Low-Light Image Enhancement

By

Shiv Kumar Ganesh

Kumuda B G Murthy

Raghava D Urs

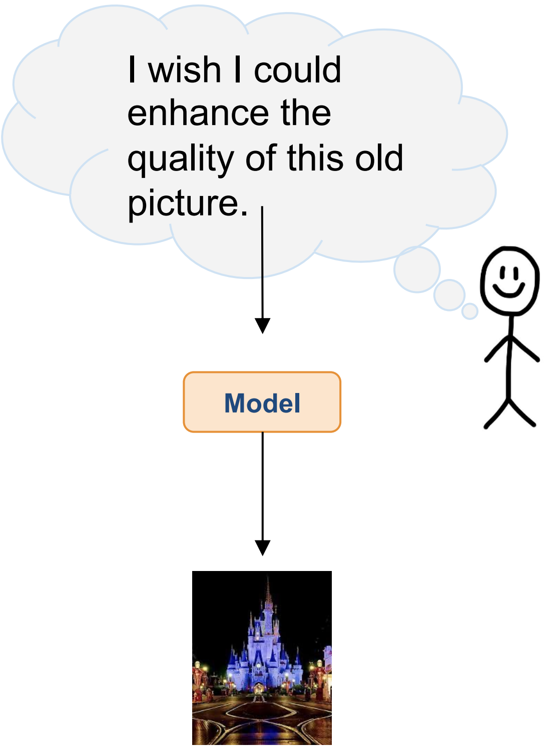
Arun Talkad

Goal

There could be multiple instances where we end up taking pictures in low or bad lighting. We would love to have these images enhanced to have better contrast and visibility.

This research introduces a new method called Zero-Reference Low Light Enhancement, which treats light enhancement as a deep network task of image-specific curve estimation.

Our method uses DCE-Net, a lightweight deep network, to predict pixel-wise and high-order curves for image dynamic range modification. Zero-DCE is intriguing because it makes no assumptions about reference images during training, i.e. it doesn't require any paired or unpaired data.



