Ultra-High-Definition Low-Light Image Enhancement: A Benchmark and Transformer-Based Method

Tao Wang[1], **Kaihao Zhang**[2], **Tianrun Shen**[1], **Wenhan Luo**[3*], **Bjorn Stenger**[4], **Tong Lu**[1*]

[1]State Key Lab for Novel Software Technology, Nanjing University, China
[2] Australian National University, Australia
[3] Shenzhen Campus of Sun Yat-sen University, China
[4] Rakuten Institute of Technology, Japan
{taowangzj, super.khzhang, whluo.china}@gmail.com, tiruns@yeah.net, bjorn@cantab.net, lutong@nju.edu.cn

Abstract

As the quality of optical sensors improves, there is a need for processing large-scale images. In particular, the ability of devices to capture ultra-high definition (UHD) images and video places new demands on the image processing pipeline. In this paper, we consider the task of low-light image enhancement (LLIE) and introduce a large-scale database consisting of images at 4K and 8K resolution. We conduct systematic benchmarking studies and provide a comparison of current LLIE algorithms. As a second contribution, we introduce LLFormer, a transformer-based low-light enhancement method. The core components of LLFormer are the axis-based multi-head self-attention and cross-layer attention fusion block, which significantly reduces the linear complexity. Extensive experiments on the new dataset and existing public datasets show that LLFormer outperforms state-of-the-art methods. We also show that employing existing LLIE methods trained on our benchmark as a pre-processing step significantly improves the performance of downstream tasks, *e.g.,* face detection in low-light conditions. The source code and pre-trained models are available at https://github.com/TaoWangzj/LLFormer.

Introduction

Images taken in low-light conditions typically show noticeable degradation, such as poor visibility, low contrast, and high noise levels. To alleviate these effects, a number of low-light image enhancement (LLIE) methods have been proposed to transform a given low-light image into a high-quality image with appropriate brightness. Traditional LLIE methods are mainly based on image priors or physical models from other tasks, such as histogram equalization-based methods (Kim 1997; Stark 2000), retinex-based methods (Kimmel et al. 2003; Wang et al. 2014) and dehazing-based methods (Dong et al. 2011; Zhang et al. 2012). Recently, many learning-based LLIE methods have been introduced, making use of large-scale synthetic datasets and achieving significant improvements in terms of performance and speed (Wei et al. 2018; Guo et al. 2020; Lim and Kim 2020; Jiang et al. 2021; Li, Guo, and Loy 2021; Liu et al. 2021b).

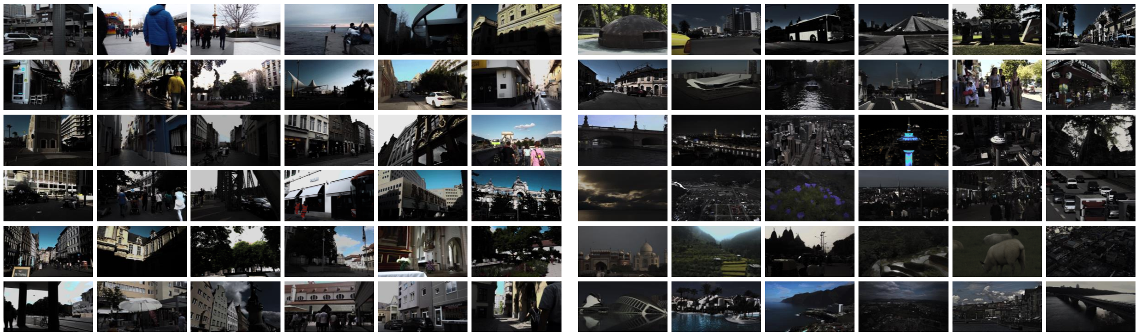
Most existing datasets, *e.g.,* LOL (Wei et al. 2018) and SID (Chen et al. 2018), consist of lower resolution images (1K or less). Thus, LLIE methods trained on these datasets are naturally constrained to low-resolution images. Sensors on modern mobile devices are able to capture images of resolutions of 4K or 8K, creating a need for algorithms designed for processing Ultra-High-Definition (UHD) images. It is difficult for existing LLIE methods to simultaneously reconcile inference efficiency and visual enhancement on UHD images. In this paper, we focus on the task of Ultra-High Definition Low-Light Image Enhancement (UHD-LLIE). We first build a large-scale benchmark dataset containing UHD images in LOw-Light conditions (UHD-LOL) to explore and evaluate image enhancement algorithms. UHD-LOL includes two subsets, UHD-LOL4K and UHD-LOL8K, containing 4K and 8K-resolution images, respectively. The UHD-LOL4K subset contains $8,099$ image pairs, $5,999$ for training and $2,100$ for testing. The subset of UHD-LOL8K includes $2,966$ image pairs, $2,029$ for training and 937 for testing. Example 4K and 8K low-light images are shown in Fig. 1.

Using this dataset, we conduct extensive benchmarking studies to compare existing LLIE methods and highlight some shortcomings in the UHD setting. We propose a novel Transformer-based method named Low-Light Transformer-based Network (LLFormer) for the UHD-LLIE task. LLFormer is composed of two basic units, an efficient Axis-based Transformer Block and a Cross-layer Attention Fusion Block. Within the Axis-based Transformer block, the axis-based self-attention unit performs the self-attention mechanism on the height and width axes of features across the channel dimension to capture non-local self-similarity and long-range dependencies with less computational complexity. Moreover, after the axis-based self-attention, we design a novel Dual Gated Feed-Forward Network, which employs a dual gated mechanism to focus on useful features. The Cross-layer Attention Fusion Block learns attention weights across features in different layers and adaptively fuses features with the learned weights to improve feature representation. The LLFormer adopts a hierarchical structure, which greatly alleviates the computational bottleneck for the UHD-LLIE task.

To summarize, the contributions of our work are as follows:

1. We build a benchmark dataset of 4K and 8K UHD images, UHD-LOL, to explore and evaluate image enhancement algorithms. To the best of our knowledge, this is the

[*]Corresponding authors.



(a) 4K images from UHD-LOL4K subset.

(b) 8K images from UHD-LOL8K subset.

Figure 1: Low-light images sampled from the proposed UHD-LOL dataset.

first large-scale UHD low-light image enhancement dataset in the literature.

2. Based on UHD-LOL, we benchmark existing LLIE algorithms to show the performance and limitations of these methods, offering new insights.

3. We propose a novel transformer model, LLFormer, for the UHD-LLIE task. In both quantitative and qualitative aspects, LLFormer achieves state-of-the-art performance on the public LOL and MIT-Adobe FiveK datasets, and our UHD-LOL benchmark.

