PixelMind

CYBER SAVANTS

Rahul Maurya

Problem:

To create a **model or algorithm** that can automatically perform **colour enhancement** on raw images, producing results similar to those achieved by professionals using Adobe Lightroom and/or Adobe Photoshop. The goal is to develop a solution that is **efficient**, **accurate**, and **capable of generalising** to a wide range of image types and scenarios

Data:

The training image dataset provided consists of pairs of images: 'before' images, which are raw images straight out of a DSLR camera, and 'after' images, which are the same images colour-corrected by a professional using Adobe Lightroom and/or Adobe Photoshop. The test image dataset contains only 'before' images, and the model's performance will be evaluated based on how well it can enhance the colours in these images.

The training dataset has been categorised into the following categories:

indoors
white background
outdoors

Approach to solution:

1. **RCNN** – Our first choice was Region-based Convolutional Neural Networks. RCNN specialise in object detection, hence not an ideal choice for image enhancement tasks.

2. **ViT** – Transformers use the attention mechanism and are best suited for NLP tasks, Vision Transformers, ViT based on the same architecture do not perform as well as CNN on tasks involving spatial data like image processing.

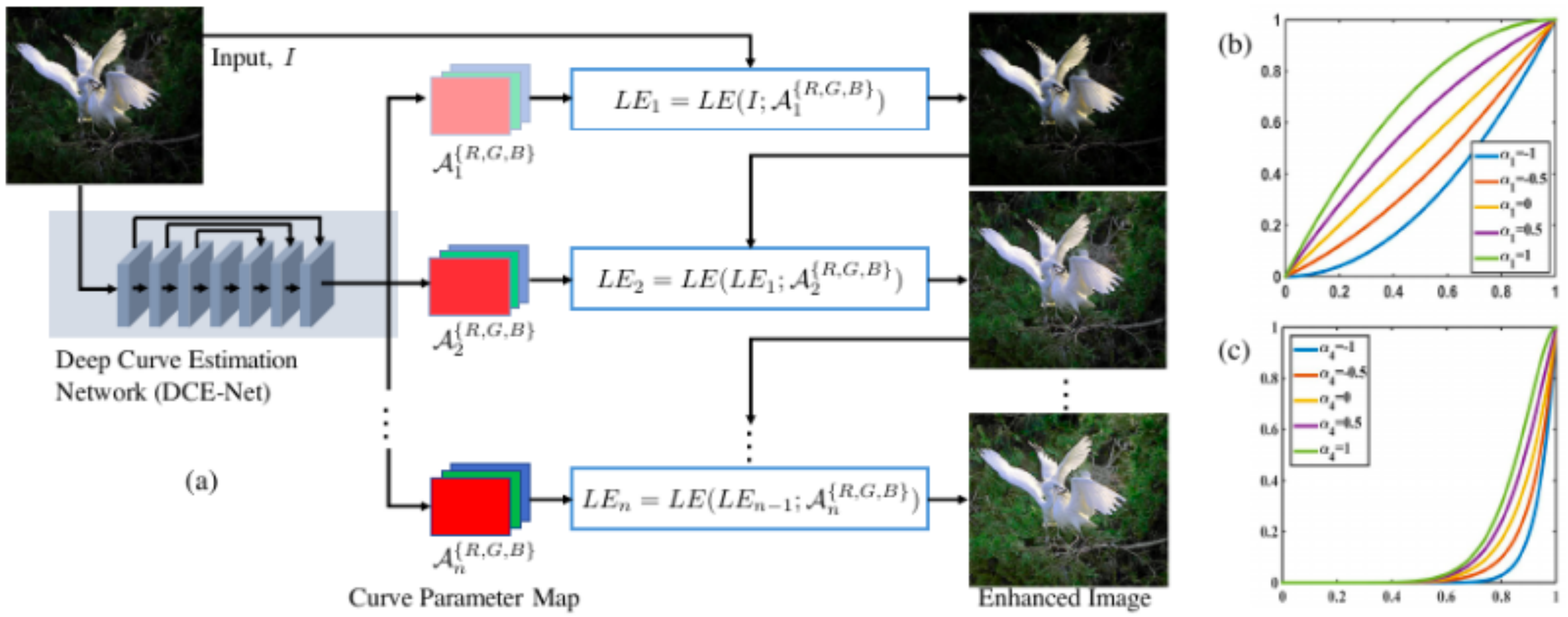
3. **Zero-DCE** – Zero-Reference Deep Curve Estimation formulates low-light image enhancement as the task of estimating an image-specific tonal curve with a deep neural network. In this example, we train a lightweight deep network, DCE-Net, to estimate pixel-wise and high-order tonal curves for dynamic range adjustment of a given image. Based on the CNN architecture, Zero-DCE is the best model for our use case.

