이화여자대학교
EWHA WOMANS UNIVERSITY

Advanced Digital Image Processing Course

Low-light image enhancement

Thoi, Trang
2023 – 05 - 14

Contents

List of Figures

1. Introduction

Low-Light Image Enhancement is a computer vision task that involves improving the quality of images captured under low-light conditions. The goal of low-light image enhancement is to make images brighter, clearer, and more visually appealing, without introducing too much noise or distortion.

Enhancing the visual quality of images captured under weak light conditions is a difficult problem. These images usually appear with low contrast, low resolution, high noise, and over-exposure, due to insufficient light and gain, which affect our visual effects seriously. Consequently, low-light image enhancement technology, which aims to improve the overall brightness, local contrast, and detailed information of raw images, has been widely studied.

Directly increasing the image brightness is the most intuitive and simplest method. Typical algorithms use histogram equalization (HE) and grayscale transformation, which enhance pixel values through mathematical functions. HE and its derivative algorithms are widely used due to their simplicity and effectiveness. Grayscale transformation uses a mathematical function to change grayscale values, and gamma correction (GC) is one of the most widely used methods. These methods enhance the entire image to the same degree regardless of whether the area is dark or not, which easily causes overexposure.

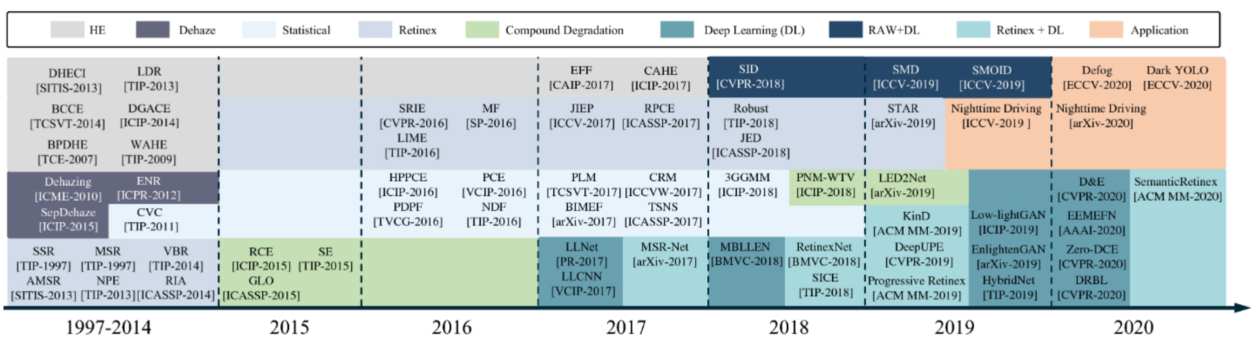


Figure 1 Milestones of single-image low-light enhancement methods: histogram equalization, dehazing, statistical model, Retinex model, deep-learning (DL), RAW+DL, Retinex+DL, compound degradation, and related applications

Retinex theory assumes that an image can be decomposed into an illumination component and a reflection component, and the latter is used as the final enhanced image. Single scale retinex (SSR), multi-scale retinex (MSR), and multi-scale retinex with color restoration (MSRCR) are typical algorithms based on retinex theory. However, these methods enhance the image by eliminating illumination, resulting in loss of information.

Deep-Learning Based Methods The era of deep-learning (DL) low-light enhancement starts in year 2017. After that, due to its distinguished performance and flexibility, this branch gradually became mainstream. Especially for EnlightenGAN (Jiang et al. 2019), unpaired learning is introduced to train a light enhancement model, which is the potential to get rid of paired dataset construction and address the domain shift problem between the training data and the practical applications. In general, with the powerful priors extracted from the large-scale data, deep learning methods achieve general superiority in performance. Some traditional ideas are injected to guide the design of the deep networks, such as Retinex model and the layer separation.

In this research, we will carry out the implementation of both General Image Processing and Deep Learning approach. The result will be compared in Qualitative and Quantitative as SSIM, NIQE, PIQE and BRISQUE.

2. Method

2.1. GIP: Histogram Equalization

2.1.1. Method Overview

Histogram equalization is a method in image processing of contrast adjustment using the image's histogram. This method usually increases the global contrast of many images, especially when the image is represented by a narrow range of intensity values. Through this adjustment, the intensities can be better distributed on the histogram utilizing the full range of intensities evenly. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the highly populated intensity values which are used to degrade image contrast.

The method is useful in images with backgrounds and foregrounds that are both bright or both dark. In particular, the method can lead to better views of bone structure in x-ray images, and to better detail in photographs that are either over or under-exposed. A key advantage of the method is that it is a straightforward technique adaptive to the input image and an invertible operator. So, in theory, if the histogram equalization function is known, then the original histogram can be recovered. The calculation is not computationally intensive. A disadvantage of the method is that it is indiscriminate. It may increase the contrast of background noise, while decreasing the usable signal.

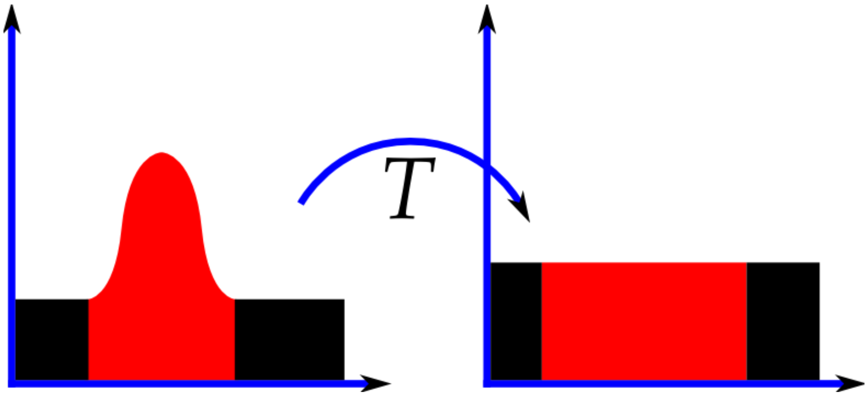


Figure 2 Histograms of an image before and after equalization

Histogram equalization often produces unrealistic effects in photographs; however, it is very useful for scientific images like thermal, satellite or x-ray images, often the same class of images to which one would apply false color.

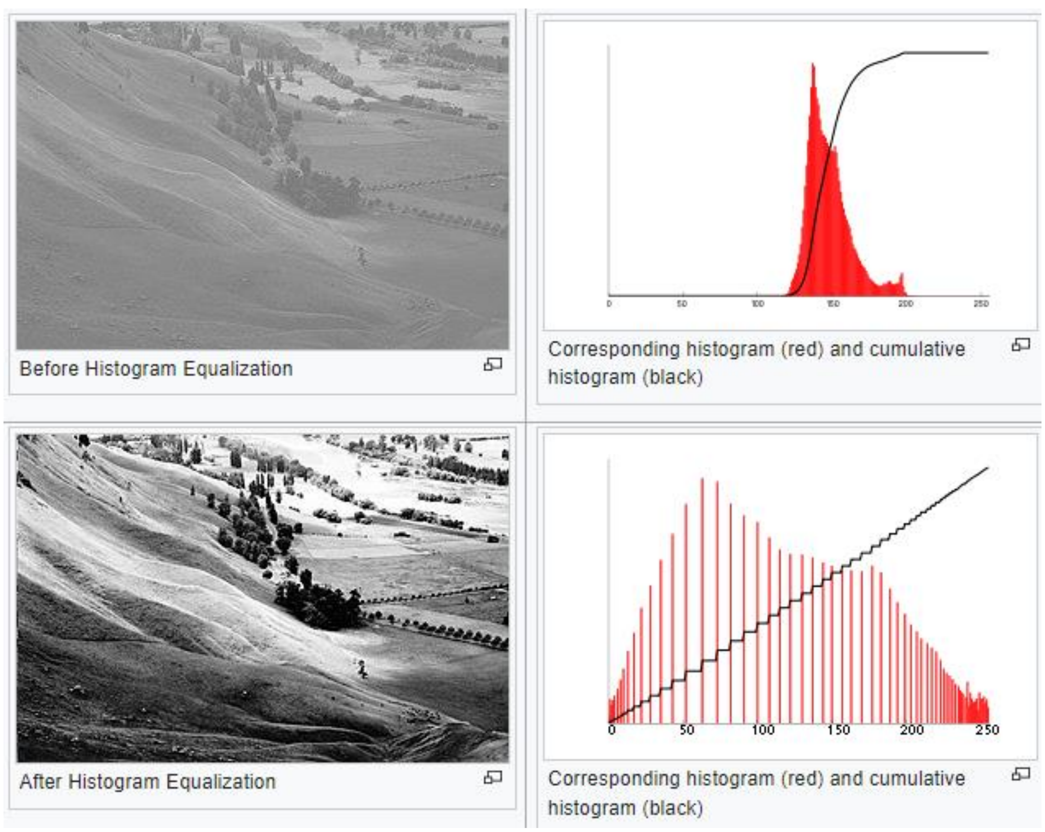


Figure 3 Before Histogram Equalization and After Histogram Equalization Image and Corresponding histogram

Also, histogram equalization can produce undesirable effects (like visible image gradient) when applied to images with low color depth. For example, if applied to an 8-bit image displayed with 8-bit gray-scale palette it will further reduce color depth (number of unique shades of gray) of the image. Histogram equalization will work the best when applied to images with much higher color depth than palette size, like continuous data or 16-bit gray-scale images.

Modifications of this method use multiple histograms, called subhistograms, to emphasize local contrast, rather than overall global contrast. Examples of such methods include adaptive histogram equalization, contrast limiting adaptive histogram equalization or CLAHE, multipeak histogram equalization (MPHE), and multipurpose beta optimized bihistogram equalization (MBOBHE). The goal of these methods, especially MBOBHE, is to improve the contrast without producing brightness mean-shift and detail loss artifacts by modifying the HE algorithm.

A signal transform equivalent to histogram equalization also seems to happen in biological neural networks to maximize the output firing rate of the neuron as a function of the input statistics. This has been proved in the fly retina.

Histogram equalization is a specific case of the more general class of histogram remapping methods. These methods seek to adjust the image to make it easier to analyze or improve visual quality (e.g., retinex)

a. Localized Brightening

Low-Light Image Enhancement is a computer vision task that involves improving the quality of images captured under low-light conditions. The goal of low-light image enhancement is to make images brighter, clearer, and more visually appealing, without introducing too much noise or distortion.

b. Contrast adjustment

Enhance the contrast of grayscale and color images using intensity value mapping, histogram equalization, and contrast-limited adaptive histogram equalization.

2.1.2. Code Review

a. Localized Brightening

Images can be highly degraded due to poor lighting conditions. These images can have low dynamic ranges with high noise levels that affect the overall performance of computer vision algorithms. To make computer vision algorithms robust in low-light conditions, use low-light image enhancement to improve the visibility of an image.

Images can be highly degraded due to poor lighting conditions. These images can have low dynamic ranges with high noise levels that affect the overall performance of computer vision algorithms. To make computer vision algorithms robust in low-light conditions, use low-light image enhancement to improve the visibility of an image.

