IllumiFusionGAN: Attention-Driven Deep Retinex Network for Superior Low-Light Enhancement

1st Hung-Manh Nguyen
Faculty of Information Technology
FPT University
HO CHI MINH City, VIET NAM
0009-0008-8858-1626

2nd Khang-Hoang Nguyen Vo
Faculty of Information Technology
FPT University
HO CHI MINH City, VIET NAM
0009-0009-3260-3002

3rd Thuan-Tran Van
Faculty of Information Technology
FPT University
HO CHI MINH City, VIET NAM
0009-0006-6123-6190

4th Tuan-Minh Le
Faculty of Information Technology
FPT University
Ho Chi Minh City, Vietnam
0009-0007-9675-9052

5th Cam-My Thi Nguyen
Faculty of Information Technology
FPT University
Ho Chi Minh City, Vietnam
mycamnt3@gmail.com

6th Thu-Le Minh Vo
Mentor guiding
FPT University
Ho Chi Minh City, Vietnam
levominhthu@gmail.com

***Abstract*—We propose a deep learning approach for low light enhancement, leveraging Retinex [41] theory to decompose images into reflectance and illumination. Our extended U-Net[42] architecture integrates residual connections, attention mechanisms, and multi-scale fusion to enhance brightness while preserving details. A four-channel input, including a maximum intensity channel, enriches feature representation. Spectral normalization stabilizes training, while adversarial training with GANs [12] refines illumination and realism. To further improve enhancement quality, we employ a diverse set of loss functions to guide the model's optimization. Implemented with PyTorch and Kornia, our model outperforms benchmarks on SSIM and PSNR, ensuring superior brightness restoration, structural consistency, and color fidelity.**

***Index Terms*—Image-Processing, Low-light image , Deep-Learning, Four-channel input.**

I. INTRODUCTION

Low-light image enhancement is challenging due to issues like noise, low contrast, and color distortion, but a deep learning approach inspired by Retinex theory can address these challenges while preserving image details[**Detailed-based**][10][13][29]. The proposed method uses an extended U-Net architecture with residual learning, attention mechanisms, and multi-scale fusion to adaptively enhance images[3].

II. RELATED WORK

A. Traditional Method

In the past years, researchers in the field of image quality improvement have proposed methods such as: Brightness Adjustment, Gamma Correction, Histogram Equalization (HE), Contrast Limited Adaptive Histogram Equalization (CLAHE)[37, 38, 28, 8, 45]. But these methods encountered a big problem when it caused the color to be incorrect with the object, high noise and loss of information in the dark area.

B. Retinex Method

To improve the above problems, we considered using Retinex theory[41] and using other architectures to denoise, restore lost details and increase image quality. According to Retinex theory, the image will be separated into 2 components: the reflection component and the illumination component.

In [48], they used Retinex to decompose the image into two components through basic convolutional layers. However, the results were not as expected when the training time was too long and the colors were not realistic.

C. Deep-learning method

To improve the existing Retinex network, we will use techniques to retain important details, restore color, and make the image as realistic as possible.

1) Residual Block: We use residual block[15] to maintain the original information of the image and support learning important features using the skip connections technique.

This technique will help the image not lose information during the feed-forward process, as well as the details in the dark regions will not be noisy and produce a better result than the basic RetinexNet.

2) U-Net: Similar to Residual block, U-Net[42] also uses Skip connections to keep important information in the image. Thanks to the Encoder-Decoder technique, U-Net can reconstruct details in the image as well as reduce noise and balance image colors.

3) Attention Mechanism: Attention Mechanism[44] is a technique that focuses on important parts of the input data by assigning the highest weight to important regions and the lowest weight to less important regions. We use two popular techniques: Channel Attention and Spatial Attention[26].

Channel Attention helps to identify the most important color channel among the three channels Red, Green, Blue using the Squeeze-and-Excitation Block mechanism. Spatial Attention helps to identify which spatial regions need to be brightened and clarified.

4) Generative Adversarial Networks: Generative Adversarial Networks (GANs)[12] are deep learning models that use image reconstruction techniques and then distinguish between the reconstructed image and the real image to improve the output image better.

GANs consist of two components: Generator and Discriminator. Generator will try to create fake images that are close to real images through deep learning layers, then Discriminator will compare to distinguish the two images and return parameters for Generator to continue learning. Repeat until both images have similar similarity.

