NTIRE 2025 Challenge on Night Photography Rendering Challenge Solution: **Versatile Dual Rendering Network**

psykhexx
Xiaoyang Ma[1], Zijun Gao[1], Leyi Xing[2]
1. China Agriculture University, 2. Beihang University
maxiaoyang2023@outlook.com, 1363328344@qq.com, xingly1018@buaa.edu.cn

March 16, 2025

Abstract

This document describes our proposed solution for the"Night Photography" challenge part of the NTIRE workshop at CVPR 2025. This paper proposes a deep rendering pipeline that integrates multiple functions of the traditional image signal processing (ISP) pipeline. The outstanding feature of this method is that it uses a deep learning model to automatically adjust the brightness, color, noise, and texture details of the image in an end-to-end manner, thus avoiding the tedious process of manually designing algorithms and parameters in traditional methods.We briefly introduce our method and summarize our key contributions as follows.

1 Methodology

This paper proposes a convenient deep learning processing flow that integrates traditional ISP, as shown in Figure 1.The pipeline consists of five main steps, each of which optimizes a different aspect of the image. First, we performed a series of corrections and pre-processing on the original input image to obtain a cropped sRGB image. Then, we use two independent neural networks, IAT[1] and SCUNet[2], to render the image with an end-to-end feature fusion strategy, borrowing from the image processing methods of professional photographers. Specifically, IAT is responsible for adjusting contrast, brightness, hue, and saturation to enhance the dynamic range and color performance of the image; while SCUNet combines noise reduction and edge enhancement to further optimize image quality and visual perception. The following sections will explain these steps in detail.

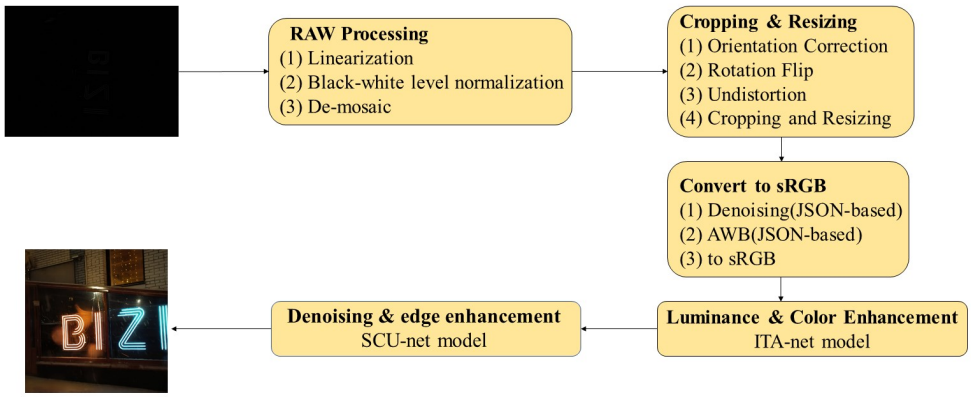

Figure 1: processing flow pipeline

1.1 Raw Processing

The raw image processing step is primarily designed to apply a series of corrections to the image in the raw domain, leveraging the metadata provided by the camera. First, the RAW image undergoes linearization and normalization based on its nominal black and white levels. Next, demosaicing is applied using the image's color filter array (CFA) to convert the single-channel RAW image into a full-color representation.Subsequently, rotational correction and horizontal flipping are performed to refine the image perspective. To further eliminate geometric distortions caused by the lens, projection transformation is employed, followed by cropping and resizing to align with the spatial resolution requirements of subsequent processing. Additionally, we utilize image metadata to conduct preliminary denoising and white balance correction, ensuring accurate color reproduction.Finally, the image is transformed into the CIE XYZ color space before being mapped to the standard sRGB color space, producing the final output in an 8-bit unsigned integer format. All processing in this stage is implemented based on the official code.

1.2 Brightness and Color Adjustment

We consolidate the brightness and hue adjustment into a unified process using the IAT network, the specific structure of the network comes from Figure 2 and 3(The image sources and most of the descriptions in this chapter are quoted from the paper [1]). We use the dual-branch feature of IAT to restore image illumination while completing color mapping and contrast adjustment.