Related Work

Low-light Image Datasets. With the advance in data-driven methods (Zhang et al. 2022, 2021b), many datasets for low-light image enhancement have been proposed. In (Vonikakis et al. 2013), Vonikakis *et al.* introduced an LLIE dataset of 225 images collected from 15 different scenes. In each scene, they captured 15 images, including 9 images under uniform-illumination conditions with different intensities and 6 images under non-uniform illumination conditions. By applying random gamma correction and Gaussian noise, Lore *et al.* (Lore, Akintayo, and Sarkar 2017) synthesized $422,500$ low-light image patches from 169 images. (Shen et al. 2017) created an LLIE dataset of $10,000$ image pairs, where $8,000$ for training and $2,000$ for testing. (Chen et al. 2018) built the See-in-the-Dark (SID) dataset. It contains $5,094$ short-exposure low-light raw images and their corresponding long-exposure ones. (Cai, Gu, and Zhang 2018) synthesized the SICE dataset from 589 image sequences with multi-exposure image fusion (MEF) or a high dynamic range (HDR) algorithm. (Wei et al. 2018) created the LOw-Light (LOL) dataset, which consists of 485 image pairs for training and 15 for testing. Based on the LOL dataset, Liu *et al.* (Liu et al. 2021a) created VE-LOL-L for training and evaluating LLIE methods, which includes $2,100$ images for training and 400 for evaluation. MIT-Adobe FiveK (Bychkovsky et al. 2011) consists of $5,000$ images captured of various indoor and outdoor scenes under different lighting conditions. In this paper, we introduce a UHD low-light image enhancement benchmark of 4K and 8K images, of sufficient size to be used for training deep models and comparing existing methods.

Low-light Image Enhancement Methods. LLIE aims to recover images from underexposed ones taken in low-light conditions (Li et al. 2022). Over the past decades, many methods have been proposed for the LLIE task, which can be roughly divided into two categories: non-learning based methods and data-driven methods. Traditional methods mainly include histogram-based methods (Lee, Lee, and Kim 2012; Xu et al. 2013; Celik 2014), Retinex-based methods (Kimmel et al. 2003; Wang et al. 2014), and dehazing-based methods (Zhang et al. 2012; Li et al. 2015). Although traditional methods obtain reasonable results, the enhanced images of these methods usually suffer from artifacts like color distortion or over enhancement. Data-driven methods (Wei et al. 2018; Zhang et al. 2020; Lim and Kim 2020; Ni et al. 2020; Zeng et al. 2020; Wang et al. 2022a, 2021; Yang et al. 2022) have been successfully applied to the LLIE task. For example, RetinexNet in (Wei et al. 2018) combines Retinex theory and deep CNNs in a unified end-to-end learning framework. DSLR (Lim and Kim 2020) applies a Laplacian pyramid scheme to enhance global and local details in multiple streams of the encoder-decoder architecture. Recently, data-driven methods based on transformers have been applied to low-level tasks: Uformer (Wang et al. 2022b) uses a modified Swin transformer block (Liu et al. 2021c) to build a U-shaped network, showing good performance in image restoration. Restormer (Zamir et al. 2022) introduces modifications of the transformer block for improved feature aggregation for image restoration. While transformers work well in many tasks, their potential for low-light image enhancement remains unexplored. In this work, we focus on designing a transformer for UHD low-light image enhancement.

Benchmark and Methodology

Benchmark Dataset

We create a new large-scale UHD-LLIE dataset called UHD-LOL to benchmark the performance of existing LLIE methods and explore the UHD-LLIE problem. UHD-LOL is composed of 4K images of $3,840 \times 2,160$ resolution and 8K images of $7,680 \times 4,320$ resolution, respectively. To build this dataset of image pairs, we use normal-light 4K and 8K images from public data (Zhang et al. 2021a). These UHD images were crawled from the web and captured by various devices. Images contain both indoor and outdoor scenes, including buildings, streets, people, animals, and natural landscapes. We synthesize corresponding low-light images following (Wei et al. 2018), which takes both the low-light degradation process and natural image statistics into consideration.

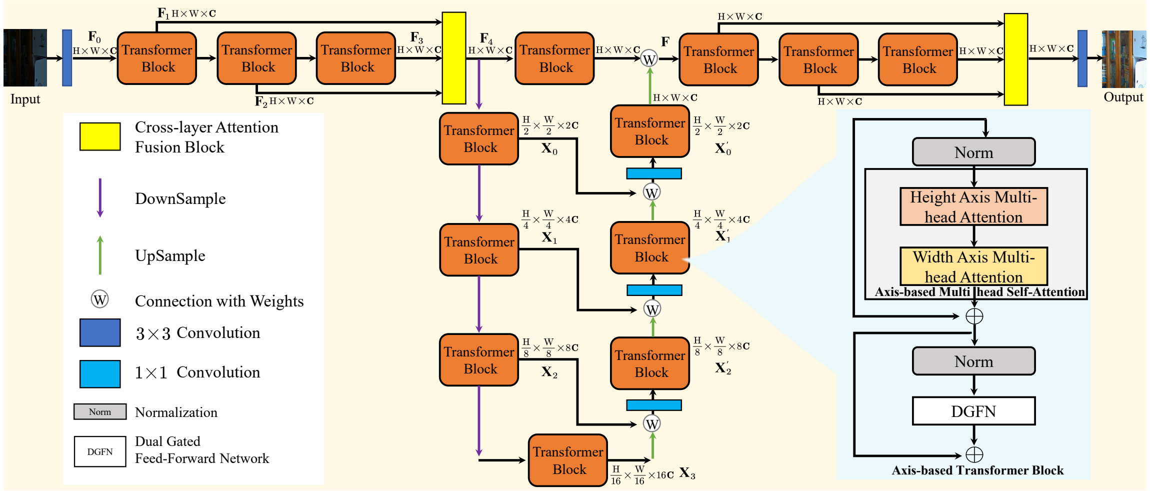


Figure 2: **LLFormer architecture**. The core design of LLFormer includes an axis-based transformer block and a cross-layer attention fusion block. In the former, axis-based multi-head self-attention performs self-attention on the height and width axis across the channel dimension sequentially to reduce the computational complexity, and a dual gated feed-forward network employs a gated mechanism to focus more on useful features. The cross-layer attention fusion block learns the attention weights of features in different layers when fusing them.

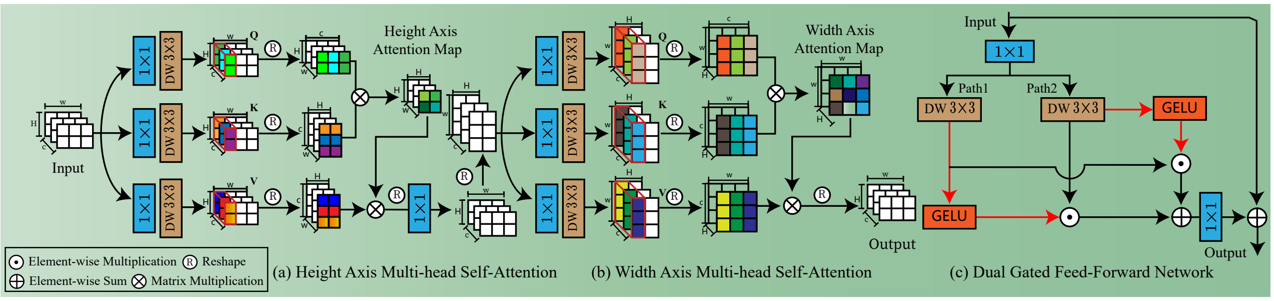
Specifically, we first generate three random variables X, Y, Z, uniformly distributed in $(0, 1)$. We use these variables to generate parameters provided by the Adobe Lightroom software. The parameters include exposure $(-5+5X^2)$, highlights $(50 \min\{Y, 0.5\} + 75)$, shadows $(-100 \min\{Z, 0.5\})$, vibrance $(-75 + 75X^2)$, and whites $(16(5 - 5X^2))$. The synthesized low-light and normal-light images make up our UHD-LOL, which consists of two subsets: UHD-LOL4K and UHD-LOL8K. The UHD-LOL4K subset contains $8,099$ pairs of 4K low-light/normal-light images. Among them, $5,999$ pairs of images are used for training and $2,100$ for testing. The UHD-LOL8K subset includes $2,966$ pairs of 8K low-light/normal-light images, which are split into $2,029$ pairs for training and 937 for testing. Example images are shown in Fig. 1.

