SN Computer Science (2026) 7:347
https://doi.org/10.1007/s42979-026-04951-4

ORIGINAL RESEARCH

An Improved Method for Enhancing Images Quality Based on Convolution Efficient Transformer

Sinh Van Nguyen[1,2] · Vinh Xuan Nguyen[1,2] · Thi-Ngoc-Hanh Le[1,2]

Received: 10 January 2026 / Accepted: 29 March 2026

Abstract
Image quality enhancement remains a critical challenge in image processing techniques, especially for images captured under suboptimal lighting conditions that can generate low contrast, color distortion, and noise. These degradations not only affect visual aesthetics but also hinder the performance of downstream image processing tasks. The primary objective of image quality enhancement is to improve the visual quality of such images to benefit subsequent processing. Despite extensive research, achieving high-quality enhanced images remains challenging. Traditional image quality enhancement techniques often address only overexposure or underexposure, which can fail when both issues are present. Deep learning has recently been increasingly adopted in image processing, demonstrating significant potential to enhance image quality with underexposure, overexposure, or a combination of both. In this paper, we enhance the image quality over prior work by incorporating a convolution-based efficient transformer (CET). The proposed approach consists of four main stages. First, we enrich the training data by building a new dataset. Second, the collected dataset is preprocessed to eliminate noisy and low-quality samples. Third, CET is integrated into the Color Shift Estimation and Correction (CSEC) architecture to enable more effective feature extraction. Fourth, an additional depth-based loss function is introduced, leveraging depth maps to improve accuracy and consistency in image correction. Finally, the enhanced CSEC model is trained on both public datasets and the newly constructed dataset. The experimental results demonstrate the effectiveness and superior performance of the proposed approach compared to existing methods.

Keywords Image processing · Quality enhancement · Illumination and resolution · Deep learning

Introduction

Image quality enhancement is a fascinating field that not only improves the quality and usability of visual data but also supports restoring objects in images. This is a fundamental process in digital image processing that aims to improve the quality of an image. The improved image is then applied in different fields such as medical image processing and photography. Whether it is bringing out details in a low-light or over-exposed photo, or clarifying the edges of objects in a scanned document. Besides, enhancing images plays a crucial role in extracting valuable information and improving the overall visual experience.

Improving Image quality involves applying various traditional techniques to adjust brightness, contrast, and sharpness [1, 2], reduce noise, correcting distortions [1, 3], and apply histogram equalization [4, 5]. During the improvement steps, noise removal and enhancing the effects of illuminating light from the image using Retinex methods [6], as well as adjusting the color balance [1, 3, 6]. Besides traditional methods, machine learning has made significant strides in the field of image processing in recent years. These methods have proven effective in improving the quality of images, whether they are underexposed, overexposed, or both [7–9].

Addressing the ongoing need for improved image quality enhancement, this paper investigates a deep learning-based approach for image quality enhancement. Our research focuses on the specific challenges of color distortion and detail loss in poorly illuminated images. First, we introduce an additional dataset specifically designed to facilitate more

✉ Sinh Van Nguyen
nvsinh@hcmiu.edu.vn

1 International University HCM-VNU, Ho Chi Minh City, Vietnam

2 Vietnam National University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

Published online: 10 April 2026

effective training of deep learning models for challenging scenarios involving poor illumination. Second, we propose an enhanced and refined image quality enhancement pipeline. Specifically, to improve the feature extraction capabilities within the DeepWBNet architecture, we strategically incorporate two Residual Blocks. This architectural modification aids in the crucial refinement of features essential for the accurate generation of weight maps and visually consistent pseudo-normal images, thereby substantially contributing to the model's ability to perform effective and accurate color shift estimation and correction. Furthermore, we integrate a new loss function (a depth-based loss function) that encourages the model to achieve not only pixel-level accuracy but also to learn perceptually relevant features. Finally, we employ the Color Shift Estimation and Correction (CSEC) method as the core training strategy for our proposed model, leveraging the benefits of our new dataset and architectural innovations. This article is an improved version of our previous paper which was selected from the conference paper [10]. We modified our paper by more than thirty percent as required. The major contributions of this paper include a new image quality enhancement dataset; an improved model by modifying the network architecture model (CET) for feature extractor with residual blocks; and an added depth-based loss function that helps increase the accuracy.

The remainder of this article is organized as follows: "Related Works" provides a review of related work in image processing techniques to enhance image quality. "Our proposed method" details the construction of our new dataset and describes our proposed image quality enhancement method. "Implementation and Results" presents the implementation and experimental results. "Discussion, Evaluation and Comparison" provides a discussion and evaluation of the proposed method. Finally, " Conclusion and Future work" concludes the paper and discusses future work.

Related Works

This section presents the techniques related to image quality enhancement area in both traditional methods and machine learning-based methods.

Brightness Adjustment: Brightness adjustment is a fundamental technique in image processing used to alter the overall lightness or darkness of an image. This is typically achieved by adding or subtracting a constant value to the pixel values of the image [1]. Mathematically, brightness adjustment operations can be expressed as follows: $(x, y)' = (x, y) + b$, where *f(x,y)* is the original pixel value at position (x, y), *f(x,y)'* is a new pixel value, and b is a brightness adjustment factor. If b is a positive value then the image will be brighter and if b is a negative value then the image brightness level will decrease.

Contrast Stretching: Contrast stretching, also known as normalization, is an image quality enhancement technique used to improve the contrast of an image by expanding the range of intensity values. This method aims to make the dark areas darker and the bright areas brighter, thereby enhancing the overall visual quality of the image [1]. Contrast can be divided into three types: low contrast, good contrast and high contrast. Images with low contrast are mostly bright or dark, with pixel values concentrated in one region of the histogram. If values are on the left, the image is darker; if on the right, the image is lighter; if in the middle, the image is neither too bright nor too dark.

Gamma (Power-Law): Transformations The gamma correction method will produce a brighter and more natural image. Unlike brightness adjustment, which is linear, gamma correction is nonlinear. When depicted in graphical form, the function of the gamma is curve-shaped. The darkest and brightest areas of the gamma graph will not have much effect on different gamma arrangements. However, the middle area of the graph will have an effect by following the arrangement. The equations of gamma can be defined as follows: $f(x, y)' = f(x, y)^1/\gamma$, where *f(x, y)'* is the image after the gamma correction process and *f(x, y)* is the image before the gamma correction process. The symbol γ is the gamma correction factor with a value range of $(0 < \gamma < 1)$.

Linear Gray Level Transformations: The gray transformation method is a spatial-domain image quality enhancement algorithm. The principle of this method is to transform the gray values of single pixels into other gray values by means of a mathematical function, which is usually called a mapping-based approach. Such a method enhances an image by modifying the distribution and dynamic range of the gray values of the pixels [3].

A linear transformation of gray values, also known as a linear stretching, is a linear function of the gray values of the input image, and the formula is as follows [3]:

$$g(x, y) = C.f(x, y) + R \tag{1}$$

where *f(x, y)* and *g(x, y)* represent the input and output images, respectively, and *C* and *R* are the coefficients of the linear transformation.