Read and display an RGB image captured in low light.

```
%Check Example Original Image.
A = imread("1.png");
imshow(A)
title("Original Image")
```

Step 1. Brighten the low-light image in proportion to the darkness of the local region, then display the brightened image. Dark regions brighten significantly. Bright regions also have a small increase in brightness, causing oversaturation. The image looks somewhat unnatural and is perhaps brightened too much.

```
%Step 1.
B = imlocalbrighten(A);
imshow(B)
title("Original Image applied imlocalbrighten ")
figure
subplot(1,2,1)
imhist(A)
title("Original Image")
subplot(1,2,2)
imhist(B)
title("Brightened Image")
```

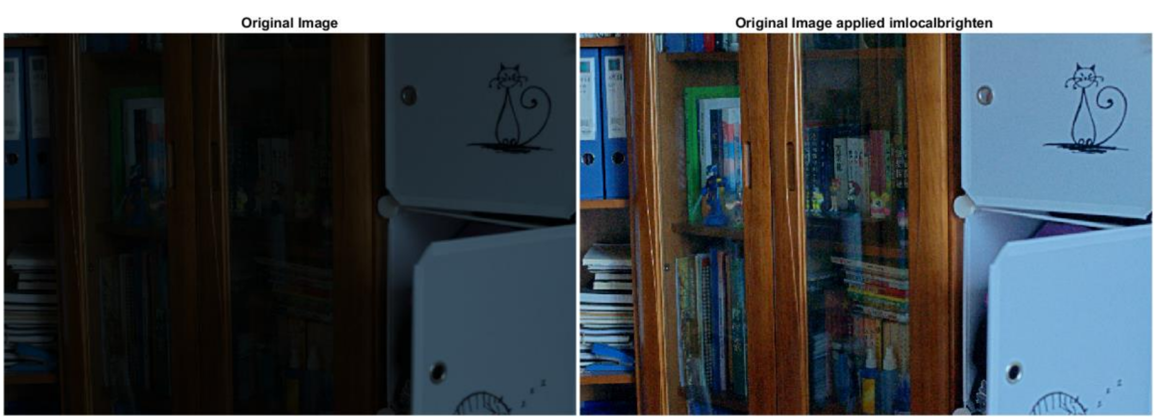


Figure 4 Original Image and image after applied imlocalbrighten

Display a histogram of the pixel values for the original image and the brightened image. For the original image, the histogram is skewed towards darker pixel values. For the brightened image, the pixel values are more evenly distributed throughout the full range of pixel values.

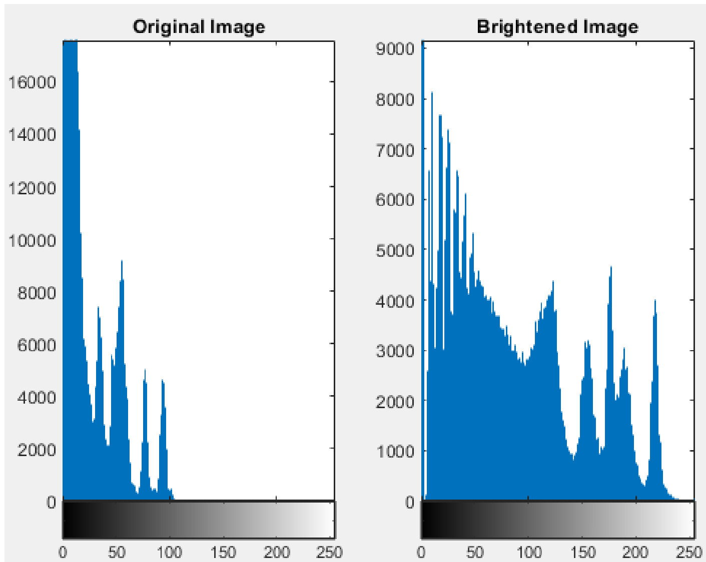


Figure 5 Original Image Histogram and image after applied imlocalbrighten Histogram.

Step 2. Brighten the original low-light image again and specify a smaller brightening amount.

Display the brightened image. The image looks more natural. The dark regions of the image are enhanced, but the bright regions by the windows are still oversaturated.

```
%Step 2.
amt = 0.5;
B2 = imlocalbrighten(A,amt);
figure
imshow(B2)
title("Image with Less Brightening")
```

Step 3. To reduce oversaturation of bright regions, apply alpha blending when brightening the image. The dark regions are brighter, and the bright pixels retain their original pixel values.

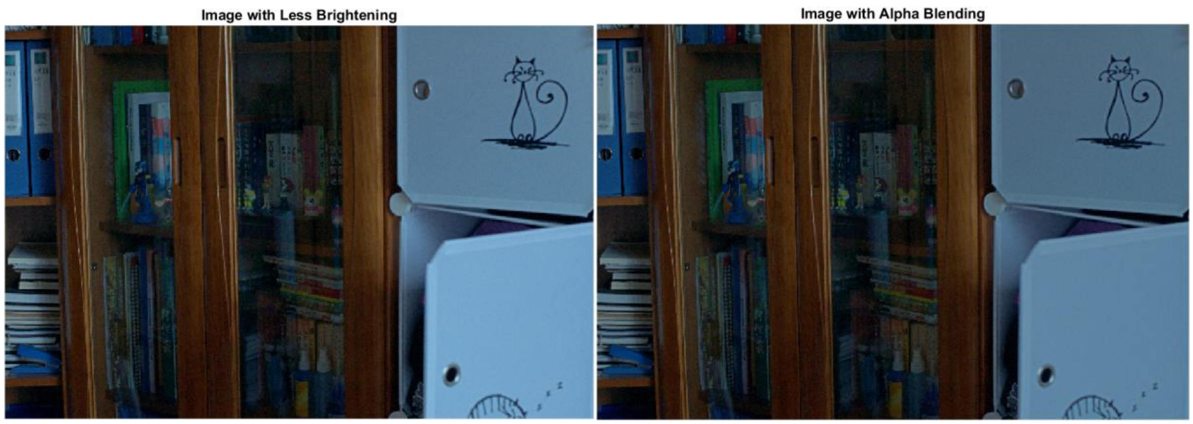


Figure 6 Image with less brightening and alpha blending

For comparison, display the three enhanced images in a montage.

Figure 7 Enhance images using imlocalbrighten and imlocalbrighten with alpha and with alpha blending

b. Contrast adjustment

Three functions are particularly suitable for contrast enhancement:

- ***imadjust*** increases the contrast of the image by mapping the values of the input intensity image to new values such that, by default, 1% of the data is saturated at low and high intensities of the input data.
- ***histeq*** performs histogram equalization. It enhances the contrast of images by transforming the values in an intensity image so that the histogram of the output image approximately matches a specified histogram (uniform distribution by default).
- ***adapthisteq*** performs contrast-limited adaptive histogram equalization. Unlike ***histeq***, it operates on small data regions (tiles) rather than the entire image. Each tile's contrast is enhanced so that the histogram of each output region approximately matches the specified histogram (uniform distribution

by default). The contrast enhancement can be limited in order to avoid amplifying the noise which might be present in the image.

Notice that ***imadjust*** had little effect on the image of the tire, but it caused a drastic change in the case of pout. Plotting the histograms of pout.tif and tire.tif reveals that most of the pixels in the first image are concentrated in the center of the histogram, while in the case of tire.tif, the values are already spread out between the minimum of 0 and maximum of 255 thus preventing ***imadjust*** from being effective in adjusting the contrast of the image.

Histogram equalization, on the other hand, substantially changes both images. Many of the previously hidden features are exposed, especially the debris particles on the tire. Unfortunately, at the same time, the enhancement over-saturates several areas of both images. Notice how the center of the tire, part of the child's face, and the jacket became washed out.

Concentrating on the image of the tire, it would be preferable for the center of the wheel to stay at about the same brightness while enhancing the contrast in other areas of the image. For that to happen, a different transformation would have to be applied to different portions of the image. The Contrast-Limited Adaptive Histogram Equalization technique, implemented in ***adapthisteq***, can accomplish this. The algorithm analyzes portions of the image and computes the appropriate transformations. A limit on the level of contrast enhancement can also be set, thus preventing the over-saturation caused by the basic histogram equalization method of ***histeq***. This is the most sophisticated technique in this example.

Contrast enhancement of color images is typically done by converting the image to a color space that has image luminosity as one of its components, such as the L*a*b* color space. Contrast adjustment is performed on the luminosity layer L* only, and then the image is converted back to the RGB color space. Manipulating luminosity affects the intensity of the pixels, while preserving the original colors.

Step 1. Read an image with poor contrast into the workspace. Then, convert the image from the RGB color space to the L*a*b* color space.

```
%Step 1.
shadow = imread("./…");
shadow_lab = rgb2lab(shadow);
```

Step 2. The values of luminosity span a range from 0 to 100. Scale the values to the range [0 1], which is the expected range of images with data type double.

```
%Step 2.
max_luminosity = 100;
L = shadow_lab(:,:,1)/max_luminosity;
```

*Figure 8 The converted image from the RGB color space to the L*a*b* color space*

Step 3. Perform the three types of contrast adjustment on the luminosity channel and keep the a* and b* channels unchanged. Convert the images back to the RGB color space.

```
%Step 3.
shadow_imadjust = shadow_lab;
shadow_imadjust(:,:,1) = imadjust(L)*max_luminosity;
shadow_imadjust = lab2rgb(shadow_imadjust);

shadow_histeq = shadow_lab;
shadow_histeq(:,:,1) = histeq(L)*max_luminosity;
shadow_histeq = lab2rgb(shadow_histeq);

shadow_adapthisteq = shadow_lab;
shadow_adapthisteq(:,:,1) = adapthisteq(L)*max_luminosity;
shadow_adapthisteq = lab2rgb(shadow_adapthisteq);
```

Step 4. Display the original image and the three contrast adjusted images as a montage.

```
%Step 4.
figure
montage({shadow,shadow_imadjust,shadow_histeq,shadow_adapthisteq},"Size",[1 4])
title("Original Image and Enhanced Images using " + ...
    "imadjust, histeq, and adapthisteq")
```

Figure 9 Original Image and Enhanced Images using imadjust, histeq, and adapthisteq

2.2.GIP: Dehaze

2.2.1. Method Overview

Footage shot in outdoor scenes can be significantly degraded due to poor lighting conditions. These images have a low dynamic range and high noise levels, which can affect the overall performance of computer vision algorithms. To ensure that your computer vision algorithms work well in low-light conditions, use low-light image enhancement to increase the visibility of your images. The pixel-by-pixel inversion histogram of a low-light image or HDR image is very similar to the histogram of a haze-stricken image. Therefore, haze removal techniques can be used to enhance low-light images.

Images can be highly degraded due to poor lighting conditions. These images can have low dynamic ranges with high noise levels that affect the overall performance of computer vision algorithms. To make computer vision algorithms robust in low-light conditions, use low-light image enhancement to improve the visibility of an image.

The dehazing algorithms in imreducehaze follow five steps:

1. Estimate the atmospheric light L using a dark channel prior.
2. Estimate the transmission map T.
3. Refine the estimated transmission map.
4. Restore the image.
5. Perform optional contrast enhancement.

Haze Removal Algorithm: algorithm takes a 3-channel low-light RGB image as input. This figure shows the block diagram of the LLE Algorithm.

