application metrics		
			PSNR↑	SSIM↑	NIQE↓	TT↓(min)	IT↓(ms)	No.P↓
Optimization-based	-	LIME [9]	14.2216	0.5144	8.5828	-	104553.82	-
		Zhang et al. [41]	14.0181	0.5130	8.6111	-	138127.48	-
Deep learning-based	Paired	RetinexNet [37]	16.7740	0.4191	9.7294	3.17	95.40	1,333,841
	Unpaired	EnlightenGAN [12]	18.5413	0.6880	5.7111	90.05	10.91	6,959,553
	Zero-reference	Zero-DCE [8]	14.9672	0.5003	8.4228	16.33	2.24	79,416
		Zero-DCE++ [17]	14.8039	0.5161	8.3412	24.12	1.51	10,561
		RUAS [20]	16.4047	0.4996	5.9297	-	8.57	3,438
		SCI [22]	14.0226	0.5080	8.3315	2563.68	1.12	258
	Task-irrelevant	NoiSER-FC (Ours)	17.5748	0.7134	3.7285	1.10	1.21	1,323
		NoiSER-ES (Ours)	17.0250	0.6563	3.7206	0.58	1.21	1,323
		NoiSER-Var3 (Ours)	14.9257	0.5998	3.6806	1.10	1.21	1,323

Table 4. Comparison of generalization performance on LSRW dataset [10]. Clearly, our NoiSER has a stronger generalization ability over different datasets than all other methods.

Methods	LSRW (Huawei)		LSRW (Nikon)	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
LIME [9]	15.3376	0.4360	14.6362	0.3777
Zhang et al. [41]	14.0984	0.4327	13.0886	0.3677
RetinexNet [37]	16.8127	0.3948	13.4853	0.2934
EnlightenGAN [12]	16.8448	0.4832	14.9071	0.4065
Zero-DCE [8]	14.2002	0.3958	11.8197	0.3550
Zero-DCE++ [17]	14.2370	0.4163	11.0893	0.3675
RUAS [20]	15.6867	0.4909	12.1426	0.4372
SCI [22]	15.2583	0.4233	14.4512	0.4092
NoiSER-FC (Ours)	15.7268	0.5407	15.5537	0.4657
NoiSER-ES (Ours)	15.6968	0.5264	15.7090	0.4584
NoiSER-Var3 (Ours)	15.3001	0.5193	15.5260	0.4672

4) although our NoiSER does not use any task-related data for training, it is still highly competitive to the other methods across all metrics. From the comparison to the zero-reference methods, it can be seen that our NoiSER perfectly outperforms them all with significantly less training time.

#### 4.2.2 Results on LSRW Dataset

To test the generalization ability, we evaluate the methods on two test subsets, LSRW (Huawei) and LSRW (Nikon). The numerical results are shown in Table 4. We can find that: 1) the models fitted using the task-related data fail to generalize well to all datasets, since they perform well on some datasets but poorly on others. For example, EnlightenGAN [12] and RetinexNet [37] obtains the best records on LSRW (Huawei), but worse on LSRW (Nikon), which is inevitable for the usage of task-related data; 2) our NoiSER generalizes generally well and is more balanced<sup>1</sup> across different datasets, indicating our approach is closer to the

<sup>1</sup>It means that NoiSER generalizes stably on different datasets, i.e., it is less likely to be extremely good on one and extremely poor on the other.

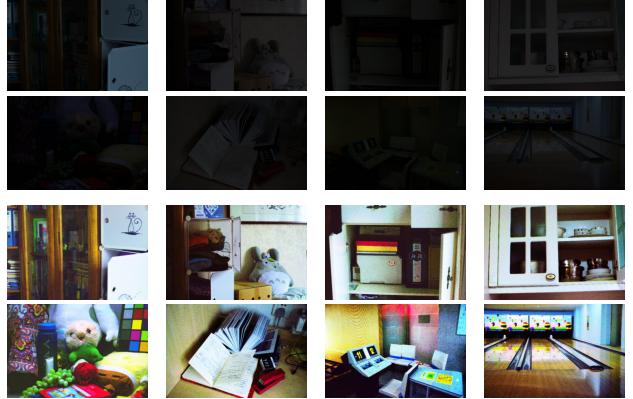


Figure 8. Intuitive visual display of our NoiSER-Var3 on most widely-used LOL dataset [37]. Just a simple and straightforward noise self-regression can yield a surprising visual effect.

essence of low-light enhancement; 3) from the comparison to those zero-reference methods that are more close to ours, our NoiSER obtains significant superiority in all metrics.

