

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/320920649>

# MSR-net:Low-light Image Enhancement Using Deep Convolutional Network

Article · November 2017

CITATIONS

81

READS

2,540

6 authors, including:



**Liang Shen**

Huazhong University of Science and Technology

5 PUBLICATIONS 99 CITATIONS

SEE PROFILE



**Shihao Liu**

Hainan University

22 PUBLICATIONS 168 CITATIONS

SEE PROFILE



**ma Jie**

Huazhong University of Science and Technology

102 PUBLICATIONS 1,333 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Doppler velocimetry pulsar navigation method in high dynamic environment [View project](#)

# MSR-net: Low-light Image Enhancement Using Deep Convolutional Network

Liang Shen\*, Zihan Yue\*, Fan Feng, Quan Chen, Shihao Liu, and Jie Ma  
Institute of Image Recognition and Artificial Intelligence  
Huazhong University of Science and Technology, Wuhan, China

## Abstract

Images captured in low-light conditions usually suffer from very low contrast, which increases the difficulty of subsequent computer vision tasks in a great extent. In this paper, a low-light image enhancement model based on convolutional neural network and Retinex theory is proposed. Firstly, we show that multi-scale Retinex is equivalent to a feedforward convolutional neural network with different Gaussian convolution kernels. Motivated by this fact, we consider a Convolutional Neural Network (MSR-net) that directly learns an end-to-end mapping between dark and bright images. Differently from existing approaches, low-light image enhancement in this paper is regarded as a machine learning problem. In this model, most of the parameters are optimized by back-propagation, while the parameters of traditional models depend on the artificial setting. Experiments on a number of challenging images reveal the advantages of our method in comparison with other state-of-the-art methods from the qualitative and quantitative perspective.

## 1. Introduction

There is no doubt that high-quality image plays a critical role in computer vision tasks such as object detection and scene understanding. Unfortunately, the images obtained in reality are often degraded in some cases. For example, when captured in low-light conditions, images always suffer from very low contrast and brightness, which increases the difficulty of subsequent high-level tasks in a great extent. Figure 1(a) provides one case, from which many details have been buried into the dark background. Due to the fact that in many cases only low-light images can be captured, several low-light image enhancement methods have been proposed to overcome this problem. In general, these methods can be categorized into two groups: histogram-based methods and Retinex-based methods.

In this paper, a novel low-light image enhancement

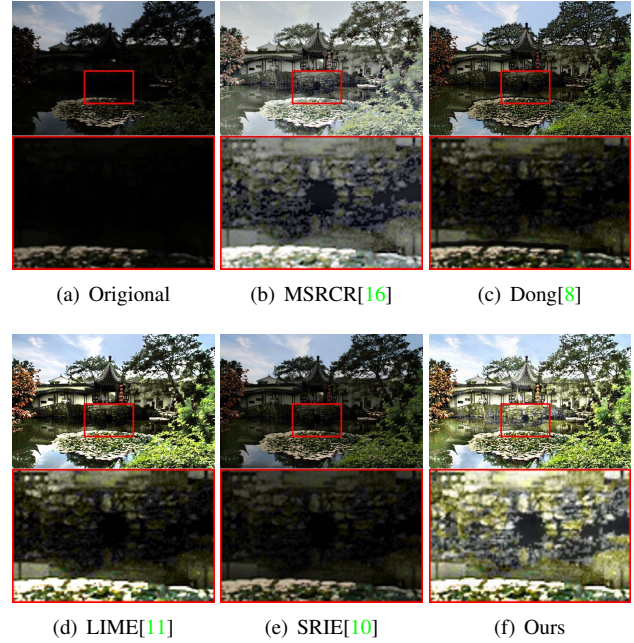


Figure 1. An example result of our image enhancement method and others state-of-the-art methods.

model based on convolutional neural network and Retinex theory is proposed. To the best of our knowledge, this is the first work of using convolutional neural network and Retinex theory to solve low-light image enhancement. Firstly, we explain that multi-scale Retinex is equivalent to a feedforward convolutional neural network with different Gaussian convolution kernels. The main drawback of multi-scale Retinex is that the parameters of kernels depend on artificial settings rather than learning from data, which makes the accuracy and flexibility of the model reduce in some way. Motivated by this fact, we put forward a Convolutional Neural Network (MSR-net) that directly learns an end-to-end mapping between dark and bright images. Our method differs fundamentally from existing approaches. We regard low-light image enhancement as a supervised learning problem. Furthermore, the surround functions in Retinex theory [19] are formulated as convolutional layers, which are

\* Authors contributed equally.

involved in optimization by back-propagation.

Overall, the contribution of our work can be boiled down to three aspects: First of all, we establish a relationship between multi-scale Retinex and feedforward convolutional neural network. Secondly, we consider low-light image enhancement as a supervised learning problem where dark and bright images are treated as input and output respectively. Last but not least, experiments on a number of challenging images reveal the advantages of our method in comparison with other state-of-the-art methods. Figure 1 gives an example. Our method achieves a brighter and more natural result with a clearer texture and richer details.

## 2. Related Work

### 2.1. Low-light Image Enhancement

In general, low-light image enhancement can be categorized into two groups: histogram-based methods and Retinex-based methods.

Directly amplifying the low-light image by histogram transformation is probably the most intuitive way to lighten the dark image. One of the simplest and most widely used technique is histogram equalization(HE), which makes the histogram of the whole image as balanced as possible. Gamma Correction is also a great method to enhance the contrast and brightness by expanding the dark regions and compressing the bright ones in the mean time. However, the main drawback of these method is that each pixel in the image is treated individually, without the dependence of their neighborhoods, which makes the result look inconsistent with real scenes. To resolve the mentioned problems above, variational methods which use different regularization terms on the histogram have been proposed. For example, contextual and variational contrast enhancement [5] tries to find a histogram mapping to get large gray-level difference.