Zero-DCE:

Zero-DCE takes a **low-light image as input** and produces **high-order tonal curves** as its **output**. These curves are then used for pixel-wise adjustment on the dynamic range of the input to obtain an enhanced image. The curve estimation process is done in such a way that it maintains the range of the enhanced image and preserves the contrast of neighbouring pixels. This curve estimation is **inspired by** curves adjustment used in photo editing software such as **Adobe Photoshop** where users can adjust points throughout an **image's tonal range.**

Zero-DCE is appealing because of its relaxed assumptions with regard to reference images: it **does not require any input/output image pairs during training**. This is achieved through a set of carefully formulated non-reference loss functions, which implicitly measure the enhancement quality and guide the training of the network.

Pipeline:



The pipeline of our method. (a) The framework of Zero-DCE. A DCE-Net is devised to estimate a set of best-fitting Light-Enhancement curves (LE-curves: LE(I(x);α)=I(x)+αI(x)(1-I(x))) to iteratively enhance a given input image. (b, c) LE-curves with different adjustment parameters *α* and numbers of iteration *n*. In (c), α_1, α_2, and α_3 are equal to -1 while *n* is equal to 4. In each subfigure, the horizontal axis represents the input pixel values while the vertical axis represents the output pixel values.

Source – https://li-chongyi.github.io/Proj_Zero-DCE.html

Highlights:

1. The first low-light enhancement network that is **independent of paired and unpaired training data**, thus avoiding the risk of overfitting. As a result, our method generalizes well to various lighting conditions.

2. An image-specific curve that is able to **approximate pixel-wise and higher-order curves** by iteratively applying itself. Such image-specific curve can effectively perform mapping within a wide dynamic range.

3. The potential of training a deep image enhancement model in the absence of reference images through task-specific non-reference loss functions that indirectly evaluate enhancement quality. It is capable of processing images in real-time (about 500 FPS for images of size 640*480*3 on GPU) and takes only **30 minutes for training**.

Source - https://li-chongyi.github.io/Proj_Zero-DCE.html

DCE-Net Architecture:

1. The **DCE-Net** is a lightweight deep neural network that learns the mapping between an input image and its **best-fitting curve parameter maps**.

2. The input to the DCE-Net is a low-light image while the outputs are a set of pixel-wise curve parameter maps for corresponding higher-order curves.