Logarithmic Transformations: A logarithmic transformation means that each pixel's value in the output image is related to the logarithm of the corresponding pixel's value in the input image. This transformation is ideal for very dark images because it can enhance the lower gray values while reducing the range of the higher gray values. The typical form of logarithmic transformations is as formula:

$$g(x,y) = log(1 + c.f(x,y)) \tag{2}$$

where c is a control parameter.

Histogram Equalization (HE): Histogram equalization is a widely used technique in image quality enhancement that aims to improve the contrast of an image by redistributing its pixel intensity values. Histogram equalization utilizes the image's histogram to enhance its quality. Given a greyscale image, the goal is to compute a transformation that, when applied to the gray values of the original image, produces a uniform distribution of the intensity values. This method makes hidden details in dark areas visible again, significantly enhancing the visual quality of the input image [3, 11].

Adaptive histogram equalization (AHE): The basic idea of AHE is to separate an image into several sub-blocks, and each sub-block is processed by histogram equalization, respectively. The AHE algorithm can be described as follows: (1) Set the size of a window, and select a sub-block of the input image according to the window. (2) Apply HE algorithm to the sub-block, and record the output. (3) Move the window horizontally or vertically and repeat #1 and #2 until all pixels in the input image are modified. (4) Organize all the enhanced sub-blocks into one image as output.

Contrast Limited Adaptive Histogram Equalization (CLAHE): The Contrast Limited Adaptive Histogram Equalization (CLAHE) method operates on the same principle as traditional histogram equalization. In CLAHE, the im-age is divided into several sub-images of size (n × n). Histogram equalization is then applied to each sub-image individually, based on the divisions made earlier. CLAHE is a kind of adaptive histogram equalization in which the contrast amplification can be limited to reduce the problem of noise amplification [12].

Mean Filter (Average Filter, Blur Filter, Box Filter Kernels): Mean filter is used to smoothing the image by calculating the average value of pixels in the image. The process involves considering the surrounding pixels. The pixels to be processed are included in an N × N matrix. It is based around a kernel, which represents the shape and size of the neighborhood to be sampled when calculating the mean. Often a 3 × 3 square kernel is used, or a 5 × 5 squares. Mathematically, the mean filtering has the same weight as the neighboring pixel defined as follows:

$$f(x,y) = \frac{1}{mn}\sum_{k=1}^{m}\sum_{l=1}^{n} U(x+k-1, y+l-1) \tag{3}$$

where *f(x, y)* represents the image of the result. while *U(x, y)* represents the input image used. The upper bound value of m and n represent the size of the row and column of the mean filtering.

Gaussian Filter: The Gaussian filter is a linear filter that uses the Gaussian function to set pixel weights, and is widely used for smoothing, blurring, and eliminating noise. In the Gaussian filter, the linear process is calculated by multiplying each neighboring pixel by a corresponding weight and summing the results to obtain the value for a specific coordinate point, denoted as (x, y). The mechanism of the linear spatial filter is to move the center of a filter mask from one point to another. In each pixel (x, y), the result of the filter at that point is the sum of the multiplication of the filter coefficients and the corresponding neighbor pixels in the filter mask range. Gaussian filters have two types of filters: one-dimensional and two-dimensional with the forms as below.

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\frac{x^2}{2\sigma^2}\right)} \tag{4}$$

where σ is the standard deviation of the distribution.

$$G(x) = \frac{1}{2\pi\sigma^2} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} \tag{5}$$

where σ is the standard deviation of the same distribution as the one-dimensional gaussian function. For x and y are expressed as coordinate points rows and columns in image pixels.

Non-Linear Filter—Median Filter: The median filter is widely used as it is very effective at removing noise while preserving edges. The median filter works by moving through the image pixel by pixel, replacing each value with the median value of neighbouring pixels. The median is calculated by first sorting all the pixel values from the matrix neighbor into numerical order, and then replacing the pixel being considered with the middle (median) pixel value. Generally the point to be processed along with the points around it is inserted into a matrix of size N x N. This matrix is called matrix neighbor (neighboring matrix), which slides, pixel by pixel, over the entire image.

Non-Linear Filter—Conservative Filter: Conservative filter is one technique to reduce existing noise on the image. In the median filter, the filter process uses the middle value of the processed neighboring pixels and pixels. In the conservative filter, the values used are the minimum and maximum values but excluding the processed middle pixels. The calculation process is performed as follows: If the middle pixel's value is within the range of the surrounding pixels' minimum and maximum values, it remains unchanged. If it exceeds the maximum value, it is replaced with the maximum value. If it is less than the minimum value, it is replaced with the minimum value.

Retinex Methods: The Retinex theory is based on the perception of color by the human eye and the modeling of color invariance [3]. The essence of this theory is to determine the reflective nature of an object by removing the effects of the illuminating light from the image. Based on Retinex theory, the human visual system processes information in a specific way during the transmission of visual information, thus removing a series of uncertain factors such as the intensity of the light source and unevenness of light. Consequently, only information that reflects essential characteristics of the object, such as the reflection coefficient, is retained. Based on the illumination-reflection model, an image can be expressed as the product of a reflection component and an illumination component: $I(x, y) = R(x, y)L(x, y)$. If L(x, y) can be estimated from I(x; y), then the reflection component can be separated from the total amount of light, and the influence of the illumination component on the image can be reduced, thus enhance the image [3].

Single-Scale Retinex (SSR): The SSR algorithm obtains a reflection image by estimating the ambient brightness. The formula is as follows: $\log R_i(x, y) = \log I_i(x, y) - \log[G(x, y) * I_i(x, y)]$, where $I(x, y)$ represents the input image, $R(x, y)$ represents the reflection image, i represents the various color channels, (x, y) represents the position of a pixel in the image, $G(x; y)$ represents the Gaussian surround function, and $*$ represents the convolution operator.

Multi-Scale Retinex (MSR): Maintain a balance between dynamic range compression and color constancy. The MSR algorithm:

$$\begin{aligned} MSR &= \log R_i(x, y) \\ &= \sum_{k=1}^{N} \omega_k \left\{ \log I_i(x, y) - \log \left[G_k(x, y) * I_i(x, y) \right] \right\} \end{aligned} \tag{6}$$

where i represents the three color channels; k represents the Gaussian surround scales; N is the number of scales, generally 3; and the ω parameters are the scale weights. Unlike the SSR algorithm, the MSR algorithm leverages the advantages offered by multiple scales, leading to enhanced image details, improved contrast, better color consistency, and an overall enhanced visual effect.