### 4.3. Visualization Evaluations

#### 4.3.1 Visual Analysis on LOL Dataset

NoiSER is easy to implement and can yield surprising results. We first show several images of LOL dataset to give a more intuitive view for the enhancement effect in Fig.8. As we can see, by simply performing a noise self-regression for training, all low-light images can be enhanced with rich texture, content and color information.

We then compare each method on an extremely dark image (from which human eyes hardly see anything) of LOL [37] dataset in Fig.9. We see that: 1) both optimization-based and deep learning methods using task-related data can enhance this very dark image to some extent, but the restored images are still inferior in terms of illumination enhancement, detail recovery and color preservation; 2) our NoiSER enhances this overdark image considerably, and

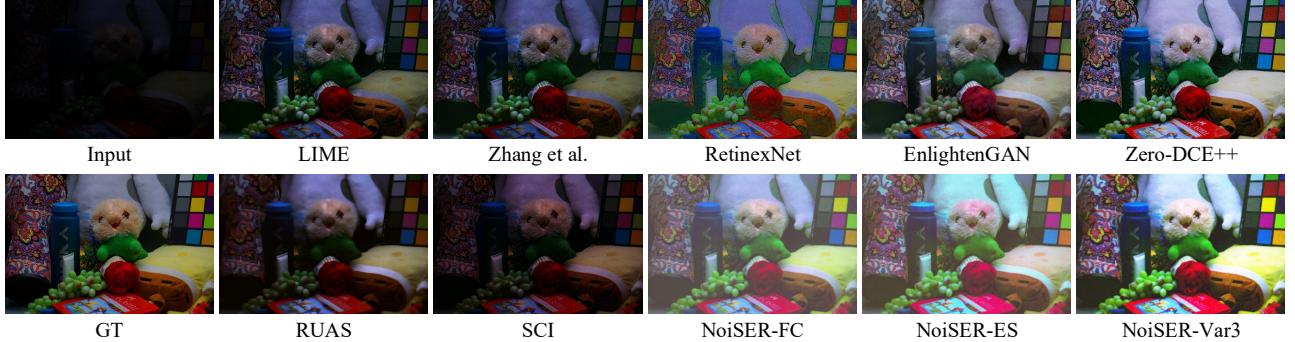


Figure 9. Visual results on LOL dataset [37], including LIME [9], Zhang et al. [41], RetinexNet [37], EnlightenGAN [12], Zero-DCE++ [17], RUAS [20], SCI [22] and our NoiSER. Clearly, our NoiSER obtains better effect, despite no task-related data are used for training.

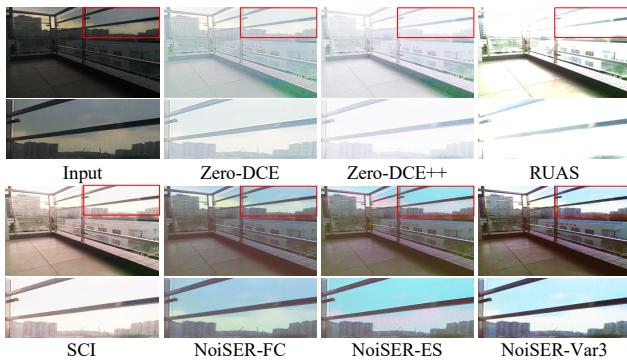


Figure 10. Visual results of different enhancement methods based on LSRW (Huawei) dataset [10]. Clearly, all other compared methods overexpose the brighter areas of the image, while our NoiSER effectively suppresses the exposure.

the restored image is even beyond the brightness and content naturalness of the ground-truth, which can be attributed to the learned gray-world mapping that always forces the channel means of a low-light image to be close to the central grey; 3) for the visual effects of several variants of our NoiSER, NoiSER-FC < NoiSER-ES < NoiSER-Var3, which is consistent with the above description.

#### 4.3.2 Visual Analysis on LSRW Dataset

To evaluate the generalization ability by visual analysis, we compare the visual results of each method on LSRW [10] dataset in Fig.10. We see that: 1) when input image is not very dark, zero-reference methods tend to produce overexposure enhanced images; 2) owing to the simple and straightforward training process, our NoiSER has a powerful generalization ability. From the comparison of the local details in Fig.10, it is clear that NoiSER recovers the texture of the images more accurately and gets better visual effect.