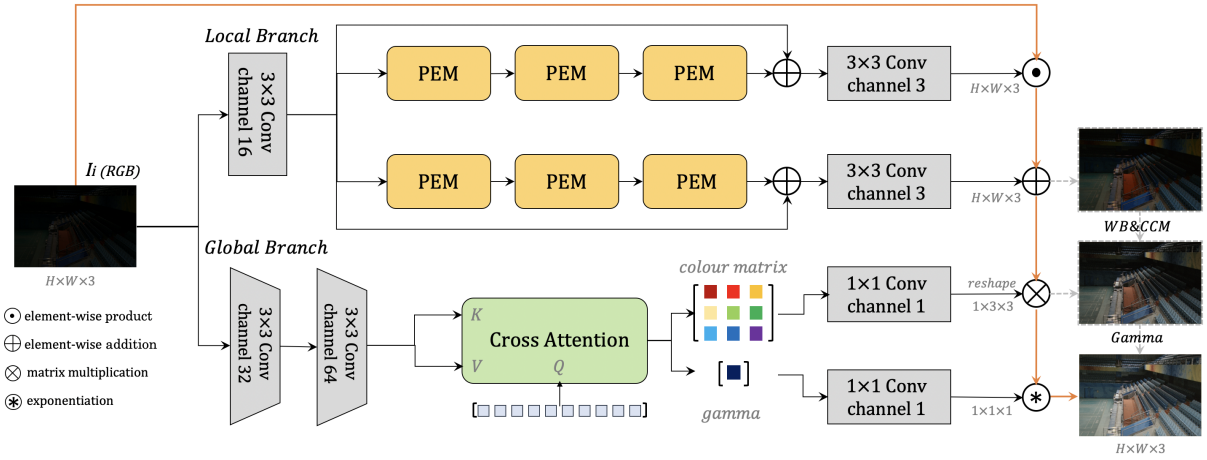

Figure 2: Structure of our Illumination Adaptive Transformer (IAT),the black line refers to the parameters generation while the yellow line refers to image processing[1]

Photometric degradation occurs in the original RGB space, so we consider processing on the original RGB data. To achieve this goal, we use a encoder-decoder structure to restore the sRGB image to the original RGB data and then edit the lighting and color. The adjustment of the input image I_i to the target image I_t is as follow:

$$I_t = g_t(f(I_i)) \tag{1}$$

where f represents the function that maps the input I_i to raw RGB data, g_t is the rendering function of the target camera, which consists of multiple decoders. We use different queries to control the ISP-related parameters (color matrix and gamma) in $g(\cdot)$. By querying and dynamically adjusting these parameters, the model can more accurately match the brightness and hue of the target image during the learning process, thereby improving the overall image quality and visual consistency. In training stage, the queries is dynamically updated in each iteration.We simplify this process into the equation (2)

$$g_t(\cdot) = \left(\max\left(\sum_{c_i} W_{c_i,c_j}(\cdot), \epsilon\right)\right)^{\gamma}, \quad c_i, c_j \in \{r, g, b\} \tag{2}$$

where W_{c_i,c_j} is a 3×3 joint color transformation matrix (including white balance and color transformation), which is controlled by 9 queries in total. γ is a gamma correction parameter, which is controlled by a separate query; to avoid numerical instability, set $\epsilon = 1e^{-8}$.The function f adopts a pixel-level least

squares model, which contains two branches to predict the multiplication mapping M and the addition mapping A namely:

$$f(I_i) = I_i \odot M + A. \tag{3}$$

The final IAT model equation 4 is as follows:

$$I_t = \left(\max \left(\sum_{c_j} W_{c_i,c_j} (I_i \odot M + A), 0 \right) \right)^{\gamma}. \tag{4}$$

The model decomposes the nonlinear operation into a local pixel-level component f and a global ISP component g_t, and designs two Transformer branches accordingly: a local adjustment branch and a global ISP branch.