Multiscale Retinex With Color Restoration (MSRCR): During image quality enhancement, the SSR or MSR algorithm is applied separately to the R, G, and B color channels. This can alter the relative proportions of these channels compared to the original image, leading to color distortion. To address this issue, the MSRCR algorithm has been developed. It includes a color recovery factor (C) for each channel, calculated based on the proportional relationships among the three color channels in the input image. This factor is then used to correct the output image's colors, eliminating color distortion. The algorithm takes advantage of the convolution operation with Gaussian functions. Dynamic range compression and color constancy are achieved for features at large, medium and small scales, thus yielding a relatively ideal visual effect. Nowadays, machine learning-based approaches are increasingly being applied and have proven to be effective solutions in image quality enhancement, and this paper will introduce some of these methods.

URetinex-Net: URetinex-Net was introduced in 2022, a Retinex-based deep unfolding network, which unfolds an optimization problem into a learnable network to decompose a low-light image into reflectance and illumination layers [13]. URetinex-Net includes three modules, i.e. initialization module, unfolding optimization module, and illumination adjustment module. The Retinex-based approach allows for precise manipulation of reflectance and illumination layers for better image quality; however, its complex network architecture requires significant computational resources and expertise. Besides, the U-Net architecture, initially designed for biomedical image segmentation by [14], which has been widely used as feature extractor across various domains. Its encoder-decoder structure with skip connections enables effective multi-scale feature learning. Over time, U-Net has been adapted for diverse applications, such as the backbone to to extract brightened and darkened features for color correction tasks [7], content feature analyzer in image/video resizing systems [15, 16]. These adaptations often modify the network's depth, width, or connectivity, demonstrating U-Net's flexibility and effectiveness in both low-level and high-level learning tasks.

Local Color Distributions Prior: When the illumination of the input image contains both over- and underexposure problems, these existing methods may not work well because they are typically designed to address either the over- or under-exposure problem in the input image. In 2022, Wang Haoyuan, Ke Xu, and Rynson WH Lau. Introduced the"Local color distributions prior"for image quality enhancement. They observe from the image statistics that the local color distributions (LCDs) of an image depending on the local illuminations, suffering from both problems tend to vary across different regions of the image. Base on this observation, they proposed LCDs as a prior for locating and enhancing the two types of regions (over-/underexposed regions) [17]. First, they utilize the LCDs to depict these regions and introduce a novel local color distribution embedded (LCDE) module that formulates LCDs in multiple scales to model correlations across different areas. Second, they propose a dual-illumination learning mechanism to enhance both types of regions. Third, they create a new dataset to support the learning process, following the camera image signal processing (ISP) pipeline to produce

standard RGB images with both under- and over-exposures from raw data [17].

Color Shift Estimation-and-Correction (CSEC): Color Shift Estimation and Correction (CSEC) method [7] was introduced by Yiyu Li et al. to enhance images with both over- and under-exposures by learning to estimate and correct colors of input image. This method first derive the color feature maps of the brightened and darkened versions of the input image via a UNet-based network, then use a pseudo-normal feature generator to produce pseudo-normal color feature maps. There is a COlor Shift Estimation (COSE) module to estimate the color shifts between the derived brightened (or darkened) color feature maps and the pseudo-normal color feature maps. The COSE module used for correcting the estimated color shifts of the over- and under-exposed regions separately. In addition, this method propose a novel COlor MOdulation (COMO) module to modulate the separately corrected colors in the over- and under-exposed regions to produce the enhanced image.

As presented in [18], the authors proposed a method for image feature extraction. This paper introduce the Convolution-based Efficient Transformer (CEFormer) which is designed to enhance the standard Transformer framework by integrating convolutional strategies to introduce crucial inductive biases, specifically translation invariance, spatial locality, and scale invariance. The CEFormer is designed to bridge the gap between the local feature extraction strengths of Convolutional Neural Networks (CNNs) and the global dependency modeling of Transformers. CEFormer provides a efficient and accurate way for the network to"see" and extract the structural features (edges, textures, and global context).

The CEFormer architecture consists of the following key components: Lightweight Convolution Module: Instead of the large, non-overlapping 16×16 patch kernels used in traditional Vision Transformers (ViT), CEFormer utilizes a lightweight module for initial image processing. This module consists of a convolution operation, followed by Batch Normalization and Max Pooling, which improves model stability and accelerates convergence. E-Attention with Translation Invariance: The model incorporates an efficient attention mechanism (E-Attention). To introduce translation invariance, it integrates the weights of depthwise convolutions directly into the attention mechanism, allowing it to better handle limited datasets. Feedforward Network (FFN) with Locality: Locality is introduced by incorporating a 3×3 depthwise convolution into the feedforward network. This allows the network to aggregate local information similarly to an inverted residual structure. Feedforward Network with Scale Invariance: To capture multi-scale context information, dilated convolutions with varying expansion rates are added to the feedforward network. This allows the model to extract features from objects of different scales without losing spatial resolution. By combining these elements, CEFormer achieves a balance between the long-range dependency modeling of Transformers and the structural advantages (inductive biases) of CNNs.

A research based on a combination of GNN and GAN is presented in [19]. GNN is first used to learn the underlying structure and features of the 3D data; then, GAN is used to generate high-quality, refined versions of the data. This approach proves to be effective in addressing the challenges associated with this type of data. Nguyen et.al. [20] proposed a deep learning method for face recognition that augments data in the training process. After pre-processing data based on image processing techniques, Retnet-v1 is used to improve recognition accuracy. At the same time as our research, a recent study on adjusting over-exposure and under-exposure of the images is presented in [21]. This research presented a new exposure method (Omnidirectional Spectral Mamba: OSMamba) for correcting under-exposed and over-exposed regions in images. The method is based on processing the diffusion model, amplitude, and phase spectra, effectively correcting illumination and recovering structures. This method obtained better results compared to the SOTA methods in SSIM and PSNR. However, the calculation of learned perceptual similarity between the images (LPIPS) is not mentioned.

Our proposed method

System overview

Our proposed framework is outlined in Fig. 1, which covers the major phases of training data preparation, pre-processing data, and training. In this system, our contributions include a custom dataset, improved pre-processing, enhancements to model architecture, and loss function. To support our model's learning under non-uniform illumination, we construct a new dataset of 350 unique scenes, each with under-exposed and over-exposed versions, totaling 700 images. To reduce noise and improve input quality, we apply a denoising filter during the pre-processing phase. In the model's architecture, we further enhance the baseline model CSEC [7] by applying CET for feature extractor, adding two Residual layers, which strengthen feature extraction and improve

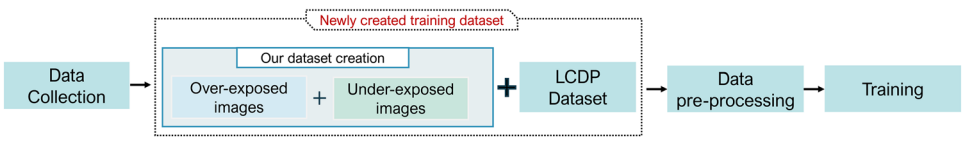

Fig. 1 Overview of our process

correction performance. Finally, in terms of optimization, we introduce an additional Mean Square Error (MSE) loss term to complement the original loss functions.

