CS512 Computer Vision – Project Report

LYT-Net: Lightweight YUV Transformer-based Network for Low-Light Image Enhancement

Team Members

- Tamilarasee Sethuraj | A20553416
- Kavin Raj Karuppusamy Ramasamy | A20564249

Research Paper

Title	:	LYT-Net: Lightweight YUV Transformer-based Network for Low-Light Image Enhancement
Authors	:	Alexandru Brateanu, Raul Balmez, Adrian Avram, Ciprian Orhei, and Cosmin Ancuti
Year	:	2024
Link	:	https://arxiv.org/abs/2401.15204

Problem Statement

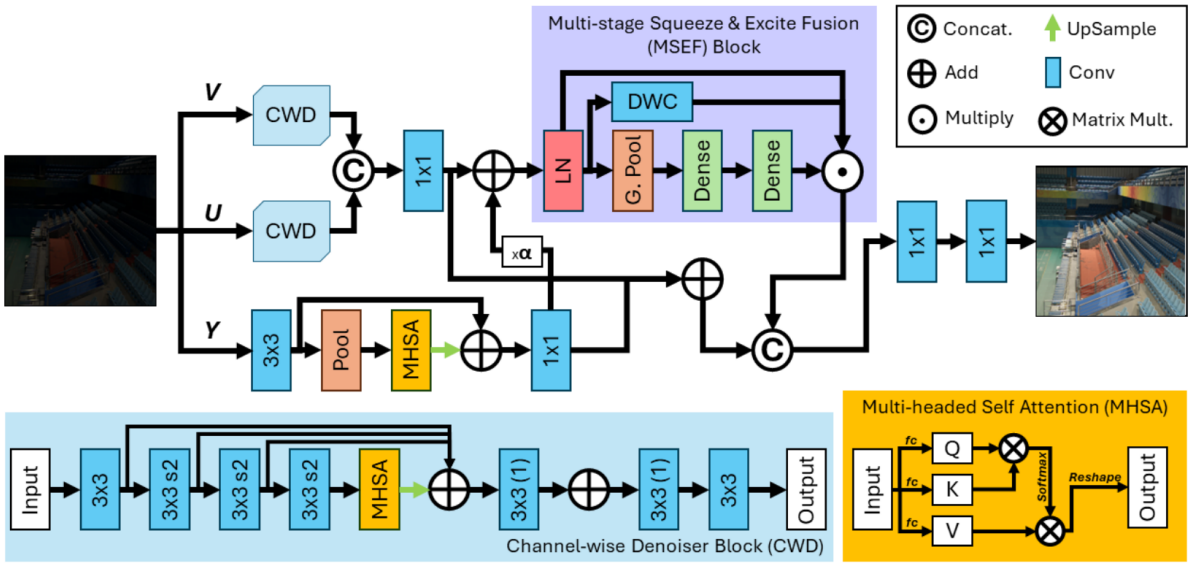
In computer vision, low-light image enhancement (LLIE) plays a crucial role in improving the quality of images captured under poor lighting conditions. These images typically suffer from multiple issues: low contrast, excessive noise, color distortion, and loss of detail. These problems not only affect image quality but also hinder the performance of downstream vision tasks such as object detection, recognition, and segmentation. While LLIE is essential for applications like surveillance, autonomous driving, and medical imaging, it presents several significant challenges. These include managing noise amplification, maintaining accurate color reproduction, and preserving fine details without introducing artifacts.

Traditional approaches like histogram equalization and Retinex-based models have limitations when dealing with complex lighting scenarios and often introduce unwanted noise or color distortions. While recent deep learning solutions, including CNN-based models like Retinex-Net and EnlightenGAN, have shown promising results, they require substantial computational resources. Similarly, transformer-based architectures like Uformer and Restormer effectively capture global image dependencies but are computationally intensive. With growing demand for real-time LLIE solutions, particularly in mobile and embedded systems, there's a clear need for lightweight models that balance performance with efficiency. Our project implements LYT-Net, an innovative lightweight transformer-based model that enhances low-light images by processing them in the YUV color space, achieving state-of-the-art results while minimizing computational overhead.