In this work, Retinex-based methods have been taken into more account. Retinex theory is introduced by Land [19] to explain the color perception property of the human vision system. The dominant assumption of Retinex theory is that the image can be decomposed into reflection and illumination. Single-scale Retinex(SSR) [17], based on the center/surround Retinex, is similar to the difference-of-Gaussian(DOG) function which is widely used in natural vision science, and it treats the reflectance as the final enhanced result. Multi-scale Retinex(MSR) [16] can be considered as a weighted sum of several different SSR outputs. However, these methods often look unnatural. Further, modified MSR [16] applies the color restoration function(CRF) in the chromaticity space to eliminate the color distortions and gray zones evident in the MSR output. Recently, the method proposed in [11] tries to estimate the illumination of each pixel by finding the maximum value

in R, G and B channel, then refines the initial illumination map by imposing a structure prior on it. Seonhee Park *et al.* [23] use the variational-optimization-based Retinex algorithm to enhance the low-light image. Fu *et al.* [10] propose a new weighted variational model to estimate both the reflection and the illumination. Different from conventional variational models, their model can preserve the estimated reflectance with more details. Inspired by the dark channel method on de-hazing, [8] finds the inverted low-light image looks like haze image. They try to remove the inverted low-light image of haze by using the method proposed in [12] and then invert it again to get the final result.

### 2.2. Convolutional Neural Network for Low-level Vision Tasks

Recently, powerful capability of deep neural network [18] has led to dramatic improvements in object recognition [13], object detection [24], object tracking [29], semantic segmentation [20] and so on. Besides these high-level vision tasks, deep learning has also shown great ability at low-level vision tasks. For instance, Dong *et al.* [7] train a deep convolutional neural network (SRCNN) to accomplish the image super-resolution tasks. Fu *et al.* [9] try to remove rain from single images via a deep detail network. Cai *et al.* [4] propose a trainable end-to-end system named DehazeNet, which takes a hazy image as input and outputs its medium transmission map that is subsequently used to recover a haze-free image via atmospheric scattering model.

## 3. CNN Network for Low-light Image Enhancement

We elaborate that multi-scale Retinex as a low-light image enhancement method is equivalent to a feedforward convolutional neural network with different Gaussian convolution kernels from a novel perspective. Subsequently, we propose a Convolutional Neural Network (MSR-net) that directly learns an end-to-end mapping between dark and bright images.

### 3.1. Multi-scale Retinex is a CNN Network

The dominant assumption of Retinex theory is that the image can be decomposed into reflection and illumination:

$$I(x, y) = r(x, y) \cdot S(x, y) \quad (1)$$

Where  $I$  and  $r$  represent the captured image and the desired recovery, respectively. Single-scale Retinex(SSR) [17], based on the center/surround Retinex, is similar to the difference-of-Gaussian(DOG) function which is widely used in natural vision science. Mathematically, this takes the form

$$R_i(x, y) = \log I_i(x, y) - \log [F(x, y) * I_i(x, y)] \quad (2)$$

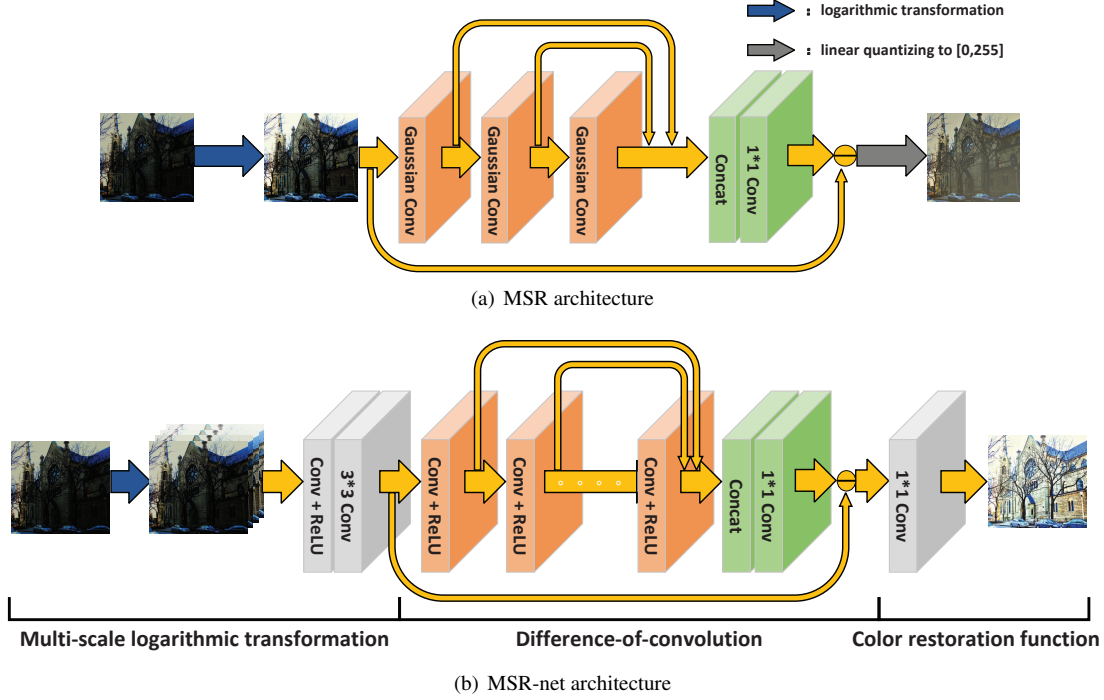


Figure 2. The architecture of MSR and our proposed MSR-net. Obviously, MSR-net is similar to the MSR to some extent. In such a network, it is divided into three parts, Multi-scale Logarithmic Transformation, Difference-of-convolution and Color Restoration Function.

