

Low Light Image Enhancement Using Transfer Learning on a Lightweight U-Net Architecture

Bruce A. Liboy

Master in Computer Science

Advisor: Jeffrey Duffany, Ph.D.

Polytechnic University of Puerto Rico

Graduate Project EXPO February 2026

Abstract — *Low-Light Image Enhancement (LLIE) plays a crucial role in photography, surveillance, autonomous systems, and scientific imaging. Traditional enhancement techniques often struggle to recover fine details and may introduce unwanted color distortions. In this project, a lightweight U-Net model will be developed for low-light image enhancement. Transfer learning will be employed by first pretraining the model on a synthetic low-light dataset, followed by fine-tuning on real paired low-light images. Using the LOw-Light (LOL) dataset on Kaggle, model performance will be evaluated before and after transfer learning using Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) image quality metrics. The lightweight design of the model makes it well-suited for deployment on edge devices and mobile platforms.*

Keywords — *Artificial Intelligence, Computer Vision, Edge ML, Low Light Image Enhancement, Low-Light Dataset, Transfer Learning, U-Net Architecture.*

INTRODUCTION

Low-light image enhancement (LLIE) has become an active area of research due to its importance in improving visual perception and downstream computer vision tasks under challenging illumination conditions. Publicly available datasets such as the LOL dataset introduced by Rakshit provide paired low- and normal-light images that have become a common benchmark for training and evaluating LLIE methods [1]. Recent deep learning approaches have demonstrated strong performance by learning complex illumination and frequency characteristics directly from data, including adaptive frequency decomposition networks that explicitly model illumination aware features for enhancement [2]. In parallel, there has been growing interest in deploying

LLIE models on resource-constrained platforms, with recent work exploring compact and real-time deep learning solutions suitable for embedded systems such as the ESP32-S3-EYE [3]. Additionally, U-Net-based architectures have been shown to be effective for smartphone-oriented low-light enhancement, balancing visual quality with architectural simplicity [4].

The research presented here pertains to a series of experiments that were conducted by utilizing a U-net architecture with the goal of producing augmented pictures for better visibility. The vision behind this theoretical work, is to eventually produce a model that is capable of taking as input raw camera footage in low light conditions and producing well lit images and video in real time for in situ operations, be it security, exploration, or something in between. To achieve this, the model needs to be lightweight enough to be able to run in constrained edge computing devices that have a limited power supply and mission critical objectives. For this reason, the U-net Architecture was selected due to its simplicity and efficiency for these kinds of tasks.

Moreover, it is also worth noting that the dataset chosen for this task is also of the utmost importance, given that it plays a major role in the training, testing, and evaluation of the model at hand. For this purposes, the LOL dataset was selected. The LOL dataset is composed of 500 low-light and well lit image pairs and is divided into 485 training pairs and 15 testing pairs. The low-light images contain noise produced during the photo capture process. Most of the images are indoor scenes. All the images have a resolution of 400×600 and the dataset was introduced in the paper Deep Retinex Decomposition for Low-Light Enhancement.

When discussing experimentation approaches, a framework that compares the difference between

a model with and without fine-tuning was chosen so that a comparison could be made between both models to evaluate if fine-tuning helps improve a U-net for the task of LLIE or not. Both models were also compared to the classical low light image enhancement technique of histogram equalization to visualize whether or not these systems rival classical approaches. This gives good insights into the benefits of fine tuning and tests the hypotheses of whether or not ML approaches are superior in quality than classical algorithms. An extensive discussion on how the models were trained, tested, and evaluated can be found on the section “Methodology” of this paper.

BACKGROUND

Low-light image enhancement (LLIE) is a fundamental problem in computer vision, aiming to improve the visibility, contrast, and perceptual quality of images captured under insufficient illumination. Low-light conditions commonly arise in real-world scenarios such as nighttime photography, surveillance, autonomous driving, remote sensing, and medical imaging. Images captured in such environments typically suffer from low contrast, color distortion, loss of detail, and amplified sensor noise, which significantly degrade both human visual perception and the performance of downstream vision tasks.