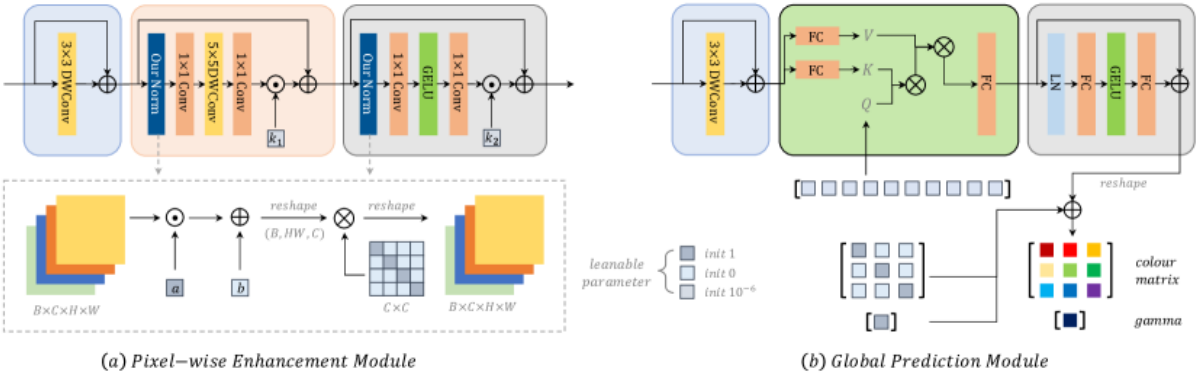

Figure 3: Detailed structure of Pixel-wise Enhancement Module (PEM) and Global Prediction Module (GPM)[1]

Local Branch This branch is used to estimate local components M and A to correct for illumination effects. Unlike the U-Net structure, this method uses a Transformer-style architecture to maintain the input resolution, which enables processing images of arbitrary resolution without resizing. Specifically, We first expand the channel dimension by a 3×3 convolution, and then pass them through two independent branches, each stacked with a pixel-level enhancement module (PEM). To reduce the amount of computation, deep convolution replaces self-attention. As shown in Figure 3(a), PEM uses a 3×3 deep convolution to encode the position information, followed by a PWConv-DWConv-PWConv sequence 3×3 deep convolutions to encode the position information, and then a PWConv-DWConv-PWConv sequence to enhance the local details. Finally, two 1×1 convolutions refine the label representation. We use lightweight normalization to replace the standard normalization layer in the transformer, which learns the scale a and bias b through two learnable parameters, and uses a learnable identity initialization matrix before fusing the channels. To further improve convergence, we apply layer scaling to scale the features by a small factor.

Global Branch Inspired by ISP operations such as gamma correction, color matrix transformation and white balance, as well as DETR, we design global component queries to predict the color matrix w and γ to control the generated sRGB images. As shown in Figure 3(b), we first use a lightweight encoder to extract global features at a lower resolution through two layers of stacked convolutions, which reduces the computation while enhancing the feature representation. Then, the encoded features are passed to the global prediction module (GPM). Different from standard DETR, our global component queries Q are initialized to 0 without multi-head self-attention. These learnable embeddings focus on the key K and value V, which are generated from the encoded features through deep convolutions to achieve resolution adaptability. After processing through a feed-forward network (FFN), we introduce two specially initialized parameters to predict the color matrix and gamma values, by initializing W to the identity matrix and γ to 1 to ensure stable training.

1.3 Denoising and Edge Enhancement

After the brightness and hue adjustment of IAT, the Luma noise caused by the exposure of the original image is not eliminated, and obvious Chroma Noise is introduced after color mapping, which seriously damages the image quality. Therefore, we plan to optimize the image noise while enhancing

the details of the image, so as to replace the denoising and sharpening process in the ISP operation. We can use the algorithm to solve the following optimization problem to achieve the above goals:

$$I^* = \arg\min_I \underbrace{\|Y(I) - Y(I_0)\|^2}_{\text{Luma Difference}} + \alpha \underbrace{\|C(I) - C(I_0)\|^2}_{\text{Chroma Difference}} + \beta \underbrace{R(I)}_{\text{Regularization Enhancement}} \tag{5}$$

where I, I_i is clean-noise image pairs; $Y(\cdot), C(\cdot)$ are operators for extracting luminance and chrominance channels in the YCbCr domain, $R(I)$ is the regularization term for detail enhancement, α, $\beta$is the trade-off parameter.This formula can be simplified to:

$$I^* = \arg\min_I \|W'(I - I_0)\|^2 + \beta\, \mathrm{R}(I) \tag{6}$$

where W' is the channel weighting matrix of YCbCr transform.Taking the deep model as the solution process, 6 can be expressed as the following bi-level optimization problem:

$$W'^* = \arg\min_{W'} \sum \mathcal{L}\left(I^*(W'), I\right), \tag{7}$$

$$\text{s.t.} \quad I^*(W') = \arg\min_I \left[\|W'(I - I_0)\|_2^2 + \beta\, R(I)\right] \tag{8}$$