Where  $R_i(x, y)$  is the associated Retinex output,  $I_i(x, y)$  is the image distribution in the  $i^{th}$  color spectral band,  $*$  denotes the convolution operation, and  $F(x, y)$  is the Gaussian surround function

$$F(x, y) = K e^{-\frac{x^2+y^2}{2c^2}} \quad (3)$$

Where  $c$  is the standard deviation of Gaussian function, and  $K$  is selected such that

$$\iint F(x, y) dx dy = 1 \quad (4)$$

By changing the position of the logarithm in the above formula and setting

$$R_i(x, y) = \log I_i(x, y) - [\log I_i(x, y)] * F(x, y) \quad (5)$$

In this way we obtained a classic high pass linear filter, but applied to  $\log I$  instead of  $I$ . The above two formulas, Equation 2 and Equation 5, are of course not equivalent in mathematical form. The former is the logarithm of ratio between the image and a weighted average of it, while the latter is the logarithm of ratio between the image and a weighted product. Actually, this amounts to choosing between an arithmetic mean and a geometric mean. Experiments show that these two methods are not much different. In this work we choose the latter for simplicity.

Further, multi-scale Retinex(MSR) [16] is considered as a weighted sum of the outputs of several different SSR outputs. Mathematically,

$$R_{MSR_i} = \sum_{n=1}^N w_n R_{n_i} \quad (6)$$

Where  $N$  is the number of scales,  $R_{n_i}$  denotes the  $i^{th}$  component of the  $n^{th}$  scale,  $R_{MSR_i}$  represents the  $i^{th}$  spectral component of the MSR output and  $w_n$  is the weight associated with the  $n^{th}$  scale.

After experimenting with one small scale (standard deviation  $c < 20$ ) and one large scale (standard deviation  $c > 200$ ), the need for the third intermediate scale is immediately apparent in order to eliminate the visible “halo” artifacts near strong edges [16]. Thus, the formula is as follows:

$$R_{MSR_i}(x, y) = \frac{1}{3} \sum_{n=1}^3 \{ \log I_i(x, y) - [\log I_i(x, y)] * F_n(x, y) \} \quad (7)$$

More concrete, we have

$$R_{MSR_i}(x, y) = \log I_i(x, y) - \frac{1}{3} \log I_i(x, y) * \left[ \sum_{n=1}^3 K_n e^{-\frac{x^2+y^2}{2c_n^2}} \right] \quad (8)$$

Noticing the fact that convolution of two Gaussian functions is still a Gaussian function, whose variance is equal to the sum of two original variance. Therefore, we can represent the above equation 8 by using the cascading structure, as Figure 2(a) shows.

The three cascading convolution layers are considered as three different Gaussian kernels. More concrete, the parameter of the first convolution layer is based on a Gaussian distribution, whose variance is  $c_1^2$ . Similarly, the variances of the second and the third convolution layers are  $c_2^2 - c_1^2$ ,  $c_3^2 - c_2^2$ , respectively. At last, the concatenation and  $1 \times 1$  convolution layers represent the weighted average. In a word, multi-scale Retinex is practically equivalent to a feedforward convolutional neural network with a residual structure.

### 3.2. Proposed Method

In the previous section, we put forward the fact that multi-scale Retinex is equivalent to a feedforward convolutional neural network. In this section, inspired by the novel fact, we consider a convolutional neural network to solve the low-light image enhancement problem. Our method outlined in Figure 2(b) differs fundamentally from existing approaches, which takes low-light image enhancement as a supervised learning problem. The input and output data correspond to the low-light and bright images, respectively. More detail about our training dataset will be explained in section 4.

Our model consists of three components: **Multi-scale Logarithmic Transformation**, **Difference-of-convolution** and **Color Restoration Function**. Compared to single-scale logarithmic transformation in MSR, our model attempts to use multi-scale logarithmic transformation, which has been verified to achieve a better performance in practice. Figure 7 gives an example. Difference-of-convolution plays an analogous role with difference-of-Gaussian in MSR, and so does color restoration function. The main difference between our model and original MSR is that most of the parameters in our model are learned from the training data, while the parameters in MSR such as the variance and other constant depend on the artificial setting.

Formally, we denote the low-light image as input  $X$  and corresponding bright image as  $Y$ . Suppose  $f_1$ ,  $f_2$ ,  $f_3$  denote three sub-functions: multi-scale logarithmic transformation, difference-of-convolution, and color restoration function. Our model can be written as the composition of three functions:

$$f(X) = f_3(f_2(f_1(X))) \quad (9)$$

**Multi-scale Logarithmic Transformation:** Multi-scale logarithmic transformation  $f_1(X)$  takes the original low-light image  $X$  as input and computes the same size output

$X_1$ . Firstly, the dark image is enhanced by several difference logarithmic transformation. The formula is as follows:

$$M_j = \log_{v_j+1}(1 + v_j \cdot X), j = 1, 2, \dots, n \quad (10)$$

Where  $M_j$  denotes the output of the  $j^{th}$  scale with the logarithmic base  $v_j + 1$ , and  $n$  denotes the number of logarithmic transformation function. Next, we concatenate these 3D tensors  $M_j$  (3 channels  $\times$  width  $\times$  height) to a larger 3D tensor  $M$  ( $3n$  channels  $\times$  width  $\times$  height) and then make it go through convolutional and ReLU layers.

$$M = [M_1, M_2, \dots, M_n] \quad (11)$$

$$X_1 = \max(0, M * W_{-1} + b_{-1}) * W_0 + b_0 \quad (12)$$

Where  $*$  denotes a convolution operator,  $W_{-1}$  is a convolution kernel that shrinks the  $3n$  channels to 3 channels,  $\max(0, \cdot)$  corresponds to a ReLU and  $W_0$  is a convolution kernel with three output channels for better nonlinear representation. As we can see from the above operation, this part is mainly designed to get a better image via weighted sums of multiple logarithmic transformations, which accelerates the convergence of the network.