Early approaches to LLIE relied on classical image processing techniques that operate directly on pixel intensity distributions. Among the most widely used methods is Histogram Equalization (HE), which enhances global contrast by redistributing pixel intensities to span the full dynamic range. Variants such as Adaptive Histogram Equalization (AHE) and Contrast Limited Adaptive Histogram Equalization (CLAHE) attempt to address some of the shortcomings of global HE by enhancing contrast locally. While these methods are computationally efficient and easy to implement, they often introduce undesirable artifacts, including noise amplification, over enhancement, and loss of natural illumination consistency.

Overall, classical LLIE methods are limited by their reliance on heuristic rules and lack the ability to adapt to complex, scene-dependent illumination variations.

With the advent of deep learning, LLIE has increasingly been formulated as a supervised image-to-image translation problem. Convolutional Neural Networks (CNNs) have demonstrated strong capability in learning complex non-linear mappings between low-light images and their corresponding well-lit ground truth counterparts. Encoder–decoder architectures, particularly U-Net–based models, have been widely adopted due to their ability to capture multi-scale contextual information while preserving fine-grained spatial details through skip connections.

Several learning-based methods train networks directly on paired low-light and normal-light datasets, such as the LOL dataset, using pixel-wise loss functions (e.g., L1 or L2 loss) and perceptual metrics. These approaches generally outperform classical methods in terms of structural preservation and perceptual quality. However, supervised training requires high-quality paired datasets, which are difficult and costly to acquire in real-world settings.

Several learning-based methods train networks directly on paired low-light and normal light datasets, such as the LOL dataset, using pixel-wise loss functions (e.g., L1 or L2 loss) and perceptual metrics. These approaches generally outperform classical methods in terms of structural preservation and perceptual quality. However, supervised training requires high-quality paired datasets, which are difficult and costly to acquire in real-world settings.

Evaluating LLIE models is challenging due to the subjective nature of visual quality and the intentional modification of image brightness. Full-reference metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) are commonly used to quantify pixel-level fidelity and structural preservation when ground truth images are available.

Despite the progress made by deep learning–based LLIE methods, a systematic comparison between classical enhancement techniques, learning

based models trained without fine-tuning, and models refined through task-specific fine-tuning remains valuable. Such comparisons help clarify the contribution of learned representations versus handcrafted enhancement rules and quantify the performance gains achieved through domain adaptation.

This work builds upon existing LLIE research by evaluating a U-Net based enhancement model under different training regimes and comparing its performance against a classical histogram equalization baseline. Performance is assessed using objective image quality metrics, providing insight into the trade-offs between classical and learning based enhancement approaches.

METHODOLOGY

In regards to methodology, the discussion will be divided by topics, starting from data loading, and then moving on to topics such as pipelines/pre-processing, model training, model testing, and model evaluation.

Data Loading

For Data Loading, considering the project was developed in Kaggle, we simply located the LOL Dataset in the datasets section of this service. The Dataset was originally submitted by the user Soumik Rakshit. This is the authentic dataset mentioned on the paper "Deep Retinex Decomposition for Low-Light Enhancement". Once the dataset was traced in the platform, a notebook was created from the location the dataset was stored in, so that the option to import the dataset was presented seamlessly on upon notebook initialization.

Pipelines/Pre-Processing

Once the notebook was initialized, two strings were created containing the origin paths for both the folder which contained the low-light images, and for the folder which contained the well lit images. Once this was put in place, two lists were created and (glob) was used to find all (.PNG) files located in these folders, such that one list contained all of the paths for the low light images, and the other

contained all paths for the well lit images. Python is the choice of language used for all former and subsequent operations discussed in this paper. Next, a train-test split was performed on the data, and 30 sample images were reserved for future validations for the samples that were going to be used for the training portion of the project, a tensorflow dataset was created to hold both the paths for the low light and well lit images alike. Furthermore, shuffling of the dataset samples was performed before batching so that each batch had random samples. It was also decided to shuffle the paths of these samples instead of the tensors themselves for the added efficiency. After, each pair of low/high images contained in the dataset were passed to a user defined function which contained all of the necessary pre-processing steps with the use of tensorflow's (.map) method. The user defined function first reads the raw bytes of the image file from disk using tensorflow's (tf.io.read_file()). After the image pair is loaded into the notebook, it's decompressed from PNG bytes to pixel tensors with the command using tensorflow's (tf.image.decode_png()). Once the image pair's been read into the notebook and transformed into pixel tensors, another transform is applied to resize the image pair into a resolution of (256X256) and normalized so that all pixel intensity values range from 0 to 1, this is achieved by using tensorflow's command (tf.image.resize(low, (256, 256)) / 255.0). Finally, after all transformations have been performed on the pair, the user defined function returns the pair for further processes down the line.