Training data preparation

The LCDP dataset [17] has recently been publicly used in the research of image quality enhancement. This dataset provides 1733 image pairs of under-exposed and over-exposed scenes, partitioned into 1415 for training, 100 for validation, and 218 for testing (available at github.com/onpix/LCDPNet). However, the LCDP dataset has drawbacks in capturing the diversity of real-world lighting conditions, lacks sufficient variation in non-uniform or naturally complex illumination, which are commonly encountered in practical applications.

To overcome the above limitations of LCDP and enhance the model's generalizability, we newly construct a dataset and combine it with the public LCDP set. More specifically, we generate an additional set comprising 700 image pairs, 350 under-exposed and 350 over-exposed, derived from 350 distinct ground truth scenes. Our new dataset is split into a 80:10:10 ratio as follows: Training: 560 images (80%), consisting of 280 over-exposed and 280 under-exposed images. Validation: 70 images (10%), consisting of 35 over-exposed and 35 under-exposed images. Testing: 70 images (10%), consisting of 35 over-exposed and 35 under-exposed images. These scenes were captured using two devices, Canon EOS Rebel SL1 camera and a Samsung M20 smartphone. The images were subsequently transformed using the OpenCV library to simulate realistic lighting imbalances. More specifically, we apply following linear transformation formulas respectively:

$$g(x,y) = alpha * f(x,y) + beta, \text{with alpha } = 1.0, \quad \text{beta} = -80, \tag{7}$$

and

$$g(x,y) = alpha * f(x,y) + beta, \text{with alpha } = 1.0, \quad \text{beta} = 80, \tag{8}$$

where $f(x, y)$ represents original input. In the context of images, $f(x, y)$ is color value of a pixel at coordinates (x, y). The dataset is then split into 80% for training, 10% for validation, and 10% for testing. By introducing more diverse images and challenging lighting variations, our dataset enriches LCDP and enables our model to better learn diverse pattern in real-world conditions.

Prior to training, a median filter with a 3×3 kernel is applied to all images in the dataset as a preprocessing step. The median filter is a widely adopted technique in image processing for noise reduction, particularly effective in eliminating salt-and-pepper noise while preserving important image features such as edges. Unlike linear filters that may blur fine details, the median filter operates by sliding a window across the image and replacing each pixel's intensity with the median value of the neighboring pixels within the window. To be specific, applying a median filter to a single image is formulated as follows:

$$medianBlur(InputArray \quad src, OutputArray \quad dst, int \quad ksize), \tag{9}$$

where *src* is the source (input) image; *dst* is the destination (output) image where the blurred image will be returned; *ksize* specifies the size of the kernel (or window). And, *medianBlur* is openCV library used to smoothen input images. This non-linear approach ensures that edge sharpness is maintained while suppressing outlier noise, resulting in cleaner input data that improves the stability and performance of the training process [1]. The algorithm for applying a median filter (Algorithm 1) to a folder of images using the OpenCV library is as follows:

```
Input: Directory name (directoryName), File filter (fileFilter)
Output: 0 for success, -1 for error
Create: directoryOutput ← directoryName + "output/"
Find: filePaths ← List of all files in directoryName matching fileFilter
count ← Number of elements in filePaths
for i = 0 to count - 1 do
    fullFilePath ← filePaths[i]
    filename ← Extract base filename from fullFilePath
    Read: image ← Image from directoryName + filename
    if image is empty then
        return -1
    end if
    Create: image_dst ← Empty image object
    Apply: Median blur filter (kernel size 3) to image and store in image_dst
    Save: image_dst to directoryOutput + filename
end for
return 0 =0
```

Algorithm 1 applyMedianFilterFoder

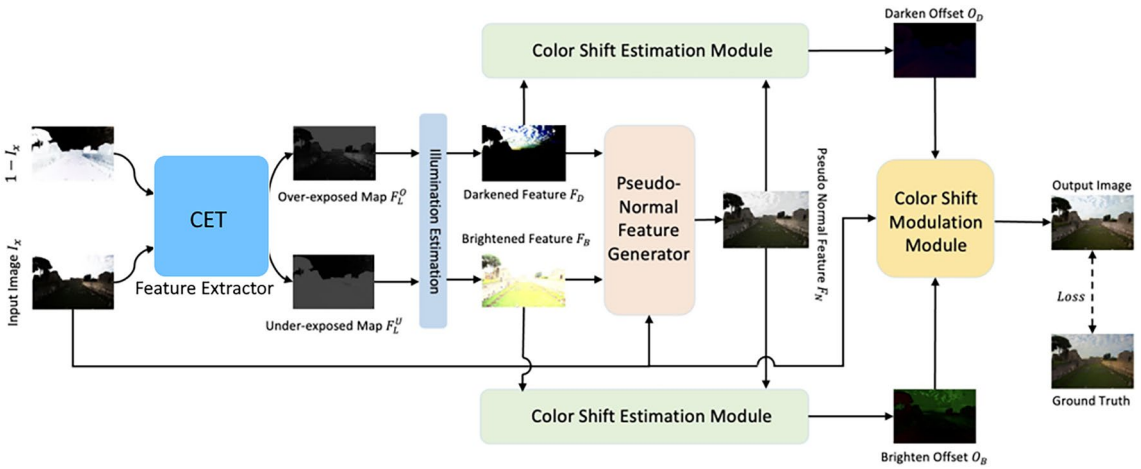

Fig. 2 Overview of our improved model based on CSEC [7]. First, use the CET feature extractor to generate darkened features F_D and brightened features F_B. Then derive a pseudo-normal feature map F_N using the generated brightened/darkened feature maps and the input image I_x. Next, further estimate the color shifts between the brightened/darkened color features F_B/F_D and the created pseudo-normal feature map FN using the proposed Color Shift Estimation (COSE) module to obtain two individual offset maps O_B (Brighten Offset) and O_D (Darken Offset). Finally, modulate the image brightness and colors using the proposed Color Modulation (COMO) module, to produce the final output image

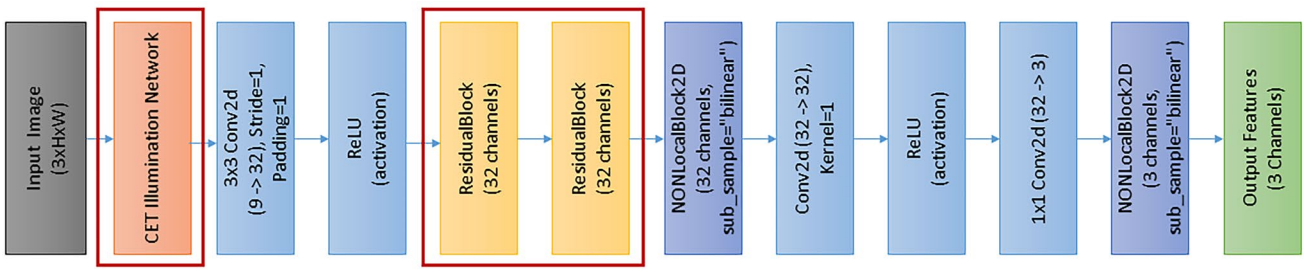

Fig. 3 Overview of feature extractor architecture with CET

Proposed training model

As aforementioned, we adopt the Color Shift Estimation and Correction (CSEC) framework (outlined in Fig. 2), proposed by [7], as the foundation for our enhancement pipeline and introduce improvements to both the network architecture and loss functions to further enhance image quality. In the CSED model, it begins by employing a CET feature extractor to generate two intermediate feature representations: a darkened feature map $\mathcal{F}_D$ and a brighted feature map $\mathcal{F}_B$. These feature maps are used along with the input image I_x to derive a pseudo-normalized feature map $\mathcal{F}_N$, which serves as a reference for estimating lighting variations. The tweaks of our design are elaborated as follows.