**Difference-of-convolution:** Difference-of-convolution function  $f_2$  takes the input  $X_1$  and computes the same size output  $X_2$ . Firstly, the input  $X_1$  passes through multi-convolutional layers.

$$H_0 = X_1 \quad (13)$$

$$H_m = \max(0, H_{m-1} * W_m + b_m), m = 1, 2, \dots, K \quad (14)$$

Where  $m$  denotes the  $m^{th}$  convolutional layer,  $K$  is equal to the number of convolutional layers. And  $W_m$  represents the  $m^{th}$  kernel. As mentioned earlier in section 3.1,  $H_1, H_2, \dots, H_K$  are considered as smooth images at different scales, then we concatenate these 3D tensors  $H_m$  to a larger 3D tensor  $H$  and get it pass the convolutional layer:

$$H = [H_1, H_2, \dots, H_K] \quad (15)$$

$$H_{K+1} = H * W_{K+1} + b_{K+1} \quad (16)$$

Where the  $W_{K+1}$  is a convolutional layer with three output channels and the  $1 \times 1$  receptive field, which is equivalent to averaging these  $K$  images. Similar to MSR, the output of  $f_2$  is the subtraction between  $X_1$  and  $H_{K+1}$ :

$$X_2 = f_2(X_1) = X_1 - H_{K+1} \quad (17)$$

**Color Restoration Function:** Considering that MSR result often looks unnatural, modified MSR [16] applies the color restoration function(CRF) in the chromaticity space to eliminate the color distortions and gray zones evident in the MSR output. In our model CRF is imitated by a  $1 \times 1$  convolutional layer with three output channels:

$$\hat{Y} = f_3(X_2) = X_2 * W_{K+2} + b_{K+2} \quad (18)$$



Where  $\hat{Y}$  is the final enhanced image. For more visualization, a low light image and the results of  $f_1, f_2, f_3$  have been shown in Figure 3 respectively.

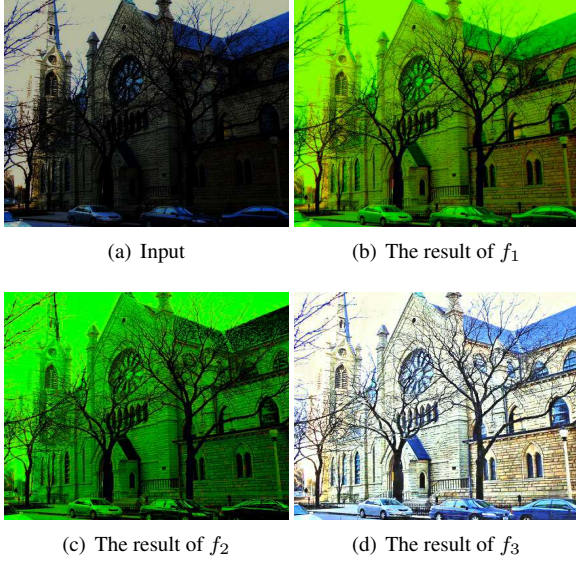


Figure 3. A low light image and the results of  $f_1, f_2, f_3$

### 3.3. Objective function

The goal of our model is to train a deep convolutional neural network to make the output  $f(X)$  and the label  $Y$  as close as possible under the criteria of Frobenius norm.

$$L = \frac{1}{N} \sum_{i=1}^N \|f(X_i) - Y_i\|_F^2 + \lambda \sum_{i=1}^{K+2} \|W_i\|_F^2 \quad (19)$$

Where  $N$  is the number of training samples,  $\lambda$  represents the regularization parameter.

Weights  $W$  and bias  $b$  are the whole parameters in our model. Besides, the regularization parameter  $\lambda$ , the number of logarithmic transformation function  $n$ , the scale of logarithmic transformation  $v$  and the number of convolutional layers  $K$ , are considered as the hyper-parameters in the model. The parameters in our model are optimized by back-propagation, while the hyper-parameters are chosen by grid-search. More detail about the sensitivity analysis of hyper-parameters will be elaborated in section 4.

## 4. Experiments

In this section, we elaborately construct an image dataset and spend about 10 hours on training the end-to-end network by using the Caffe software package [15]. To evaluate the performance of our method, we use both the synthetic test data, the public real-world dataset and compare with four recent state-of-the-art low-light image enhancement methods. At the same time, we analyse the running

time and evaluate the effect of hyper-parameters to the final results.