Once these transformations have been applied to all images from the dataset, these are cached in memory for faster subsequent epochs with the tensorflow command (dataset.cache()), and batches of 8 samples are created with the command (dataset.batch(8)). Lastly, prefetching is configured on the dataset so that subsequent batches may be loaded while the current batch is training; this is achieved with the command (dataset.prefetch(tf.data.AUTOTUNE)).

After this step, more user defined functions are created to apply random synthetic darkening on the

well lit images so that the pre-trained model can train on synthetic low light and true well lit image pairs. This is accomplished by performing transformations on the well lit images to apply random gamma darkening, random brightness reduction, and added noise to achieve a realistic interpretation of the ground truth low light image counterparts of the LOL dataset.

Once these user defined functions were established, the dataset used to pre-train the model was formed in the same manner as the initial training dataset explained above, with the caveat that, instead of utilizing ground truth low light images, the pre-train dataset contains only synthetic low and well lit pairs. It is also worth noting that the testing dataset was also transformed in the same manner.

Model Architecture

The U-Net architecture of choice is comprised of the input layer, which is a 256X265X3 layer, which accounts for all color channels of a 256X256 image and the convolutional blocks, which main purpose is to extract features from the input image, such as edges, textures, and brightness patterns, among others. A standard kernel size of 3X3 was chosen, and the padding setting was set to the "same" option, which produces output sizes equal to the input size. This is important to preserve spatial resolution. The activation chosen for the convolutional blocks was that of "relu", which introduces non-linearity so that the network can model complex transformations. Moreover, the "two convolution layers in a block" is a common U-Net design that captures richer features before down-sampling.

In addition to the convolutional blocks, there are also the up-sampling blocks, whose main purpose is to restore the original image resolution after down-sampling and merge high-resolution features from the encoder. Upsampling doubles the height and width of samples. The skip connections in these blocks connect the upsampled output to the corresponding layer in the encoder. This helps recover fine details lost during down-sampling. Additionally, there are the encoder blocks in the

form of max pooling layers, whose main function is to extract feature maps and to down-sample the input by halving both the height and the width of the input passed to the layer (e.g., 256X256 to 128X128). A full diagram of the architecture used is provided in Figure 1.