Proposed Solution

The LYT-Net architecture is a lightweight, transformer-based model designed specifically for Low-Light Image Enhancement (LLIE). It efficiently processes images in the YUV color space, where the luminance (Y) and chrominance (U and V) channels are treated separately. This approach allows the model to focus on adjusting illumination and removing noise more effectively while preserving color fidelity. LYT-Net comprises several critical components, including:

1. Channel-Wise Denoiser (CWD)
2. Multi-Headed Self-Attention (MHSA)
3. Multi-Stage Squeeze & Excite Fusion (MSEF)



The LYT-Net architecture follows a dual-path design, separating the Y (luminance) channel from the U and V (chrominance) channels. This design allows the network to handle illumination corrections and reduce noise more effectively in low-light images. The Y channel, which controls brightness, is processed with attention mechanisms for illumination adjustments. Meanwhile, the U and V channels, which carry color data, are denoised using specialized blocks.

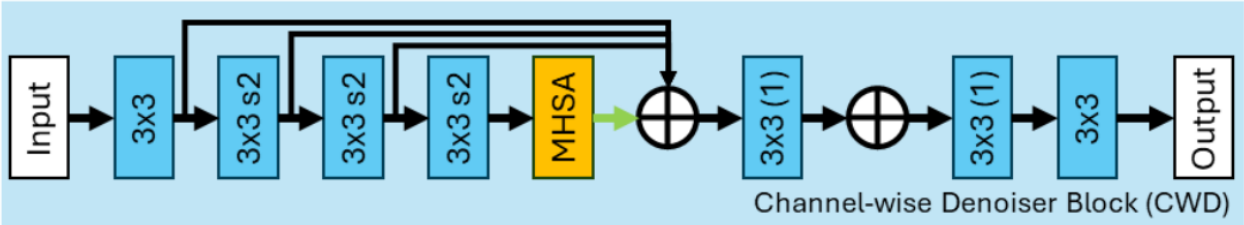
The YUV format is a color encoding system that splits an image into luminance (Y) and chrominance (U and V) components. This format is suited for efficient image processing and compression. The Y (luminance) component captures the overall light intensity, while the U and V channels represent chrominance, encoding differences between the blue and red channels,

respectively. Unlike the RGB model, where red, green, and blue channels contribute equally to both color and brightness, YUV prioritizes brightness, aligning more closely with human visual perception.

The advantage of YUV lies in its ability to compress images efficiently. Since the human eye is more sensitive to brightness than color, chrominance data can be downsampled without noticeable quality loss. In low-light image enhancement tasks, like LYT-Net, YUV helps by enabling independent processing of brightness (Y) and color (U and V). This separation allows targeted improvements to luminance while minimizing color distortions, enhancing computational efficiency.

Channel-Wise Denoiser (CWD)

The Channel-Wise Denoiser (CWD) is designed to reduce noise in the chrominance channels (U and V) while ensuring that fine image details are preserved. It employs a U-shaped architecture with multiple convolutional layers and a Multi-Headed Self-Attention (MHSA) mechanism at the bottleneck. The U-shaped structure is ideal for low-level vision tasks, as it captures multi-scale features and uses skip connections to prevent the loss of fine details during upsampling.



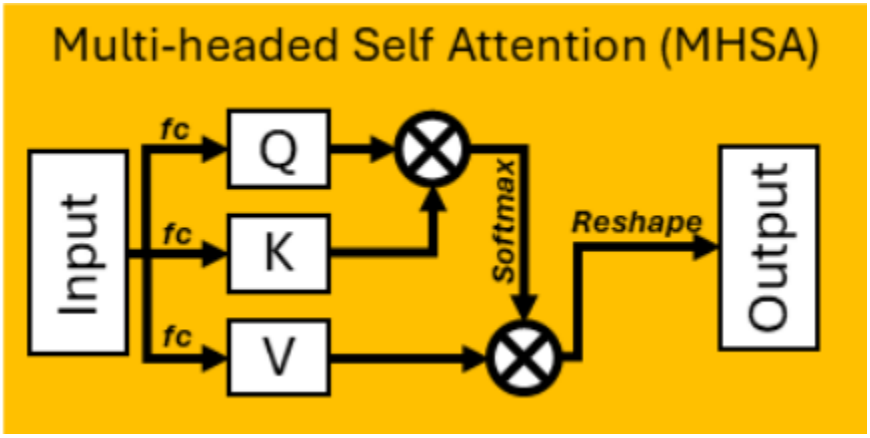
The CWD consists of four convolutional layers:

- The first layer processes features with a stride of 1 to extract initial spatial features.
- The next three layers use strides of 2 to downsample the feature map, progressively capturing features at multiple scales.
- At the bottleneck, the MHSA mechanism captures long-range dependencies within the image, enhancing its ability to focus on important regions of the chrominance channels.
- After the MHSA block, the feature map is upsampled using interpolation-based methods (instead of transposed convolutions), reducing the number of parameters while maintaining performance.

By applying the CWD block to the U and V channels, LYT-Net effectively reduces noise while retaining important color information. This separation allows for more focused noise reduction, avoiding the risk of over-processing the luminance channel, which is critical for maintaining natural lighting in the enhanced image.

Multi-Headed Self-Attention (MHSA)

The Multi-Headed Self-Attention (MHSA) block is applied to the Y (luminance) channel, which is responsible for handling contrast and illumination. The MHSA block is inspired by the Vision Transformer (ViT) architecture and is designed to capture long-range dependencies between different regions of the image. This is crucial for adjusting lighting in low-light images, where global context plays an important role in enhancing poorly lit areas.



In the MHSA block:

- The input feature map is projected into query (Q), key (K), and value (V) matrices using linear layers.
- The self-attention mechanism is applied across multiple heads, enabling the model to attend to different parts of the image simultaneously.
- The attention outputs from all heads are concatenated and passed through a final linear layer to restore the original feature size.

Mathematically, the self-attention mechanism is defined as:

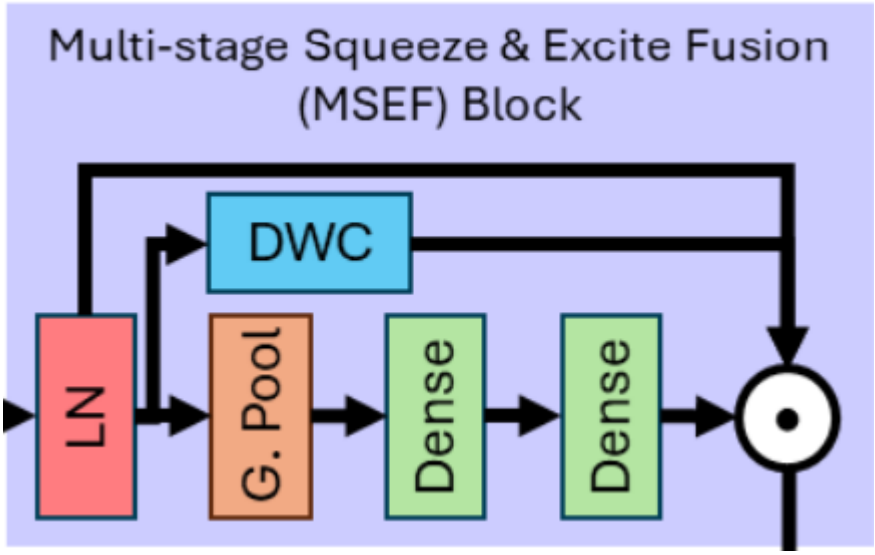
$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})\, V$$

where d_k is the dimensionality of the key vectors. The softmax operation ensures that the model focuses on the most relevant parts of the image for enhancing illumination.

Once the self-attention mechanism has been applied, the output is reshaped back to its original dimensions and passed to the next stage. The MHSA block allows LYT-Net to adjust illumination while preserving fine details, ensuring that the enhanced image looks natural without introducing artifacts.