Data Collection

Source : LOL Dataset
Data Size : 485 images
Data Format : PNG images

Original Image

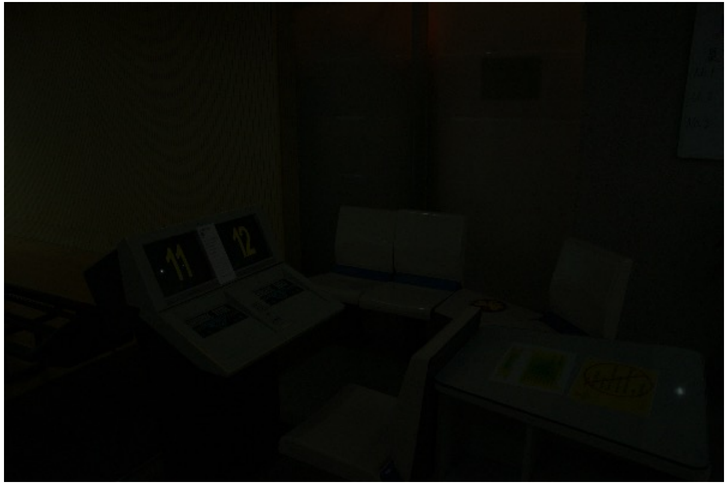


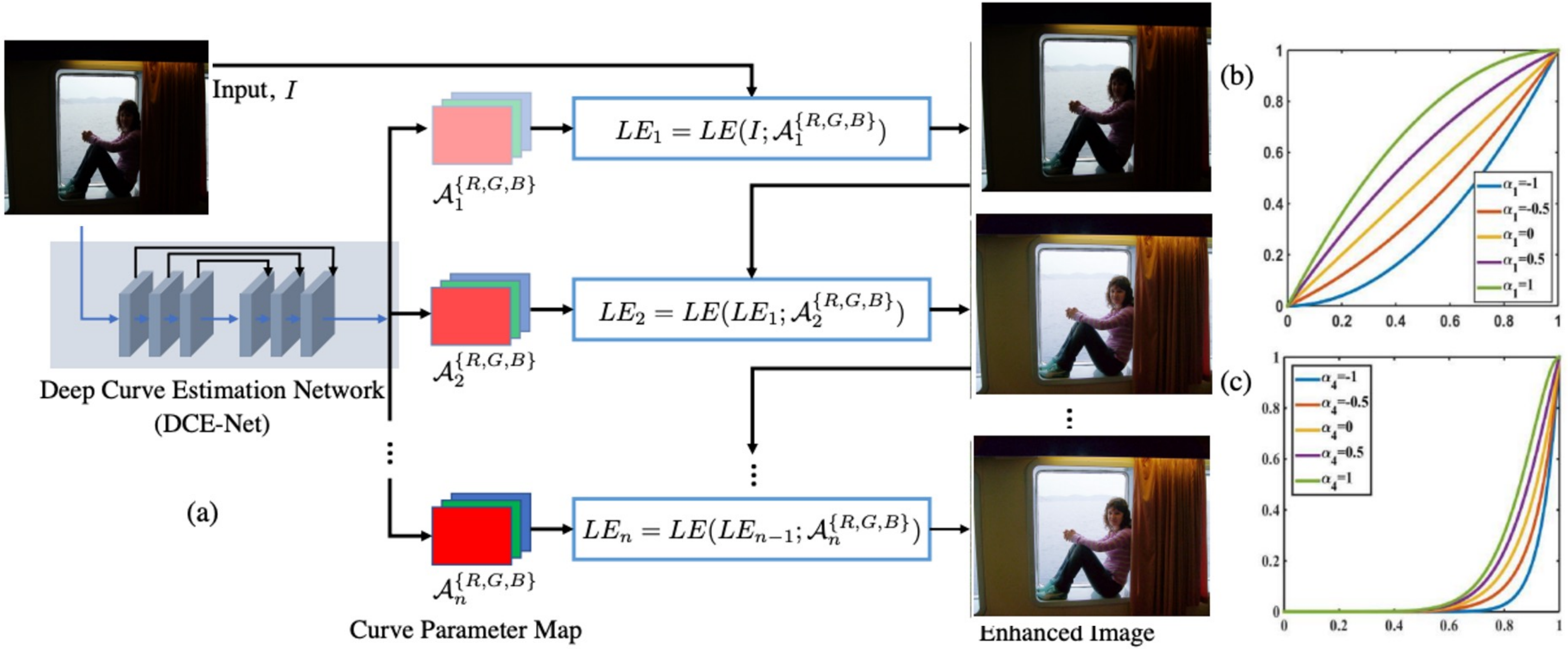
Image Pre-processing

- High quality images were considered for this experiment.
- Images were pre-processed to reduce the resolution to 256 X 256 pixel.
- Color information and the color curves for each individual image were extracted in order to feed Zero-DCE model.
- Deep curve estimation was done in each image. This helped to extract the low light RBG information that was fed further to enlighten GAN.

Zero-DCE

- The goal of DCE-Net is to estimate a set of best-fitting light-enhancement curves (LE-curves) given an input image.
- The framework then maps all pixels of the input's RGB channels by applying the curves iteratively to obtain the final enhanced image.
- Color information and the color curves for each individual image were extracted in order to feed Zero-DCE model.
- Deep curve estimation was done in each image. This helped to extract the low light RBG information that was fed further to enlighten GAN.