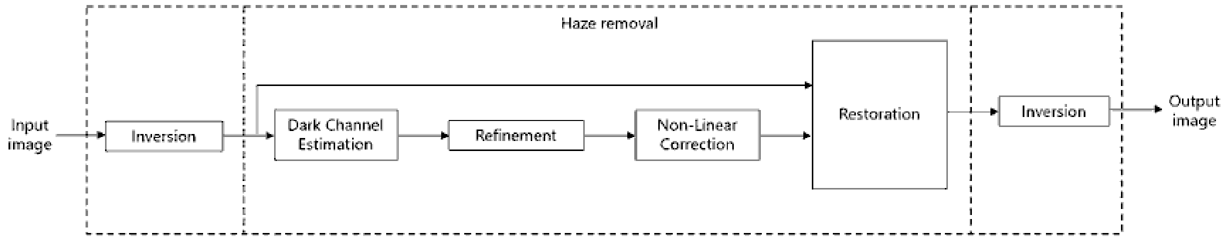


Figure 10 Process of applying Haze Removal to enhance low light image.

The algorithm consists of six stages.

Step 1. Scaling and Inversion

- Input image $\boldsymbol{I^c(x, y)}$, $c \in [r, g, b]$ is converted to range [0,1] by dividing by 255 -> inverting pixel-wise
- $I^c_{scal}(x, y) = \frac{\boldsymbol{I^c(x,y)}}{255}$ $\quad I^c_{inv}(x, y) = 1 - I^c_{scal}(x, y)$

Step 2. Dark Channel Estimation

- The dark channel is estimated by finding the pixel-wise minimum across all three channels of the inverted image.
- The minimum value is multiplied by a haze factor z (z : amount of haze to remove, $z \in [0,1]$)
- Higher value: more haze will be removed from the image.
- $I_{air}(x, y) = z \times min_{c \in [r,g,b]} I^c_{inv}(x, y)$

Step 3. Refinement

- The airlight image from the previous stage is refined by iterative smoothing -> smoothing strengthens the details of the image after enhancement.
- Five filter iterations with a 3-by-3 kernel for each stage
- Refined image is stored in $I_{refined}(x, y)$

$I_{refined(n+1)}(x, y) = I_{refined(n)}(x, y) * h \quad , \quad n = [0,1,2,3,4] \;\; \& \;\; I_{refined(0)} = I_{air} \quad where\ h = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

Step 4. Non-Linear Correction

- To reduce over-enhancement, the refined image is corrected using a non-linear correction equation

- m : mid-line of changing the dark regions of the airlight map from dark to bright values
- $I_{nlc}(x, y) = \frac{[I_{refined}(x,y)]^4}{[I_{refined}(x,y)]^4 + m^4}$

Step 5. Restoration

- Restoration is performed pixel-wise across the three channels of the inverted and corrected image I_{nlc}
- $I^c_{restore}(\mathrm{x}, \mathrm{y}) = \frac{I^c_{scal}(x,y) - I_{nlc}(x,y)}{1 - I_{nlc}(x,y)}$

Step 6. Inversion

- Invert the output of the restoration stage, and scales to the range [0,255]
- $\boldsymbol{I^c_{enhanced}}(x, y) = 255 \text{ x } (1 - I^c_{restore})$

2.2.2. Code Review

The process of enhancing low-light images using haze removal techniques consists of three steps:

Step 1: Invert the low-light image.

Step 2: Apply the haze removal algorithm to the inverted low-light image.

Step 3: Invert the enhanced footage.

imreducehaze uses two different dehazing algorithms, simpledcp and approxdcp. These methods both rely on a dark channel prior, which is based on the observation that nonhazy images of outdoor scenes usually contain some pixels that have low signal in one or more color channels. The methods differ in how they estimate the dark channel prior and atmospheric light. Specify optional pairs of arguments as Name1=Value1,...,NameN=ValueN, where Name is the argument name and Value is the corresponding value. Name-value arguments must appear after other arguments, but the order of the pairs does not matter.

- **Method – Techniques to reduce haze:** Technique used to reduce haze, specified as one of these values:
    - "simpledcp" — Simple dark channel prior method [2]. This technique uses a per-pixel dark channel to estimate haze and quadtree decomposition to estimate the atmospheric light.

- "approxdcp" — Approximate dark channel prior method [1]. This technique uses both per-pixel and spatial blocks when computing the dark channel and does not use quadtree decomposition.

- **AtmosphericLight - Maximum value to be treated as haze**, specified as a 1-by-3 numeric vector for RGB images or a numeric scalar for grayscale images. Values must be in the range [0, 1]. Atmospheric light values greater than 0.5 tend to give better results. If you do not specify AtmosphericLight, then the imreduzehaze function estimates a value depending on the value of Method.
- **ConstrastEnhancement** - Contrast enhancement technique, specified as "global", "boost", or "none".
- **BoostAmount** - Amount of per-pixel gain to apply as postprocessing, specified as a positive number in the range (0, 1]. This argument is only supported if ContrastEnhancement is specified as "boost".

Output Arguments include:

- J-Dehazed image: Amount of per-pixel gain to apply as postprocessing, specified as a positive number in the range (0, 1]. This argument is only supported if ContrastEnhancement is specified as "boost".
- T-Haze thickness: Haze thickness estimated at each pixel, returned as a numeric array.
- L – Estimated atmospheric light: Estimated atmospheric light, returned as a numeric array. L represents the value of the brightest nonspecular haze.

Algorithm:

The model to describe a hazy image I is $I(x) = J(x)T(x) + L(1-T(x))$. I is the observed intensity, J is the scene radiance, L is atmospheric light, and T is a transmission map describing the portion of light that reaches the camera. Dehazing algorithms recover the scene radiance (dehazed image) J from an estimation of the transmission map and atmospheric light according to: $J(x) = (I(x) - A)/(max(t(x), t0)) + A$.

Step 1. Use haze removal algorithms to enhance low-light images.

```
%Invert the image and see how low-light areas of the original image appear blurry.
AInv = imcomplement(A);
imshow(AInv);
title("low-light areas of the original image appear blurry")
%imreducehaze function to remove haze.
BInv = imreducehaze(AInv);
```

```
imshow(BInv);
title("Haze removed image")
```

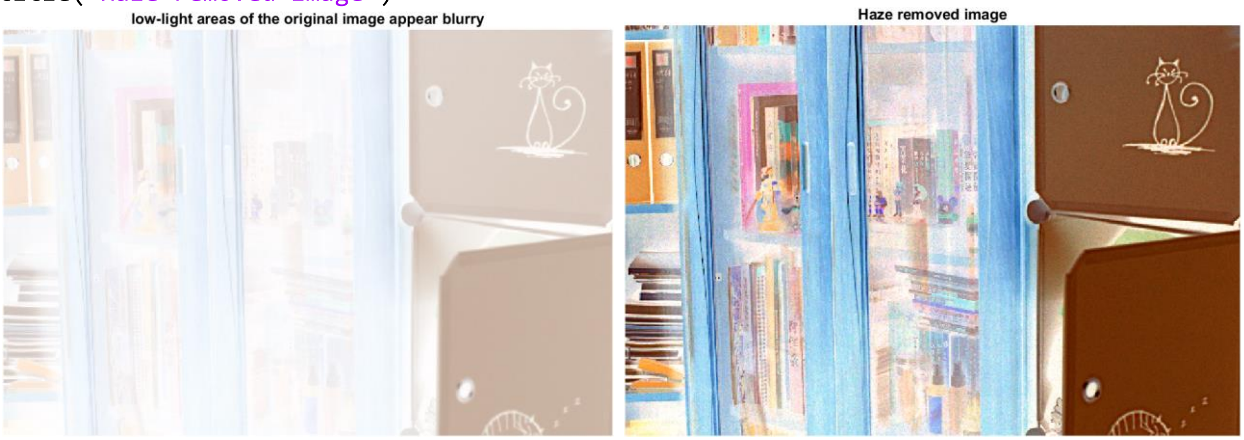


Figure 11 Inverted image and after haze removed inverted image.

```
%The result is reversed to obtain an improved image.
B = imcomplement(BInv);
```

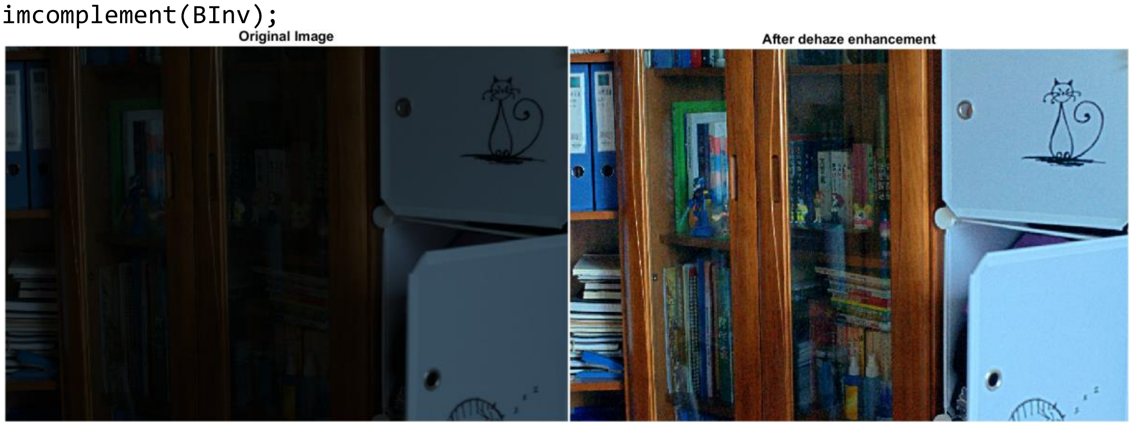


Figure 12 Original image and after applied dehaze enhancement.

Step 2.2 imreducehaze: use optional parameters for better results.

```
%2. imreducehaze Use optional parameters for better results
BInv = imreducehaze(AInv, 'Method','approx','ContrastEnhancement','boost');
```

Step 2.1 Use a different color space to reduce color distortion.

```
%3. Use a different color space to reduce color distortion
Lab = rgb2lab(A);
LInv = imcomplement(Lab(:,:,1) ./ 100);
LEnh = imcomplement(imreducehaze(LInv,'ContrastEnhancement','none'));
LabEnh(:,:,1)   = LEnh .* 100;
LabEnh(:,:,2:3) = Lab(:,:,2:3) * 2; % Increase saturation
B = lab2rgb(LabEnh);
```

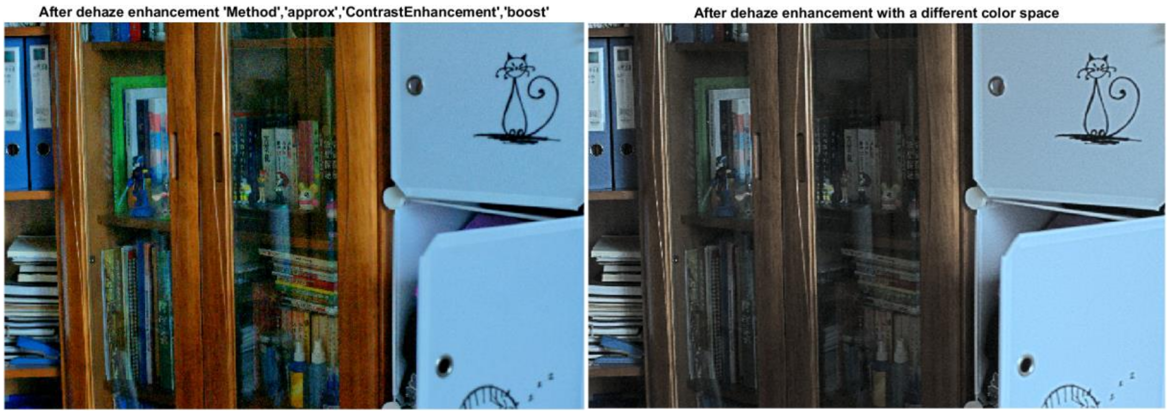


Figure 13 imreducehaze with optional parameters and dehaze enhancement with a different color space

Step 3. Improve Step 2 results with noise cancellation.

imguidedfilter(): Guided filtering of images. Smooth the image using imguidedfilter. In this syntax, imguidedfilter uses the image itself as the guidance image. Image to be filtered, specified as a binary, grayscale, or RGB image. Guide image, specified as a binary, grayscale, or RGB image of the same height and width as image A. Guide image, specified as a binary, grayscale, or RGB image of the same height and width as image A.