Multi-Stage Squeeze & Excite Fusion (MSEF)

The Multi-Stage Squeeze & Excite Fusion (MSEF) block is applied after the luminance (Y) and chrominance (U, V) channels have been processed separately. The MSEF block enhances both spatial and channel-wise features by employing a combination of global average pooling and excitation mechanisms.



The squeeze operation compresses the feature map into a reduced set of descriptors that capture global context. This operation is followed by an excitation step, which re-expands the descriptors to their original dimensions while selectively emphasizing the most important features. This process helps the network focus on the most relevant spatial and channel-wise details, improving the overall quality of the enhanced image.

The MSEF block works as follows:

- The input feature map undergoes layer normalization to stabilize the learning process.
- Global average pooling is applied to capture the global spatial context.
- The pooled features are passed through fully connected layers with ReLU and Tanh activations, which produce a descriptor that emphasizes important features.
- The feature map is then re-expanded to its original dimensions and recombined with the input feature map using a residual connection.

This approach allows the model to enhance both local and global features, ensuring that the final image is not only well-lit but also retains important structural and color details.

Hybrid Loss Function

To train the LYT-Net model, a hybrid loss function is employed, combining several loss components to ensure that the enhanced images are both perceptually pleasing and quantitatively accurate. The hybrid loss function is defined as:

$$L = L_S + \alpha_1 L_{prec} + \alpha_2 L_{hist} + \alpha_3 L_{psnr} + \alpha_4 L_{color} + \alpha_5 L_{MS-SSIM}$$

Where:

- L_S : Smooth L1 loss, which helps minimize pixel-wise errors between the target and predicted images.
- L_{prec} : Perceptual loss, which ensures that high-level features extracted by a pre-trained VGG19 network are similar between the target and predicted images.
- L_{hist} : Histogram loss, which aligns the pixel intensity distributions of the enhanced and target images, ensuring proper contrast.
- L_{psnr} : PSNR loss, which penalizes reconstruction errors based on the Peak Signal-to-Noise Ratio (PSNR).
- L_{color} : Color loss, which ensures that the color balance of the enhanced image matches that of the target.
- $L_{MS-SSIM}$: Multi-Scale SSIM loss, which evaluates structural similarity across multiple scales, ensuring that both local and global structures are preserved.

This hybrid loss function ensures that the model optimizes various aspects of image quality, leading to enhanced images that are visually pleasing and maintain high structural and color fidelity.

Dataset:

For this project, we will use the LOw-Light (LOL) paired dataset for training and evaluation, as described in *Deep Retinex Decomposition for Low-Light Enhancement by Chen Wei, Wenjing Wang, Wenhan Yang and Jiaying Liu (BMVC, 2018)*. Our implementation utilizes both LOL-v1 and LOL-v2 datasets.

The LOL-v1 dataset provides a foundational set of paired low-light and normal-light images, comprising 485 training pairs and 15 testing pairs. While relatively small, this dataset offers a controlled environment for evaluating enhancement techniques.

LOL-v2 expands upon its predecessor with two distinct subsets:

- LOL-v2-real: Contains 689 training pairs and 100 testing pairs captured in real-world low-light conditions
- LOL-v2-synthetic: Includes 900 training pairs and 100 testing pairs generated synthetically to simulate various low-light scenarios

This comprehensive dataset structure allows us to evaluate LYT-Net's performance across both natural and synthetic low-light conditions, providing a robust assessment of the model's capabilities.

Dataset Google Drive link:

- LOLv1 - https://drive.google.com/file/d/1vhJg75hIpYvsmryyaxdygAWeHuiY_HWu/view
- LOLv2 - https://drive.google.com/file/d/1OMfP6Ks2QKJcru1wS2eP629PgvKqF2Tw/view

Dataset Source: https://daooshee.github.io/BMVC2018website

Implementation:

The LYT-Net model was trained using image pairs from the Low-Light (LOL) dataset, each training dataset was further split into 80% for training and 20% for validation. All images were resized to a uniform size of 256x256 pixels. The RGB channels were combined to form an input of size 256x256x3. The pixel values were normalized to the range 0-1 by dividing by 255. This normalization ensures the input data is in a consistent format, preparing it for model training.