Model Architecture

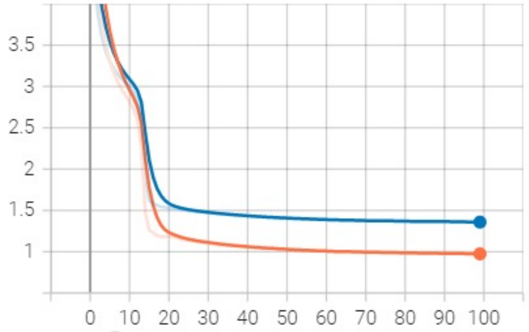


Training and Validation

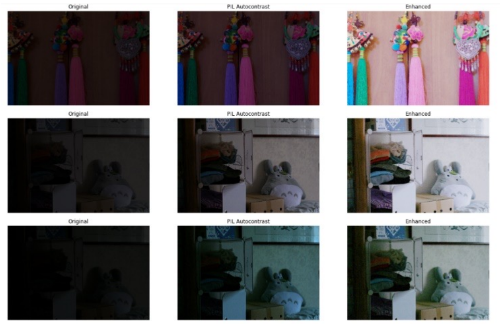
- After hyper parameter tuning, neural network of 8 convolutional layers was considered for prediction.
- The first five convolutional layers have ReLU activation function, and the last convolutional layer has tanh.
- The training set is fed to the model built above for around 100 epochs with a learning rate of 1e-4

Loss	Value
Total	**1.3587**
Illumination Smoothness	**0.0239**
Spatial Consistency	**0.2618**
Color Constancy	**0.0429**
Exposure	**1.0301**