LLFormer Architecture

As illustrated in Fig. 2, the overall architecture of LLFormer is a hierarchical encoder-decoder structure. Given a low-light image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, LLFormer first employs a 3×3 convolution as a projection layer to extract shallow feature $\mathbf{F_0} \in \mathbb{R}^{H\times W\times C}$. Next, $\mathbf{F_0}$ is fed into three sequential Transformer blocks to extract deeper features. More specifically, intermediate features outputted from transformer blocks are denoted as $\mathbf{F_1}, \mathbf{F_2}, \mathbf{F_3} \in \mathbb{R}^{H\times W\times C}$. These features $\mathbf{F_1}, \mathbf{F_2}, \mathbf{F_3}$ pass through the proposed cross-layer attention fusion block to be aggregated and transformed into the enhanced image features $\mathbf{F_4}$. Second, four stages in an encoder are used for deep feature extraction on $\mathbf{F_4}$. To be specific, each stage contains one downsampling layer and multiple transformer blocks. From top to bottom stages, the number of transformer blocks increases. We use the pixel-unshuffle operation (Shi et al. 2016) to downscale the spatial size and double the channel number. Therefore, features in the i-th stage of the encoder can be denoted as $\mathbf{X}_i \in \mathbb{R}^{\frac{H}{2^i}\times\frac{W}{2^i}\times 2^i C}$ and $i = 0, 1, 2, 3$ corresponding to the four stages. Subsequently, the low-resolution latent feature $\mathbf{X}_3$ passes through a decoder which contains three stages and takes $\mathbf{X}_3$ as input and progressively restores the high-resolution representations. Each stage is composed of an upsampling layer and multiple transformer blocks. Features in the i-th stage of decoder are denoted as $\mathbf{X}'_i \in \mathbb{R}^{\frac{H}{2^i}\times\frac{H}{2^i}\times 2^{i+1}C}$, $i = 0, 1, 2$. We apply the pixel-shuffle operation (Shi et al. 2016) for upsampling. To alleviate the information loss in the encoder and for features to be well recovered in the decoder, we use the weighted skip connection with a 1×1 convolution for feature fusion between the encoder and decoder, which can flexibly adjust the contributions of the features from encoder and decoder. Third, after the decoder, the deep feature $\mathbf{F}$ in turn passes through three transformer blocks and a cross-layer attention fusion block to generate the enhanced features for image reconstruction. Finally, LLFormer applies a 3×3 convolution on the enhanced features to yield the enhanced images $\hat{I}$. We optimize LLFormer using a smooth L_1 loss (Girshick 2015).

Axis-based Transformer Block

Transformers were shown to have advantages in modeling non-local self-similarity and long-range dependencies compared to CNNs. However, as discussed in (Vaswani et al. 2017; Liu et al. 2021c), the computational cost of the standard Transformer is quadratic with respect to the spatial size of input feature maps $(H \times W)$. Moreover, it often becomes infeasible to apply transformers to high-resolution images especially UHD images. To address this problem, we propose an axis-based multi-head self-attention (A-MSA) mechanism in the transformer block. The computational complexity of A-MSA is linear in spatial size, which greatly reduces the computational complexity. Further, we introduce a dual gated mechanism in the plain transformer feed-forward network and propose the dual gated feed-forward network (DGFN) to capture more important information in features. We integrate our A-MSA and DGFN with the plain transformer units to


Input
1×1
DW 3×3
Q
K
V
R
Height Axis Attention Map
Width Axis Attention Map
Output
Path1
Path2
GELU
⊙ Element-wise Multiplication Ⓡ Reshape
⊕ Element-wise Sum ⊗ Matrix Multiplication

(a) Height Axis Multi-head Self-Attention (b) Width Axis Multi-head Self-Attention (c) Dual Gated Feed-Forward Network

Figure 3: The architecture of our Axis-based Multi-head Self-Attention and Dual Gated Feed-Forward Network. From left to right, the components are Height Axis Multi-head Attention, Width Axis Multi-head Attention, and Dual Gated Feed-Forward Network.

build the Axis-based Transformer Block (ATB). As shown in Fig. 2, an ATB contains an A-MSA, a DGFN, and two normalization layers. The formula of ATB is:

$$\begin{aligned}\mathbf{F}' &= \text{A-MSA}\left(\text{LN}\left(\mathbf{F_{in}}\right)\right) + \mathbf{F_{in}},\\ \mathbf{F}_{\text{out}} &= \text{DGFN}\left(\text{LN}\left(\mathbf{F}'\right)\right) + \mathbf{F}',\end{aligned} \tag{1}$$

where $\mathbf{F_{in}}$ denotes the input of ATB. $\mathbf{F}'$ and $\mathbf{F}_{\text{out}}$ are the outputs of A-MSA and DGFN, respectively. LN is the layer normalization (Ba, Kiros, and Hinton 2016). In the following, we provide details of A-MSA and DGFN.

Axis-based Multi-head Self-Attention. The computational complexity of the standard self-attention is quadratic with the resolution of input, *i.e.*, $\mathcal{O}\left(W^2H^2\right)$ for $H \times W$ feature maps. Instead of computing self-attention globally, we propose A-MSA, as illustrated in Fig. 2, to compute self-attention on the height and width axes across the channel dimension sequentially. Thanks to this operation, the complexity of our A-MSA is reduced to linear. Moreover, to alleviate the limitation of transformers in capturing local dependencies, we employ depth-wise convolutions to help A-MSA focus on the local context before computing a feature attention map (Zamir et al. 2022; Wang et al. 2022b). Since the mechanisms of height and width axis multi-head self-attention are similar, we thus only introduce the details of height axis multi-head self-attention for ease of illustration.

For height axis multi-head attention, as shown in Fig. 3 (a), given feature $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ output from the normalization layer, we at first apply 1×1 convolutions to enhance the input feature $\mathbf{X}$, and use 3×3 depth-wise convolutions to obtain features with enriched local information. Then, the output features from 3×3 depth-wise convolutions are query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$, as $\mathbf{Q} = W^Q_{3\times3}W^Q_{1\times1}\mathbf{X}$,$\mathbf{K} = W^K_{3\times3}W^K_{1\times1}\mathbf{X}$ and $\mathbf{V} = W^V_{3\times3}W^V_{1\times1}\mathbf{X}$, where $W_{1\times1}$ and $W_{3\times3}$ denote 1×1 convolution and 3×3 depth-wise convolution, respectively. After that, the query and key are reshaped for conducting dot-product to generate height axis attention map $\mathbf{A} \in \mathbb{R}^{H\times H\times W}$. To achieve multi-head self-attention, we split the reshaped $\hat{\mathbf{Q}}$, $\hat{\mathbf{K}}$ and $\hat{\mathbf{V}}$ into k heads along the feature channel dimension respectively, as $\hat{\mathbf{Q}} = [\hat{q}_1, \ldots, \hat{q}_k]$, $\hat{\mathbf{K}} = \left[\hat{k}_1, \ldots, \hat{k}_k\right]$, $\hat{\mathbf{V}} = [\hat{v}_1, \ldots, \hat{v}_k]$, where the dimension of each head is $d_k = C/k$. The height axis multi-head self-attention for the j-th head can be formulated as:

$$\text{SA}\left(\hat{q}_j, \hat{k}_j, \hat{v}_j\right) = \hat{v}_j\text{softmax}\left(\hat{q}_j\hat{k}_j/\boldsymbol{\alpha}\right), \tag{2}$$

where $\hat{q}_j \in \mathbb{R}^{H\times d_k\times W}$, $\hat{k}_j \in \mathbb{R}^{d_k\times H\times W}$ and $\hat{v}_j \in \mathbb{R}^{d_k\times H\times W}$ denote the j-th head of $\hat{\mathbf{Q}}$, $\hat{\mathbf{K}}$ and $\hat{\mathbf{V}}$, respectively. $\boldsymbol{\alpha}$ is a scale factor. The output feature $\mathbf{X}'$ can be obtained by:

$$\mathbf{X}' = W_{1\times1}\mathbf{Concat}^k_{j=0}\left(\text{SA}\left(\hat{q}_j, \hat{k}_j, \hat{v}_j\right)\right), \tag{3}$$

where $\mathbf{Concat}$ represents the concatenation operation. Finally, we reshape $\mathbf{X}'$ to obtain the output feature $\mathbf{X}_{out} \in \mathbb{R}^{H\times W\times C}$ of height axis multi-head attention. $\mathbf{X}_{out}$ is forwarded to the width axis multi-head attention (see Fig. 3 (b)) to compute self-attention along the width axis.

Dual Gated Feed-Forward Network. Previous work suggests that Feed-Forward Networks (FFN) demonstrate a limitation in capturing local context (Vaswani et al. 2017; Dosovitskiy et al. 2021). For efficient feature transformations, we introduce a dual gated mechanism and local information enhancement in FFN, and propose a novel dual gated feed-forward network (DGFN). As shown in Fig. 3 (c), for the dual gated mechanism, we first apply dual GELU and element-wise product in two parallel paths to filter the less informative features and then fuse useful information from two paths with an element-wise sum. Further, we apply a 1×1 convolution ($W_{1\times1}$) and a 3×3 depth-wise convolution ($W_{3\times3}$) in each path to enrich the local information. Given $\mathbf{Y} \in \mathbb{R}^{H\times W\times C}$ as input, the complete DGFN is formulated as:

$$\begin{aligned}\text{DG} &= \phi\left(W^1_{3\times3}W^1_{1\times1}\mathbf{Y}\right) \odot \left(W^2_{3\times3}W^2_{1\times1}\mathbf{Y}\right)\\ &\quad + \left(W^1_{3\times3}W^1_{1\times1}\mathbf{Y}\right) \odot \phi\left(W^2_{3\times3}W^2_{1\times1}\mathbf{Y}\right),\\ \hat{\mathbf{Y}} &= W_{1\times1}\text{DG}(\mathbf{Y}) + \mathbf{Y},\end{aligned} \tag{4}$$

where $\hat{\mathbf{Y}} \in \mathbb{R}^{H\times W\times C}$ represents the output features, DG denotes the dual gated mechanism, $\odot$ is the element-wise multiplication operation, and ϕ is the GELU activation function.

Cross-layer Attention Fusion Block

Recent transformer-based methods adopt feature connections or skip connections to combine features from different layers (Zamir et al. 2022; Wang et al. 2022b). However, these operations do not fully exploit dependencies across different layers, limiting the representation capability. To address this, we propose a novel cross-layer attention fusion block (CAFB), which adaptively fuses hierarchical features with learnable correlations among different layers. The intuition

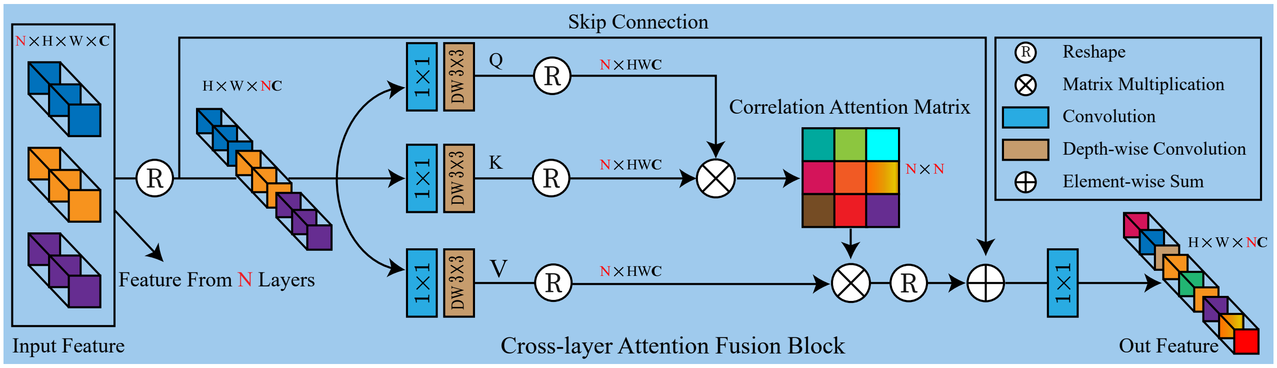


Figure 4: The architecture of the proposed Cross-layer Attention Fusion Block. This block efficiently integrates features from different layers with a layer correlation attention matrix.

behind CAFB is that activations at different layers are a response to a specific class, and feature correlations can be adaptively learned using a self-attention mechanism.

The CAFB architecture is shown in Fig. 4. Given concatenation features ($\mathbf{F_{in}} \in \mathbb{R}^{N\times H\times W\times C}$) from N successive layers ($N = 3$ in the experiments), we first reshape $\mathbf{F_{in}}$ into $\hat{\mathbf{F}}_{\mathbf{in}}$ with dimensions $H \times W \times NC$. Like self-attention in ATB, we employ 1×1 convolutions to aggregate pixel-wise cross-channel context followed by 3×3 depth-wise convolutions to yield $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$. We then reshape the query and key into 2D matrices of dimensions $N \times HWC$ ($\hat{\mathbf{Q}}$) and $HWC \times N$ ($\hat{\mathbf{K}}$) to calculate the layer correlation attention matrix $\mathbf{A}$ of size $N \times N$. Finally, we multiply the reshaped value $\hat{\mathbf{V}} \in \mathbb{R}^{HWC\times N}$ by the attention matrix $\mathbf{A}$ with a scale factor $\boldsymbol{\alpha}$, and add the input features $\mathbf{F_{in}}$. The CAFB process is formulated as:

$$\begin{aligned}\hat{\mathbf{F}}_{\mathbf{out}} &= W_{1\times 1}\text{Layer\_Attention}\left(\hat{\mathbf{Q}}, \hat{\mathbf{K}}, \hat{\mathbf{V}}\right) + \hat{\mathbf{F}}_{\mathbf{in}}, \\ \text{Layer\_Attention}\left(\hat{\mathbf{Q}}, \hat{\mathbf{K}}, \hat{\mathbf{V}}\right) &= \hat{\mathbf{V}}\text{softmax}(\hat{\mathbf{Q}}\hat{\mathbf{K}}/\boldsymbol{\alpha}),\end{aligned} \tag{5}$$

where $\hat{\mathbf{F}}_{\mathbf{out}}$ is the output feature that focuses on informative layers of the network. In practice, we place the proposed CAFB in the symmetric position of the head and tail in the network, so that CAFB helps capture long-distance dependencies among hierarchical layers in both feature extraction and image reconstruction processes.