The DegreeOfSmoothing argument specifies a soft threshold on variance for the given neighborhood. If a pixel's neighborhood has variance much lower than the threshold, it will see some amount of smoothing. If a pixel's neighborhood has variance much higher than the threshold it will have little to no smoothing.

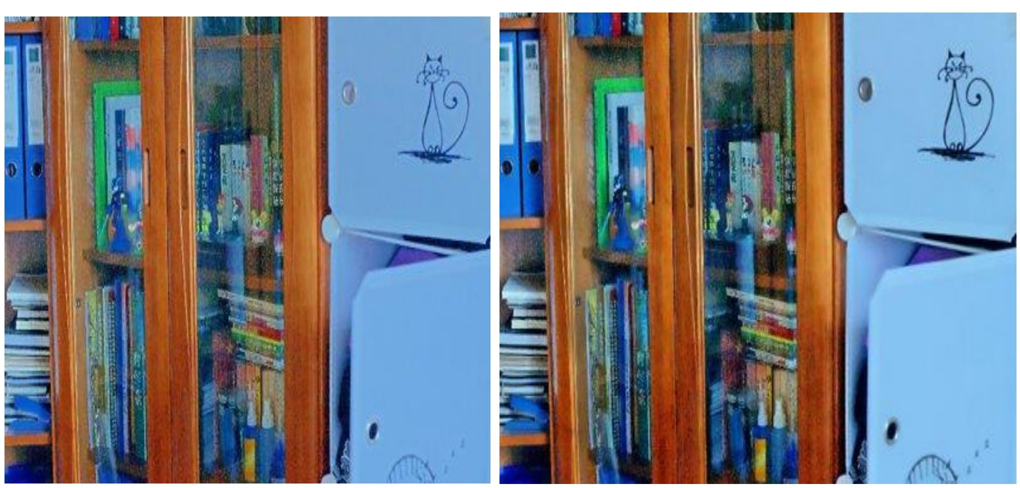


Figure 14 Dehaze enhancement image after applied noise cancellation.

Input images A and G can be of different classes. If either A or G is of class integer or logical, then imguidedfilter converts them to floating-point precision for internal computation.

Input images A and G can have different number of channels.

- If both A and G are RGB images, then imguidedfilter filters each channel of A independently using the corresponding channel of G.
- If A is an RGB image and G is a single-channel image, then imguidedfilter filters each channel of A independently using the same guidance image, G.
- If A is a single-channel image and G is an RGB image, then imguidedfilter filters A using the combined color statistics of all the three channels of G.

2.3. GIP: multiscale retinex with color restoration (MSRCR)

2.3.1. Method Overview

The multiscale retinex with color restoration (MSRCR) combines the dynamic range compression of the small-scale retinex and the tonal rendition of the large scale. The idea of the retinex was conceived by Land as a model of the lightness and color perception of human vision. Through the years, Land evolved the concept from a random walk computation to its last form as a center/surround spatially opponent operation, which is related to the neurophysiological functions of individual neurons in the primate retina, lateral geniculate nucleus, and cerebral cortex. Previously defined a single-scale retinex (SSR) that can either provide dynamic range compression (small scale), or tonal rendition (large scale), but not both simultaneously. The multiscale retinex with color restoration (MSRCR) combines the dynamic range compression of the small-scale retinex and the tonal rendition of the large scale.

The MSR, comprised of three scales (small, intermediate, and large), was found to synthesize dynamic range compression, color consistency, and tonal rendition, and to produce results that compare favorably with human visual perception, except for scenes that contain violations of the gray-world assumption. Even when the gray-world violations were not dramatic, some desaturation of color was found to occur. A color restoration scheme was defined that produced good color rendition even for severe gray-world violations, but at the expense of a slight sacrifice in color consistency. In retrospect, the form of the color restoration is a virtual spectral analog to the spatial processing of the retinex. This may reflect some underlying principle at work in the neural computations of consciousness; perhaps, even that the visual representation of lightness, color, and detail is a highly compressed mesh of contextual relationships, a world of relativity and relatedness that is more often associated with higher levels of visual processing such as form analysis and pattern recognition. While there is no firm theoretical or mathematical basis for proving the generality of this color restored MSR, we have tested it successfully on numerous diverse scenes and images, including some known to contain severe gray-world violations.

2.3.2. Code Review

The MSRCR has been developed to be used with provided Matlab function code.

MSRetinex(im, sigmaS, nscale, scalefactor, saturatedpix, precision) with

- im is the input grayimage that estimated the image illumination or simply rgb2gray() of the color image
- sigmaS is the size of the gaussian kernel in each scales
- nscale is the number of scales
- scalefactor is the scalefactor between each scale
- saturatedpix is 1x2 scalar vector precising the percentage of pixels to be saturated in each sides
- precision is the precision for im scaling

The algorithm in the function convert im to precision (8, 16 or 32 bit) scale and add 1 for log, process over each scale with nscale > 1, pad to the next power of scalefactor and convolve with gaussian kernel and pad output to the next power of scalefactor, then substract it to the image. Rescale to [0, 1] and apply the simplest color balance algorithm to get better output with calculations of lower percentage cutoff, upper percentage cutoff, use a maximum of 1e6 bins even if precision is set to 32bit. Finally, rescale ret to the precision scale.

2.4.NN architecture: EnlightenGAN

2.4.1. Method Overview

EnlightenGAN is the first work that successfully introduces unpaired training to low-light image enhancement. EnlightenGAN is demonstrated to be easily adaptable to enhancing real-world images from various domains and outperfoms recent methods under a variety of metrics in terms of visual quality and subjective survey

Inspired by unsupervised image-to-image translation, we adopt generative adversarial networks (GANs) to build an unpaired mapping between low and normal light image spaces without relying on exactly paired images. That frees us from training with only synthetic data or limited real paired data captured in controlled settings. We introduce a lightweight, yet effective one-path GAN named EnlightenGAN, without using cycle-consistency as prior works and therefore enjoying the merit of much shorter training time. Due to the lack of paired training data, we incorporate several innovative techniques. We first propose a dual discriminator to balance global and local low-light enhancement. Further, owing to the absence of ground-truth supervision, a self-regularized perceptual loss is proposed to constrain the feature distance between the low-light input image and its enhanced version, which is subsequently adopted both locally and globally together with the adversarial loss for training EnlightenGAN. We also propose to exploit the illumination information of the low-light input as a self-regularized attentional map in each level of deep features to regularize the unsupervised learning. Thanks to the unsupervised setting, we show that EnlightenGAN can be very easily adapted to enhancing real-world low-light images from different domains.

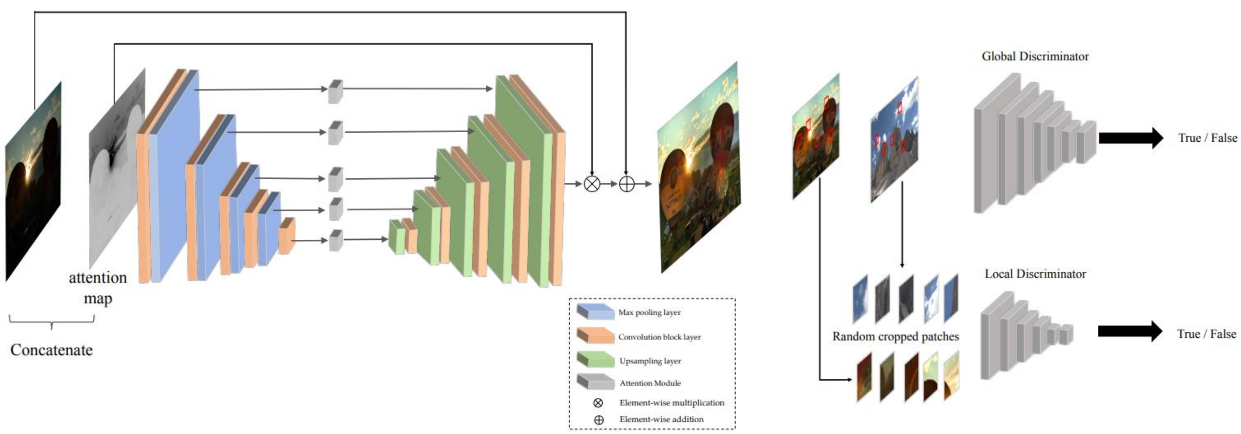


Figure 15 The overall architecture of EnlightenGAN. In the generator, each convolutional block consists of two 3 × 3 convolutional layers followed by batch normalization and LeakyRelu. Each attention module has the feature map multiply with a (resized) attention map.

The notable innovations of EnlightenGAN are:

• EnlightenGAN is the first work that successfully introduces unpaired training to low-light image enhancement. Such a training strategy removes the dependency on paired training data and enables us to train with largervarieties of images from different domains. It also avoids overfitting any specific data generation protocol or imaging device that previous works [15], [5], [16] implicitly rely on, hence leading to notably improved real-world generalization.

• EnlightenGAN gains remarkable performance by imposing (i) a global-local discriminator structure that handles spatially varying light conditions in the input image; (ii) the idea of self-regularization, implemented by both the self-feature preserving loss and the self-regularized attention mechanism. The self-regularization is critical to our model success, because of the unpaired setting where no strong form of external supervision is available.

• EnlightenGAN is compared with several state-of-the art methods via comprehensive experiments. The results are measured in terms of visual quality, no-referenced image quality assessment, and human subjective survey.

All results consistently endorse the superiority of EnlightenGAN. Moreover, in contrast to existing paired trained enhancement approaches, EnlightenGAN proves particularly easy and flexible to be adapted to enhancing real-world low-light images from different domains.

As shown in Fig., our proposed method adopts an attention-guided U-Net as the generator and uses the dual discriminator to direct the global and local information. The model also uses a self-feature preserving loss to guide the training process and maintain the textures and structures. In this section we first introduce two important building blocks, i.e., the global-local discriminators and the self-feature preserving loss, then the whole network in detail.