The input images are converted from RGB to YUV color space for processing. The Y channel (luminance) handles brightness, while the U and V channels (chrominance) manage color information. This separation allows the model to more effectively adjust brightness and reduce noise.

The Y channel is processed using the Multi-Headed Self-Attention (MHSA) block, which focuses on enhancing spatial features. The MHSA block splits the input into multiple heads, applies self-attention, and then concatenates the results. This mechanism allows the model to attend to different parts of the image simultaneously, improving spatial feature representation. The MHSA block uses an embedding size of 32 and four heads which was found to be the best fit based on the trials of different values. The attention mechanism is applied using scaled dot-product attention.

The U and V channels are processed using the Convolutional Weight Denoising (CWD) block. This block applies a series of convolutional layers followed by upsampling to enhance and denoise the chrominance channels. The CWD block consists of four consecutive 3x3 convolutional layers, each with 16 filters and a stride of 2 for downsampling. After the MHSA block, the output is upsampled using bilinear interpolation to match the original size of the U and V channels.

The processed Y, U, and V channels are then combined. The Multi-Stage Squeeze & Excite Fusion (MSEF) block is applied to enhance both spatial and channel-wise features. The MSEF block uses layer normalization, depthwise convolution, global average pooling, and fully connected layers to emphasize the most relevant features. The MSEF block uses a 3x3 depthwise convolution, global average pooling, and two fully connected layers.

In summary, architecture includes the following steps:

- Convert the input RGB image to YUV color space.
- Process the Y channel using the MHSA block.
- Process the U and V channels using the CWD block.
- Combine the processed Y, U, and V channels.
- Apply the MSEF block to the combined channels.
- Convert the enhanced YUV image back to RGB color space.

The model was trained using the Adam optimizer with parameters beta1 = 0.9 and beta2 = 0.999. The initial learning rate was set to 2e-4. PSNR and SSIM were the primary performance metrics used for evaluation. Early stopping was implemented to halt training if the validation loss did not improve for 10 consecutive epochs. The model with the best validation loss was saved and used for testing.

After training, the model was evaluated on the test dataset, and the average PSNR and SSIM values were calculated. The results were visualized using matplotlib, showing side-by-side comparisons of the input, predicted, and ground truth images. These visual comparisons helped assess the model's effectiveness in enhancing low-light images.

The LYT-Net model effectively enhances low-light images by processing the Y, U, and V channels separately before combining them for final output. This modular design, incorporating advanced techniques like MHSA and MSEF, allows the model to efficiently improve brightness, reduce noise, and maintain color accuracy. The model was trained and evaluated on three distinct datasets (LOLv1, LOLv2 Real, and LOLv2 Synthetic), and the results demonstrate its superior performance in enhancing low-light images.

For comparison and visualization, each dataset was run in a separate notebook, allowing for easy comparison and output visualization. Training on the LOLv1 dataset, for example, took approximately 38 minutes per epoch. This version is now clear, concise, and free of redundancy. It coherently presents the implementation details while maintaining a focus on the core aspects of the model and its training process.

How to Run the Program

Our code is written in Python notebook files to make it easy to read and understand. We recommend using a GPU for training the model, as it significantly reduces training time. We ran the code on our local machine because Google Colab sessions sometimes disconnect, which can interrupt the training process.

Steps to Run the Program:

1. Download the Project Folder

   First, download the project folder from the Bitbucket repository. Ensure that Python and Jupyter Notebook are installed on your local machine.

2. Setting Up the Environment

   The necessary Python packages are listed in the requirements.txt file located in the data folder. To install these packages, run the following command in your terminal:

   ```
   pip install -r data/requirements.txt
   ```

3. Download and Prepare the Dataset

   Download the dataset from Google Drive links and extract the files into the data folder.