Experiments and Analysis

Implementation Details

The LLFormer is trained on 128×128 patches with a batch size of 12. For data augmentation, we adopt horizontal and vertical flips. We use the Adam optimizer with an initial learning rate of 10^{-4} and decrease it to 10^{-6} using cosine annealing. The numbers of encoder blocks in the LLFormer from stage 1 to stage 4 are $\{2, 4, 8, 16\}$, and the number of attention heads in A-MSA are $\{1, 2, 4, 8\}$. The numbers corresponding to decoders from stage 1 to 3 are $\{2, 4, 8\}$ and $\{1, 2, 4\}$. For benchmarking, we compare 16 representative LLIE methods, including seven traditional non-learning methods (BIMEF (Ying, Li, and Gao 2017), FEA (Dong et al. 2011), LIME (Guo, Li, and Ling 2016), MF (Fu et al. 2016a), NPE (Wang et al. 2013), SRIE (Fu et al. 2016b), MSRCR (Jobson, Rahman, and Woodell 1997)), three supervised CNN-based methods (RetinexNet (Wei et al. 2018), DSLR (Lim and Kim 2020), KinD (Zhang, Zhang, and Guo 2019)), two unsupervised CNN-based methods (ELGAN (Jiang et al. 2021), RUAS (Liu et al. 2021b)), two zero-shot learning-based methods (Z_DCE (Guo et al. 2020), Z_DCE++ (Li, Guo, and Loy 2021)) and two supervised transformer-based methods (Uformer (Wang et al. 2022b), Restormer (Zamir et al. 2022)). For each method, we use the publicly available code and train each learning-based method for 300 epochs. For ELGAN, we directly use its pre-trained model for testing. Performance is evaluated with the PSNR, SSIM, LPIPS, and MAE metrics.

Benchmarking Study for UHD-LLIE

UHD-LOL4K Subset. We test 16 different state-of-the-art LLIE methods and our proposed LLFormer on the UHD-LOL4K subset. The quantitative results are reported in Table 1. According to Table 1, we can find that traditional LLIE algorithms (BIMEF, FEA, LIME, MF, NPE, SRIE, MSRCR) generally do not work well on UHD-LOL4K. Among them, the quantitative scores (PSNR, SSIM, LPIPS, MAE) of some methods are even worse than those of unsupervised learning methods (RUAS, ELGAN). The results of CNN-based supervised learning methods (see RetiunexNet, DSLR, and KID) are better than unsupervised learning-based and zero-shot learning-based methods, which is expected. Among the CNN-based methods, DSLR obtains the best performance in terms of PSNR, SSIM, LPIPS, and MAE. Compared with CNN-based supervised learning methods, the performances of transformer-based supervised learning methods (Uformer, Restormer, and LLFormer) are greatly improved. Among these, the proposed LLFormer obtains the best performance, achieving a 0.42 dB improvement in PSNR compared to Restormer. A visual comparison is shown in Fig. 5. The image recovered by LLFormer contains vivid colors and is closer to the ground truth.

UHD-LOL8K Subset. We also conduct benchmarking experiments on the UHD-LOL8K subset by partitioning each 8K image into 4 patches of 4K resolution. The last four columns of Table 1 show the evaluation results. Deep learning methods RetinexNet, DSLR, Uformer, Restormer, and LLFormer achieve better performance on both pixel-wise and perceptual metrics. Transformer-based methods achieve top ranks for all evaluation metrics with LLFormer outperforming other methods. As shown in Fig. 5, LLFormer produces visually pleasing results with more details.

Methods	UHD-LOL4K				UHD-LOL8K			
	PSNR ↑	SSIM ↑	LPIPS ↓	MAE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MAE ↓
input images	11.9439	0.5295	0.3125	0.2591	13.7486	0.6415	0.3104	0.2213
BIMEF† (Ying, Li, and Gao 2017)	18.1001	0.8876	0.1323	0.1240	19.5225	0.9099	0.1825	0.1048
FEA† (Dong et al. 2011)	18.3608	0.8161	0.2197	0.0986	15.3301	0.7699	0.3696	0.1700
LIME† (Guo, Li, and Ling 2016)	16.1709	0.8141	0.2064	0.1285	13.5699	0.7684	0.3055	0.2097
MF† (Fu et al. 2016a)	18.8988	0.8631	0.1358	0.1111	18.2474	0.8781	0.2158	0.1258
NPE† (Wang et al. 2013)	17.6399	0.8665	0.1753	0.1125	16.2283	0.7933	0.3214	0.1506
SRIE† (Fu et al. 2016b)	16.7730	0.8365	0.1495	0.1416	19.9637	0.9140	0.1813	0.0975
MSRCR† (Jobson, Rahman, and Woodell 1997)	12.5238	0.8106	0.2136	0.2039	12.5238	0.7201	0.4364	0.2352
RetinexNet‡ (Wei et al. 2018)	21.6702	0.9086	0.1478	0.0690	21.2538	0.9161	0.1792	0.0843
DSLR‡ (Lim and Kim 2020)	27.3361	0.9231	0.1217	0.0341	21.9406	0.8749	0.2661	0.0805
KinD‡ (Zhang, Zhang, and Guo 2019)	18.4638	0.8863	0.1297	0.1060	17.0200	0.7882	0.1739	0.1538
Z_DCE§ (Guo et al. 2020)	17.1873	0.8498	0.1925	0.1465	14.1593	0.8141	0.2847	0.1914
Z_DCE++§ (Li, Guo, and Loy 2021)	15.5793	0.8346	0.2223	0.1701	14.6837	0.8348	0.2466	0.1904
RUAS$^{\triangle}$ (Liu et al. 2021b)	14.6806	0.7575	0.2736	0.1690	12.2290	0.7903	0.3557	0.2445
ELGAN$^{\triangle}$ (Jiang et al. 2021)	18.3693	0.8642	0.1967	0.1011	15.2009	0.8376	0.2293	0.1713
Uformer* (Wang et al. 2022b)	29.9870	0.9804	0.0342	0.0262	28.9244	0.9747	0.0602	0.0344
Restormer* (Zamir et al. 2022)	36.9094	0.9881	0.0226	0.0117	35.0568	0.9858	0.0331	0.0195
LLFormer*	37.3340	0.9889	0.0200	0.0116	35.4313	0.9861	0.0267	0.0194

Table 1: Benchmarking study on the UHD-LOL4K and UHD-LOL8K subsets. $\dagger$, $\ddagger$, $\S$, $\triangle$ and $\star$ indicate the traditional methods, supervised CNN-based methods, unsupervised CNN-based methods, zero-shot methods and transformer-based methods. The top three results are marked in red, blue and purple, respectively.