We use GANs to help images have natural brightness and color, increase the ability to recover details in the region and make the image sharper.[43]

III. METHODOLOGY

A. Network Architecture

The Retinex theory-based decomposition network leverages an extended U-Net architecture to separate an image into its reflectance (R) and illumination (I) components. This network employs an encoder-decoder structure enhanced with attention mechanisms and residual connections to improve performance. The network outputs a 4-channel tensor, with 3 channels representing reflectance (R) and 1 channel representing illumination (I).

The input to the network consists of 4 channels: the red, green, and blue color channels (R, G, B) of the original image, along with an additional channel called "input_max". The "input_max" channel contains the maximum pixel value across the color channels for each pixel location. This additional channel provides the network with supplementary information about the image's intensity, potentially aiding in the separation of reflectance and illumination.

To further refine the illumination component, an "IlluminationEnhancer" network is utilized to improve I. Additionally, a Generative Adversarial Network (GAN) is incorporated to distinguish between real images and the images reconstructed by the decomposition network. This adversarial training helps to ensure that the reconstructed images are realistic and perceptually similar to the original input images.

1) Multi-scale Feature Fusion Module: To integrate information from varying resolution setups, a Multi-scale Fusion Module is employed to merge the outputs of multiple cascade networks that have different numbers of input channels[40]. This module utilizes a 2D convolution with a 1×1 kernel to reduce the number of channels after aggregating features from multiple height resolutions[40].

Initially, all input features are normalized to a common spatial dimension using bilinear interpolation, ensuring consistent input sizes before merging[40][35][11][1]. Subsequently, the features are concatenated along the channel direction and processed through a 1×1 convolutional layer to decrease the channel count to the desired level[40]. To stabilize the training process, a Batch Normalization layer is applied [40][34][33][7][5]. Finally, the ReLU activation function is used to introduce nonlinearity into the model[36][5][40][21].

2) Deep Attention Residual Blocks:

a) Channel Attention: The Channel Attention mechanism focuses on emphasizing significant channels within the input feature map using two statistical aggregation methods: *Global Average Pooling* (GAP) and *Global Max Pooling* (GMP). These techniques extract global information from each channel, which is then processed by a neural network consisting of two linear layers. The resulting attention weights are applied to the input channels, either enhancing or minimizing the influence of each channel during feature extraction. This allows the network to focus on the most relevant features and suppress irrelevant ones, improving overall performance.

$$A_c(x) = \sigma(\text{MLP}(\text{AvgPool}(x)) + \text{MLP}(\text{MaxPool}(x))) \tag{1}$$

b) Spatial Attention: The Spatial Attention mechanism is designed to pinpoint significant regions within the spatial dimensions of an input feature map. Unlike channel attention, this mechanism extracts spatial information by computing both the mean and maximum values along the channel dimension. By focusing on important areas in the image, the Spatial Attention mechanism enhances the model's ability to detect useful features. This allows the network to prioritize relevant spatial locations, improving its overall understanding and performance.

$$A_s(x) = \sigma(\text{Conv}_{77}([\text{AvgPool}_c(x); \text{MaxPool}_c(x)])) \tag{2}$$

c) Deep Attention Block: The Deep Attention Block is a module that combines both Channel Attention and Spatial Attention to enhance feature learning. Its structure consists of a convolutional layer followed by Batch Normalization (BN) and a ReLU activation function. The feature maps are then adjusted using Channel Attention and Spatial Attention before being combined with the original input via a residual connection. The formulation of the Deep Attention Block is as follows:

$$\mathbf{Y} = \text{ReLU}\big((\mathbf{X}' \cdot \mathbf{A}_{\text{channel}}) \cdot \mathbf{A}_{\text{spatial}} + \mathbf{X}\big) \tag{3}$$

d) Residual Block With Deep Attention: The Residual Block With Deep Attention consists of a sequence of multiple Deep Attention Blocks to leverage the benefits of Residual Learning. Stacking multiple blocks allows the model to capture more complex features while maintaining gradient stability. The general formulation of this residual block is:

$$\mathbf{Y} = \text{DeepAttentionBlocks}(\mathbf{X}) + \mathbf{X} \tag{4}$$

3) Illumination Enhancement Network: The IlluminationEnhancer is a deep learning model designed to improve lighting in dark images, utilizing an encoder-decoder architecture enhanced with deep attention mechanisms[2]. It employs a *ResidualBlockWithDeepAttention* to boost feature extraction by integrating Channel Attention and Spatial Attention[24]. The encoder extracts information through three levels, and the bottleneck learns a deep representation of the image[23].