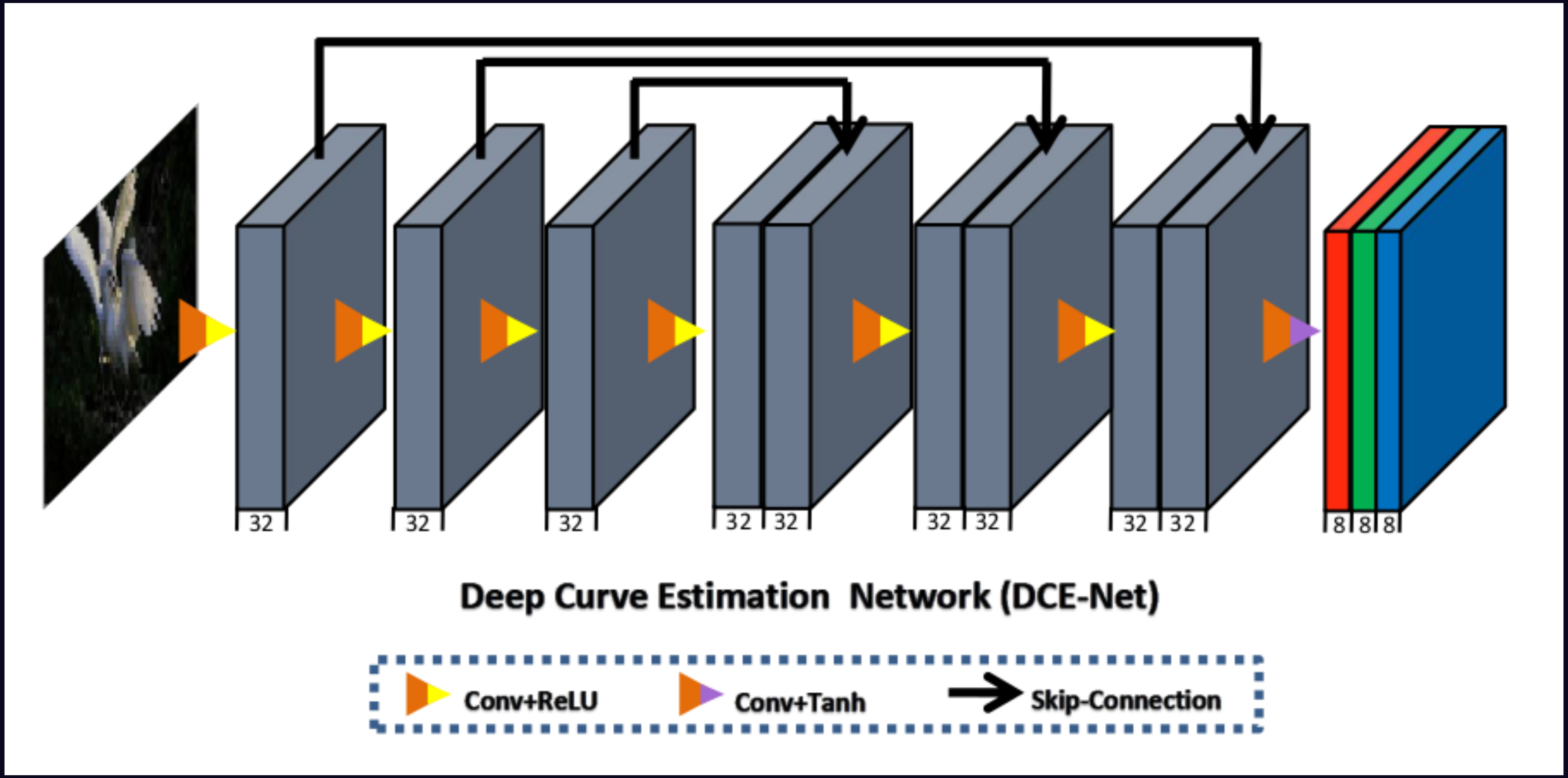
3. It is a **plain CNN** of **seven convolutional layers** with symmetrical concatenation.

4. Each layer consists of **32 convolutional kernels** of **size 3x3** and **stride 1** followed by the ReLU activation function.