4. Running the Model

   There will be three notebooks in the src folder for each dataset (LOLv1, LOLv2Real and LOLv2Synthethic).Open one of the notebooks and execute the cells in sequence. The notebook will:

   - Load the dataset from the appropriate data folder.
   - Train the model or load pre-trained weights from Google drive - https://drive.google.com/drive/folders/1ft_vHN3L4KYns_AO2OXn4aE6-LAOBQgW?usp=sharing. For loading the pre-trained weights follow this Save and load Keras models tutorial.

5. Visualizing Results

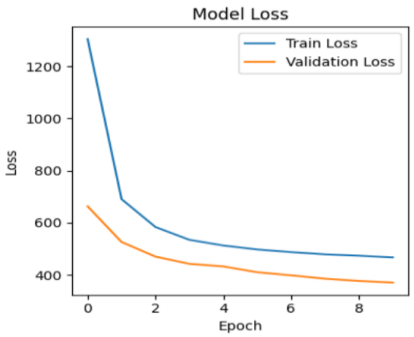
   Once the model has been trained or predictions have been made, the results will be displayed using matplotlib. You will see side-by-side comparisons of the input image, the predicted enhanced image, and the ground truth image (if available).

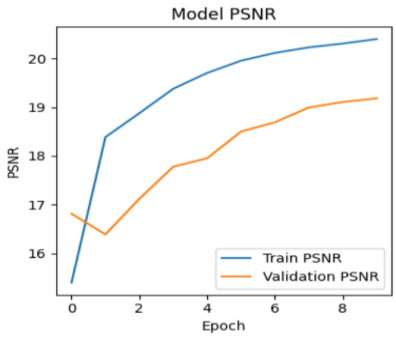
Results

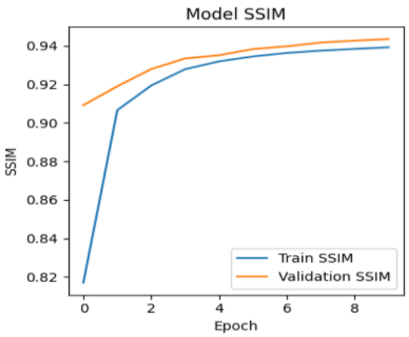
To evaluate the results gathered from the experiment, it's important to have a quantitative evaluation metric. We have used PSNR and SSIM for validating the level of model performance in different datasets.

As we can see from below graphs, the loss value has decreased over the epoch and the PSNR and SSIM values continued to increase with each epoch. The higher values of validation PSNR and SSIM values shows that the model has learnt to generalize the prediction.