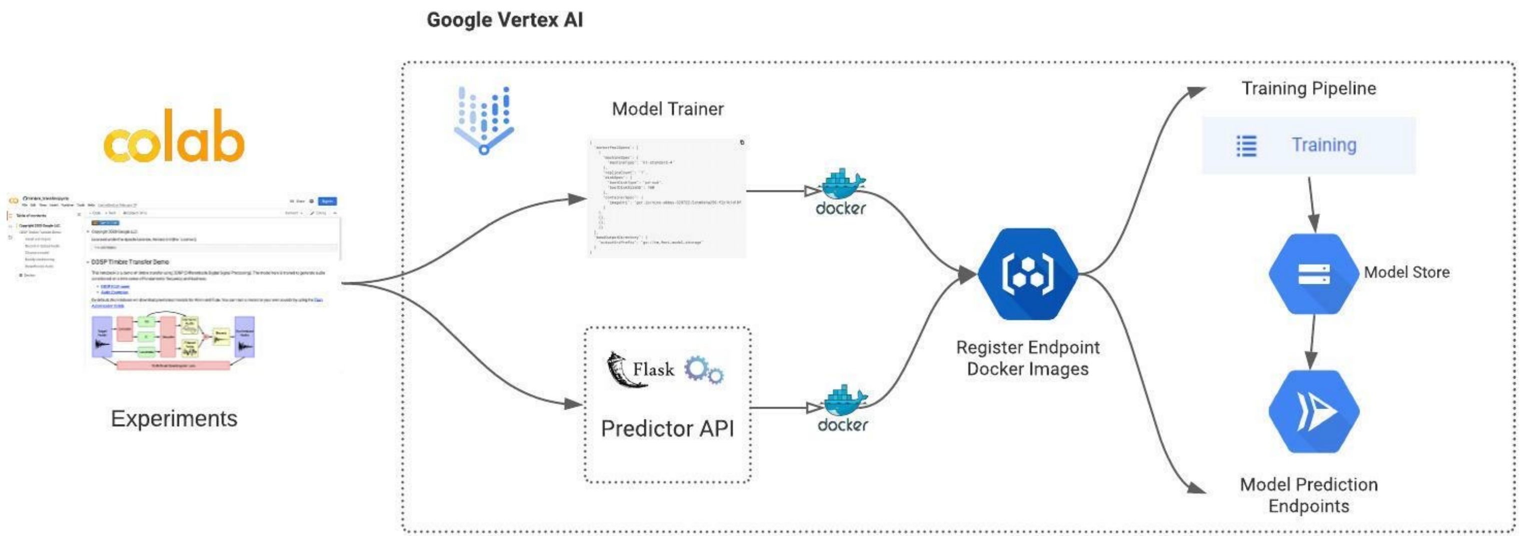
Training and Validation Loss



Original vs Baseline vs Model

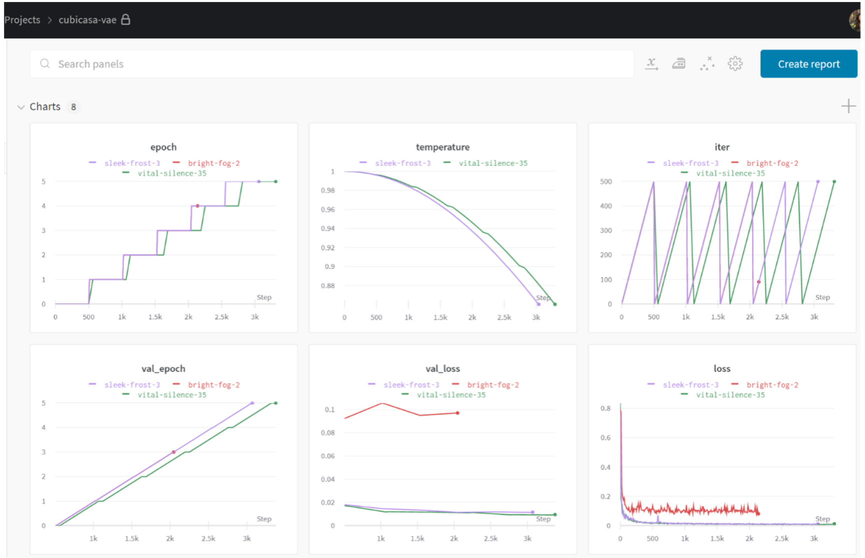


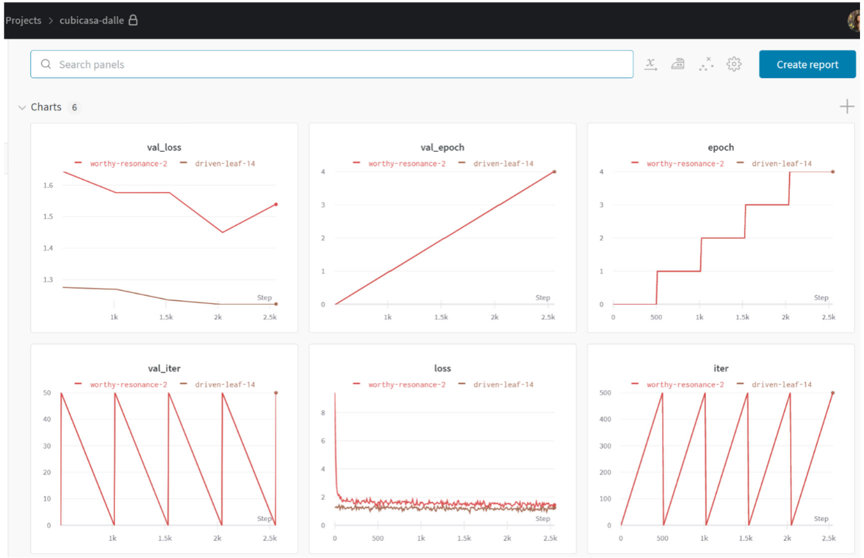
MLOps Architecture



MLOps - [WandB]()