Feature extractor Integrating CET as the feature extractor within the CSEC framework replaces the original U-Net backbone with a hybrid architecture that combines the strengths of convolutional neural networks (CNNs) for capturing local details and Transformers for modeling global context. This synergy enables the framework to better perceive scene-level relationships, resulting in more uniform color transitions and improved consistency across regions with disparate exposure levels. By leveraging CET's efficient attention mechanism and depthwise convolutions, the model more accurately generates the "pseudo-normal" reference maps required to correct complex color shifts. Visually, this leads to sharper structural details and more realistic textures in both recovered highlights and deep shadow regions when compared to the original CNN-based approach. Overall, the CET-based hybrid architecture yields more stable outputs and effectively harmonizes the divergent color distributions present in mixed-exposure images (Fig. 3).

Color Shift Estimation Module The Color Shift Estimation (COSE) module is a core component of the CSEC framework, responsible for estimating color shifts in under- and over-exposed image regions. Unlike brightness correction, which adjusts pixel intensities, color shift correction involves modeling directional changes in RGB color space, making it a more complex task. The COSE addresses this by leveraging deformable convolution (DConv), which predicts flexible sampling offsets that can capture local

variations in color distribution. While conventional methods apply DConv only in the spatial domain, CSEC innovatively extends it to operate in both spatial and color spaces, enabling the model to jointly learn brightness adjustments and chromatic shifts. This dual-domain approach enhances the model's ability to restore natural color balance under non-uniform lighting conditions [7]. Detail of of COSE is presented in Algorithm 2.

Color Modulation Module The Color Modulation (COMO) module utilizes the learned offset maps O_B and O_D, which capture the color differences between the brightened/darkened features (F_B, F_D) and the pseudo-normal feature map F_N to adjust the brightness and color of the input image and generate the final enhanced output I_y. The algorithm of COMO is in Algorithm 3. To ensure coherent and natural color restoration, COMO incorporates non-local context modeling, allowing it to consider broader spatial dependencies. Unlike standard self-affinity mechanisms, COMO extends this to a cross-affinity formulation, enabling effective information exchange between both overexposed and underexposed regions. This design allows the network to synthesize more balanced and visually harmonious images under complex illumination conditions [7].

Loss Function The Color Shift Estimation and Correction (CSEC) framework employs two main loss functions L_{pseudo} and L_{output} to guide the network training. The L_{pseudo} provides intermediate supervision for the generation of the pseudo-normal feature map F_N, helping the network accurately model color shifts. It is defined as the L_1 distance between the predicted pseudo-normal feature map and the ground truth image:

$$L_{pseudo} = \| F_N - GT \|_1 . \tag{10}$$

To supervise the final image enhancement output, L_{output} in the original CSEC formulation combines four components: L_1 loss, cosine similarity loss L_{cos}, structural Similarity Index (SSIM) loss L_{ssim}, and perceptual loss L_{vgg}, based on VGG feature distances.

In our work, we specifically model the L_{output} with two more losses, Mean Squared Error loss (L_{MSE}) and depth-base loss L_{depth}. Mean Squared Error loss (L_{MSE}) is to explicitly penalize pixel-wise differences between the enhanced image and ground truth. This inclusion promotes greater pixel-level accuracy and encourages the network to produce outputs that more closely match the true color distribution. Meanwhile, the depth-base loss (L_{depth}) is to improve accuracy and structural consistency in the image correction process. In real world, depth information reflects the spatial arrangement of objects, which is closely related to illumination variation and color distribution in real scenes. We incorporate a depth loss that leverages geometric cues from depth maps.

To formulate it, we obtain depth maps estimated by the pre-trained MiDaS monocular depth model [22]. Given the predicted depth maps of the enhanced image D_g and the ground truth image D_r, both normalized to [0, 1], the depth loss is defined as:

$$\begin{aligned}\mathcal{L}_{depth} &= \frac{1}{HW}\sum_{x,y}|D_g(x,y) - D_r(x,y)| \\ &+ \lambda_{\text{grad}}\frac{1}{HW}\sum_{x,y}\left(|\nabla_x D_g - \nabla_x D_r| + |\nabla_y D_g - \nabla_y D_r|\right),\end{aligned} \tag{11}$$

where H and W denote the depth map resolution and λ_{grad} balances the gradient constraint. In this equation, the first term ensures global depth alignment between the enhanced image and the ground-truth image. While the second term preserves local structural boundaries and surface transitions. This loss preserves both global depth alignment and local structural boundaries, improving the stability of illumination and color correction.

The overall loss formulation for output supervision thus becomes:

$$\begin{aligned}L_{output} &= \lambda_1 L_1 + \lambda_2 L_{cos} \\ &+ \lambda_3 L_{ssim} + \lambda_4 L_{vgg} + \lambda_5 L_{MSE} + \lambda_6 L_{depth}\end{aligned} \tag{12}$$

where λ_1, λ_2, λ_3, λ_4, λ_5 and λ_6 balancing hyperparameters that control the contribution of each loss component. The overall loss function is then:

$$L = \lambda_p L_{pseudo} + \lambda_o L_{output} \tag{13}$$

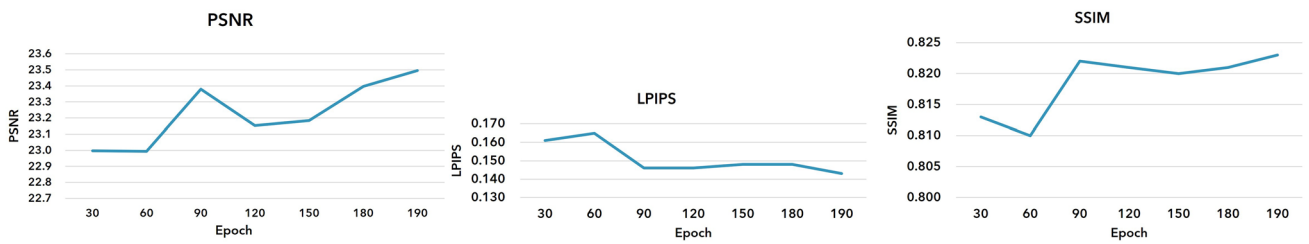

Fig. 4 Evolution of image quality metrics during training

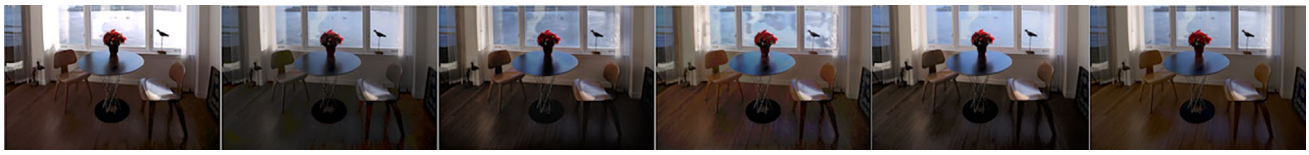

Fig. 5 Result of under-exposed images

Input: $input\_image$: Tensor representing the input image
Output: $illumination\_map$: Tensor representing the estimated illumination map of $input\_image$,
$inverse\_illumination\_map$: Tensor representing the estimated illumination map of the inverse of $input\_image$,
$brighten\_input$: Tensor representing the decomposed "brighten" component of $input\_image$,
$darken\_input$: Tensor representing the decomposed "darken" component of $input\_image$

Feature Extraction
$illumination\_map \leftarrow \text{FeatureExtractor}(input\_image)$
Calculate Inverse Image
$inverse\_input\_image \leftarrow 1 - input\_image$
Feature Extraction on Inverse Image
$inverse\_illumination\_map \leftarrow \text{FeatureExtractor}(inverse\_input\_image)$
Convert Illumination Maps to Grayscale
$illumination\_map \leftarrow \text{RGBToGrayscale}(illumination\_map)$
$inverse\_illumination\_map \leftarrow \text{RGBToGrayscale}(inverse\_illumination\_map)$
Decompose for "Brighten" Component
$brighten\_input \leftarrow \text{DecomposeBrighten}(input\_image, illumination\_map)$
Decompose for "Darken" Component
$darken\_input \leftarrow \text{DecomposeDarken}(inverse\_input\_image, inverse\_illumination\_map)$
$darken\_input \leftarrow 1 - darken\_input$
Return Outputs
return $illumination\_map$, $inverse\_illumination\_map$, $brighten\_input$, $darken\_input$ =0

Algorithm 2 ColorShiftEstimation

Input: $input\_image$: Tensor representing the original input image,
$brighten\_input$: Tensor representing the decomposed "brighten" component,
$darken\_input$: Tensor representing the decomposed "darken" component,
$weight\_map$: Tensor representing the weights for combining inputs.
Output: $output\_image$: Tensor representing the color-modulated output image

#Extract Weights from Weight Map
$weight\_1 \leftarrow \text{ExtractChannel}(weight\_map, 0)$ {Weight for input_image}
$weight\_2 \leftarrow \text{ExtractChannel}(weight\_map, 1)$ {Weight for brighten_input}
$weight\_3 \leftarrow \text{ExtractChannel}(weight\_map, 2)$ {Weight for darken_input}
Weighted Combination of Image Components
$output\_image \leftarrow (input\_image \times weight\_1) + (brighten\_input \times weight\_2) + (darken\_input \times weight\_3)$
return $output\_image$ =0

Algorithm 3 ColorModulation

Implementation and Results

Training The model is implemented and trained using Python, with PyTorch for deep learning framework. To complement this Pytorch, we employ Kornia, a library offering differentiable computer vision operations, specifically for computing perceptual loss metrics such as SSIM and PSNR. Besides, we use NumPy to support efficient numerical operations and array manipulations. For specific image processing tasks, particularly tone mapping within the base model, OpenCV (cv2) is utilized. Additionally, the torchvision library, a component of the PyTorch ecosystem, supplies essential utilities for image loading, transformation, and saving throughout the training pipeline. To ensure fair evaluation, we train the LCDP, CSEC, and our proposed models on the same dataset, each for a total of 190 epochs. The experiments were performed on a dedicated server with an 8-core CPU, a single A100 80GB GPU, and 64GB of RAM. Summary of training configuration: Optimizer Type: Adam. Learning Rate: 10^{-4} (0.0001). Batch Size: 1. Scheduler Type: Cosine Annealing with Warm Restart with T_max: 10 epochs. Training Batch Size: 1. Validation Batch Size: 1. Loss function weighting: L_{L1} : 1.0, L_{cos} : 0.5, L_{ssim} : 0.8, L_{vgg} : 0.01, L_{MSE} : 1.0, L_{depth}: 0.05.

Evaluation metrics To assess the performance of our proposed method, we employ three widely used quantitative

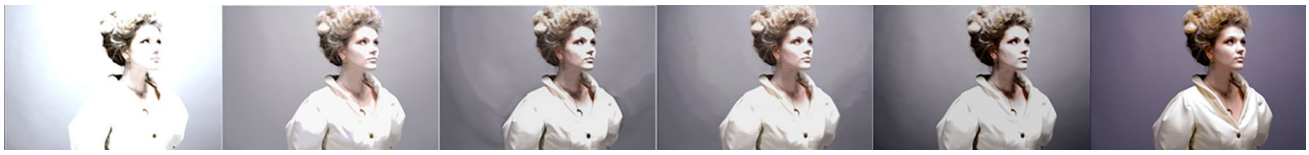

Fig. 6 Result of over-exposed images

Fig. 7 Result of image with both under-exposed and over-exposed

metrics: Peak Signal-to-Noise Ratio (PSNR) [19, 23], Structural Similarity Index Measure (SSIM) [19, 20, 23], and Learned Perceptual Image Patch Similarity (LPIPS) [24]. PSNR measures the ratio between the maximum possible signal power and the power of the noise that distorts the image. It is a conventional metric in image enhancement tasks, where higher PSNR values indicate better image quality [25, 26]. SSIM, introduced by [27], evaluates the similarity between two images by considering luminance, contrast, and structural information. Unlike traditional metrics such as MSE or PSNR, SSIM aligns more closely with human visual perception. The SSIM index ranges from -1 to 1, where 1 denotes perfect structural similarity. LPIPS is a deep learning-based perceptual metric that quantifies the visual similarity between two images based on how they are perceived by humans. It compares features extracted from deep neural networks and provides a more perceptually aligned evaluation. Lower LPIPS values indicate higher perceptual similarity between the compared images.

To supports the effectiveness of your training procedure and shows consistent improvement in reconstruction, structure, and perceptual quality, we provide the evolution of image quality metrics during training in Fig. 4. As shown in this figure, PSNR and SSIM gradually increase while LPIPS consistently decreases as training progresses, indicating continuous improvements in reconstruction accuracy, structural similarity, and perceptual quality. After approximately 90 epochs, the metrics become relatively stable with only minor fluctuations, suggesting that the model has reached a stable convergence state. This behavior demonstrates the effectiveness of the proposed training strategy in achieving reliable and stable optimization.