This process is similar to blind image denoising based on Maximum A Posteriori (MAP). We adopted this idea to design SCUNet[2], as shown in Figure 4. SCUNet uses SC blocks Swin Transformer (SwinT), blocks and residual convolution (RConv) blocks, implemented by 1×1 convolution, split and splice operations, and residual connections. The input feature tensor X is first processed by 1×1 convolution and then evenly split into two feature map groups X_1 and X_2. Then, X_1 and X_2 are input to SwinT blocks and RConv blocks respectively to obtain Y_1 and Y_2.Finally, Y_1 and Y_2 are spliced as inputs of 1×1 convolution and residually connected with input X to obtain the final output of the SC block.

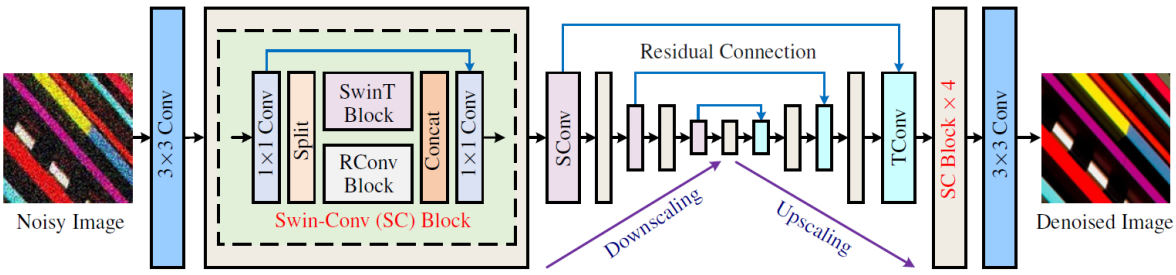

Figure 4: The architecture of Swin-Conv-UNet (SCUNet) denoising network.[2]

2 Experiments

We used the official training data as the training set, the second evaluation dataset as the validation set, and the first evaluation dataset as the test set, with 300, 150, and 50 image pairs, respectively. We first processed all the images raw and used them as input for IAT model training. Both of our models were trained on a single GeForce RTX 4090 GPU.

For the IAT model, the model was optimized using the Adam optimizer with a weight decay of 1e-4. We adopted a two-stage training strategy, first cropping the training images into 512×512 blocks and training them for 100 epochs with a batch size of 8, with an initial learning rate of 1e-4. The validation images were cropped into 1024×1024 and trained for 10 epochs with a batch size of 4 as fine-tuning, with an initial learning rate of 1e-5. Finally, the full-resolution images were tested on the test data and the PSNR and SSIM scores were calculated. The model parameters with the best score were selected and the output images were calculated as the input for SCUNet training. It is worth noting that no image augmentation techniques are applied during IAT training. The loss function used is L1 loss measuring the difference between the input image and the target image. To mitigate overfitting, we adopt a cosine learning rate schedule.

For the SCUNet model, we crop the training images into 256×256 patches and apply random horizontal and vertical flips to augment the data. The validation set is then used to fine-tune the images

cropped into 512×512 patches with batch sizes of 4 and 2, respectively. Similar to the IAT model, we use L1 loss as the loss function. The model is trained using the Adam optimizer with an initial learning rate of 5e-5 and a weight decay of 1e-4. We also adopt a cosine learning rate schedule.

3 Acknowledge

Some of codes and models in this article are borrowed from GitHub repositories(The following is a hyperlink):

Illumination-Adaptive-Transformer

SCUNet

nightimaging25

We would like to thank them for their beautiful work and great open source.

Thanks to the Key Lab of Smart Agriculture Systems(Key Lab of Smart Agriculture Systems, Ministry of Education, China Agricultural University, Beijing 100083, China) for providing computing power support.

References

[1] Ziteng Cui, Kunchang Li, Lin Gu, Shenghan Su, Peng Gao, ZhengKai Jiang, Yu Qiao, and Tatsuya Harada. You only need 90k parameters to adapt light: a light weight transformer for image enhancement and exposure correction. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022.

[2] Kai Zhang, Yawei Li, Jingyun Liang, Jiezhang Cao, Yulun Zhang, Hao Tang, Deng-Ping Fan, Radu Timofte, and Luc Van Gool. Practical blind image denoising via swin-conv-unet and data synthesis. *Machine Intelligence Research*, 20(6):822–836, 2023.