**AIP Final Project Proposal**

**Group Members: Arnab Roy (25563) and Abhay Agarwal (24249)**

**Zero-Reference Deep Curve Estimation for Low-Light Image Enhancement**

**1. Introduction**

Low-light image enhancement is essential in computer vision for improving visibility in challenging lighting conditions. Traditional methods struggle with overexposure and noise, while supervised deep learning approaches require paired data, limiting real-world applicability. This project explores **Zero-Reference Deep Curve Estimation (Zero-DCE)**, which formulates enhancement as a curve estimation problem, learning to adjust brightness adaptively without reference images. In this project we will implement and extend the paper “***Zero-Reference Deep Curve Estimation for Low-Light Image Enhancement by Guo et al. (2020)”***.

**2. Objectives**

* Identify and address limitations of Zero-DCE, including noise amplification and loss of fine details.
* Integrate advanced noise suppression by modifying the existing loss function.
* Enhance output by trying different backbone architectures.

**3. Methodology**

**3.1 Dataset Preparation**

* We’ll use *LOL dataset (*[*https://www.kaggle.com/datasets/soumikrakshit/lol-dataset*](https://www.kaggle.com/datasets/soumikrakshit/lol-dataset)*)* first to train the model.
* The dataset contains 485 low light images for training, 15 low light images for validation along with ground truth.
* In LOL dataset there are very few samples where the main object in the image is well illuminated and the background is dark. So, model performance will be checked for such examples randomly chosen from *ExDark dataset* (<https://github.com/cs-chan/Exclusively-Dark-Image-Dataset>).
* If the performance of the model on such images is found unsatisfactory then model will be re-trained with both LOL dataset and a subset of ExDark dataset.

**3.2 Implementation with extension Details**

Following experiments will be done -

* Implement the Zero-DCE paper by using the model architecture and loss functions proposed, and check the performance of the model, especially PSNR and SSIM with ground truth on the evaluation set.
* Use a different architecture (e.g., U-Net) and same loss functions to train the model to check whether there is any improvement in the performance.
* Try additional loss functions (e.g., total variation loss, structural similarity loss) to improve model performance.
* Add classical image processing methods (e.g., histogram equalization) at downstream pre-processing pipeline.

**4. Expected Outcomes**

* Improved low-light image enhancement with better noise suppression and detail preservation.
* Higher perceptual quality measured using SSIM and other evaluation metrics.