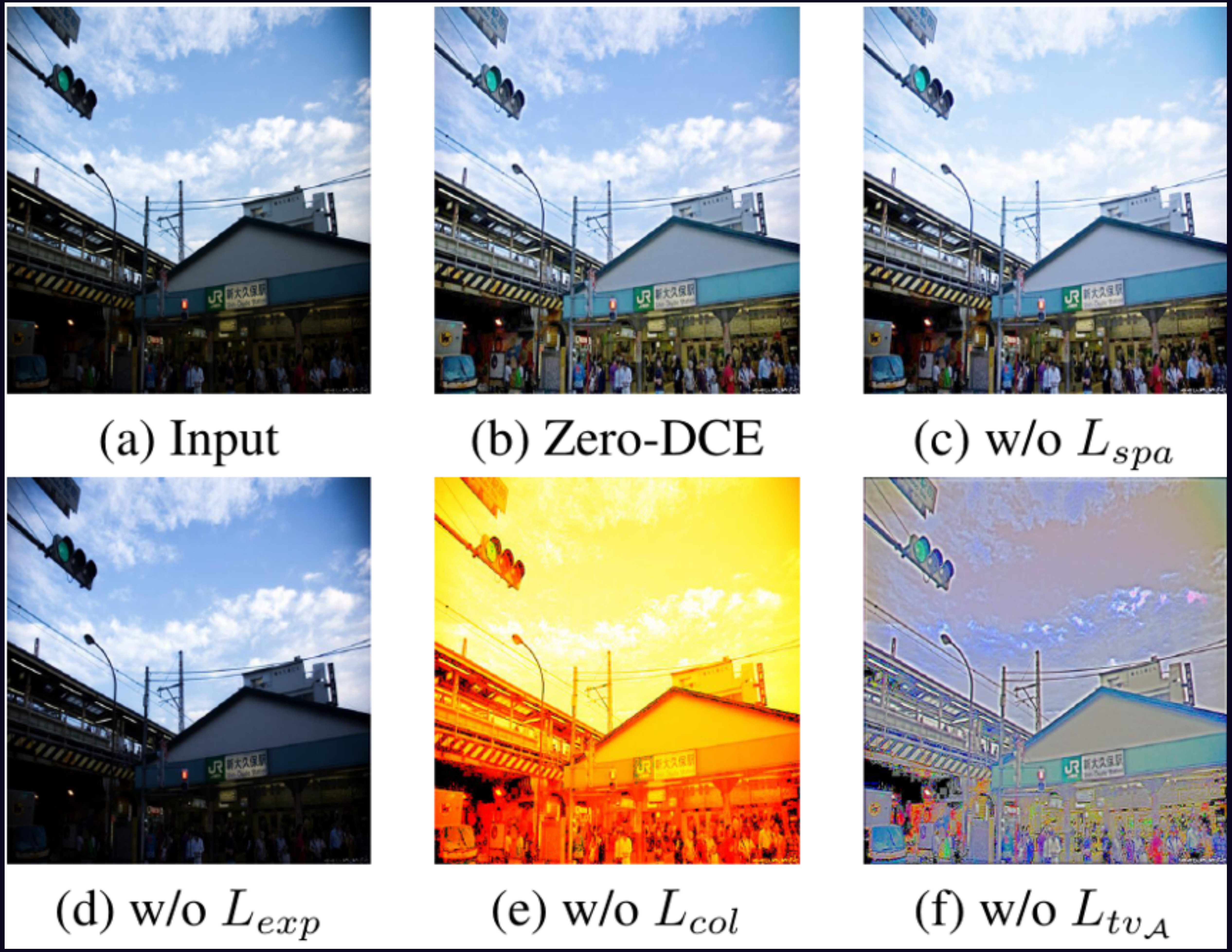
5. The last convolutional layer is followed by the Tanh activation function, which produces 24 parameter maps for 8 iterations, where each iteration requires three curve parameter maps for the three channels.