#### 4.4. Automated Overexposure Suppression

Finally, we evaluate each method to process the overexposed image of SICE dataset. As shown in Fig.11, existing methods Zero-DCE [8], Zero-DCE++ [17], RUAS [20] and SCI [22] obtain poor results and the processed images are



Figure 11. Comparison of the overexposure suppression effects on SICE dataset [2]. We see that all other competitors failed to handle the overexposed inputs appropriately, while our NoiSER has an automated exposure suppression capability.

almost completely corrupted, since their aim is to learn a mapping from low to high illumination. One possible solution is to use a large scale multi-exposure dataset to train the model, so that the model can handle both overdark and overexposed images. However, it is still tricky to handle the extremely complex illumination in reality. Therefore, we are delighted to say that our NoiSER has an inborn ability to suppress the overexposure, since it learns a gray-world mapping which can automatically turn extreme light or dark into moderates, as shown in “NoiSER-Var3” in Fig.11.

#### 4.5. Ablation Study.

To evaluate the importance and the effects of the instance normalization (IN) layers in remediating the overall magnitude/lighting of the input image, we conduct experiments by removing the IN layers with other setting unchanged, and the visual effects are shown in Fig.12. As can be seen, after removing the IN layers, NoiSER cannot work normally, which indicates the importance of the IN layer in our model.

### 5. Conclusion and Future Work

We have discussed a new problem, i.e., how to enhance a low-light image by deep learning without any task-related data. We have taken a bold and crazy perspective in thinking about the solution and the final result offers plenty of

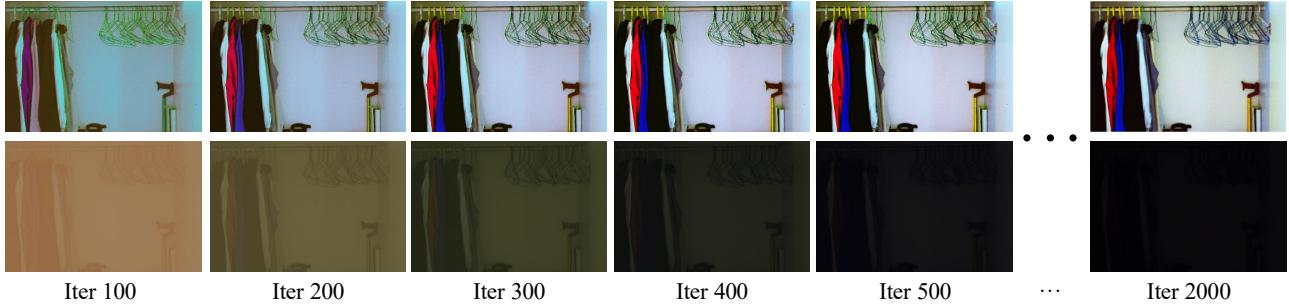


Figure 12. Visual ablation study on the instance normalization (IN) in SRM. Top row denotes the training process of NoiSER-Var3, while the bottom row denotes the training process of NoiSER-Var3 without IN. Clearly, the enhancement effect becomes very poor when IN is removed, i.e., IN is very important to naturally remediate the overall magnitude/lighting of the input image in our model.

surprises. We prove that the problem can easily solved by a straightforward noise self-regression approach that learns a simple convolutional neural network equipped with the instance-normalization layers by taking random noise as the input for training. Extensive experiments show our method can obtain highly-competitive and even better performance than current SOTA methods with different task-related data, in terms of enhancement ability, stable generalization capability, automated exposure suppression and negligible computational consumption. In the future, we will further think about how to optimize the noise for improving visual effect.

## Acknowledgment

The work described in this paper is partially supported by the National Natural Science Foundation of China (62072151, 61732007, 61932009, 62020106007) and the Anhui Provincial Natural Science Fund for the Distinguished Young Scholars (2008085J30). Zhao Zhang is the corresponding author of this paper.