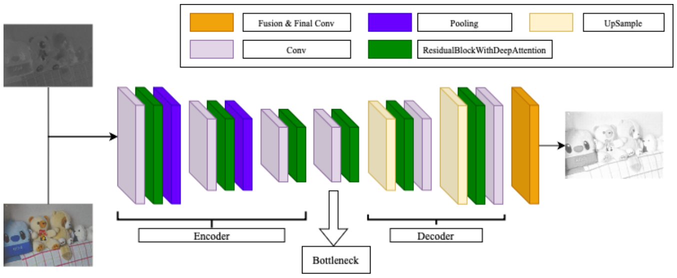

Fig. 1. Architectural model **Lighting enhancer**. The model consists of three main parts: **Encoding**, **Bottleneck** and **Decoding**. Colored blocks represent important components, including Conv (purple), Pooling (blue), ResidualBlockWithDeepAttention (green), UpSample (light yellow), and Fusion & Final Conv (orange). The model takes as input a dark image and returns an image that has been enhanced with brightness

The decoder reconstructs the image using features combined from the encoder via residual connections. *MultiScaleFusion* is applied to aggregate information from different levels, which refines the output illumination map.

$$I_\delta = \Gamma(\text{MSF}(\text{Dec}_1, \text{Up}(\text{Dec}_2))) \tag{5}$$

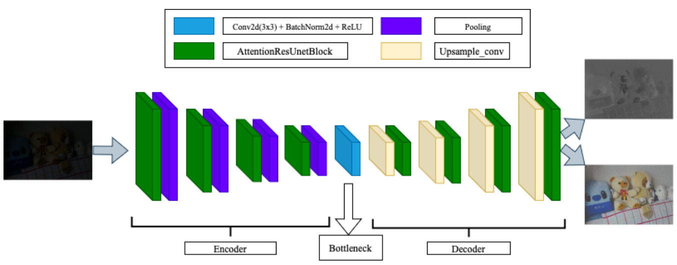

Fig. 2. The DecomUnet architecture features an encoder for feature extraction, a bottleneck for processing information, and a decoder for reconstructing the output image. Specific enhancements are achieved using the AttentionResUnetBlock, and Upsample_conv assists in restoring image dimensions.

4) Decomposition Network: The encoder in the described architecture uses four stacked Attention Residual Blocks, each succeeded by a max pooling operation, to capture and compress key features from input data while decreasing spatial dimensions[50][6]. This design enables effective extraction and compression of salient features while reducing the spatial dimensions of the input data[50][6]. The bottleneck layer then refines these features, which improves their quality prior to upsampling.

$$R, I = \text{Sigmoid}(\text{Split}(\text{DecomUnet}(x \oplus \max_c(x))) \tag{6}$$

B. Adversarial Training Framework

1) Generator Architecture: The generator consists of two main components: the **DecomNet** and the **Enhancer**. The **DecomNet** decomposes the input image I_{input} into a reflectance component R_{low} and an illumination component I_{low}:

$$R_{\text{low}}, I_{\text{low}} = \text{DecomNet}(I_{\text{input}}) \tag{7}$$

Next, the **Enhancer** refines the illumination by processing

$$I_\delta = \text{Enhancer}(I_{\text{low}}, R_{\text{low}}) \tag{8}$$

Finally, the enhanced image is reconstructed by element-wise multiplication:

$$I_{\text{enhanced}} = R_{\text{low}} \odot I_\delta \tag{9}$$

2) Discriminator Design:

$$\text{Attention}(Q, K, V) = \text{Softmax}(QK^T/\sqrt{d})V \tag{10}$$

where:

- $\mathbf{Q}$ represents the query matrix.
- $\mathbf{K}$ represents the key matrix.
- $\mathbf{V}$ represents the value matrix.
- d is the dimensionality of the key vectors.

The process computes the dot product of $\mathbf{Q}$ and $\mathbf{K}$, scales it by $\sqrt{d}$, applies Softmax for normalization, and weights $\mathbf{V}$ accordingly. This enables the model to focus on key input regions, improving its ability to capture dependencies and relationships.

The equation represents the Feature Matching Loss, a crucial component in training generative adversarial networks (GANs). This loss function quantifies the discrepancy between the feature representations of real and generated (fake) samples within the discriminator network. By minimizing this loss, the generator is incentivized to produce samples that closely resemble real data in terms of their feature distributions.

$$\mathcal{L}_{\text{FM}} = \mathbb{E}\left[\sum_{i=1}^{L} \frac{1}{N_i} \|D^{(i)}(x_{\text{real}}) - D^{(i)}(x_{\text{fake}})\|_1\right] \tag{11}$$

C. Multi-objective Loss Function

To improve the training process of the network, we need to optimize the model so that the output image is improved. To do that, we will divide it into two parts based on the working mechanism of GANs.