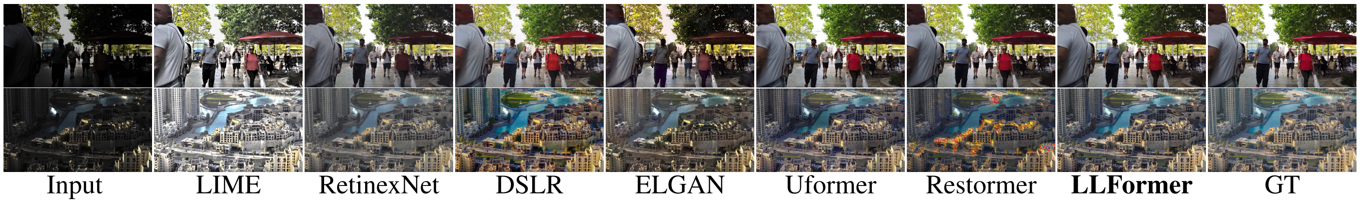
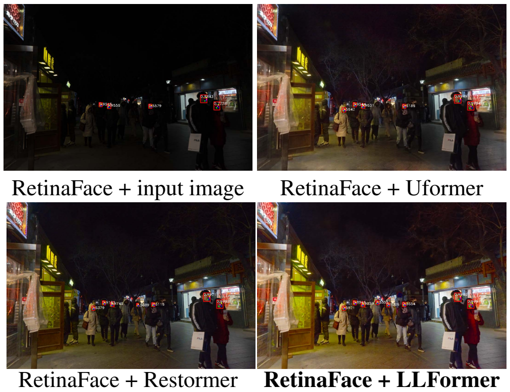


Figure 5: Visual results of different methods on the proposed dataset. The top row is from the UHD-LOL4K subset, and the bottom is from the UHD-LOL8K subset. **Zoom in for a better view**.


RetinaFace + input image

RetinaFace + Uformer

RetinaFace + Restormer

RetinaFace + LLFormer

Figure 6: Enhanced visual results and face detection results.

Improving Downstream Tasks. To verify whether LLIE is beneficial for downstream tasks, we randomly select 300 images from the DARK FACE dataset (Yang et al. 2020) and pre-process these images using the top three methods in our benchmark study. We then detect faces using RetinaFace (Deng et al. 2020). When using the pre-processing step, the average precision (AP) values for Uformer, Restormer, and LLFormer improve by 67.06%, 68.11%, and **71.2%**, respectively. Visual results are shown in Fig. 6. Pre-trained LLIE models not only generate images with adequate color balance, but also help improve the performance of downstream tasks.

Comparison results on Public Datasets

We benchmark LLFormer on the LOL (Wei et al. 2018) and MIT-Adobe FiveK (Bychkovsky et al. 2011) datasets, comparing it with 14 methods specifically designed for LLIE and two transformer-based methods. We use published code to retrain Uformer and Restormer on these datasets, respectively. Results are shown in Table 2. LLFormer achieves significantly higher performance on the LOL dataset, obtaining higher PSNR, SSIM, and MAE scores than Restormer. In terms of LPIPS, LLFormer ranks in second place. On the MIT-Adobe FiveK dataset, transformer-based methods rank at the top and LLFormer achieves the best results on all metrics. Among the best three transformer-based methods, the overhead (parameters and multiply-accumulate operations) for Uformer, Restormer and LLFormer are 38.82M/76.67G, 26.10M/140.99G and **24.52M/22.52G** (measured on 256 × 256 images), respectively. This shows that the proposed LLFormer achieves the best performance with efficient use of resources. This is due to the design of LLFormer, where the axis-based multi-head self-attention and hierarchical structure help to decrease the computational complexity. A visual comparison is shown in Fig. 7. LLFormer produces images with adequate saturation as well as color and texture fidelity.

Ablation Studies

We conduct ablation studies by measuring the contributions of the following factors: (1) Axis-based Multi-head Self Attention; (2) Dual Gated Feed-Forward Network; (3) Weighted skip connection; (4) Cross-layer Attention Fusion Block; (5) Width and depth of the network. Experiments are performed on the UHD-LOL4K subset, and models are trained on image patches of size 128 × 128 for 100 epochs.

A. Axis-based Transformer Block. We measure the impact of the proposed axis-based multi-head self attention and dual gated feed-forward network (FFN), see Table 3. Com-

Methods	LOL				MIT-Adobe FiveK			
	PSNR ↑	SSIM ↑	LPIPS ↓	MAE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MAE ↓
BIMEF (Ying, Li, and Gao 2017)	13.8752	0.5950	0.3264	0.2063	17.9683	0.7972	0.1398	0.1134
FEA (Dong et al. 2011)	16.7165	0.4784	0.3847	0.1421	15.2342	0.7161	0.1949	0.1512
LIME (Guo, Li, and Ling 2016)	16.7586	0.4449	0.3945	0.1200	13.3031	0.7497	0.1319	0.2044
MF (Fu et al. 2016a)	16.9662	0.5075	0.3796	0.1416	17.6271	0.8143	0.1204	0.1194
NPE (Wang et al. 2013)	16.9697	0.4839	0.4049	0.1290	17.3840	0.7932	0.1320	0.1224
SRIE (Fu et al. 2016b)	11.8552	0.4954	0.3401	0.2571	18.6273	0.8384	0.1047	0.1030
MSRCR (Jobson, Rahman, and Woodell 1997)	13.1728	0.4615	0.4350	0.2067	13.3149	0.7515	0.1767	0.1993
RetinexNet (Wei et al. 2018)	16.7740	0.4250	0.4739	0.1256	12.5146	0.6708	0.2535	0.2068
DSLR (Lim and Kim 2020)	14.9822	0.5964	0.3757	0.1918	20.2435	0.8289	0.1526	0.0880
KinD (Zhang, Zhang, and Guo 2019)	17.6476	0.7715	0.1750	0.1231	16.2032	0.7841	0.1498	0.1379
Z_DCE (Guo et al. 2020)	14.8607	0.5624	0.3352	0.1846	15.9312	0.7668	0.1647	0.1426
Z_DCE++ (Li, Guo, and Loy 2021)	14.7484	0.5176	0.3284	0.1801	14.6111	0.4055	0.2309	0.1539
RUAS (Liu et al. 2021b)	16.4047	0.5034	0.2701	0.1534	15.9953	0.7863	0.1397	0.1426
ELGAN (Jiang et al. 2021)	17.4829	0.6515	0.3223	0.1352	17.9050	0.8361	0.1425	0.1299
Uformer (Wang et al. 2022b)	18.5470	0.7212	0.3205	0.1134	21.9171	0.8705	0.0854	0.0702
Restormer (Zamir et al. 2022)	22.3652	0.8157	0.1413	0.0721	24.9228	0.9112	0.0579	0.0556
LLFormer	23.6491	0.8163	0.1692	0.0635	25.7528	0.9231	0.0447	0.0505

Table 2: Comparison results on LOL and MIT-Adobe FiveK datasets in terms of PSNR, SSIM, LPIPS and MAE. The top three results are marked in red, blue and purple, respectively. Same as (Zamir et al. 2020), we consider images from expert C for the MIT-Adobe FiveK dataset.