## References

- [1] Gershon Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin institute*, 310(1):1–26, 1980. [1](#), [3](#)
- [2] Jianrui Cai, Shuhang Gu, and Lei Zhang. Learning a deep single image contrast enhancer from multi-exposure images. *IEEE Trans. Image Process.*, 27(4):2049–2062, 2018. [8](#), [10](#)
- [3] Turgay Çelik and Tardi Tjahjadi. Contextual and variational contrast enhancement. *IEEE Trans. Image Process.*, 20(12):3431–3441, 2011. [1](#)
- [4] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. *CoRR*, abs/2204.04676, 2022. [1](#)
- [5] Ziteng Cui, Guo-Jun Qi, Lin Gu, Shaodi You, Zenghui Zhang, and Tatsuya Harada. Multitask AET with orthogonal tangent regularity for dark object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada*, pages 2533–2542, 2021. [3](#)
- [6] Dengxin Dai and Luc Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems, Maui, HI, USA, November 4–7, 2018*, pages 3819–3824, 2018. [3](#)
- [7] Ruicheng Feng, Chongyi Li, Shangchen Zhou, Wenxiu Sun, Qingpeng Zhu, Jun Jiang, Qingyu Yang, Chen Change Loy, and Jinwei Gu. MIPI 2022 challenge on under-display camera image restoration: Methods and results. *CoRR*, abs/2209.07052, 2022. [1](#)
- [8] Chunle Guo, Chongyi Li, Jichang Guo, Chen Change Loy, Junhui Hou, Sam Kwong, and Runmin Cong. Zero-reference deep curve estimation for low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA*, pages 1777–1786, 2020. [1](#), [2](#), [3](#), [8](#), [9](#), [10](#)
- [9] Xiaojie Guo. LIME: A method for low-light image enhancement. In *Proceedings of the ACM Conference on Multimedia Conference, Amsterdam, The Netherlands*, pages 87–91, 2016. [1](#), [8](#), [9](#), [10](#)
- [10] Jiang Hai, Zhu Xuan, Ren Yang, Yutong Hao, Fengzhu Zou, Fang Lin, and Songchen Han. R2rnet: Low-light image enhancement via real-low to real-normal network. *CoRR*, abs/2106.14501, 2021. [7](#), [8](#), [9](#), [10](#)
- [11] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006. [2](#)
- [12] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. Enlightengan: Deep light enhancement without paired supervision. *IEEE Trans. Image Process.*, 30:2340–2349, 2021. [1](#), [2](#), [8](#), [9](#), [10](#)
- [13] Daniel J. Jobson, Zia-ur Rahman, and Glenn A. Woodell. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Trans. Image Process.*, 6(7):965–976, 1997. [1](#)
- [14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations, San Diego, CA, USA*, 2015. [8](#)
- [15] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations, Banff, AB, Canada*, 2014. [2](#)
- [16] Chulwoo Lee, Chul Lee, and Chang-Su Kim. Contrast enhancement based on layered difference representation of 2d