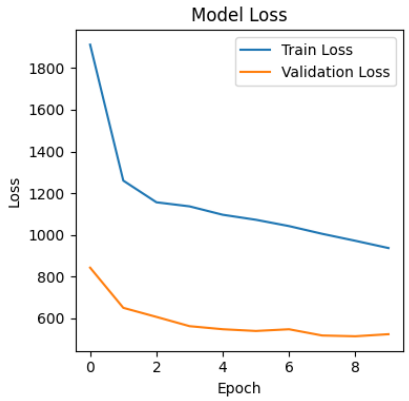
LOLv1

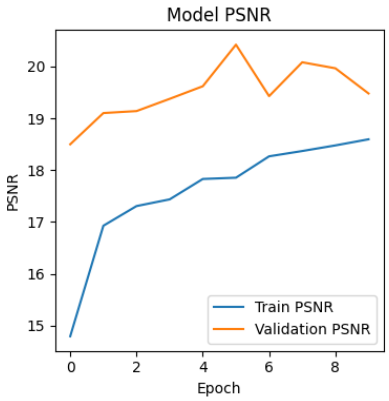


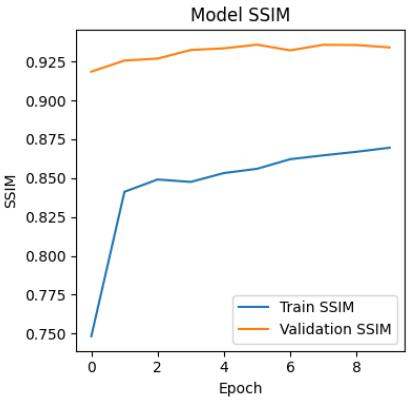


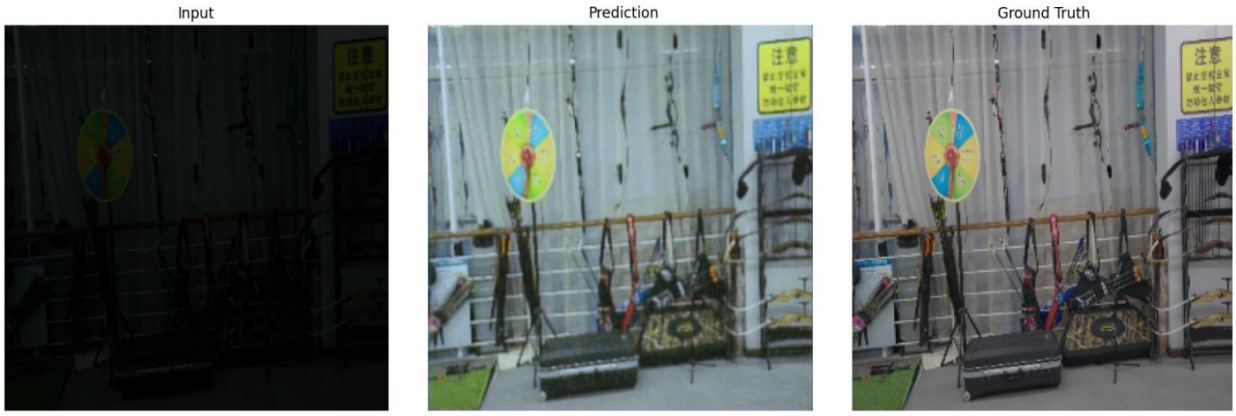


LOLv2 Real

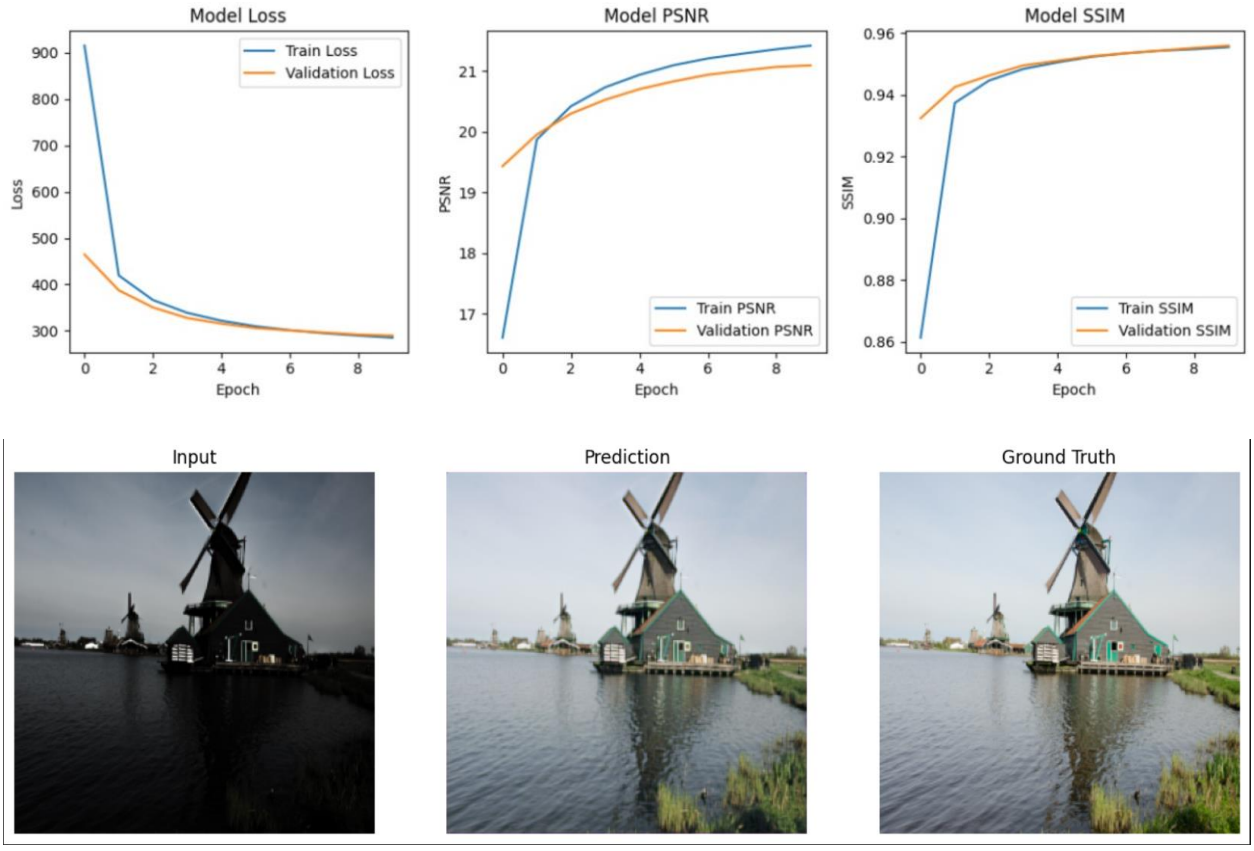








LOLv2 Synthetic



As we can see from above result images, the predicted images are close to the original image in terms of clarity and contrast of the images.

The below table shows the comparison between the original research paper outputs and our implementation results across the datasets.

Model	Epoch	Param (M)	LOLv1		LOLv2 - Real		LOLv2 - Synthetic	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Paper	1000	0.045	27.23	0.853	27.8	0.873	29.38	0.94
Ours	10	0.054	19.69	0.785	18.08	0.781	21.1	0.868

Conclusion

In this project, we successfully implemented LYT-Net, a lightweight transformer-based network for low-light image enhancement. By leveraging the YUV color space and separating luminance and chrominance channels, the model efficiently enhances brightness while reducing noise and preserving color fidelity. The use of Multi-Headed Self-Attention (MHSA) and Channel-Wise Denoiser (CWD) blocks allows for effective feature extraction and noise reduction, while the Multi-Stage Squeeze & Excite Fusion (MSEF) block enhances both spatial and channel-wise features.