Figure 5 provides a compelling visual comparison that demonstrates the effectiveness of our proposed method on an under-exposed image. The input image is extreme darkness, obscuring details in the shadows and making the overall scene difficult to see. While URetinex-Net brightens the scene, it often washes out colors and struggles with over-exposed areas like the window. LCDP offers improved exposure, but results in colors that are noticeably darker and less vibrant than the reference image. The CSEC method marks a significant improvement, producing clearer details and more natural colors than LCDP, though some darker areas still lack full detail. Our method delivers the most visually compelling result. It successfully brightens the image and recovers details from the deepest shadows while preserving natural, accurate colors that closely match the ground truth image. Ours (CET) is better texture retention, sharper edges and finer details like wood grain or fabric patterns.

Figure 6 effectively highlights our method's capability in restoring images suffering from over-exposure. The input image (a) is notably washed out, especially on the subject's face, hair, and clothing, resulting in a significant loss of texture and detail. URetinex-Net (b) struggles with this, potentially worsening the over-exposure and leading to a complete loss of detail in those areas. Both LCDP (c) and CSEC (d) demonstrate progressive improvements, managing to tame the highlights and recover some of the lost detail. However, our method (e) demonstrates a superior ability to correct the over-exposed regions. It successfully recovers intricate details in the subject's hair and the texture of her clothing, which were lost in the outputs of other methods. Furthermore, the color and skin tone appear much more natural and are a closer match to the ground truth. With Ours (CET), it achieves finer detail, more consistent and seamless lighting transitions across the entire frame while retaining micro-textures better than other methods.

Table 1 Quality comparison of different methods

Methods	Effective with	Quality comparison
URetinex-Net	Under-exposed only (low light)	The method yields good results specifically for low-light images and under-exposed regions; however, it has a significant limitation with over-exposed regions, often exacerbating the over-exposure and causing a complete loss of detail in those areas
LCDP	Under and over-exposed	LCDP is more effective than URetinex-Net in that it handles both under- and over-exposed images well. When an image contains both types of regions, LCDP enhances them both. However, the resulting quality is lower than that of CSEC and our method, with the color appearing darker and less clear
CSEC	Under and over-exposed	CSEC is also effective with both under- and over-exposed images, enhancing both regions when present. Compared to LCDP, it shows a better result with clearer detail and more natural color. However, there are still some dark areas with unclear detail
Ours	Under and over-exposed	Similar to CSEC, our method shows better overall results than other methods. By incorporating improved steps based on CSEC, it provides clearer detail and more natural color, surpassing CSEC in these aspects
Ours (CET) and L_{depth}	Under and over-exposed	Integrating CET and depth-based loss function as the feature extractor within the CSEC leverages E-Attention and depthwise convolutions to accurately generate 'pseudo-normal' reference maps. Consequently, the model produces sharper structural details and more realistic textures in over-exposed and under-exposed areas compared to other methods. The obtained results of SSIM, PSNR, and LPIPS have confirmed that our proposed method is effective

Figure 7 illustrates the method's performance on image with both under-exposed and over-exposed. The input image (a) contains both under-exposed shadows on the building and over-exposed highlights from the sun. While methods like URetinex-Net, LCDP, and CSEC struggle to produce a balanced result, the proposed method (e) excels at it. It effectively brightens the shadowed areas of the building to reveal details while simultaneously controlling the highlights in the sky and on the sunlit walls. The result is a natural-looking image with excellent color fidelity across the entire dynamic range. Ours (CET) produces a more visually sophisticated result, achieving superior structural sharpness and harmonized global exposure while effectively reducing noise and preserving intricate textures.

Table 2 Quantitative comparison of our method and different methods

Methods	Dataset	PSNR	SSIM	LPIPS
1. CSEC	LCDP	21.234	0.821	0.144
2. CSEC with Median filter	LCDP	21.248	0.768	0.200
3. CSEC	LCDP + Our	20.999	0.805	0.173
4. CSEC with Median filter	LCDP + Our	21.888	0.781	0.199
5. Our method (ResidualBlocks, L_{MSE}, CSEC)	LCDP + Our	22.115	**0.818**	**0.153**
6. Our method (ResidualBlocks, L_{MSE}, CSEC, **Median filter**)	LCDP + Our	**22.632**	0.767	0.211
7. Our method (ResidualBlocks, L_{MSE}, CSEC, **CET**)	LCDP + Our	**23.151**	**0.822**	**0.146**
8. Our method (ResidualBlocks, L_{MSE}, CSEC, **CET**, L_{depth})	LCDP + Our	**23.495**	**0.823**	**0.143**

The bold values indicate better results than other existing methods.

Discussion, Evaluation and Comparison

Figures 5, 6, and 7 exhibits visual comparison between our improved method with CET and state-of-the-art methods. As shown in the summary of Table 1, our improved method consistently produces outputs that are perceptually closer to the ground truth compared to other methods.

Visually, the enhanced images exhibit better overall quality and more natural color correction.

Our approach includes an optional preprocessing step involving a median filter, which can be enabled or disabled depending on the input image characteristics. The effect of the median filter varies with image quality. For noisy input images, the filter effectively suppresses noise and improves visual clarity. However, when applied to high-quality images, the median filter may inadvertently blur fine details, leading to a loss of structural information.

The quantitative results in Table 2 further support this observation. While applying the median filter tends to improve the PSNR metric due to reduced pixel-level noise, it can also lead to a decrease in SSIM and an increase in LPIPS. This is because the median filter alters local pixel values, affecting structural similarity and perceptual

features. SSIM, which assesses luminance, contrast, and structure, penalizes such structural modifications. Similarly, LPIPS increases when perceptual features deviate from the original. In particular, in cases of severe noise, filtering may reduce LPIPS by restoring perceptual similarity, despite structural alterations. As we can see in rows 7 and 8, where we applied CET, with and without L_{depth}. The obtained results of SSIM and PSNR are increasing, while LPIPS is decreasing.

Ablation Study To systematically evaluate the impact of each proposed component, we conducted a comprehensive ablation study. We found that the incorporation of the new dataset, the median filter preprocessing, the inclusion of Residual Blocks and the MSE loss function each contributed significantly to the model's overall performance. For instance, models trained without the additional Residual Blocks showed a notable decrease in the clarity of fine details, while removing the MSE loss resulted in less accurate pixel-level color reproduction, especially in heavily overexposed areas. This study validates that the combination of these elements is essential for achieving our method's superior results.

Residual Blocks help create clearer images by improving the model's ability to learn and propagate information effectively. In deep networks, as layers are added, the vanishing gradient problem can occur, where gradients become too small to effectively update the network weights, hindering learning. Residual Blocks address this by introducing "skip connections" that allow the input from a previous layer to be directly added to the output of a later layer [28]. This facilitates identity mapping and ensures better gradient flow, leading to more stable and efficient convergence during training. By strengthening the learning of contextual features, these blocks help the network capture more complex color shifts and illumination variations, which is essential for producing more visually consistent and color-accurate enhanced output. Besides, the MSE loss function contributes to clearer images by enforcing pixel-level accuracy. MSE directly calculates the average of the squared differences between the pixels of the enhanced image and the ground truth image. This explicit penalty for pixel-wise differences encourages the network to produce outputs that are more precise and closely match the true color distribution, leading to sharper details and overall better image quality.