### 4.1. Image Dataset Generation

On the one hand, in order to learn the parameters of the MSR-net, we construct a new image dataset, which contains a great amount of high quality(HQ) and low-light(LL) natural images. An important consideration is that all the image should be selected in real world scenes. We collect more than 20,000 images from the UCID dataset [26], the BSD dataset [2] and Google image search. Unfortunately, many of these images suffer significant distortions or contain inappropriate content. Images with obvious distortions such as heavy compression, strong motion blur, out of focus blur, low contrast, underexposure or overexposure and substantial sensor noise are deleted firstly. After this, we exclude inappropriate images such as too small or too large size, cartoon and computer generated content to obtain 1,000 better source images. Then, for each image, we use Photoshop method [1] to figure out the ideal brightness and contrast settings, then process them one by one to get the high quality(HQ) images with the best visual effect. At last, each HQ image is used to generate 10 low-light(LL) images by reducing brightness and contrast randomly and using gamma correction with stochastic parameters. So we attain a dataset containing 10,000 pairs of HQ/LL images. Further, 8,000 images in the dataset are randomly selected to generate one million  $64 \times 64$  HQ/LL patch pairs for training. And the remaining 2,000 images pairs are used to test the trained network during training(please see more details about the dataset generation in the supplemental materials).

On the other hand, in order to evaluate our approach objectively and impartially, we choose the real-world low-light images from the public MEF dataset [21, 33], NPE dataset [30] and VV dataset [27, 28].

### 4.2. Training Setup

We set the depth of MSR-net to  $K = 10$ , and use Adam with weight decay of  $10^{-6}$  and a mini-batch size of 64. We start with a learning rate of  $10^{-4}$ , dividing it by 10 at 100K and 200K iterations, and terminate training at 300K iterations. During our experiments, we found that the network with multi-scale logarithmic transformation performs better than that with single-scale logarithmic transformation, so we set the number of logarithmic transformation function  $n = 4$  and  $v = 1, 10, 100, 300$  respectively. The size of convolution kernel has been described partially in the previous section 3.2, and the specific values are shown in the Table 3.

### 4.3. Results on synthetic test data

Figure 4 shows visual comparison for three synthesized low light images. As we can see, the result of MSRRCR [16]

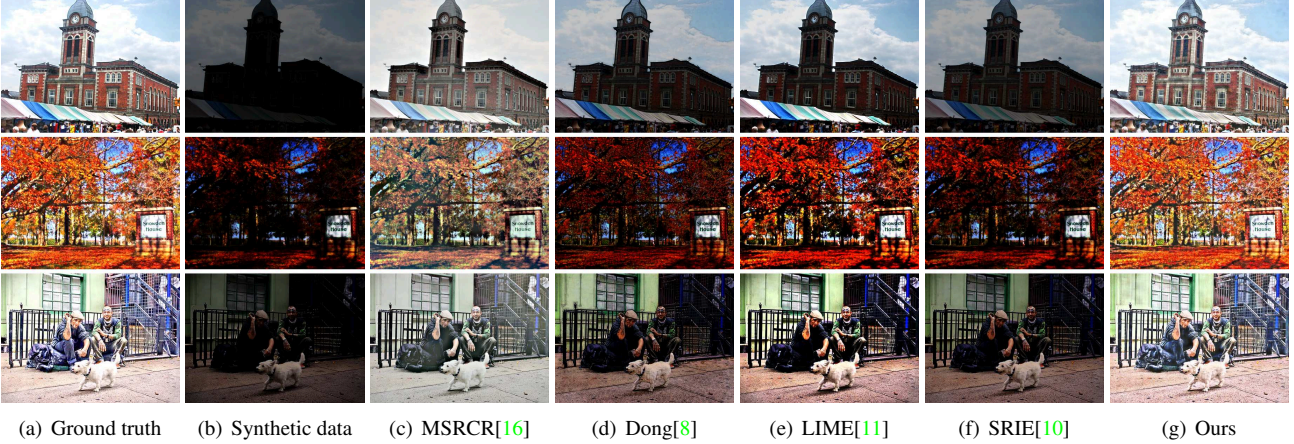


Figure 4. Results using different methods on synthesized test images

Table 1. Quantitative measurement results using SSIM/NIQE on synthesized test images

Dataset	Ground truth	Synthetic image	MSRCR[16]	Dong[8]	LIME[11]	SRIE[10]	Ours
1st row	1/2.69	0.35/6.44	0.89/3.29	0.64/3.57	0.74/3.04	0.58/3.72	<b>0.91/2.54</b>
2nd row	1/3.17	0.23/4.43	0.90/3.61	0.42/4.69	0.68/4.23	0.39/3.94	<b>0.94/3.54</b>
3rd row	1/3.46	0.26/3.73	0.81/3.34	0.47/3.75	0.62/3.89	0.48/3.57	<b>0.91/3.33</b>
2,000 test images	1/3.67	0.74/3.53	0.90/3.50	0.69/4.16	0.84/3.89	0.63/3.66	<b>0.92/3.46</b>

looks unnatural, the method proposed by Dong [8] always generates unexpected black edge and the result of SRIE [10] tends to be dark in some extent. LIME [11] has a similar result to our method, while ours achieves better performance in dark regions.

Since the ground truth is known for the synthetic test data, we use SSIM [31] for a quantitative evaluation and NIQE [22] to assess the natural preservation. A higher SSIM indicates that the enhanced image is closer to the ground truth, while a lower NIQE value represents a higher image quality. All the best results are boldfaced. As shown in Table 1, our method achieves higher SSIM and lower NIQE average than other methods for 2,000 test images.

#### 4.4. Results on real-world data

Figure 5 also shows the visual comparison for three real-world low-light images. As shown in every red rectangle, our method MSR-net always achieves better performance in dark regions. More specifically, in the first and second image we get brighter result. In the third image we achieve more natural result, for instance, the tree has been enhanced to be bright green. Besides, from the Garden image in Figure 1, our result gets a clearer texture and richer details than other methods.