1) Generator Loss: Generator is used to reconstruct dark image into bright image based on Retinex image decomposition mechanism and using additional techniques to balance image brightness and color. The loss function of Generator is expressed as follows:

$$\begin{aligned}\mathcal{L}_{\text{g}} = \mathcal{L}_{\text{recon}} + \lambda_1 \mathcal{L}_{\text{adv}} + \lambda_2 \mathcal{L}_{\text{color}} + \lambda_3 \mathcal{L}_{\text{fm}} \\ + \lambda_4 \mathcal{L}_{\text{I\_smooth}} + \lambda_5 \mathcal{L}_{\text{R\_smooth}} + \lambda_6 \mathcal{L}_{\text{SSIM}}\end{aligned} \tag{12}$$

In this paper, We use *L1* norm to calculate most of the loss values and the lambdas will be automatically adjusted in the range of $[0.01, 3.0]$ to find the weights that fit the model.. For reconstruction loss, we will combine the loss function of low image and high image:

$$\mathcal{L}_{\text{recon\_low}} = \|\text{R}_{\text{Low}} \cdot \text{L}_{\text{Low}} - \text{I}_{\text{Low}}\|_1 \tag{13}$$

$$\mathcal{L}_{\text{recon\_high}} = \|\text{R}_{\text{High}} \cdot \text{L}_{\text{High}} - \text{I}_{\text{High}}\|_1 \tag{14}$$

$$\mathcal{L}_{\text{recon}} = \lambda_{\text{low}} \mathcal{L}_{\text{recon\_low}} + \lambda_{\text{high}} \mathcal{L}_{\text{recon\_high}} \tag{15}$$

In the formula (13) and (14), we will calculate to find the difference between the input and output images of low image and high image. (15) is a combination of both loss functions above with lambdas automatically adjusted between $[1.0, 3.0]$ to find the right weights. For color loss, we will convert from RGB to LAB color space because we want to separate the light and color parts separately:

$$\Delta \mathrm{X} = \mathrm{X}_{\text{fake}}^{\text{LAB}} - \mathrm{X}_{\text{real}}^{\text{LAB}} = (\Delta L, \Delta A, \Delta B) \tag{16}$$

$$\mathcal{L}_{\text{color}} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i=1}}^{\mathrm{N}} \sqrt{(\Delta L_i)^2 + (\Delta A_i)^2 + (\Delta B_i)^2} \tag{17}$$

In equation (16), we calculate the difference between the fake image generated by the Generator and the image. Finally, we calculate the color loss based on the 3 LAB color channels according to the L2 norm in equation (17). For Feature Matching loss (11), we also use L1 norm to force Generator to generate images with features that are close to real images. To improve the smoothness of the two main components in Retinex, we use the Sobel gradient method to detect details in the image and reduce noise. It is calculated according to the following formula:

$$\mathcal{L}_{\text{I\_smooth}} = \|\Delta I^2 \circ \exp(-\Delta R^2)\|_1 \tag{18}$$

$$R_{low}^{softmax} = \frac{\infty}{\beta} \log \sum_{\mathrm{c}} \exp(\beta R_{low,c}) \tag{19}$$

$$\mathcal{L}_{R\_smooth} = \frac{1}{N} \sum (R_{low}^{softmax} - I_{eq})^2 \tag{20}$$

In (18), ΔI and ΔR are partially differentiated to find the edge features using the Sobel method. In (19) and (20), we calculate the softmax loss of the reflectance component and the image has applied Histogram Equalization (HE) to find the difference of the two components. For adversarial loss, we calculate to estimate how similar the Discriminator output is to the real image via Binary Cross Entropy with Logits:

$$\mathcal{L}_{adv} = -\frac{1}{N} \sum \left[y \cdot \log \sigma(D_{out}) + (1-y) \cdot \log\left(1 - \sigma(D_{out})\right) \right] \tag{21}$$

Where y is a value of 0 or 1 representing fake and real images, D_{out} is the output of the Disciminator in GANs. Finally, there is the SSIM loss which is a measure of similarity between enhanced image and high image based on texture, brightness and contrast:

$$\mathcal{L}_{SSIM} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Where x and y are the enhanced image and high image respectively. muy and sigma are the mean and covariance of the two images. C1 and C2 are small constants to avoid division by zero.