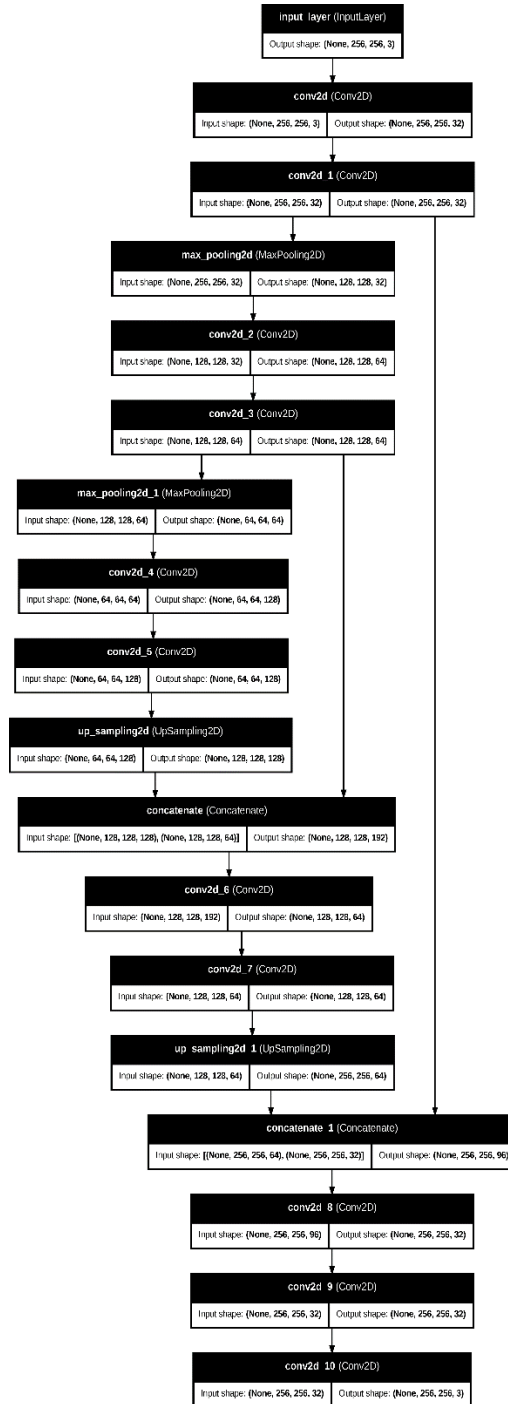


Figure 1
U-Net Model Architecture

Model Training

For the training aspect of this project, 2 models were trained with a U-Net Architecture. One of the models only received general training (synthetic low light image, and ground truth well lit image pair), and the other model was pre-trained on the same data as the former model, but with the additional fine-tuning step where, on top of the pre-training data, the model was also trained on ground truth low light images, and ground truth well lit images.

The training was done in Kaggle, and GPU compute resources were used to accelerate the training process due to time constraints, but this model is small enough that it is possible to train solely on CPU; it just takes more time. The GPU resource used on Kaggle was a GPU P100, and both the compilation of the models and the training were done using these compute resources. The optimizer that was used for the model was an Adaptive Gradient Descent (Adam) at a learning rate of 1e-4, which is generally considered conservative and a good choice for image-to-image tasks. The loss function that was used to train these models was the Mean Absolute Error function (MAE). MAE is common for LLIE tasks because it tends to produce sharper images than MSE, and it also penalizes large pixel errors less aggressively; on top of that, it also produces a model that is more robust to outliers.

Finally, a simple strategy of 20 epochs was chosen to fit the model to the training data for the model that only trained on the synthetic/Ground Truth pair. For the model that was pretrained on the synthetic/Ground truth pair and fine-tuned on the full training dataset, the model was trained for 20 epochs on the pre-training dataset and then another 20 epochs on the actual training dataset. This approach was simple enough to test the hypothesis of whether transfer learning affects the accuracy of the model in a positive manner or not.

Model Testing

To test the model, a handful of useful metrics were chosen to evaluate the quality of the pictures produced by both models. The two metrics that were chosen for this task were SSIM and PSNR.

Peak Signal to Noise Ratio (PSNR) measures pixel-wise fidelity between two images. The metric answers the question "How close is each pixel value in the enhanced image to the ground truth?". PSNR Tells us how much noise, overexposure, or distortion the enhancement introduced. A high PSNR means Less pixel distortion, better brightness recovery, and fewer artifacts (haloing, noise amplification).

$$PSNR = 10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \quad (1)$$

Where:

- I = ground truth image
- \hat{I} ground truth image
- H, W = image height and width
- MAX_I = maximum possible pixel value = 1.0

On the other hand, the Structural Similarity Index (SSIM) measures perceptual similarity, not raw pixel accuracy. It compares images based on luminance (brightness), contrast, and structure (edges, textures, shapes). SSIM answers the question "Does the enhanced image preserve the same structure and visual information as the ground truth?" In LLIE, SSIM is arguably more important than PSNR. a higher SSIM means edges preserved, no over-smoothing, and natural-looking enhancement.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (2)$$

Where:

- x, y : image patches from the enhanced image and the ground truth image, respectively
- μ_x, μ_y : mean intensity of image patches x and y
- σ^2, σ^2 : variance of image patches x and y
- σ_{xy} : covariance between image patches x and y
- C_1, C_2 : small constants introduced to stabilize the division when the denominators are close to zero

Typically:

$$C_1 = (k_1L)^2, C_2 = (k_2L)^2 \quad (3)$$

With:

- $k_1 = 0.01, k_2 = 0.03$

- L = dynamic range of pixel values (1 or 255)

The most important takeaway here is that PSNR evaluates pixel-wise fidelity between enhanced images and ground truth, providing a quantitative measure of distortion introduced during enhancement. However, PSNR does not fully reflect perceptual quality. Therefore, SSIM is also reported to assess structural preservation and perceptual similarity. In the context of LLIE, SSIM is particularly important as it captures edge consistency, texture retention, and overall visual realism.