Besides the NIQE to evaluate the image quality, we assess the detail enhancement through the Discrete Entropy [32]. A higher discrete entropy shows that the color is richer and the outline is clearer. We delete the high-light images and only keep the low-light images on the MEF

dataset [21, 33], NPE dataset [30] and VV dataset [27, 28] to evaluate our method. As shown in Table 2, for different dataset, MSR-net can also obtain lower NIQE and higher discrete entropy.

Considering the fact that dealing with real-world low light images sometimes causes noise, we attempt to use a denoising algorithm BM3D [6] as a post-processing. An example is shown in Figure 6, where removing the noise after our deep network can further improve the visual quality on real-world low light image.

#### 4.5. Color Constancy

In addition to enhance the dark image, our model also does a good job in correcting the color. Figure 4 provides some examples. As we can see, our enhanced image is much more similar to the ground truth. To evaluate the performance of the different algorithms, the angular error [14] between the ground truth image  $Y$  and model result  $\hat{Y}$  is used:

$$\varepsilon = \arccos \left( \frac{\langle Y, \hat{Y} \rangle}{\|Y\| \cdot \|\hat{Y}\|} \right) \quad (20)$$

A lower error indicates that the color of enhanced image is closer to the ground truth. Table 4 gives some specific result of the images in Figure 4.

#### 4.6. Running time on test data

Compared with other non-deep methods, our approach processes the low-light images efficiently. Table 5 shows



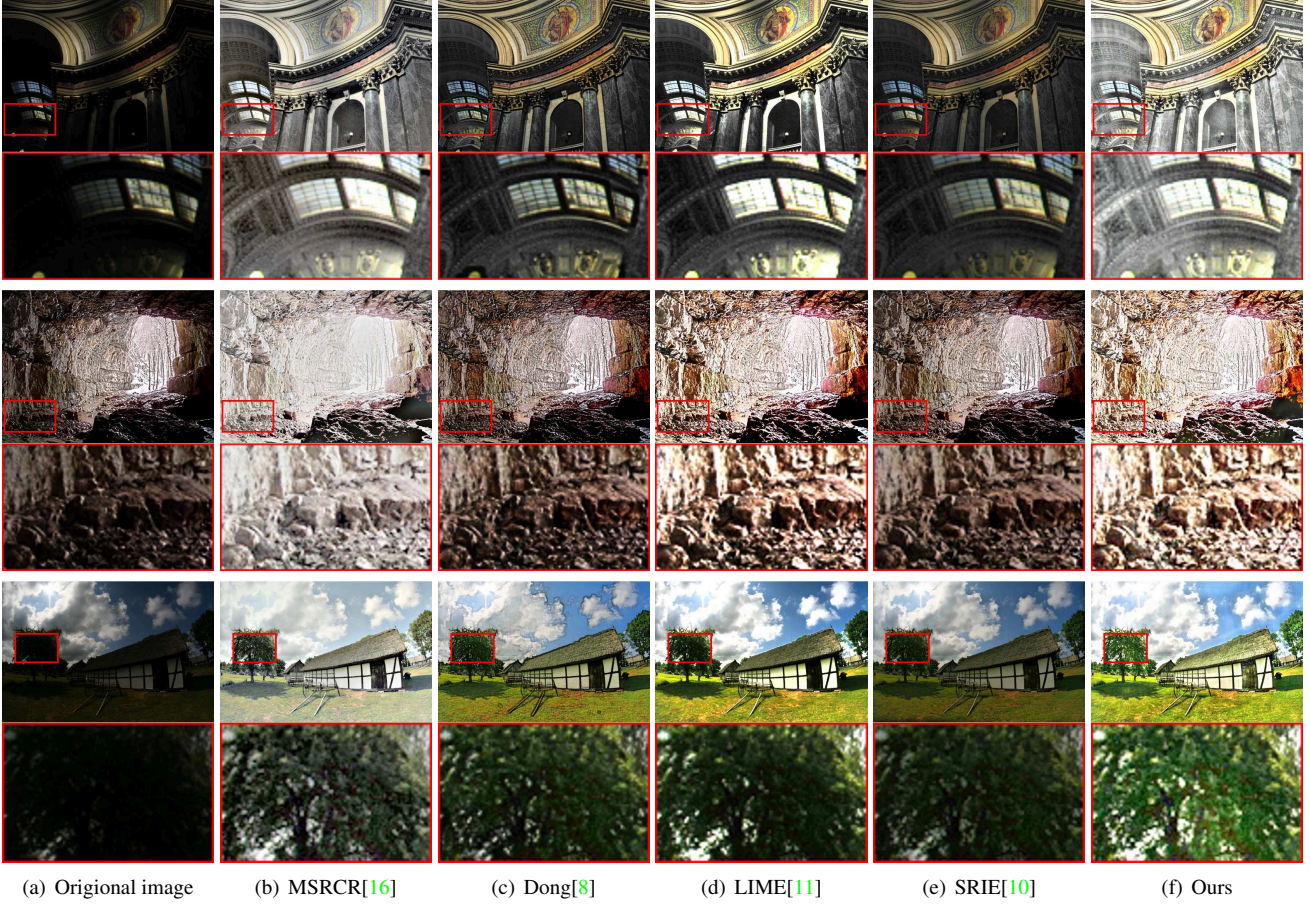


Figure 5. The results of real-world images using different methods and zoomed in region: (top-to-bottom): “Madison”, “Cave”, “Kluki”.