2) Discriminator Loss: For the Discriminator, we will use Hinge loss and Gradient Penalty (WGAN-GP) to optimize the distinction between real and fake images:

$$\mathcal{L}_D = \mathcal{L}_{D\_real} + \mathcal{L}_{D\_fake} + \mathcal{GP} \tag{22}$$

We use Hinge Loss to optimize the distinction between real and fake images, if the value is close to 0 it means that the two images do not have too much difference and are relatively similar, which is defined as follows:

$$\mathcal{L}_{D\_fake} = \max(0, 1 + D_{fake}) \tag{23}$$

$$\mathcal{L}_{D\_real} = \max(0, 1 - D_{real}) \tag{24}$$

Gradient Penalty is an improvement in Wasserstein GAN[49] that helps the model generate better and more detailed images:

$$\mathcal{GP} = \lambda \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}} \left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2 \right] \tag{25}$$

Where, $\hat{x}$ is the interpolation pattern between the real and fake images, $\nabla_{\hat{x}} D(\hat{x})$ is that of the Discriminator on the interpolation pattern. In (25), we set lambda = 10 based on the results obtained in the experiment.

IV. EXPERIMENT

A. Dataset and Evaluation Metrics

LOLv1 has been widely used as a benchmark to evaluate various low-light image enhancement techniques. Researchers often evaluate their models on this dataset using metrics such as PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index)[9]. The dataset provides a standardized testbed that allows fair comparisons between different enhancement methods. In the LOLv1 dataset, we used 485 pairs of images for training, and 15 pairs of images for testing to evaluate the model performance.

LOLv2 has become a benchmark to evaluate low-light image enhancement algorithms[16]. Researchers often train their models on training datasets and evaluate their performance using metrics such as PSNR, SSIM, and LPIPS on test datasets[18]. The diversity and practicality of LOLv2 make it particularly valuable for developing algorithms that can generalize to real-world low-light conditions. We train over 600 image pairs on the LOLv2 dataset, and evaluate the model performance using the LOLv1 test set.

To evaluate RURnetGAN, we use several metrics to check its performance. We use Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) for image quality when we have original images to compare against[19]. For datasets without original images, we use Naturalness Image Quality Evaluator (NIQE)[19].

PSNR measures the ratio of maximum signal power to noise power, showing image quality compared to noise[39]. SSIM checks how similar the structure of an image is to the original, considering luminance, contrast and structure[39]. NIQE assesses image quality by looking at naturalness without needing a reference image[19].

Using these metrics gives a complete view of RURnetGAN's performance[19]. PSNR and SSIM provide quality scores relative to original images, while NIQE works for datasets where originals are unavailable[19]. Together, these metrics help to thoroughly measure RURnetGAN's effectiveness[19].

B. Result

The quantitative comparison results, as detailed in table I, demonstrate that RURnetGAN outperforms benchmark methods on both LOLv1 and LOLv2 datasets. RURnetGAN achieved the highest PSNR and SSIM values, indicating superior image enhancement performance compared to other methods.

On the LOLv1 dataset, RURnetGAN reached a PSNR of 24.07 dB and an SSIM of 0.81. These values significantly exceed those of the second-ranked method, IAT (PSNR 23.33, SSIM 0.80), as well as other methods like KinD and RUAS. The data highlight RURnetGAN's effectiveness in enhancing low-light images, surpassing existing techniques in both pixel recovery and structural information preservation.

Similarly, RURnetGAN leads on the LOLv2 dataset with a PSNR of 26.0 dB and an SSIM of 0.77. The consistently high and stable performance across both datasets demonstrates RURnetGAN's robustness and generalization capabilities. Overall, the quantitative results confirm the superiority of the RURnetGAN method in enhancing low-light images, proving its advanced ability to recover pixel and structural information compared to current state-of-the-art methods.

	LOLv1		**LOLv2**	
	PSNR ↑	**SSIM ↑**	**PSNR ↑**	**SSIM ↑**
RURnetGAN	**24.07**	**0.81**	**26.0**	**0,77**
LIME [14]	16.67	0.56	15.24	0.47
LightenNet [22]	10.53	0.44	-	-
RetinexNet [41]	16.77	0.56	18.37	0.72
MBLLEN [27]	17.90	0.72	18.00	0.72
EnGAN [20]	17.48	0.65	18.23	0.62
KinD [47]	20.87	0.79	19.74	0.76
RUAS [30]	18.23	0.72	18.37	0.72
IAT [4]	23.33	0.80	23.50	0.82