The model was trained and evaluated on three versions of the LOL dataset: LOLv1, LOLv2 Real, and LOLv2 Synthetic. The results show that LYT-Net achieves competitive performance, with high PSNR and SSIM values, while maintaining low computational complexity. This makes it a suitable solution for real-time applications in resource-constrained environments, demonstrating its potential for practical use in low-light image enhancement tasks.

Team Member Responsibilities (in %)

Task	Tamilarasee	Kavin
Data Preprocessing and Augmentation	50	50
Model Implementation	50	50
Evaluation and Analysis	50	50
Experimentation and Improvement	50	50
Report Writing	50	50
Presentation Preparation	50	50

References:

- C. Wei, W. Wang, W. Yang, and J. Liu, “Deep retinex decomposition for low-light enhancement,” in Proceedings of the British Machine Vision Conference (BMVC), 2018
- S. Park, S.Yu, B.Moon, S.Ko, and J. Paik, “Low-light image enhancement using variational optimization-based retinex model,” IEEE Transactions on Consumer Electronics, vol. 63(2), 2017.
- Shansi Zhang, Nan Meng and Edmund Y. Lam, “LRT: An Efficient Low-Light Restoration Transformer for Dark Light Field Images,” IEEE Transactions on Image Processing, vol. 32, 2023.
- Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, “Residual dense network for image restoration,” In IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.
- Y. Jiang, X. Gong, D. Liu, Y. Cheng, C. Fang, X. Shen, J. Yang, P. Zhou, and Z. Wang, “Enlightengan: Deep light enhancement without paired supervision,” IEEE Transactions on Image Processing, vol. 30, pp. 2340–2349, 2021.