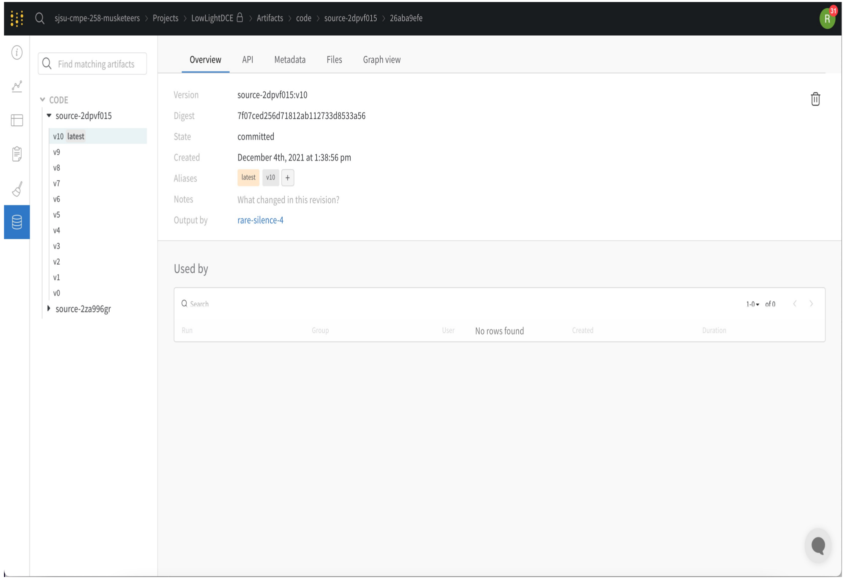
Zero-DCE Pipeline Runs



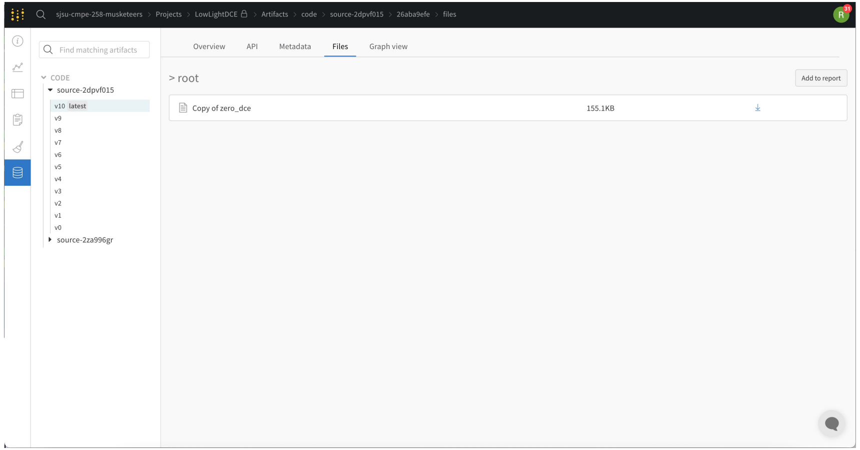


MLOps - WandB

Model Tracking/Logging



Model Versioning/Repository



Demo in Streamlit

The inference can be viewed on providing low light image and get improved image as a result.
Application URL: https://share.streamlit.io/raghavadevarajeurs/low-light-image-enhancement/main/inference.py

Low-Light Image Enhancement

Upload a png/ jpeg file to improve the image resolution

Choose a file

Drag and drop file here
Limit 200MB per file

Browse files

23.png 296.4KB

Low resolution Input

High resolution Output

Technical Difficulty

- Finding appropriate problem scope
  - *changed our dataset from original larger scope*
  - *restricted type of images to smaller size images*

- Understanding of domain which dealt with image manipulation and Image related algorithms.

- RAM/memory usage
  - *paying for Colab Pro helped*

Lesson Learnt

- Generative tasks require a lot of data and computing resources
- We needed to keep our scope small for what type of images to generate

Teamwork

- Slack was used as the medium of communication across the team.
- We followed agile methodology for project management and its execution. We met as a team every Wednesday at 8PM.
- The version control system used was git and the project was pushed to GitHub to be stored.
- Team communication played an important role in development of the project.

Code and Version Control

Packages	: Tensorflow, matplotlib, WandB, PIL, Streamlit
Input	: High resolution low light images
Output	: High resolution good contrast images
DL Model	: Zero-DCE
Tools Used	: Google Colab Pro, Weights and Biases (WandB)
Google Colab	: Low Light Image Enhancement
Version Ctrl	: GitHub
MLOps	: Vertex AI

Thank You