The testing of the models was made with the reserved 30 images that were not used for training and the metrics computed were an average of the performance of the model on these 30 images reserved for this task.

RESULTS AND DISCUSSION

The Results and Discussion section is structured to evaluate the performance of the proposed approaches through both quantitative metrics and qualitative analysis. The quantitative results subsection focuses on objective image quality measures, specifically PSNR and SSIM, to provide a standardized comparison of reconstruction fidelity and structural preservation across models and the classical baseline.

Complementing this analysis, the qualitative observations subsection adopts the perspective of a human observer to discuss perceptual differences in the enhanced images. This discussion emphasizes visual attributes such as tonal balance, edge definition, color rendition, and texture, highlighting characteristic behaviors of data-driven models—such as increased softness and contrast smoothing—as well as limitations of classical methods like histogram equalization, particularly its tendency toward desaturation and unnatural tonal distributions.

Quantitative Results

After training the models and performing the testing on both models, results show that the model

without fine-tuning achieved a PSNR Mean of 14.291, with a PSNR Standard Deviation of 3.611 and a SSIM Mean of 0.709 with an SSIM Standard Deviation of 0.133 which are great as a baseline to compare the fine-tuned model to Figure 2. shows the output of this model for a visual assessment of image quality.

On the other hand, the fine-tuned model achieved numbers that outperformed the simpler model without the fine-tuning. The fine-tuned model got a PSNR Mean of 16.251, which is a step up from the former model and indicates better pixel-wise fidelity, a PSNR Standard Deviation of 2.917, which indicates that results are less spread than the basic model, a SSIM Mean of 0.776, which indicates a small boost in structural preservation and perceptual similarity, and a SSIM Standard Deviation of 0.094. These results prove that transfer learning enhances models that are aimed for the task of Low Light Image enhancement, and that incorporating the such may boost PSNR and SSIM quality metrics over the conventional models with no transfer learning. Figure 2 shows the output of this model for a visual assessment of image quality.

Lastly, it's also worth comparing the different models' output with a classical approach to LLIE. For this reason, it was opted to pick Histogram Equalization to compare the results yielded by said approach and compare the results with those of a data-driven approach to observe if the results of the ML-driven approach are better or worse than conventional/classical approaches to LLIE. When testing on the same set of images as those used for the validation of the ML models, it was observed that Histogram Equalization achieved a PSNR Mean of 15.134, a PSNR Standard Deviation of 2.403, an SSIM Mean of 0.463, and an SSIM Standard Deviation of 0.112. This indicates, according to the SSIM Mean, which is one of the most important metrics for LLIE, that ML-driven approaches tend to do much better at this task than the classical approach discussed in this paper. Figure 2 shows the output of histogram equalization for a visual assessment of image quality. Table 1 shows these results in a more summarized manner for better comparison.

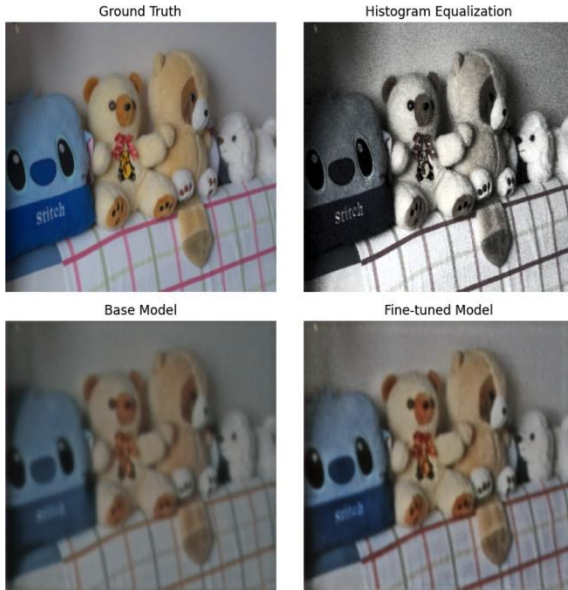


Figure 2
A Comparative 2X2 Figure that Depicts the Output Difference of Each Approach VS the Ground Truth (Upper Right)