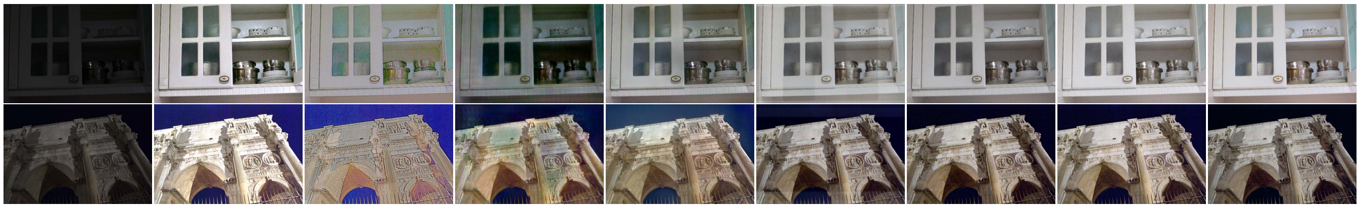


Figure 7: Visual comparison. The top row is from the LOL dataset and the bottom is from the MIT-Adobe FiveK dataset. **Zoom in for a better view**.

Variant	Component	MACs	Params	PSNR/SSIM
Base	(a) model with Resblock (Lim et al. 2017)	11.90G	13.87M	31.92/0.9771
Multi-head attention	(b) A-MSA (Height) + DGFN	13.60G	14.77M	35.15/0.9836
	(c) A-MSA (Width) + DGFN	13.60G	14.77M	34.98/0.9832
	(d) A-MSA + DGFN	16.26G	19.78M	35.31/0.9843
	(e) A-MSA + FFN (Vaswani et al. 2017)	18.90G	20.62M	23.33/0.9111
Feed-Forward Network	(f) A-MSA + DGFN	21.47G	24.18M	35.83/0.9846
	(g) A-MSA + DGFN	25.52G	24.52M	32.78/0.9786
LLFormer	(h) A-MSA + DGFN	22.52G	24.52M	**36.20/0.9867**

Table 3: Ablation study on Transformer Block. (d) refers to A-MSA without depth-wise convolution, (f) is DGFN without depth-wise convolution, and (g) is DGFN without the dual gated mechanism.

Variant	Component	MACs	Params	PSNR/SSIM
Skip	w/o skip	22.52G	24.52M	35.45/0.9844
	w/o conv	22.47G	24.50M	35.90/0.9852
	w/o skip+conv	22.47G	24.50M	35.12/0.9847
CAFB	w/o head CAFB	21.76G	24.51M	35.57/0.9847
	w/o tail CAFB	21.76G	24.51M	35.81/0.9852
	w/o CAFB	21.00G	24.50M	35.10/0.9835
LLFormer	contain all	22.52G	24.52M	**36.20/0.9867**

Table 4: Ablation study on connection and fusion.

pared with the base model using Resblock (Lim et al. 2017), our A-MSA (either height or width) and DGFN significantly contribute to the improvements. When using depth-wise convolution to enhance locality in self-attention (compare (d) and (h)) or the feed-forward network (compare (f) and (h)), the improvements in terms of PSNR are 0.89, 0.75, respectively. By applying the dual gated mechanism, PSNR and SSIM are improved by 3.42 and 0.0081 (see (g) (h)). Using the dual gated mechanism together with locality yields the best results. In contrast, combining A-MSA with the conventional FFN (Vaswani et al. 2017), degrades the performance (Table 3 (e)). This indicates that designing an appropriate FFN is critical for the transformer block.

B. Skip Connection and Fusion Block. To validate the weighted connection and cross-layer attention fusion block, we conduct ablation studies by progressively removing the corresponding components: (1) skip, (2) 1×1 convolution, (3) skip with 1×1 convolution, (4) head CAFB, (5) tail CAFB, (6) all CAFBs. Table 4 shows the results in terms of PSNR and SSIM, which indicate that each component helps improve the results. The model improves significantly when including CAFB and weighted skip connections. We observe a minor gain when applying 1×1 convolutions.

Variant	W/D	MACs	Params	PSNR/SSIM	Speed
LLFormer	**16/4**	22.52G	24.52M	**36.20/0.9867**	0.063 s
Wider	**32**/4	81.92G	95.63M	36.91/0.9871	0.120 s
	48/4	111.22G	114.49M	37.44/0.9880	0.152 s
	64/4	311.16G	377.64M	38.00/0.9881	0.193 s
Deeper	16/**3**	14.88G	3.51M	20.19/0.9432	0.054s
	16/**5**	29.53G	106.77M	36.09/0.9844	0.142 s
	16/**6**	36.45G	432.25M	35.62/0.9847	0.181 s
	16/**7**	43.32G	1727.08M	35.41/0.9845	0.217 s

Table 5: "Wider vs. Deeper" analysis.

C. Wider *vs.* Deeper. To understand the effect of width and depth in the network, we conduct ablation experiments to gradually increase the width (channels) and depth (number of encoder stages) of our LLFormer. Table 5 shows the results in terms of overhead, performance, and speed. The results demonstrate that LLFormer strikes the best tradeoff between performance and complexity (36.20/0.9867, 22.52G, 24.52M, 0.063s), compared to its wider or deeper counterparts.

Conclusion

In this paper, we build the first large-scale low-light UHD image enhancement benchmark dataset, which consists of UHD-LOL4K and UHD-LOL8K subsets. Based on this dataset, we conduct comprehensive experiments for UHD-LLIE. To the best of our knowledge, this is the first attempt to specifically address the UHD-LLIE task. We propose the first transformer-based baseline network called LLFormer for UHD-LLIE. Extensive experiments show that LLFormer significantly outperforms other state-of-the-art methods. The UHD-LOL dataset, together with LLFormer, will benefit the community in serving as a benchmark for LLIE and UHD-LLIE tasks.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61672273, 61832008), in part by Shenzhen Science and Technology Program (No. JSGG20220831093004008, JCYJ20220818102012025).

References

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bychkovsky, V.; Paris, S.; Chan, E.; and Durand, F. 2011. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR*, 97–104.

Cai, J.; Gu, S.; and Zhang, L. 2018. Learning a deep single image contrast enhancer from multi-exposure images. *IEEE TIP*, 27(4): 2049–2062.

Celik, T. 2014. Spatial entropy-based global and local image contrast enhancement. *IEEE TIP*, 23(12): 5298–5308.

Chen, C.; Chen, Q.; Xu, J.; and Koltun, V. 2018. Learning to see in the dark. In *CVPR*, 3291–3300.

Deng, J.; Guo, J.; Ververas, E.; Kotsia, I.; and Zafeiriou, S. 2020. Retinaface: Single-shot multi-level face localisation in the wild. In *CVPR*, 5203–5212.

Dong, X.; Wang, G.; Pang, Y.; Li, W.; Wen, J.; Meng, W.; and Lu, Y. 2011. Fast efficient algorithm for enhancement of low lighting video. In *ICME*, 1–6.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

Fu, X.; Zeng, D.; Huang, Y.; Liao, Y.; Ding, X.; and Paisley, J. 2016a. A fusion-based enhancing method for weakly illuminated images. *Signal Processing*, 129: 82–96.

Fu, X.; Zeng, D.; Huang, Y.; Zhang, X.-P.; and Ding, X. 2016b. A weighted variational model for simultaneous reflectance and illumination estimation. In *CVPR*, 2782–2790.

Girshick, R. 2015. Fast r-cnn. In *ICCV*, 1440–1448.

Guo, C.; Li, C.; Guo, J.; Loy, C. C.; Hou, J.; Kwong, S.; and Cong, R. 2020. Zero-reference deep curve estimation for low-light image enhancement. In *CVPR*, 1780–1789.