TABLE I
PERFORMANCE COMPARISON OF RURNETGAN AND OTHER LOW-LIGHT IMAGE ENHANCEMENT METHODS ON LOLV1 AND LOLV2 DATASETS

TableII presents the comparison results of the NIQE index, which evaluates the naturalness of the augmented images. The results show that KinD achieved the lowest NIQE value of 3.650, indicating that the images enhanced by KinD have the highest naturalness according to this index among the compared methods. EnGAN also achieved a relatively low NIQE (3.688), which is comparable to KinD. RURnetGAN achieved a NIQE of 3.975, ranking third in terms of naturalness according to NIQE, which is still significantly better than LIME, RetinexNet, and RUAS. The LIME, RetinexNet, and RUAS methods have significantly higher NIQE values, indicating that the augmented images from these methods may be less natural. These results show that deep learning methods such as KinD, EnGAN, and RURnetGAN tend to produce more natural-looking augmented images than other methods in this comparison, as assessed by the NIQE index.

Model	**NIQE**
RURnetGAN	**3.975**
LIME [14]	4.408
RetinexNet [41]	4.351
EnGAN [20]	3.688
KinD [47]	3.650
RUAS [30]	4.434

TABLE II
RURNETGAN'S NIQE REVIEW AND MATCHING LOW-LIGHT IMAGE ENHANCEMENT METHODS

V. CONLUSION

This paper introduces a Retinex theory-based deep learning method for low-light image enhancement[31][25]. The approach decomposes images into reflective and illuminated components, utilizing an expanded U-Net architecture with residual connections, attention mechanisms, and multi-scale fusion to restore brightness while preserving natural color and fine details[46][31]. Spectral normalization combined with GAN-based adversarial training ensures stable, realistic outputs, and a four-channel input enhances feature representation[32][31]. The model outperforms traditional methods on benchmark datasets like LOLv1 and LOLv2 in terms of SSIM, PSNR, and perceptual quality[17][31]. Future work will focus on reducing latency and complexity and expanding applicability for more realistic scenarios[17][31].

REFERENCES

[1] Omni Calculator. 2024. Bilinear interpolation calculator. (June 2024). https://www.omnicalculator.com/math/bilinear-interpolation.

[2] Benjamin Cham. 2023. Breaking through the darkness: how to enhance low-light images ... (May 2023). https://medium.com/htx-s-s-coe/breaking-through-the-darkness-how-to-enhance-low-light-images-with-deep-learning-techniques-257c98daf96f.

[3] HengShuai Cui, Jinjiang Li, Zhen Hua, and Linwei Fan. 2022. Attention-guided multi-scale feature fusion network for low-light image enhancement. (Mar. 2022). https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2022.837208/full.

[4] Z. Cui, K. Li, L. Gu, S. Su, P. Gao, Z. Jiang, Y. Qiao, and T. Harada. 2022. You only need 90k parameters to adapt light: a lightweight transformer for image enhancement and exposure correction. In *Proceedings of the 33rd British Machine Vision Conference (BMVC)*. BMVA Press, London, UK, (Nov. 2022).

[5] DanB. 2018. Rectified linear units (relu) in deep learning - kaggle. (2018). https://www.kaggle.com/code/dansbecker/rectified-linear-units-relu-in-deep-learning.

[6] DhanushKumar. 2023. Max pooling. the pooling operation involves sliding... - medium. (Nov. 2023). https://medium.com/@danushidk507/max-pooling-ef545993b6e4.

[7] Ketan Doshi. 2021. Batch norm explained visually - how it works, and why neural ... (May 2021). https://towardsdatascience.com/batch-norm-explained-visually-how-it-works-and-why-neural-networks-need-it-b18919692739/.

[8] Vinay Dubey and R. Katarya. 2021. Adaptive histogram equalization based approach for sar image enhancement: a comparative analysis. (May 2021). https://www.semanticscholar.org/paper/723673bbadc4a3843c719df3c70aad139f30bdc2.

[9] dvlab-research. 2022. Dvlab-research/snr-aware-low-light-enhance - github. (Mar. 2022). https://github.com/dvlab-research/SNR-Aware-Low-Light-Enhance.

[10] Ed Erkes. 2023. Dealing with noise: salvaging a severely underexposed high iso ... (Aug. 2023). https://erkesphoto.com/photography-technique/dealing-with-noise-salvaging-a-severely-underexposed-high-iso-image/.

[11] GeeksforGeeks. 2024. What is bilinear interpolation? - geeksforgeeks. (Sept. 2024). https://www.geeksforgeeks.org/what-is-bilinear-interpolation/.

[12] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. (2014). https://arxiv.org/abs/1406.2661 arXiv: 1406.2661 [stat.ML].