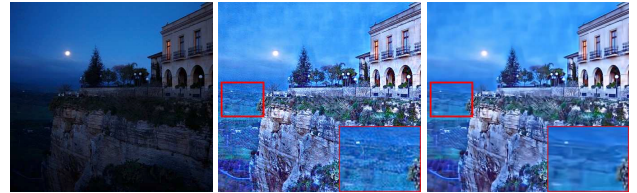
Table 2. Quantitative measurement results using Discrete Entropy/NIQE on the real-world images

	Real-world images	MSRCR[16]	Dong[8]	LIME[11]	SRIE[10]	Ours
Madison	9.56/3.05	15.01/3.13	11.19/3.63	10.74/3.43	10.97/4.18	<b>15.49/3.06</b>
Cave	13.75/3.21	14.91/5.91	14.54/6.35	12.66/6.19	14.66/4.44	<b>15.75/5.84</b>
Kluki	10.36/3.19	14.62/2.33	12.71/2.43	12.86/2.78	12.60/2.42	<b>15.91/1.81</b>
MEF dataset	11.02/4.31	15.47/3.32	13.27/4.06	13.52/3.69	13.16/3.63	<b>16.28/3.03</b>
NPE dataset	12.54/4.13	14.61/4.37	14.41/4.12	13.93/4.25	14.12/4.05	<b>15.34/3.73</b>
VV dataset	12.47/3.52	17.87/2.54	14.80/2.76	15.79/2.48	14.25/2.78	<b>18.26/2.29</b>

Table 3. MSR-net network configuration. The convolutional layer parameters are denoted as “conv(receptive field size)-(numbers of output channels)”. The ReLU activation is not shown for brevity

$W_{-1}$	conv1-32	$W_{11}$	conv1-3
$W_0$	conv3-3	$W_{12}$	conv1-3
$W_i, i = 1, \dots, 10$	conv3-32		

the average running time of processing a test image for three different sizes, and each averaged 100 testing images. These experiments are tested on a PC running Windows 10 OS with 64G RAM, 3.6GHz CPU and Nvidia GeForce GTX 1080 GPU. All codes of these methods are run in Mat-



(a) Real-world image      (b) Without denoising      (c) With denoising

Figure 6. Results without and with denoising

lab, which ensures the fairness of time comparison. Methods [16],[8],[11],[10] are implemented using CPU, while



Table 4. Angular error  $\varepsilon$  (degree) of different model

	Dong[8]	LIME[11]	SRIE[10]	Ours
1st row	18.67	16.49	20.01	<b>5.89</b>
2nd row	33.62	25.40	35.51	<b>6.55</b>
3rd row	25.77	21.55	24.73	<b>6.20</b>
2K images	18.45	13.53	17.59	<b>5.79</b>

our method is tested on both CPU and GPU. Because our method is a completely feedforward process after network training, we can find that our approach on GPU processes significantly faster than methods [16],[8],[10] except [11].

#### 4.7. Study of MSR-net Parameters

The number of logarithmic transformation function  $n$  and the number of convolutional layers  $K$  are two main hyper-parameters in MSR-net. In this subsection, we try to experiment on the effect of these hyper-parameters on the final results. As we all know, the effectiveness of deeper structures for low-level image tasks is found not as apparent as that shown in high-level tasks [3, 25]. Specifically, we test for the number of logarithmic transformation function  $n \in \{1, 2, 4\}$  and  $v = 300; v = 100, 300; v = 1, 10, 100, 300$  respectively. At the same time, we set the number of convolutional layers  $K \in \{6, 10, 21\}$ . For the sake of fairness, all these networks are iterated 100K times and 100 synthetic images are used to measure the result by averaging their SSIM.

As shown in Table 6, adding more hidden layers obtains higher SSIM and achieves better results. We believe that, with an appropriate design to avoid over-fitting, deeper structure can improve the network’s nonlinear capacity and learning ability. In Figure 7, from the color of the boy’s skin and the clothes, the network using multi-scale logarithmic transformation performs better. It is also essential for MSR-net to improve nonlinear capacity by using multi-scale logarithmic transformation. To get better performance within the running time and hardware limits, we finally chose the number of logarithmic transformation function  $n = 4$  and the depth of convolutional layers  $K = 10$  for our experiments above.

## 5. Conclusion

In this paper, we propose a novel deep learning approach for low-light image enhancement. It shows that multi-scale Retinex is equivalent to a feedforward convolutional neural network with different Gaussian convolution kernels. After this, we construct a Convolutional Neural Network(MSR-net) that directly learns an end-to-end mapping between dark and bright images with little extra pre/post-processing beyond the optimization. Experiments on synthetic and real-world data reveal the advantages of our method in comparison with other state-of-the-art methods from the quali-

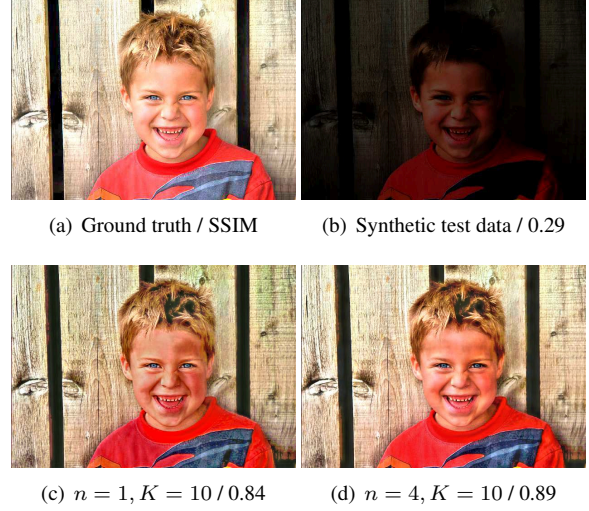


Figure 7. Results using different parameter setting

tative and quantitative perspective. Nevertheless, there are still some problems with this approach. Because of the limited receptive field in our model, very smooth regions such as clear sky are sometimes attacked by halo effect. Enlarging receptive field or adding hidden layers may solve this problem.