Guo, X.; Li, Y.; and Ling, H. 2016. LIME: Low-light image enhancement via illumination map estimation. *IEEE TIP*, 26(2): 982–993.

Jiang, Y.; Gong, X.; Liu, D.; Cheng, Y.; Fang, C.; Shen, X.; Yang, J.; Zhou, P.; and Wang, Z. 2021. Enlightengan: Deep light enhancement without paired supervision. *IEEE TIP*, 30: 2340–2349.

Jobson, D. J.; Rahman, Z.-u.; and Woodell, G. A. 1997. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE TIP*, 6(7): 965–976.

Kim, Y.-T. 1997. Contrast enhancement using brightness preserving bi-histogram equalization. *IEEE TCE*, 43(1): 1–8.

Kimmel, R.; Elad, M.; Shaked, D.; Keshet, R.; and Sobel, I. 2003. A variational framework for retinex. *IJCV*, 52(1): 7–23.

Lee, C.; Lee, C.; and Kim, C.-S. 2012. Contrast enhancement based on layered difference representation. In *ICIP*, 965–968.

Li, C.; Guo, C.; Han, L.-H.; Jiang, J.; Cheng, M.-M.; Gu, J.; and Loy, C. C. 2022. Low-light image and video enhancement using deep learning: A survey. *IEEE TPAMI*, 44(12): 9396–9416.

Li, C.; Guo, C. G.; and Loy, C. C. 2021. Learning to Enhance Low-Light Image via Zero-Reference Deep Curve Estimation. *IEEE TPAMI*, 44(8): 4225–4238.

Li, L.; Wang, R.; Wang, W.; and Gao, W. 2015. A low-light image enhancement method for both denoising and contrast enlarging. In *ICIP*, 3730–3734.

Lim, B.; Son, S.; Kim, H.; Nah, S.; and Mu Lee, K. 2017. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 136–144.

Lim, S.; and Kim, W. 2020. Dslr: Deep stacked laplacian restorer for low-light image enhancement. *IEEE TMM*, 23: 4272–4284.

Liu, J.; Xu, D.; Yang, W.; Fan, M.; and Huang, H. 2021a. Benchmarking low-light image enhancement and beyond. *IJCV*, 129(4): 1153–1184.

Liu, R.; Ma, L.; Zhang, J.; Fan, X.; and Luo, Z. 2021b. Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement. In *CVPR*, 10561–10570.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021c. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 10012–10022.

Lore, K. G.; Akintayo, A.; and Sarkar, S. 2017. LLNet: A deep autoencoder approach to natural low-light image enhancement. *PR*, 61: 650–662.

Ni, Z.; Yang, W.; Wang, S.; Ma, L.; and Kwong, S. 2020. Towards unsupervised deep image enhancement with generative adversarial network. *IEEE TIP*, 29: 9140–9151.

Shen, L.; Yue, Z.; Feng, F.; Chen, Q.; Liu, S.; and Ma, J. 2017. Msr-net: Low-light image enhancement using deep convolutional network. *arXiv preprint arXiv:1711.02488*.

Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A. P.; Bishop, R.; Rueckert, D.; and Wang, Z. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 1874–1883.

Stark, J. A. 2000. Adaptive image contrast enhancement using generalizations of histogram equalization. *IEEE TIP*, 9(5): 889–896.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*, 5998–6008.

Vonikakis, V.; Chrysostomou, D.; Kouskouridas, R.; and Gasteratos, A. 2013. A biologically inspired scale-space for illumination invariant feature detection. *Measurement Science and Technology*, 24(7): 074024.

Wang, L.; Xiao, L.; Liu, H.; and Wei, Z. 2014. Variational Bayesian method for retinex. *IEEE TIP*, 23(8): 3381–3396.

Wang, S.; Zheng, J.; Hu, H.-M.; and Li, B. 2013. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE TIP*, 22(9): 3538–3548.

Wang, T.; Zhang, X.; Jiang, R.; Zhao, L.; Chen, H.; and Luo, W. 2021. Video deblurring via spatiotemporal pyramid network and adversarial gradient prior. *Computer Vision and Image Understanding*, 203: 103135.

Wang, Y.; Wan, R.; Yang, W.; Li, H.; Chau, L.-P.; and Kot, A. 2022a. Low-light image enhancement with normalizing flow. In *AAAI*, 2604–2612.

Wang, Z.; Cun, X.; Bao, J.; and Liu, J. 2022b. Uformer: A general u-shaped transformer for image restoration. In *CVPR*, 17683–17693.

Wei, C.; Wang, W.; Yang, W.; and Liu, J. 2018. Deep retinex decomposition for low-light enhancement. In *BMVC*.

Xu, H.; Zhai, G.; Wu, X.; and Yang, X. 2013. Generalized equalization model for image enhancement. *IEEE TMM*, 16(1): 68–82.

Yang, C.; Jin, M.; Jia, X.; Xu, Y.; and Chen, Y. 2022. AdaInt: Learning Adaptive Intervals for 3D Lookup Tables on Real-time Image Enhancement. In *CVPR*, 17522–17531.

Yang, W.; Yuan, Y.; Ren, W.; Liu, J.; Scheirer, W. J.; Wang, Z.; Zhang, T.; Zhong, Q.; Xie, D.; Pu, S.; et al. 2020. Advancing image understanding in poor visibility environments: A collective benchmark study. *IEEE TIP*, 29: 5737–5752.

Ying, Z.; Li, G.; and Gao, W. 2017. A bio-inspired multi-exposure fusion framework for low-light image enhancement. *arXiv preprint arXiv:1711.00591*.

Zamir, S. W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F. S.; and Yang, M.-H. 2022. Restormer: Efficient Transformer for High-Resolution Image Restoration. In *CVPR*, 5728–5739.

Zamir, S. W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F. S.; Yang, M.-H.; and Shao, L. 2020. Learning enriched features for real image restoration and enhancement. In *ECCV*, 492–511.

Zeng, H.; Cai, J.; Li, L.; Cao, Z.; and Zhang, L. 2020. Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time. *IEEE TPAMI*, 44(4): 2058–2073.

Zhang, K.; Li, D.; Luo, W.; Ren, W.; Stenger, B.; Liu, W.; Li, H.; and Yang, M.-H. 2021a. Benchmarking Ultra-High-Definition Image Super-resolution. In *ICCV*, 14769–14778.

Zhang, K.; Luo, W.; Zhong, Y.; Ma, L.; Stenger, B.; Liu, W.; and Li, H. 2020. Deblurring by realistic blurring. In *CVPR*, 2737–2746.

Zhang, K.; Ren, W.; Luo, W.; Lai, W.-S.; Stenger, B.; Yang, M.-H.; and Li, H. 2022. Deep image deblurring: A survey. *IJCV*, 130(9): 2103–2130.

Zhang, X.; Shen, P.; Luo, L.; Zhang, L.; and Song, J. 2012. Enhancement and noise reduction of very low light level images. In *ICIP*, 2034–2037.

Zhang, X.; Wang, T.; Luo, W.; and Huang, P. 2021b. Multi-level fusion and attention-guided CNN for image dehazing. *TCSVT*, 31(11): 4162–4173.

Zhang, Y.; Zhang, J.; and Guo, X. 2019. Kindling the darkness: A practical low-light image enhancer. In *ACMMM*, 1632–1640.