PROJECT REPORT

LOW LIGHT IMAGE ENHANCEMENT

Submitted By: Mayur Rajpal

EE 3rd Year, 22115093

Introduction

Low-light image enhancement is a crucial subfield of computer vision focused on improving image clarity captured in dim lighting. Conventional techniques often demand reference images or manual adjustments, which are not always practical. The Zero-DCE (Zero-Reference Deep Curve Estimation) framework offers a groundbreaking solution by enhancing low-light images without needing reference images or extensive preprocessing. This framework's main advantage is its ability to function without paired low-light and normal-light images, making it highly versatile. Additionally, Zero-DCE is lightweight, making it ideal for devices with limited computational power, such as smartphones. It achieves fine-tuned enhancement by estimating pixel-wise and high-order tonal curves from the input image. By iteratively refining these curves, Zero-DCE effectively manages various lighting conditions within a single frame.

PSNR : 

Architecture

The DCE network employs a series of convolutional layers interspersed with concatenation layers, enabling the integration of fine-grained and high-level features. The ReLU activation functions used in the intermediate layers assist in learning complex patterns.

Here is the process breakdown:

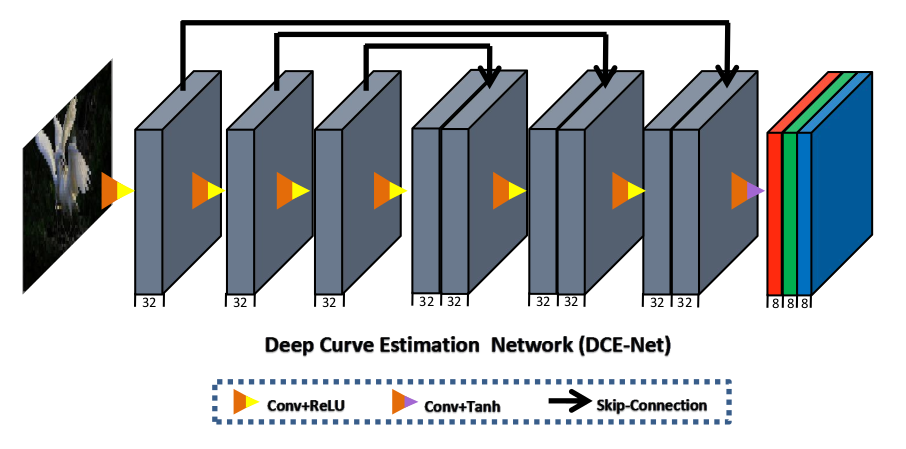
1. **Input Layer:** Receives a low-light image needing enhancement.
2. **Feature Extraction with Convolutional Layers:** Analyzes the image at a pixel level to detect elements such as edges and textures.
3. **Curve Estimation Layers:** Uses tonal curves to adjust the brightness and contrast of each pixel.
4. **Integration Layer:** Merges outputs from the curve estimation layers to form a comprehensive tonal curve map for the entire image, guiding subsequent pixel intensity adjustments.
5. **Pixel-wise Adjustment:** Applies the estimated curves to each pixel, customizing brightness and contrast enhancements based on individual pixel requirements.
6. **Output Layer:** Generates an image with improved brightness and contrast while preserving the integrity of the original content.

**Architecture Details**:

* **First Six Layers**: These layers are convolutional with 32 filters, a 3x3 kernel size, and a stride of 1. They are followed by ReLU activation functions.
* **Final Layer**: The final convolutional layer has a variable number of filters, calculated as the number of iterations multiplied by three. For example, if the iteration count is set to eight, it results in 24 curve parameter maps, as each iteration produces three parameter maps for the RGB channels.

**Training Methodology**:

* **No Paired Data Required**: Unlike traditional methods, Zero-DCE does not require paired input/output image data for training. Instead, it uses non-reference loss functions that evaluate the enhancement quality of the images.
* **Non-Reference Loss Functions**: These functions are designed to be differentiable and can assess the quality of the enhanced image without needing reference images. They effectively guide the network's learning process, ensuring the output images are optimally enhanced.



**Exposure Loss**

This function ensures the enhanced image has appropriate overall brightness. It calculates the loss based on deviations of the mean brightness of non-overlapping patches from a target value EEE. Initially, the RGB channels are averaged to get the overall brightness of each pixel. The loss is then computed by applying average pooling over patches and comparing their brightness to EEE:

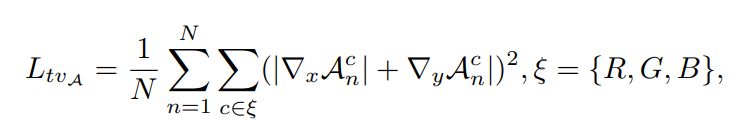
A mathematical equation with numbers and symbols

Description automatically generated with medium confidence

Here, Y is the average intensity value of a local region in the enhanced image. Setting E to 0.6, this function keeps the image’s brightness level within a desirable range.

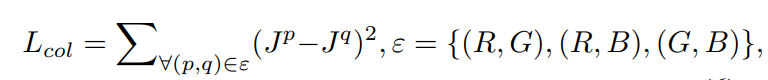
**Illumination Smoothness Loss**

This function promotes smooth and gradual changes in brightness across the image, avoiding abrupt transitions. It calculates the differences in brightness between neighboring pixels vertically (∇y) and horizontally (∇x), then averages these differences over all pixels and images in the batch:

 Where N is the number of pixels. This results in more natural-looking enhanced images by ensuring smooth illumination transitions

**Color Constancy Loss**

The goal of this loss function is to maintain a balanced color representation in the enhanced image. It computes the loss based on the mean color values of the red, green, and blue channels, penalizing deviations from ideal color balance. The function calculates the squared differences between the mean values of each pair of channels (p and q), and then returns the square root of the sum of these differences:



Where Jp denotes the average intensity value of channel p in the enhanced image. This ensures the loss is differentiable, aiding in gradient-based optimization.

**Spatial Consistency Loss**

This loss function focuses on maintaining the spatial coherence of feature maps across consecutive training epochs. It ensures that the average intensity values of local regions in the enhanced image (denoted as Y) are consistent with those in the input image (denoted as I). By preserving the relative differences between neighboring regions, this loss function helps enhance the image without losing spatial consistency:

A black and white math equation

Description automatically generatedThis formula sums the squared differences between average intensities of corresponding regions in the enhanced and original images, promoting spatial coherence.

Findings

* Zero-Reference Deep Curve Estimation (Zero-DCE) enhances low-light images by generating high-order tonal curves specifically for each image.
* The method significantly improves the quality, color correction, and overall accuracy of low-light images
* Zero-DCE formulates the enhancement task as image-specific curve estimation, addressing the unique lighting conditions of each image
* Effective for various low-light scenarios without the need for paired training data, making it versatile and efficient

Methods to Improve

* Incorporate a diverse set of low-light conditions in the training dataset to improve the robustness of the model.
* Adjust hyperparameters and layers to capture subtle lighting nuances better.
* Integrate a feedback mechanism where users can adjust results in real-time, helping the model learn preferred enhancements.
* Apply contrast enhancement methods post-enhancement to further refine the visual quality

Reference

* Research paper : <https://arxiv.org/pdf/2103.00860>
* [Super Resolution explained — Ep1. Super Resolution of images has been a… | by Manas Satish Bedmutha | Medium](https://medium.com/@manasbedmutha98/super-resolution-explained-ep1-a08cf287051e)