[13] Jiawei Guo, Jieming Ma, Angel F Garcia-Fernandez, Yungang Zhang, and Haining Liang. 2023. A survey on image enhancement for low-light images. (Mar. 2023). https://pubmed.ncbi.nlm.nih.gov/37025779/.

[14] X. Guo, Y. Li, and H. Ling. 2016. Lime: low-light image enhancement via illumination map estimation. *IEEE Transactions on Image Processing*, 26, 2, 982–993. DOI: 10.1109/TIP.2016.2639450.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *CoRR*, abs/1512.03385. http://arxiv.org/abs/1512.03385 arXiv: 1512.03385.

[16] here. 2024. Zero-shot low light image enhancement with diffusion prior - arxiv. (Dec. 2024). https://arxiv.org/html/2412.13401v1.

[17] Yi Huang, Xiaoquang Tu, Gui Fu, Tingting Liu, Bokai Liu, Ming Yang, and Ziliang Feng. 2023. Low-light image enhancement by learning contrastive representations in spatial and frequency domains. (Mar. 2023). https://arxiv.org/abs/2303.13412.

[18] Md Tanvir Islam, Inzamamul Alam, Simon S. Woo, Saeed Anwar, IK Hyun Lee, and Khan Muhammad. 2024. Loli-street: benchmarking low-light image enhancement and beyond. (Oct. 2024). https://arxiv.org/html/2410.09831v1.

[19] 2022 Jānis Iesalnieks (QA Engineer)June 7. 2022. Full-reference quality metrics: vmaf, psnr and ssim - testdevlab. (June 2022). https://www.testdevlab.com/blog/full-reference-quality-metrics-vmaf-psnr-and-ssim.

[20] Y. Jiang, X. Gong, D. Liu, Y. Cheng, C. Fang, X. Shen, J. Yang, P. Zhou, and Z. Wang. 2021. Enlightengan: deep light enhancement without paired supervision. *IEEE Transactions on Image Processing*, 30, 2340–2349. DOI: 10.1109/TIP.2021.3051462.

[21] Bharath Krishnamurthy. 2022. Relu activation function explained - built in. (Oct. 2022). https://builtin.com/machine-learning/relu-activation-function.

[22] C. Li, J. Guo, F. Porikli, and Y. Pang. 2018. Lightennet: a convolutional neural network for weakly illuminated image enhancement. *Pattern Recognition Letters*, 104, 15–22. DOI: 10.1016/j.patrec.2018.02.003.

[23] Tie Zhong Litian Wang Liquan Zhao and Chunming Wu. 2024. Low-light image enhancement using generative adversarial networks. (Aug. 2024). https://www.nature.com/articles/s41598-024-69505-1.

[24] Fangjin Liu, Zhen Hua, Jinjiang Li, and Linwei Fan. 2022. Low-light image enhancement network based on recursive network. (Mar. 2022). https://pmc.ncbi.nlm.nih.gov/articles/PMC8961027/.

[25] Xinyi Liu, Qi Xie, Qian Zhao, Hong Wang, and Deyu Meng. 2022. Low-light image enhancement by retinex based algorithm unrolling and adjustment. (Feb. 2022). https://arxiv.org/abs/2202.05972.

[26] E Lu and X Hu. 2022. Image super-resolution via channel attention and spatial attention. (Jan. 2022). https://link.springer.com/article/10.1007/s10489-021-02464-6.

[27] F. Lv, F. Lu, J. Wu, and C. Lim. 2018. Mbllen: low-light image/video enhancement using cnns. In *Proceedings of the 29th British Machine Vision Conference (BMVC)* number 1. Vol. 220, 4. DOI: 10.5244/C.32.220.

[28] Jianqiang Mei, Fan Jia, Wanyan Chen, and Biyuan Li. 2023. Deeps: a novel framework for image quality improvement of x-ray non-destructive testing. (Apr. 2023). https://www.semanticscholar.org/paper/9e202dfa08daac86e5b7e420b4fa174b2777abdd.

[29] Varina Patel. 2022. Mastering low contrast light in nature photography. (Dec. 2022). https://visualwilderness.com/fieldwork/mastering-low-contrast-light-in-nature-photography.

[30] L. Risheng, M. Long, Z. Jiaao, F. Xin, and L. Zhongxuan. 2021. Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[31] S. Rostami, F. Zare, M. Amirian, and M. Amiri. 2024. Fpga-based implementation for spiking neural networks with real-time learning. (Nov. 2024). https://www.nature.com/articles/s41598-024-80339-9.