Conclusion and Future work

Our improved method maintains a favorable balance between high performance and computational efficiency. Although many state-of-the-art models in image quality enhancement have large parameter counts, our proposed architecture is designed to be lightweight and scalable. This is achieved by building on a convolution-based efficient transformer (CET) with efficient Residual Blocks, which allows for effective feature extraction without the high computational cost often associated with models like Vision Transformers. As a result, our network is well suited for deployment in real-world applications where rapid inference on diverse and large-scale datasets is a critical requirement. The primary contribution of this research is not simply applying a known deep learning technique to a new dataset. Instead, we present a novel holistic solution that addresses the complex challenge of simultaneously correcting underexposed and overexposed regions in a single image. This is accomplished through the strategic combination of a newly created dataset to enhance model generalization, an improved network architecture that leverages Residual Blocks, and a refined loss function (both L_{MSE} and L_{depth}). This integrated approach, which tackles a problem that is largely overlooked by traditional methods, represents a significant step forward in the field of image quality enhancement. In the future, we plan to refine the architecture and training strategy to further improve the results on metrics like PSNR and LPIPS.

Acknowledgements This research is also supported by The Central Interdisciplinary Laboratory in Electronics and Information Technology (AI and Cooperation Robot), International University—VNU-HCM. We would like to thank for supporting the machines in experiments.

Author Contributions NVS initialized concepts and directions. NXV and TNHL conceived experiments, conducted experiments and analyzed results. NVS and TNHL provided critical updates and suggestions that significantly enhanced the scope and direction of the research. NVS wrote the paper with important contributions. All authors (NVS, NXV and TNHL) reviewed and approved the final manuscript.

Funding Not applicable.

Data Availability The datasets used in this study are publicly available in the LCDPNet repository at https://github.com/onpix/LCDPNet/tree/main.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Research involving human and/or animals Not applicable.

Informed consent Not applicable.

References

1. Putra RD, Purboyo TW, Prasasti AL. A review of image enhancement methods. Int J Appl Eng Res. 2017;12(23):13596–603.

2. Sigger N, Vien Q-T, Nguyen SV, Tozzi G, Nguyen TT. Unveiling the potential of diffusion model-based framework with transformer for hyperspectral image classification. Sci Rep. 2024;14(8438):1–11.
3. Wang W, Wu X, Yuan X, Gao Z. An experiment-based review of low-light image enhancement methods. Ieee Access. 2020;8:87884–917.
4. Dyke RM, Hormann K. Histogram equalization using a selective filter. Vis Comput. 2023;39(12):6221–35.
5. Mustafa WA, Abdul Kader MMM. A review of histogram equalization techniques in image enhancement application. In: Journal of Physics: Conference Series, vol. 1019. IOP Publishing; 2018. p. 012026.
6. Chien CC, Kinoshita Y, Shiota S, Kiya H. A retinex-based image enhancement scheme with noise aware shadow-up function, in International Workshop on Advanced Image Technology (IWAIT) 2019, vol. 11049. SPIE; 2019. pp. 501–506.
7. Li Y, Xu K, Hancke GP, Lau RW. Color shift estimation-and-correction for image enhancement. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2024. pp. 25389–25398.
8. Chen J, Wang X, Guo Z, Zhang X, Sun J. Dynamic region-aware convolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021. pp. 8064–8073.
9. Sinh Van Nguyen HMT, Le TS. Application of geometric modeling in visualizing the medical image dataset. Spring Nat Comput Sci. 2020;1(254):01–11.
10. Nguyen SV, Nguyen VX, Ngoc HLT. A method for enhancing images quality based on machine learning. In: Intelligent Systems and Data Science. ISDS 2025. Communications in Computer and Information Science, vol. 2714. Springer; 2026. p. 398–416.
11. Ningsih DR, et al. Improving retinal image quality using the contrast stretching, histogram equalization, and clahe methods with median filters. Int J Image Gr Signal Process. 2020;14(2):30.
12. Guo J, Ma J, García-Fernández ÁF, Zhang Y, Liang H. A survey on image enhancement for low-light images. Heliyon. 2023;9(4):e14558.
13. Wu W, Weng J, Zhang P, Wang X, Yang W, Jiang J. Uretinex-net: Retinex-based deep unfolding network for low-light image enhancement. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022. pp. 5901–5910.
14. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18. Springer; 2015. pp. 234–241.
15. Le T-N-H, Huang H, Chen Y-R, Lee T-Y. Retargeting video with an end-to-end framework. IEEE Trans Vis Comput Gr. 2024;30(9):6164–76.
16. Le T-N-H, Lee T-Y, Lin S-S, Dong W. Deep learning-based importance map for content-aware media retargeting. Multimedia Tools Appl. 2024;83(30):74301–22.
17. Wang H, Xu K, Lau RW. Local color distributions prior for image enhancement. In: European conference on computer vision. Springer; 2022. pp. 343–359.
18. Yin L et. al, Convolution-transformer for image feature extraction. In: CMES-Computer modeling in engineering & sciences, vol. 141, Tech Science Press; 2024. pp. 87–106.
19. Nguyen LDV, Van Nguyen S, Le ST, Tran MK, Maleszka M. Processing the 3d heritage data samples based on combination of gnn and gan. In: International conference on computational collective intelligence. Springer; 2024. pp. 295–307.
20. Nguyen LDV, Van Chau V, Van Nguyen S. Face recognition based on deep learning and data augmentation. In: International conference on future data and security engineering. Springer; 2022. pp. 560–573.
21. Li G, Chen B, Zhao C, Zhang L, Zhang J. Osmamba: omnidirectional spectral mamba with dual-domain prior generator for exposure correction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR). 2025. pp. 7480–7490.
22. Birkl R, Wofk D, Müller M. Midas v3. 1—a model zoo for robust monocular relative depth estimation. 2023. arXiv preprint arXiv:2307.14460.
23. Le ST, Nguyen SV, Tran MK, Nguyen LDV. Graphics and vision's camera calibration and applications to neural radiance fields. In: Asian conference on intelligent information and database systems. Springer; 2024. pp. 118–129.
24. Zhang R, Isola P, Efros AA, Shechtman E, Wang O. The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018. pp. 586–595.
25. Sara U, Akter M, Uddin MS. Image quality assessment through fsim, ssim, mse and psnr—a comparative study. J Comput Commun. 2019;7(3):8–18.
26. Hore A, Ziou D. Image quality metrics: Psnr vs. ssim. In: 2010 20th international conference on pattern recognition. IEEE; 2010. pp. 2366–2369.
27. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process. 2004;13(4):600–12.
28. Jin X. Analysis of residual block in the resnet for image classification. In: Proceedings of the 1st international conference on data analysis and machine learning, Changsha, China. 2023. pp. 28–30.