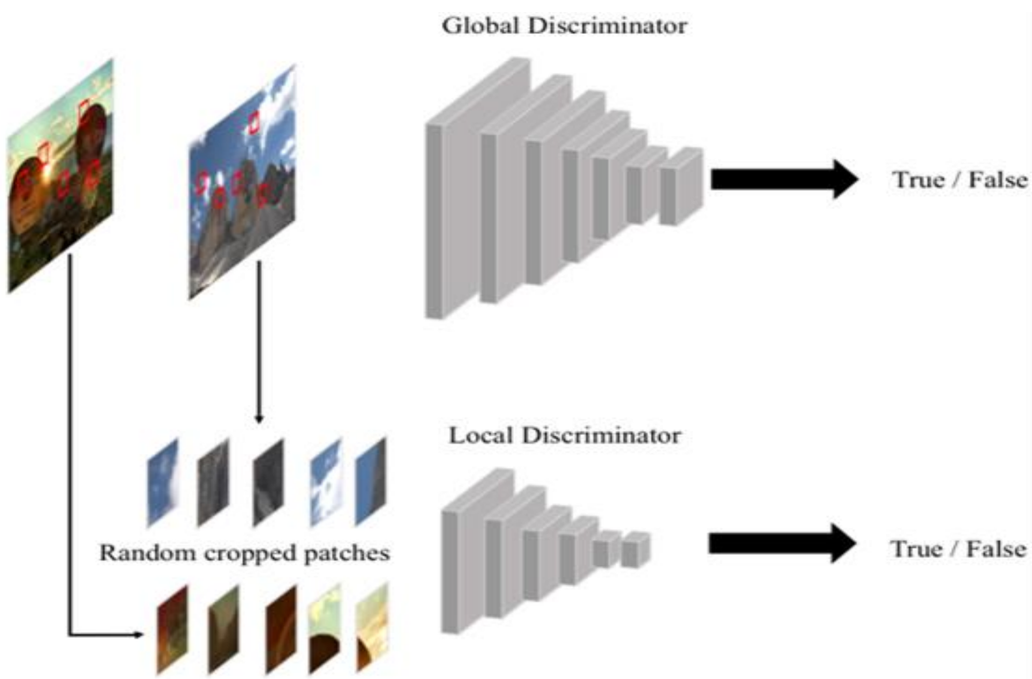


Figure 16 Global Discriminator and Local Discriminator

GAN (Generative Adversarial Network) consists of a generator and discriminator. Generator with Input: low-light image / Output: light enhanced image / Target: X (unsupervised learning). Discriminator with Input: real or fake image / Output: Probability / Target: True or False.

- Global-Local Discriminators
- Self-Feature Preserving Loss
- U-Net Generator Guided with Self-Regularized Attention

First example: EnlightenGAN successfully suppresses the noise in black sky and produces the best visible details of yellow wall. Second example: NPE and SRIE fail to enhance the background details. LIME introduces over-exposure on the woman's face. LLNet generate severe color distortion. However, EnlightenGAN not only restores the background details but also avoids over-exposure artifacts, distinctly outperforming other methods. Third example: EnlightenGAN produces a visually pleasing result while avoiding over-exposure artifacts in the car and cloud. Others either do not enhance dark details enough or generate over-exposure artifacts. Please zoom in to see more details.

Figure 17 Comparison with other state-of-the-art methods. Zoom-in regions are used to illustrate the visual differences. Three examples are listed from the top to the bottom rows.

Comparing the performance of Enlighten-GAN with current state-of-the-art methods. We conduct a list of experiments including visual quality comparison, human subjective review and no-referenced image quality assessment (IQA), which are elaborated on next. Visual Quality Comparison by comparing the visual quality of EnlightenGAN with several recent competing methods. Results are demonstrated in Fig. 4, where the first column shows the original low-light images, and the second to fifth columns are the images enhanced by: a vanilla CycleGAN trained using our unpaired training set, RetinexNet, SRIE, LIME, NPE, LLNet, and CycleGAN.

The last column shows the results produced by EnlightenGAN. We next zoom in on some details in the bounding boxes. LIME easily leads to over-exposure artifacts, which makes the results distorted and glaring with some information missing. The results of SRIE and NPE are generally darker compared with others. CycleGAN and RetinexNet generate unsatisfactory visual results in terms of both brightness and naturalness. In contrast, EnlightenGAN successfully not only learns to enhance the dark area but also preserves the texture details and avoids over-exposure artifacts.

2.4.2. Code Review

EnlightenGAN is first trained from the scratch for 100 epochs with the learning rate of 1e-4, followed by another 100 epochs with the learning rate linearly decayed to 0. We use the Adam optimizer, and the batch size is set to be 32. Thanks to the lightweight design of one-path GAN without using cycle-consistency, the training time is much shorter than cycle-based methods. The whole training process takes 3 hours on 3 Nvidia 1080Ti GPUs.

```python
import time
import os
from options.test_options import TestOptions
from data.data_loader import CreateDataLoader
from models.models import create_model
from util.visualizer import Visualizer
from pdb import set_trace as st
from util import html

opt = TestOptions().parse()
opt.nThreads = 1   # test code only supports nThreads = 1
opt.batchSize = 1  # test code only supports batchSize = 1
opt.serial_batches = True  # no shuffle
opt.no_flip = True  # no flip

data_loader = CreateDataLoader(opt)
dataset = data_loader.load_data()
model = create_model(opt)
visualizer = Visualizer(opt)
# create website
web_dir = os.path.join("./ablation/", opt.name, '%s_%s' % (opt.phase, opt.which_epoch))
webpage = html.HTML(web_dir, 'Experiment = %s, Phase = %s, Epoch = %s' % (opt.name, opt.phase, opt.which_epoch))
# test
print(len(dataset))
for i, data in enumerate(dataset):
    model.set_input(data)
    visuals = model.predict()
    img_path = model.get_image_paths()
    print('process image... %s' % img_path)
    visualizer.save_images(webpage, visuals, img_path)

webpage.save()
```

Figure 18 Test Code Implementation of the EnlightenGAN Model

3. Result

3.1. Evaluation Metrics

3.1.1. SSIM

Structural similarity (SSIM) index is an image quality metric that assesses the visual impact of three characteristics of an image: luminance, contrast and structure.

SSIM algorithms. The SSIM Index quality assessment index is based on the computation of three terms, namely the luminance term, the contrast term and the structural term. The overall index is a multiplicative combination of the three terms.

$$SSIM(x, y) = [l(x, y)]^{\alpha} \cdot [c(x, y)]^{\beta} \cdot [s(x, y)]^{\gamma}$$

where

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1},$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2},$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

where μx, μy, σx,σy, and σxy are the local means, standard deviations, and cross-covariance for images x, y. If $\alpha = \beta = \gamma = 1$ (the default for Exponents), and C3 = C2/2 (default selection of C3).

SSIM MATLAB function is used for measuring image quality require 2 inputs:

1. Image for quality measurement, specified as a numeric array or a dlarray (Deep Learning Toolbox) object. If A is not a 2-D grayscale image or 3-D grayscale volume, such as an RGB image or stack of grayscale images, specify the DataFormat name-value argument. Do not specify the DataFormat name-value argument if A is a formatted dlarray object.
2. Image for quality measurement, specified as a numeric array or a dlarray (Deep Learning Toolbox) object. If A is not a 2-D grayscale image or 3-D grayscale volume, such as an RGB image or stack of grayscale images, specify the DataFormat name-value argument. Do not specify the DataFormat name-value argument if A is a formatted dlarray object.

SSIM MATLAB function output:

1. global SSIM value for the image.
2. local SSIM values for each pixel: Display the local SSIM map. Include the global SSIM value in the figure title. Small values of local SSIM appear as dark pixels in the local SSIM map. Regions with small local SSIM value correspond to areas where the blurred image noticeably differs from the reference image. Large values of local SSIM value appear as bright pixels. Regions with large local SSIM correspond to uniform regions of the reference image, where blurring has less of an impact on the image.

3.1.2. NIQE

Naturalness Image Quality Evaluator (NIQE) no-reference image quality score. calculates the no-reference image quality score for image A using the Naturalness Image Quality Evaluator (NIQE). NIQE compares A to a default model computed from images of natural scenes. A smaller score indicates better perceptual quality.

NIQE algorithm: NIQE measures the distance between the NSS-based features calculated from image A to the features obtained from an image database used to train the model. The features are modeled as multidimensional Gaussian distributions.

NIQE MATLAB function is used for measuring image quality require

1. Input image, specified as a 2-D grayscale or RGB image.
2. Custom model of image features, specified as a niqeModel object. model is derived from natural scene statistics.

NIQE MATLAB function output: No-reference image quality score, returned as a nonnegative scalar. Lower values of score reflect better perceptual quality of image A with respect to the input model.

3.1.3. PIQE

Perception based Image Quality Evaluator (PIQE) no-reference image quality score.

PIQE Algorithm: PIQE calculates the no-reference quality score for an image through block-wise distortion estimation, using these steps:

1. Compute the Mean Subtracted Contrast Normalized (MSCN) coefficient for each pixel in the image using the algorithm proposed by N. Venkatanath and others.

2. Divide the input image into nonoverlapping blocks of size 16-by-16.
3. Identify high spatially active blocks based on the variance of the MSCN coefficients.
4. Generate activityMask using the identified high spatially active blocks.
5. In each block, evaluate distortion due to blocking artifacts and noise using the MSCN coefficients.
6. Use threshold criteria to classify the blocks as distorted blocks with blocking artifacts, distorted blocks with Gaussian noise, and undistorted blocks.
7. Generate noticeableArtifactsMask from the distorted blocks with blocking artifacts and noiseMask from the distorted blocks with Gaussian noise.
8. Compute the PIQE score for the input image as the mean of scores in the distorted blocks.
9. The quality scale of the image based on its PIQE score is given in this table. The quality scale and respective score range are assigned through experimental analysis on the dataset in LIVE Image Quality Assessment Database Release 2.

PIQE MATLAB function requires input image, specified as a 2-D grayscale image of size m-by-n or 2-D RGB image of size m-by-n-by-3.

PIQE MATLAB function output are:

- Score - Score for the input image A, returned as a nonnegative scalar in the range [0, 100]. The PIQE score is the no-reference image quality score, and it is inversely correlated to the perceptual quality of an image. A low score value indicates high perceptual quality and a high score value indicates low perceptual quality.
- activityMask - Spatial quality mask of active blocks, returned as a 2-D binary image of size m-by-n, where m and n are the dimensions of the input image A. The activityMask is composed of high spatially active blocks in the input image. The high spatially active blocks in the input image are the regions with more spatial variability due to factors that include compression artifacts and noise. The high spatially active blocks are assigned a value '1' in the activityMask.
- NoticeableArtifactsMask - Spatial quality mask of noticeable artifacts, returned as a 2-D binary image of size m-by-n, where m and n are the dimensions of the input image A. The noticeableArtifactsMask is composed

of blocks in activityMask that contain blocking artifacts (due to compression) or sudden distortions.

- noiseMask - Spatial quality mask of Gaussian noise, returned as a 2-D binary image of size m-by-n, where m and n are the dimensions of the input image A. The noiseMask is composed of blocks in activityMask that contain Gaussian noise.

To calculate PIQE score for an image and the corresponding distorted images. First, display the results with their corresponding image. Read an image into the workspace. Generate distorted images by adding noise and blur. Use imnoise function to generate the noisy image and imgaussfilt function to generate the blurred image. To display the spatial quality mask, generate spatial quality masks that indicate the high spatially active blocks, noticeable artifacts blocks, and noise blocks in the image. Visualize the spatial quality masks by overlaying them on the distorted image. Display the image with and without the masks and the PIQE score for the image.

3.1.4. BRISQUE

Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) no-reference image quality score.