[32] Zhixu Shu and Kewang Zhang. 2024. Spectral normalization for generative adversarial networks for artistic image transformation. (Oct. 2024). https://onlinelibrary.wiley.com/doi/full/10.1155/2024/6644706.

[33] Keras Team. 2015. Batchnormalization layer - keras. (Jan. 2015). https://keras.io/api/layers/normalization_layers/batch_normalization/.

[34] Contributors to Wikimedia projects. 2018. Batch normalization - wikipedia. (Apr. 2018). https://en.wikipedia.org/wiki/Batch_normalization.

[35] Contributors to Wikimedia projects. 2004. Bilinear interpolation - wikipedia. (May 2004). https://en.wikipedia.org/wiki/Bilinear_interpolation.

[36] Contributors to Wikimedia projects. 2012. Rectifier (neural networks) - wikipedia. (Dec. 2012). https://en.wikipedia.org/wiki/Rectifier_(neural_networks).

[37] K. Venkatesh and P. Prasad. 2018. Filter based enhancement of x-ray image through hardware description language. (July 2018). https://www.semanticscholar.org/paper/54f659711887f16adcb0e3e6f849ce521eb49569.

[38] S. Vishnyakov, Vadim Lazarev, and Y. Vishnyakova. 2024. Introduction of the image processing and machine learning techniques for pupils. (Apr. 2024). https://www.semanticscholar.org/paper/d3137636d761782706bc7bca4afc0995bb45baeb.

[39] Visionular-admin. 2024. Making sense of psnr, ssim, vmaf - visionular. (July 2024). https://visionular.ai/vmaf-ssim-psnr-quality-metrics/.

[40] Xuesong Wang. 2020. Msfm: multi-scale fusion module for object detection - openreview. (Oct. 2020). https://openreview.net/forum?id=6IVdytR2W90.

[41] C. Wei, W. Wang, W. Yang, and J. Liu. 2018. Deep retinex decomposition for low-light enhancement. In *Proceedings of the 29th British Machine Vision Conference (BMVC)* number 1. Vol. 220. DOI: 10.5244/C.32.220.

[42] Frauke Wilm, Jonas Ammeling, Mathias Öttl, R. Fick, M. Aubreville, and Katharina Breininger. 2024. Rethinking u-net skip connections for biomedical image segmentation. (Feb. 2024). https://www.semanticscholar.org/paper/f1c31e46f8fd2bcd30ae84cab2f931aaa5a1837a.

[43] Bolin Xu, Dong-Sheng Zhou, and Weijing Li. 2022. Image enhancement algorithm based on gan neural network. (Jan. 2022). https://www.semanticscholar.org/paper/a4f1960fca0a1e64c560157ee65d675092612bcf.

[44] X Yang. 2020. An overview of the attention mechanisms in computer vision. (Jan. 2020). https://iopscience.iop.org/article/10.1088/1742-6596/1693/1/012173/meta.

[45] Febi Yanto, Nanda Jannata, Lestari Handayani, and Eka Pandu Cynthia. 2024. Pengaruh contrast limited adaptive histogram equlization dalam klasifikasi ct-scan tumor ginjal menggunakan deep learning. (June 2024). https://www.semanticscholar.org/paper/3971c2abbd4cd7613bb9ef138a6fd51677651c99.

[46] Guangtao Zhang, Weiwei Liu, Xiaohua Xia, and Lei Zhao. 2023. Spectral enhanced retinex model for low-light image enhancement with optimized weighted least squares. (Sept. 2023). https://www.mdpi.com/2076-3417/13/18/10336.

[47] Y. Zhang, J. Zhang, and X. Guo. 2019. Kindling the darkness: a practical low-light image enhancer. In *Proceedings of the 27th ACM International Conference on Multimedia, MM '19*. ACM, New York, NY, USA, 1632–1640. DOI: 10.1145/3343031.3350931.

[48] Yu Zhang, Xiaoguang Di, Bin Zhang, and Chunhui Wang. 2020. Self-supervised image enhancement network: training with low light images only. *CoRR*, abs/2002.11300. https://arxiv.org/abs/2002.11300 arXiv: 2002.11300.

[49] Yue Zhao and G. Gan. 2021. Research on network security situation prediction based on improved gan. (Oct. 2021). https://www.semanticscholar.org/paper/8a7e3edc1c8eac056d6b184d5986535ee209a283.

[50] Zubair. 2024. Residual attention network overview — by zubair - medium. (July 2024). https://medium.com/@zakhtar2020/residual-attention-network-overview-7a349212a7f1.