Table 1
Performance Comparison of LLIE Methods

Metric	Base Model	Fine-tuned Model	Hist. Eq.
PSNR Mean	14.291	16.251	15.134
PSNR Std. Dev.	3.611	2.917	2.403
SSIM Mean	0.709	0.776	0.463
SSIM Std. Dev.	0.133	0.094	0.112

Qualitative Observations

When analyzing the visual characteristics of the images produced by the base model, it is evident that the model performs well at substantially increasing image brightness while largely preserving the original color palette, especially when compared to the classical approach. In addition to brightness enhancement, the base model maintains a reasonable balance between highlights and shadows, avoiding both overexposed regions and excessive shadow clipping. However, despite these strengths, the resulting images appear noticeably soft and lack contrast, leading to a flat visual appearance that reduces overall perceptual appeal.

The fine-tuned model effectively addresses these limitations without sacrificing the strengths of the base model. In particular, it produces images with improved color richness, enhanced sharpness, and better-defined edges, while maintaining equal or superior brightness levels. As a result, the fine-tuned

model yields outputs that are both visually more appealing and structurally more faithful to the original scene, overcoming the loss of detail observed in the base model.

In contrast, the histogram equalization approach produces images of inferior visual quality relative to the data-driven methods. Although it increases brightness and apparent sharpness, it significantly degrades color fidelity and introduces noticeable noise and artifacts. These shortcomings lead to unnatural-looking results and highlight the limitations of classical enhancement techniques when compared to learning-based LLIE approaches.

On the other hand, while effective at increasing global contrast, Histogram Equalization does not take into account the relationships between color channels or the semantic structure of the scene. When applied to low-light natural images, this results in output that often appears desaturated and unnatural. This effect arises because the independent stretching of each RGB channel disrupts the original channel balance, reducing chroma and causing colors to wash out. Furthermore, low-light images typically contain significant sensor noise, which is amplified alongside the signal during Histogram Equalization, further degrading color fidelity and visual realism. Consequently, while Histogram Equalization can improve overall brightness and local contrast, it frequently produces images that lack the color richness and perceptual quality observed in learning-based methods. Figure 3 displays this phenomenon in a comparative 2X2 grid.

Despite these limitations, Histogram Equalization exhibits strong performance in high-contrast, structure-dominated scenarios such as text or document images. In such cases, the primary objective is not color fidelity but rather the enhancement of edges and local contrast to distinguish foreground from background. Since text is largely monochromatic and relies on luminance differences, Histogram Equalization's aggressive contrast stretching effectively enhances readability without concern for color distortion. This explains why Histogram Equalization often outperforms learning-based approaches in tasks such as optical character recognition or license plate detection, where the

preservation of edge sharpness is more critical than natural appearance. Figure 4 demonstrates how Histogram Equalization outperforms learning based methods in preserving edges and text sharpness.

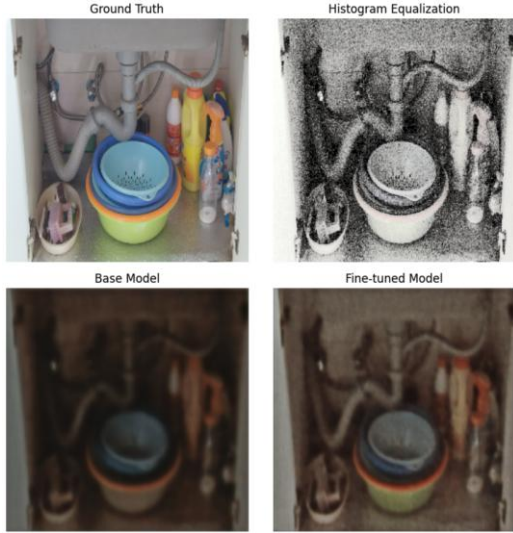


Figure 3
A Comparative 2X2 Figure that Depicts How Histogram Equalization Degrades Color Saturation



Figure 4
A Comparative 2X2 Figure that Depicts How Histogram Equalization Outperforms Learning Based Approaches in Images that Contain some Sort of Text

CONCLUSION

In conclusion, after reviewing the performance metrics for all 3 approaches, it can be said that data-driven approaches, on average, tend to perform

better at the task of low-light image Enhancement than their classical methods counterparts. Moreover, it was also confirmed that the addition of transfer learning increases model performance for models only pre-trained on partially synthetic data. The difference of SSIM between Histogram Equalization and the Fine-tuned model was 0.313, which is a significant boost in performance when taking into consideration the simplicity and size of the final model.