Using BRISQUE MATLAB function, it's possible to compute the BRISQUE score for a natural image and its distorted versions using the default model. To use Custom Feature Model, Train a custom BRISQUE model from a set of quality-aware features and corresponding human opinion scores. Use the custom model to calculate a BRISQUE score for an image of a natural scene. Save images from an image datastore.

BRISQUE algorithm - brisque predicts the BRISQUE score by using a support vector regression (SVR) model trained on an image database with corresponding differential mean opinion score (DMOS) values. The database contains images with known distortion such as compression artifacts, blurring, and noise, and it contains pristine versions of the distorted images. The image to be scored must have at least one of the distortions for which the model was trained.

BRISQUE MATLAB inputs are input images, specified as a 2-D grayscale or RGB image and custom model trained on a set of quality-aware features, specified as a brisqueModel object. model is derived from natural scene statistics.

BRISQUE MATLAB output is Custom model trained on a set of quality-aware features, specified as a brisqueModel object. model is derived from natural scene statistics.

3.2.GIP: Histogram Equalization

When applying local brightening to an image, the histogram undergoes changes. The histogram of an image represents the distribution of pixel intensities across different brightness levels. Initially, the histogram of the image may have a concentration of values towards the darker end, indicating a lack of brightness or contrast.

After applying local brightening, the histogram will exhibit changes. The pixel intensities in the brighter areas of the image will be increased, causing a shift towards higher brightness levels in the histogram. This shift indicates an increase in the number of pixels with higher intensities, resulting in a brighter appearance for the specific regions targeted by the local brightening.

The extent of the histogram changes will depend on the intensity adjustments made during the local brightening process. If the adjustments are subtle, the shift in the histogram will be moderate. On the other hand, if more significant adjustments are applied, the shift in the histogram will be more pronounced, indicating a greater enhancement of brightness and contrast in the localized areas.

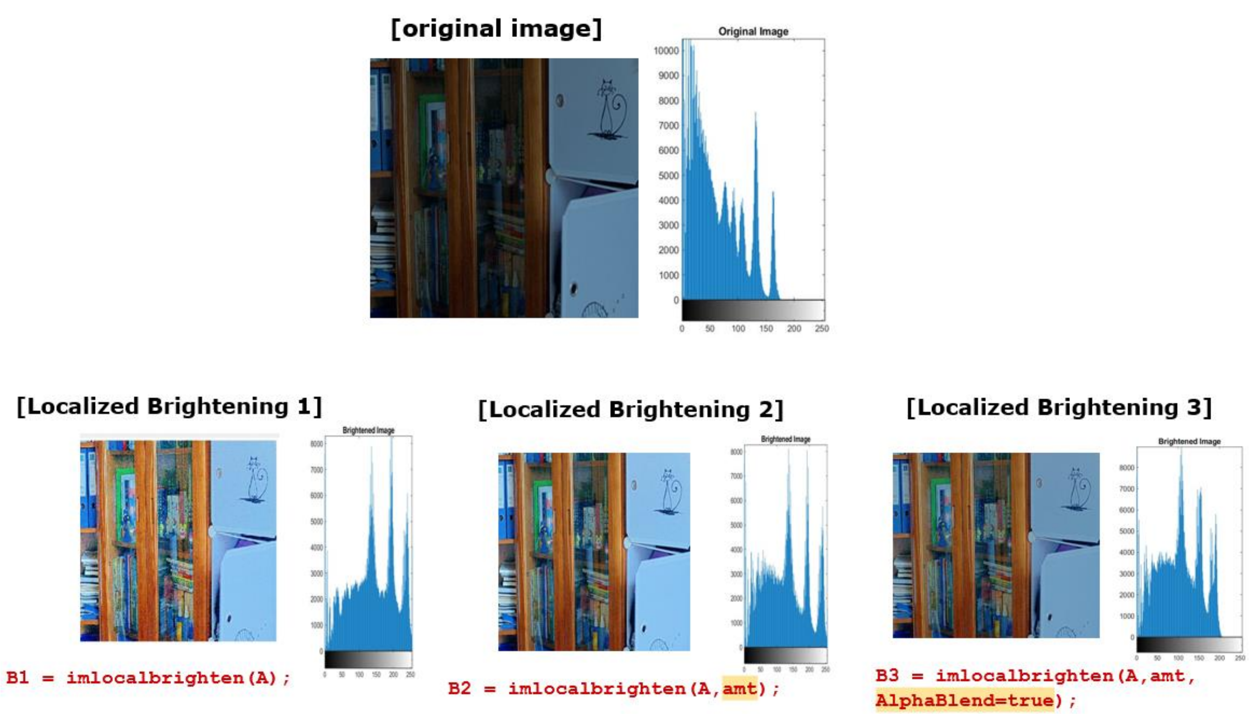


Figure 19 Image Enhancement result after Localized Brightening

Overall, local brightening helps to balance the brightness and contrast within an image, making it visually appealing and improving the visibility of details in specific areas. The histogram changes serve as a visual representation of the adjustments made during the process, showing the redistribution of pixel intensities and the resulting improvement in image quality.

When applying contrast adjustment to an image, the histogram undergoes changes as well. The histogram represents the distribution of pixel intensities across different brightness levels. Initially, the histogram may have a concentration of values in a specific range, indicating a limited range of contrast in the image.

After the contrast adjustment, the histogram will exhibit noticeable changes. The adjustments made to the pixel intensities result in a spread of values across a wider range of brightness levels. This spread causes the histogram to expand, indicating an increased diversity of pixel intensities.

If the contrast adjustment increases the difference between the darkest and brightest areas of the image, the histogram will show a broader distribution, with more pixel values spanning a larger range of brightness levels. This expansion indicates a significant enhancement in the contrast of the image.

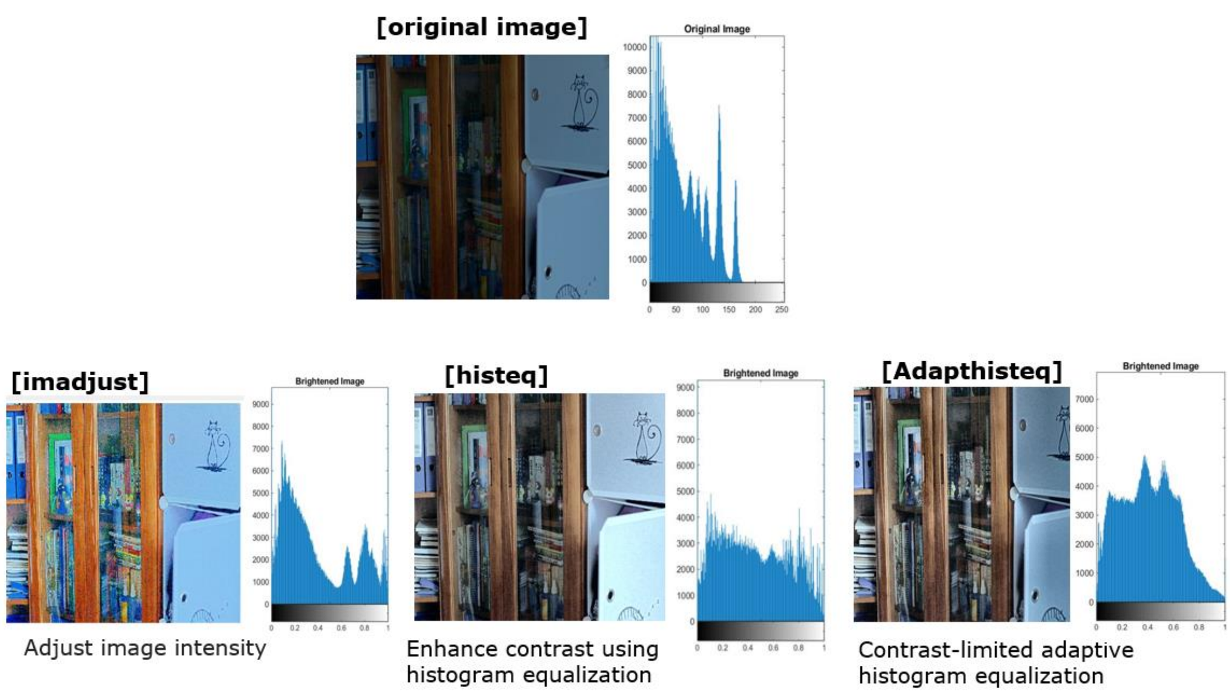


Figure 20 Image Enhancement result after Contrast Adjustment

On the other hand, if the contrast adjustment reduces the difference between the darkest and brightest areas, the histogram will show a narrower distribution, with fewer pixel values spread across a smaller range of brightness levels. This compression indicates a decrease in contrast, resulting in a more evenly distributed range of intensities.

The extent of the histogram changes will depend on the magnitude of the contrast adjustment applied. A subtle adjustment will cause a moderate shift or expansion/compression in the histogram, while a more significant adjustment will lead to a more pronounced and noticeable change.

Overall, contrast adjustment aims to improve the visual impact and clarity of an image by enhancing the difference between its darkest and brightest areas. The histogram changes provide a visual representation of the adjustments made, showing how the distribution of pixel intensities has been modified to achieve the desired contrast enhancement.

3.3.GIP: Dehaze

When applying a dehaze method to an image, the histogram undergoes changes as well. The histogram represents the distribution of pixel intensities across different brightness levels. Initially, the histogram of a hazy image may have a concentration of values towards the lower end of the brightness range, indicating reduced visibility and a lack of contrast.

After the dehazing process, the histogram will exhibit noticeable changes. The dehaze algorithm analyzes the image and estimates the amount of haze present in the scene. It then removes or reduces the haze, resulting in improved visibility and restoration of details.

The histogram changes depend on the specific dehaze method used, but generally, they involve a redistribution of pixel intensities to improve contrast and visibility. The concentration of values towards the lower end of the brightness range is reduced, and the histogram expands to cover a wider range of brightness levels.

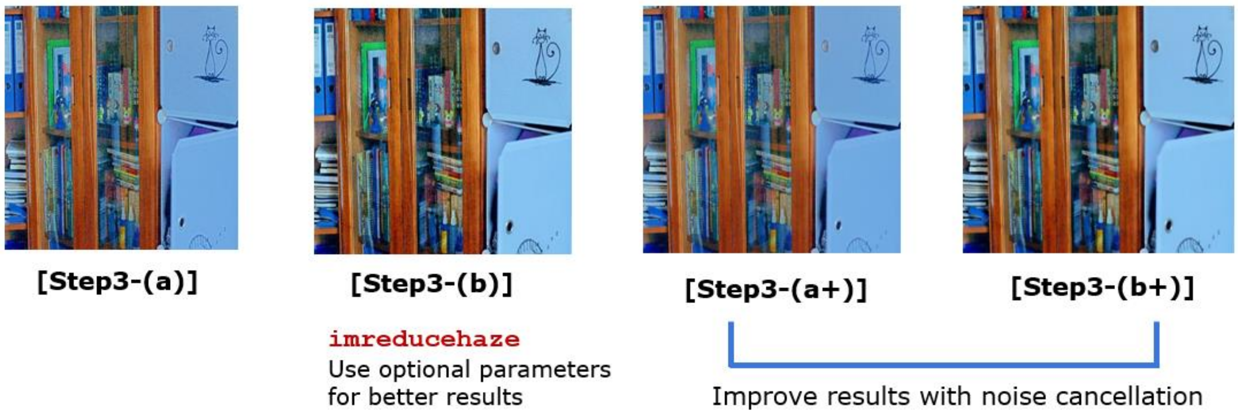


Figure 21 Image Enhancement result after Dehaze Enhancement

The dehaze method increases the intensity of pixel values that were affected by the haze, leading to a shift towards higher brightness levels in the histogram. This shift indicates an enhancement in the visibility of details and an increase in overall image contrast.