Source - https://li-chongyi.github.io/Proj_Zero-DCE.html

DCE-Net Architecture:

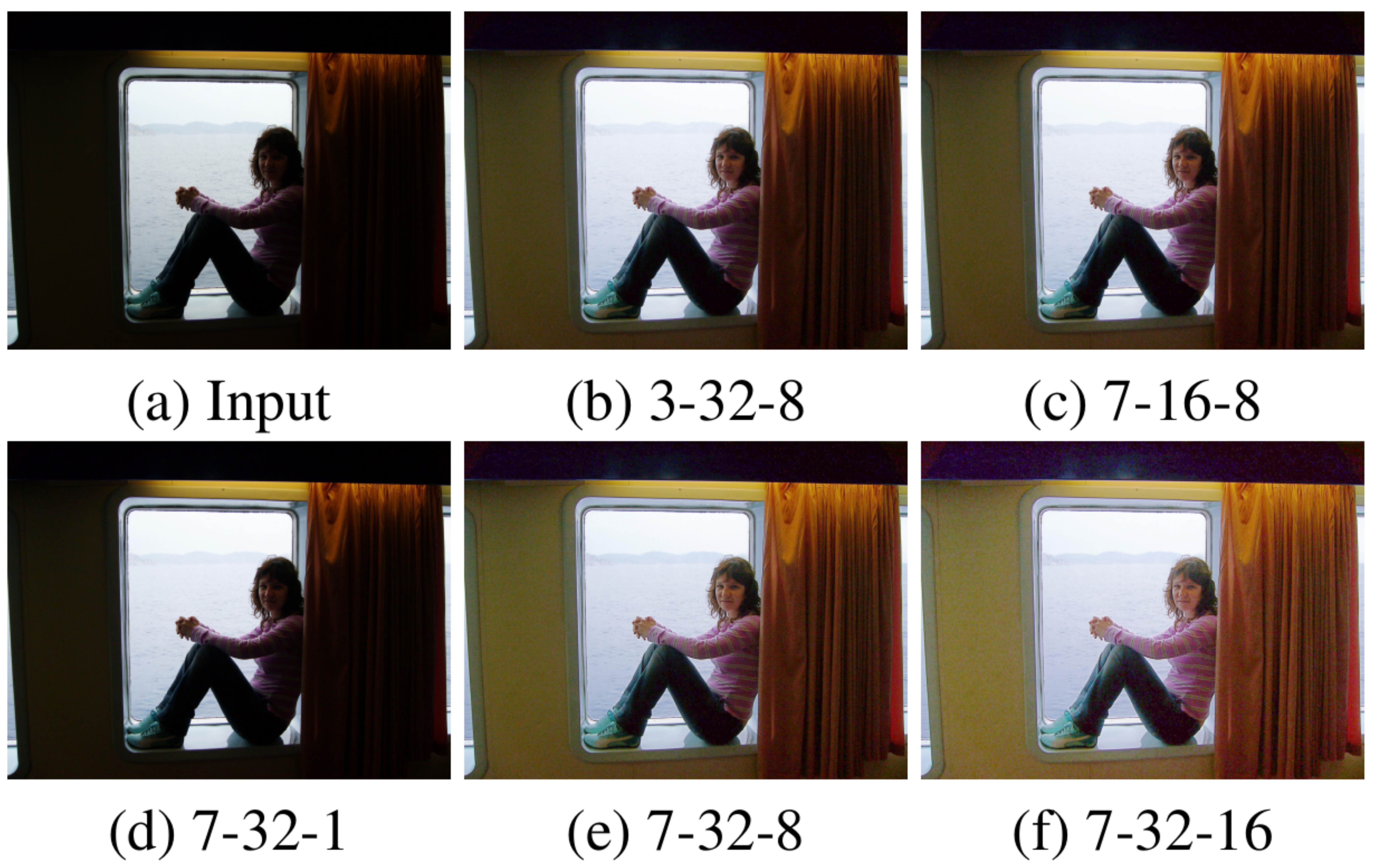


Results:



Source – https://li-chongyi.github.io/Proj_Zero-DCE.html

Results:



Source – https://li-chongyi.github.io/Proj_Zero-DCE.html

RhythmFit

Team:

Rahul Maurya

Team Lead

IITM Roll Number: 23f1002653
Foundational Level

Parth Tiwari

ML

IITM Roll Number: 22f3002726
Foundational Level

Harikesh Shukla

Development

IITM Roll Number: 22ds2000099
Foundational Level