It is also worth noting that, although learning based methods perform better for LLIE, Histogram equalization does have an edge at the task of OCR under low light conditions due to the nature of how this classical approach enhances images to boost clarity.

Learning-based low-light image enhancement models, particularly those fine-tuned on real low-light datasets, simultaneously optimize luminance, color fidelity, and structural consistency. Fine-tuned models preserve the chromatic relationships and avoid over-amplifying noise, resulting in images that are both perceptually natural and visually appealing. The comparison between Histogram Equalization and data-driven approaches highlights the trade-offs between classical and learning-based methods: while Histogram Equalization maximizes contrast in a purely algorithmic manner, learning-based models provide a more balanced enhancement that maintains color richness, edge sharpness, and overall scene realism, making them better suited for general low-light photography.

FUTURE WORKS

Some areas of future improvement for the development of this lightweight LLIE model would be to experiment more with different model configurations and to compare and contrast different model architectures to evaluate the optimal one for the task. In addition, there is also the potential of including other performance metrics that evaluate different aspects of the model output; metrics like Learned Perception Image Patch Similarity (LPIPS), L1 Loss, RMSE Natural Image Quality

Evaluator (NIQE), and BRISQUE, among others. Moreover, the addition of images and different image transformation pipelines is also worth noting as a component that could yield different results and in theory improve performance metrics along the line of LLIE. There is also the opportunity to experiment with different numbers of epochs, model pruning, distillation, and/or quantization to reduce the model size, and even implementation on board for real-time image enhancement. Last but not least, there's also an opportunity to experiment with other loss functions to observe whether changing them could potentially improve model performance altogether.

ACKNOWLEDGMENTS

I would like to thank my Advisor, Dr. Jeffrey Duffany, for providing me with the necessary support to complete this article. I would also like to thank Alexandra Artusio-Glimpse for provisioning an Overleaf template for submitting a 2-page summary paper to the CPEM 2024 conference, following the recommended formatting to be submitted to the IEEE Xplore database. Lastly, I would like to thank the Kaggle team for provisioning tools to easily access and utilize the LOL dataset inside their Jupyter notebooks, and I would also like to thank the team behind NotebookLM for providing tools that make scientific research much easier.

REFERENCES

- [1] Soumik Rakshit. *LOL Dataset*. Kaggle. Accessed November 29, 2025. [Online]. Available: <https://www.kaggle.com/datasets/soumikrakshit/lol-dataset>.
- [2] Liang, X., Chen, X., Ren, K. et al. *Low-light image enhancement via adaptive frequency decomposition network*. *Sci Rep* 13, 14107 (2023). [Online]. Available: <https://doi.org/10.1038/s41598-023-40899-8>.
- [3] Nguyen, T. et al. (2026). *Towards a Real-Time and Compact Low-Light Image Enhancement Method Using Deep Learning on ESP32-S3-EYE*. In: Quan, T.T., Sombatheera, C., Pham, HA., Tran, N.T. (eds) *Multi-disciplinary Trends in Artificial Intelligence. MIWAI 2025. Lecture Notes in Computer Science*, vol 16355. Springer, Singapore. [Online]. Available: https://doi.org/10.1007/978-981-95-4963-4_4.

- [4] Tahia Binte Chowdhury; Md. Khaliluzzaman; Md. Jashim Uddin; Karnish Khanam; Md. Monirul Islam. *Low Light Enhancer: A Low Light Image Enhancement Model Based on U-Net using Smartphone*. 2022 International Conference on Innovations in Science, Engineering and Technology (ICISSET), pp. 26-27, Feb. 2022. Available: <https://ieeexplore.ieee.org/document/9775888>.