## References

- [1] <http://www.photoshopesentials.com/photo-editing/adding-a-brightness-contrast-adjustment-layer-in-photoshop.html>. 5
- [2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011. 5
- [3] J. Bruna, P. Sprechmann, and Y. Lecun. Image super-resolution using deep convolutional networks. *Computer Science*, 2015. 8
- [4] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, 2016. 2
- [5] T. Celik and T. Tjahjadi. Contextual and variational contrast enhancement. *IEEE Transactions on Image Processing*, 20(12):3431–3441, 2011. 2
- [6] K. Dabov, A. Foi, V. Katkovnik, and K. O. Egiazarian. Image restoration by sparse 3d transform-domain collaborative filtering. In *Image Processing: Algorithms and Systems*, page 681207, 2008. 6
- [7] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016. 2
- [8] X. Dong, Y. A. Pang, and J. G. Wen. Fast efficient algorithm for enhancement of low lighting video. In *ACM SIGGRAPH 2010 Posters*, page 69. ACM, 2010. 1, 2, 6, 7, 8, 9

Table 5. Comparison of average running time on 100 images(seconds)

Image size	MSRCR[16]	Dong[8]	LIME[11]	SRIE[10]	Ours(CPU)	Ours(GPU)
500 × 500	0.761	0.321	<b>0.188</b>	1.370	2.422	0.320
750 × 750	1.266	0.681	<b>0.359</b>	3.090	5.043	0.678
1,000 × 1,000	1.668	1.103	<b>0.521</b>	5.755	8.962	0.824

Table 6. Average SSIM for different network parameters

	$K = 6$	$K = 10$	$K = 21$
$n = 1$	0.879	0.897	0.904
$n = 2$	0.890	0.902	0.914
$n = 4$	0.893	0.905	0.920

- [9] X. Fu, J. Huang, D. Z. Y. Huang, X. Ding, and J. Paisley. Removing rain from single images via a deep detail network. **2**
- [10] X. Fu, D. Zeng, Y. Huang, X.-P. Zhang, and X. Ding. A weighted variational model for simultaneous reflectance and illumination estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2782–2790, 2016. **1, 2, 6, 7, 8, 9**
- [11] X. Guo, Y. Li, and H. Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE Transactions on Image Processing*, 26(2):982–993, 2017. **1, 2, 6, 7, 8, 9**
- [12] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2011. **2**
- [13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. **2**
- [14] S. D. Hordley and G. D. Finlayson. Re-evaluating colour constancy algorithms. 1(1):76–79, 2004. **6**
- [15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. **5**
- [16] D. J. Jobson, Z.-u. Rahman, and G. A. Woodell. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Transactions on Image processing*, 6(7):965–976, 1997. **1, 2, 3, 4, 5, 6, 7, 8, 9**
- [17] D. J. Jobson, Z.-u. Rahman, and G. A. Woodell. Properties and performance of a center/surround retinex. *IEEE transactions on image processing*, 6(3):451–462, 1997. **2**
- [18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. **2**
- [19] E. H. Land. The retinex theory of color vision. *Scientific American*, 237(6):108–129, 1977. **1, 2**
- [20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. **2**
- [21] K. Ma, K. Zeng, and Z. Wang. Perceptual quality assessment for multi-exposure image fusion. *IEEE Transactions on Image Processing*, 24(11):3345–3356, 2015. **5, 6**
- [22] A. Mittal, R. Soundararajan, and A. C. Bovik. Making a completely blind image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013. **6**
- [23] S. Park, S. Yu, B. Moon, S. Ko, and J. Paik. Low-light image enhancement using variational optimization-based retinex model. *IEEE Transactions on Consumer Electronics*, 63(2):178–184, 2017. **2**
- [24] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. **2**
- [25] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M. H. Yang. Single image dehazing via multi-scale convolutional neural networks. pages 154–169, 2016. **8**
- [26] G. Schaefer and M. Stich. Ucid: An uncompressed color image database. In *Storage and Retrieval Methods and Applications for Multimedia 2004*, volume 5307, pages 472–481. International Society for Optics and Photonics, 2003. **5**
- [27] V. Vonikakis, D. Chrysostomou, R. Kouskouridas, and A. Gasteratos. Improving the robustness in feature detection by local contrast enhancement. In *Imaging Systems and Techniques (IST), 2012 IEEE International Conference on*, pages 158–163. IEEE, 2012. **5, 6**
- [28] V. Vonikakis, D. Chrysostomou, R. Kouskouridas, and A. Gasteratos. A biologically inspired scale-space for illumination invariant feature detection. *Measurement Science and Technology*, 24(7):074024, 2013. **5, 6**
- [29] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3119–3127, 2015. **2**
- [30] S. Wang, J. Zheng, H.-M. Hu, and B. Li. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE Transactions on Image Processing*, 22(9):3538–3548, 2013. **5, 6**
- [31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. **6**
- [32] Z. Ye, H. Mohamadian, and Y. Ye. Discrete entropy and relative entropy study on nonlinear clustering of underwater and arial images. In *Control Applications, 2007. CCA 2007. IEEE International Conference on*, pages 313–318. IEEE, 2007. **6**
- [33] K. Zeng, K. Ma, R. Hassen, and Z. Wang. Perceptual evaluation of multi-exposure image fusion algorithms. In *Quality of Multimedia Experience (QoMEX), 2014 Sixth International Workshop on*, pages 7–12. IEEE, 2014. **5, 6**