The extent of the histogram changes depends on the severity of the haze in the original image and the strength of the dehaze algorithm applied. A stronger dehaze effect will cause a more significant shift and expansion of the histogram, indicating a greater improvement in visibility and contrast.

Overall, the dehaze method helps to restore the visibility and clarity of a hazy image by reducing the effects of atmospheric haze or fog. The histogram changes reflect the redistribution of pixel intensities and the restoration of details, showing the improvement in contrast and visibility achieved through the dehazing process.

3.4.GIP: MSRCR

When applying multiscale Retinex with color restoration to an image, the histogram undergoes changes as well. The histogram represents the distribution of pixel intensities across different brightness levels and color channels. Initially, the histogram may have gaps or peaks indicating underexposed or overexposed regions and color imbalances.

The multiscale Retinex algorithm operates by decomposing the image into multiple scales, where each scale represents different levels of spatial frequency. It then applies a series of operations to enhance the dynamic range and color balance at each scale. This includes adjusting the intensity values, restoring color information, and enhancing local contrast.

The histogram changes will depend on the specific operations performed during the multiscale Retinex process. The intensity adjustments aim to equalize the distribution of pixel intensities, filling gaps and reducing peaks in the histogram. This results in an improved dynamic range, where details in both the darker and brighter areas become more visible.

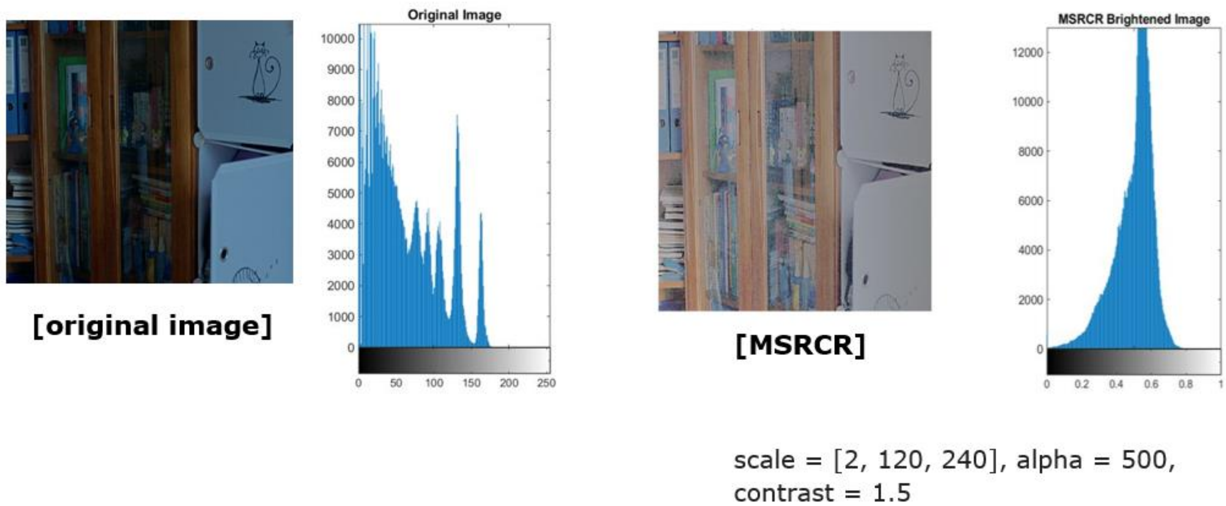


Figure 22 Image Enhancement result after MSRCR Enhancement

The color restoration step helps to correct color imbalances present in the image. By adjusting the color channels, the algorithm can restore natural and vibrant colors. The changes in the color channels will be reflected in the color histogram of the image, indicating a more balanced and accurate color representation.

The extent of the histogram changes will depend on the original image's characteristics and the strength of the multiscale Retinex with color restoration algorithm applied. Stronger adjustments will result in more pronounced changes in the histogram, indicating a significant improvement in the dynamic range and color rendition of the image.

Overall, multiscale Retinex with color restoration enhances the appearance of an image by improving its dynamic range, color balance, and overall visual quality. The histogram changes reflect the adjustments made to pixel intensities and color channels, demonstrating the enhancement in contrast, color accuracy, and overall image quality achieved through this technique.

3.5.NN architecture: EnlightenGAN

When applying EnlightenGAN to an image, the algorithm analyzes the input image and generates a modified version that appears more visually pleasing in terms of lighting and overall quality. The generator network of the GAN is responsible for generating the enhanced image, while the discriminator network provides feedback to guide the training process.

The EnlightenGAN algorithm operates by learning from a large dataset of images with diverse lighting conditions. During the training phase, the generator network learns to understand the relationship between the input image and the desired enhanced output. It adjusts the pixel values of the input image to improve its lighting, enhance details, and create a more visually appealing result.

The specific modifications made by EnlightenGAN can vary depending on the input image and the training of the network. The algorithm may adjust the brightness and contrast, reduce shadows, correct color imbalances, and enhance the overall visual quality of the image.

The resulting enhanced image produced by EnlightenGAN will typically exhibit improvements in terms of lighting conditions, overall brightness, contrast, and color balance. It aims to create an image that is visually pleasing, well-lit, and aesthetically enhanced compared to the original input.

EnlightenGAN can be particularly useful for improving images that suffer from poor lighting conditions, uneven illumination, or other visual imperfections. By leveraging the power of GANs and deep learning techniques, EnlightenGAN offers an effective solution for enhancing the appearance of images and creating visually stunning results.

Figure 23 Image Enhancement result after Enlightening GAN

3.6.Summary and comparison

3.6.1. LOL Dataset

The LOL dataset is composed of 500 low-light and normal-light image pairs and divided into 485 training pairs and 15 testing pairs. The low-light images contain noise produced during the photo capture process. Most of the images are indoor scenes. All the images have a resolution of 400×600.

In this research, 14 testing images are applied to compare the performance of models.

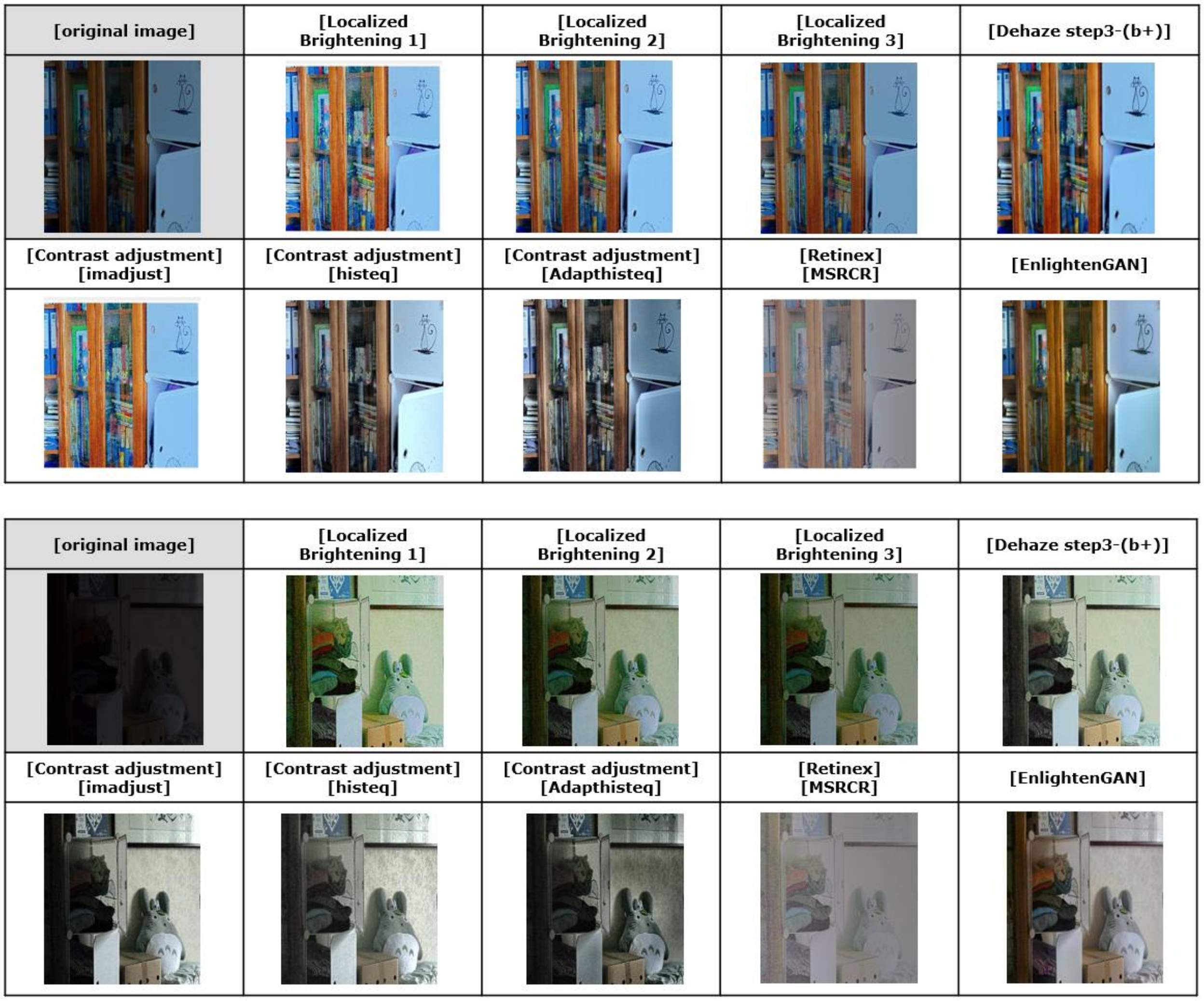


Figure 24 Comparison of the qualitive of the image after applied enhancement in studied methods in LOL dataset.