- histograms. *IEEE Trans. Image Process.*, 22(12):5372–5384, 2013. 1
- [17] Chongyi Li, Chunle Guo, and Chen Change Loy. Learning to enhance low-light image via zero-reference deep curve estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(8):4225–4238, 2022. 1, 2, 3, 8, 9, 10
- [18] Dong Liang, Ling Li, Mingqiang Wei, Shuo Yang, Liyan Zhang, Wenhan Yang, Yun Du, and Huiyu Zhou. Semantically contrastive learning for low-light image enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence, Virtual*, pages 1555–1563, 2022. 3
- [19] Jinxiu Liang, Jingwen Wang, Yuhui Quan, Tianyi Chen, Jiaying Liu, Haibin Ling, and Yong Xu. Recurrent exposure generation for low-light face detection. *IEEE Trans. Multim.*, 24:1609–1621, 2022. 3
- [20] Risheng Liu, Long Ma, Jiaao Zhang, Xin Fan, and Zhongxuan Luo. Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, virtual*, pages 10561–10570, 2021. 2, 8, 9, 10
- [21] Kin Gwn Lore, Adedotun Akintayo, and Soumik Sarkar. Llnet: A deep autoencoder approach to natural low-light image enhancement. *Pattern Recognit.*, 61:650–662, 2017. 1
- [22] Long Ma, Tengyu Ma, Risheng Liu, Xin Fan, and Zhongxuan Luo. Toward fast, flexible, and robust low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA*, pages 5627–5636, 2022. 1, 2, 8, 9, 10
- [23] Candy McCabe. Mirror visual feedback therapy. a practical approach. *Journal of hand therapy*, 24(2):170–179, 2011. 3
- [24] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a “completely blind” image quality analyzer. *IEEE Signal Process. Lett.*, 20(3):209–212, 2013. 8
- [25] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA*, pages 257–265, 2017. 1
- [26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. 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 *Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada*, pages 8024–8035, 2019. 8
- [27] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA*, pages 779–788, 2016. 1
- [28] Jiahuan Ren, Zhao Zhang, Richang Hong, Mingliang Xu, Yi Yang, and Shuicheng Yan. Seeing through the noisy dark: Toward real-world low-light image enhancement and denoising. *CoRR*, abs/2210.00545, 2022. 3
- [29] Xutong Ren, Wenhan Yang, Wen-Huang Cheng, and Jiaying Liu. LR3M: robust low-light enhancement via low-rank regularized retinex model. *IEEE Trans. Image Process.*, 29:5862–5876, 2020. 1
- [30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention, Germany*, volume 9351, pages 234–241, 2015. 1
- [31] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA*, pages 9446–9454, 2018. 2
- [32] Yecong Wan, Yuanshuo Cheng, Mingwen Shao, and Jordi González. Image rain removal and illumination enhancement done in one go. *Knowl. Based Syst.*, 252:109244, 2022. 3
- [33] Bo Wang, Zhao Zhang, Mingbo Zhao, Xiaojie Jin, Mingliang Xu, and Meng Wang. OSIC: A new one-stage image captioner coined. *CoRR*, abs/2211.02321, 2022. 1
- [34] Wenjing Wang, Xinhao Wang, Wenhan Yang, and Jiaying Liu. Unsupervised face detection in the dark. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. 3
- [35] Wenjing Wang, Wenhan Yang, and Jiaying Liu. Hla-face: Joint high-low adaptation for low light face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, virtual*, pages 16195–16204, 2021. 3
- [36] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004. 8
- [37] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. In *Proceedings of the British Machine Vision Conference, Newcastle, UK*, page 155, 2018. 1, 7, 8, 9, 10
- [38] Yanyan Wei, Zhao Zhang, Yang Wang, Mingliang Xu, Yi Yang, Shuicheng Yan, and Meng Wang. Deraincyclegan: Rain attentive cyclegan for single image deraining and rain-making. *IEEE Trans. Image Process.*, 30:4788–4801, 2021. 1
- [39] Ke Xu, Xin Yang, Baocai Yin, and Rynson W. H. Lau. Learning to restore low-light images via decomposition-and-enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA*, pages 2278–2287, 2020. 1
- [40] Syed Waqas Zamir, Aditya Arora, Salman H. Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, virtual*, pages 14821–14831, 2021. 1
- [41] Qing Zhang, Yongwei Nie, and Wei-Shi Zheng. Dual illumination estimation for robust exposure correction. *Comput. Graph. Forum*, 38(7):243–252, 2019. 1, 8, 9, 10
- [42] Yonghua Zhang, Xiaojie Guo, Jiayi Ma, Wei Liu, and Jiawan Zhang. Beyond brightening low-light images. *Int. J. Comput. Vis.*, 129(4):1013–1037, 2021. 1

- [43] Yonghua Zhang, Jiawan Zhang, and Xiaojie Guo. Kindling the darkness: A practical low-light image enhancer. In *Proceedings of the ACM International Conference on Multimedia, Nice, France*, pages 1632–1640, 2019. 1
- [44] Zhao Zhang, Huan Zheng, Richang Hong, Mingliang Xu, Shuicheng Yan, and Meng Wang. Deep color consistent network for low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA*, pages 1889–1898, 2022. 1
- [45] Suiyi Zhao, Zhao Zhang, Richang Hong, Mingliang Xu, Yi Yang, and Meng Wang. FCL-GAN: A lightweight and real-time baseline for unsupervised blind image deblurring. In *Proceedings of the ACM International Conference on Multimedia, Lisboa, Portugal*, pages 6220–6229, 2022. 1
- [46] Suiyi Zhao, Zhao Zhang, Richang Hong, Mingliang Xu, Haifan Zhang, Meng Wang, and Shuicheng Yan. Crnet: Unsupervised color retention network for blind motion deblurring. In *Proceedings of the ACM International Conference on Multimedia, Lisboa, Portugal*, pages 6193–6201, 2022. 1
- [47] Yuzhi Zhao, Yongzhe Xu, Qiong Yan, Dingdong Yang, Xuehui Wang, and Lai-Man Po. D2hnet: Joint denoising and deblurring with hierarchical network for robust night image restoration. *CoRR*, abs/2207.03294, 2022. 3
- [48] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy*, pages 2242–2251, 2017. 1
- [49] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proc. IEEE*, 109(1):43–76, 2021. 3