As the LOL Dataset has a target high light level dataset, calculate the score for reference. The result shows that NIQE = **4.2526**, BRISQUE= **18.1645** and PIQE= **35.6819.**

Category	**Methods**	**SSIM**	**NIQE**	**BRISQUE**	**PIQE**
Histogram Equalization	Contrast adjustment	0.5111108	9.77666	36.9093	53.5622
		0.4292451	10.0759	39.9343	57.7805

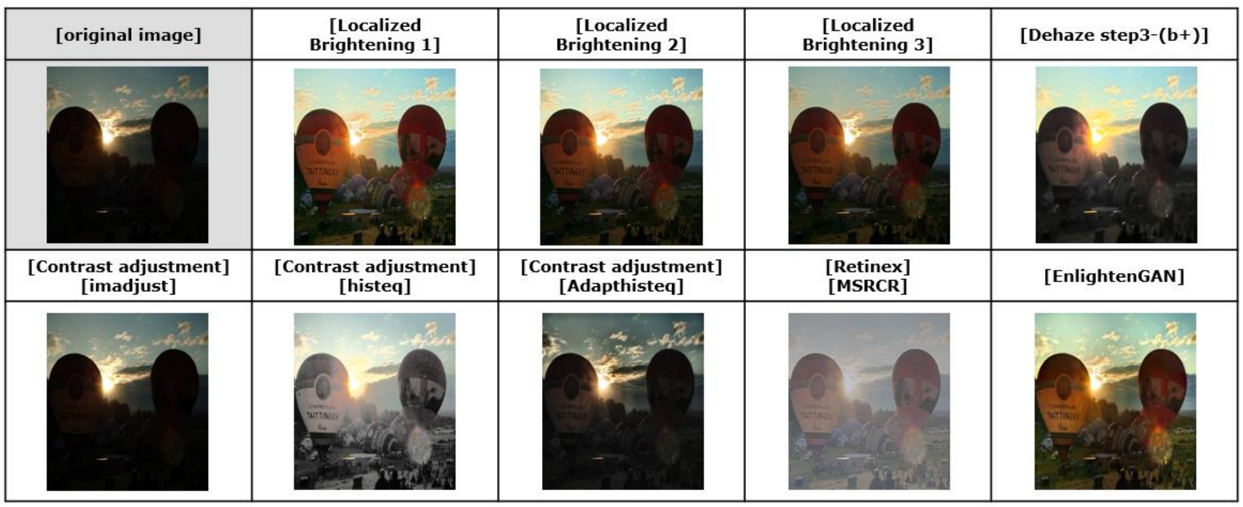
		0.4645869	10.3384	39.6008	57.7821
	Localized Brightening	0.4389536	10.6225	39.413	59.2722
		0.4971988	10.1943	38.0217	56.3881
		0.5277159	10.0454	37.3499	54.4252
Dehaze	Use haze removal algorithms to enhance low-light images	0.4518117	10.7686	39.6819	59.7893
		0.4718311	9.89757	37.7117	55.6493
		0.4399195	11.3579	40.9582	62.06
	Use haze removal algorithms to enhance low-light images with Noise reduction	0.5572641	7.41552	26.5828	39.4974
		0.5766776	6.71176	26.3761	36.7308
		0.5591991	7.76131	28.8303	43.5972
Retinex	Multiscale Retinex	0.4191170	8.09221	28.7679	39.5529
Deep Learning	EnlightenGAN	**0.6500**	**4.8891**	**22.6811**	**30.4929**

To conclude, the score for the Deep Learning method shows the best result in all 4 metrics SSIM, NIQE, BRISQUE and PIQE. The Retinex model shows low Similarity with target compared with other method, but regarding the quality of output image, it’s better than Histogram equalization and dehaze method. The Dehaze method has better results compared to the Histogram Equalization method.

3.6.2. MEF Dataset

Multi-exposure image fusion (MEF) is considered an effective quality enhancement technique widely adopted in consumer electronics, but little work has been dedicated to the perceptual quality assessment of multi-exposure fused images. In this paper, we first build an MEF database and carry out a subjective user study to evaluate the quality of images generated by different MEF algorithms. There are several useful findings. First, considerable agreement has been observed among human subjects on the quality of MEF images. Second, no single state-of-the-art MEF algorithm produces the best quality for all test images. Third, the existing objective quality models for general image fusion are very limited in predicting perceived quality of MEF images. Motivated by the lack of appropriate objective models, we propose a novel objective image quality assessment (IQA) algorithm for MEF images based on the principle of the structural similarity approach and a novel measure of patch structural consistency. Our experimental results on the subjective database show that the proposed model well correlates with subjective judgments and significantly outperforms the existing IQA models for general image fusion. Finally, we demonstrate the potential application of the proposed model by automatically tuning the parameters of MEF algorithms.

In this research, 17 testing images are applied to compare the performance of models.



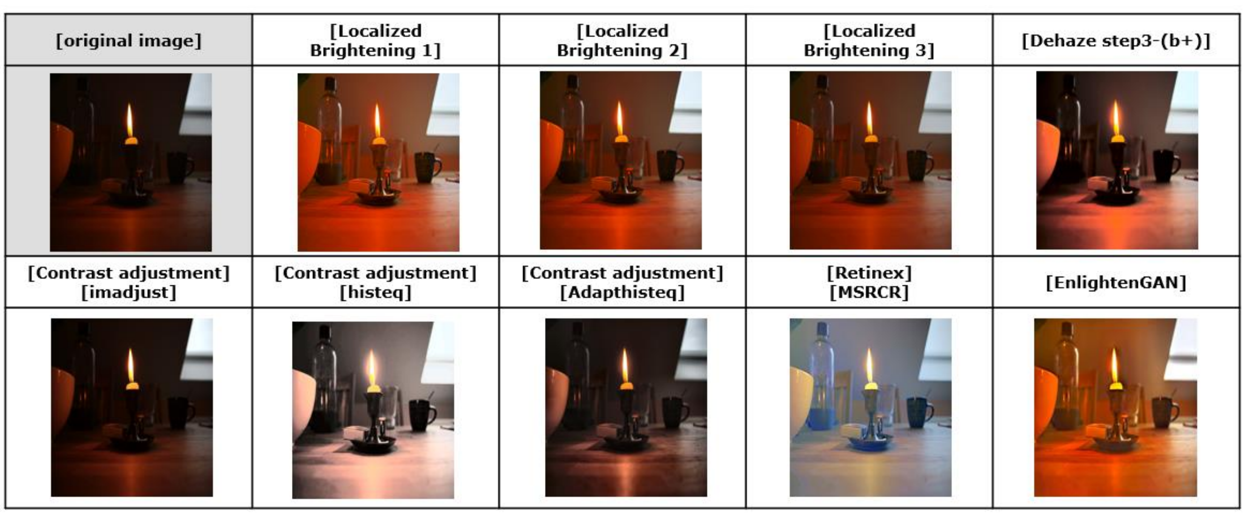


Figure 25 Comparison of the qualitive of the image after applied enhancement in studied methods in MEF dataset.

Category	Methods	NIQE	BRISQUE	PIQE
Histogram Equalization	Contrast adjustment	7.69593	36.1297	59.8091
		7.8383	40.21	64.1037
		7.81795	34.9389	58.3548
	Localized Brightening	7.8418	34.348	55.5775
		7.72586	34.1196	53.8428
		7.66902	35.332	53.8909
Dehaze	Use haze removal algorithms to enhance low-light images	3.63692	19.9324	32.7945
		3.62341	23.4696	41.1055
		3.7026	19.5738	34.9504
	Use haze removal algorithms to enhance low-light images with Noise reduction	4.05015	40.5357	55.2059
		3.87855	41.1819	55.4139
		4.20154	41.1159	58.5526
Retinex	Multiscale Retinex	7.66597	33.5825	54.5813
Deep Learning	EnlightenGAN	2.8956	23.6407	32.2574

4. References

Gonzalez, Rafael C. (2018). Digital image processing. Richard E. Woods (4th ed.). New York, NY: Pearson. pp. 138–140. ISBN 978-1-292-22304-9. OCLC 991765590

Wang, L.; Liu, Z.; Siu, W.; Lun, D.P.K. Lightening Network for Low-Light Image Enhancement. IEEE Trans. Image Process. 2020, 29, 7984–7996.

Wang, W.; Wu, X.; Yuan, X.; Gao, Z. An Experiment-Based Review of Low-Light Image Enhancement Methods. IEEE Access 2020, 8, 87884–87917.

Wang, Y.; Liu, H.; Fu, Z. Low-Light Image Enhancement via the Absorption Light Scattering Model. IEEE Trans. Image Process. 2019, 28, 5679–5690.

Lu, Y.; Jung, S.-W. Progressive Joint Low-Light Enhancement and Noise Removal for Raw Images. IEEE Trans. Image Process. 2022, 31, 2390–2404.

Lee, S.; Kim, N.; Paik, J. Adaptively partitioned block-based contrast enhancement and its application to low light-level video surveillance. SpringerPlus 2015, 4, 431.

Reza, A.M. Realization of the Contrast Limited Adaptive Histogram Equalization (CLAHE) for Real-Time Image Enhancement. J. VLSI Signal Process. Syst. Signal Image Video Technol. 2014, 38, 35–44.

Kim, Y. Contrast enhancement using brightness preserving bi-histogram equalization. IEEE Trans. Consum. Electron. 1997, 43, 1–8.

Kim, M.; Chung, M. Recursively separated and weighted histogram equalization for brightness preservation and contrast enhancement. IEEE Trans. Consum. Electron. 2008, 54, 1389–1397.

Jiang, J.; Zhang, Y.; Hu, M. Local Histogram Equalization with Brightness Preservation. Acta Electron. Sin. 2006, 34, 861–866.

Srinivas, K.; Bhandari, A. Low light image enhancement with adaptive sigmoid transfer function. IET Image Process. 2020, 14, 668–678.

Kim, W.; Lee, R.; Park, M.; Lee, S.-H. Low-light image enhancement based on maximal diffusion values. IEEE Access 2019, 7, 129150–129163.

Huang, S.; Cheng, F.; Chiu, Y. Efficient Contrast Enhancement Using Adaptive Gamma Correction with Weighting Distribution. IEEE Trans. Image Process. 2013, 22, 1032–1041.

Jobson, D.J.; Rahman, U.; Woodell, G.A. Properties and performance of a center/surround retinex. IEEE Trans. Image Process. 1997, 6, 451–462.

Jobson D J, Rahman Z U, Woodell G A. A multiscale retinex for bridging the gap between color images and the human observation of scenes[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 1997, 6(7):965-76

Jobson, D.J. Retinex processing for automatic image enhancement. J. Electron. Imaging 2004, 13, 100–110.

An alternative technique for the computation of the designator in the retinex theory of color vision," in Proc. Nat. Acad. Sci., vol. 83, pp. 3078–3080, 1986.

Zhou, W., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. "Image Quality Assessment: From Error Visibility to Structural Similarity." IEEE Transactions on Image Processing. Vol. 13, Issue 4, April 2004, pp. 600–612.

Mittal, A., R. Soundararajan, and A. C. Bovik. "Making a Completely Blind Image Quality Analyzer." IEEE Signal Processing Letters. Vol. 22, Number 3, March 2013, pp. 209–212.

N. Venkatanath, D. Praneeth, Bh. M. Chandrasekhar, S. S. Channappayya, and S. S. Medasani. "Blind Image Quality Evaluation Using Perception Based Features", In Proceedings of the 21st National Conference on Communications (NCC). Piscataway, NJ: IEEE, 2015

Mittal, A., A. K. Moorthy, and A. C. Bovik. "No-Reference Image Quality Assessment in the Spatial Domain." IEEE Transactions on Image Processing. Vol. 21, Number 12, December 2012, pp. 4695–4708.

Mittal, A., A. K. Moorthy, and A. C. Bovik. "Referenceless Image Spatial Quality Evaluation Engine." Presentation at the 45th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, November 2011.

https://ww2.mathworks.cn/help/images/low-light-image-enhancement.html

Fast efficient algorithm for enhancement of low lighting video | IEEE Conference Publication | IEEE Xplore https://kr.mathworks.com/help/images/low-light-image-enhancement.html

https://kr.mathworks.com/help/visionhdl/ug/low-light-enhancement.html

https://link.springer.com/article/10.1007/s11263-020-01418-8

https://github.com/Lilin2015/MSRCR

https://ieeexplore.ieee.org/document/597272

https://ieeexplore.ieee.org/document/4895264