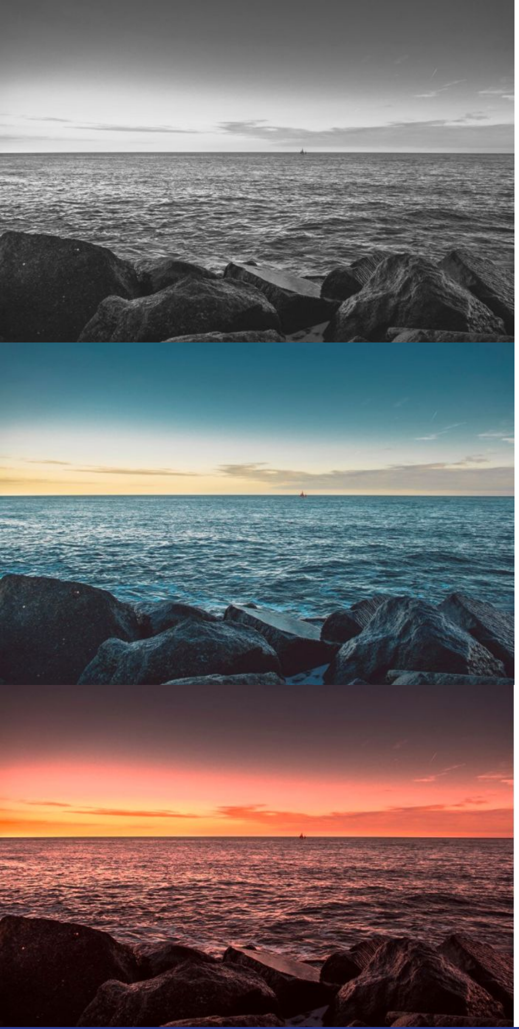
Data Science Project Automated Colorization of Images

Bharat Prakash, Deepanjan Bhattacharyya & Harsh Vashistha

Agenda

- Introduction
- Problem Statement
- Approach
- Model
- Results
- Hiccups
- Future Work
- Reference



Introduction

- Converting a black & white image to color is a tedious task
- Problems:
  - Too many objects and varying textures.
  - Each object or part of image can take on different color
- Problem under-constrained and can have multiple solutions.

Problem Statement

Automatically colorize grayscale images.

Why ?

Automated conversion of old school black white pictures.

Improvement in evaluating black and white images on color models

Approach

- Convert RGB images to Lab color space
  - L(grayscale) stands for lightness, and a and b for the color spectra green–red and blue–yellow.
- Input is the grayscale(L) and we want to predict a and b layers
- The colorization pipeline is basically is an autoencoder, the first half of the pipeline is an Encoder to extract important features and the second half is a Decoder to recreate the image from those features.
- These two halves are connected via a fusion layer which also takes embeddings generated by Inception-ResNet-V2 model (pre-trained on ImageNet) on a separate branch.
- The output is a two channel image representing a*b* channels, which is then merged with the L channel provided as input, which is then converted back to RGB.

Dataset

Source: ImageNet (First 12k images from fall_11 published urls)

- Images in the dataset largely are colored (RGB) and with varying content.
- Resolution of the images is also not fixed and hence, the images are preprocessed to (299x299) for inception-resnet-v2 and (256x256) for the coloring pipeline.

Preprocessing and Cleaning

Varied Resolution to Fixed Resolution

RGB → L*a*b*

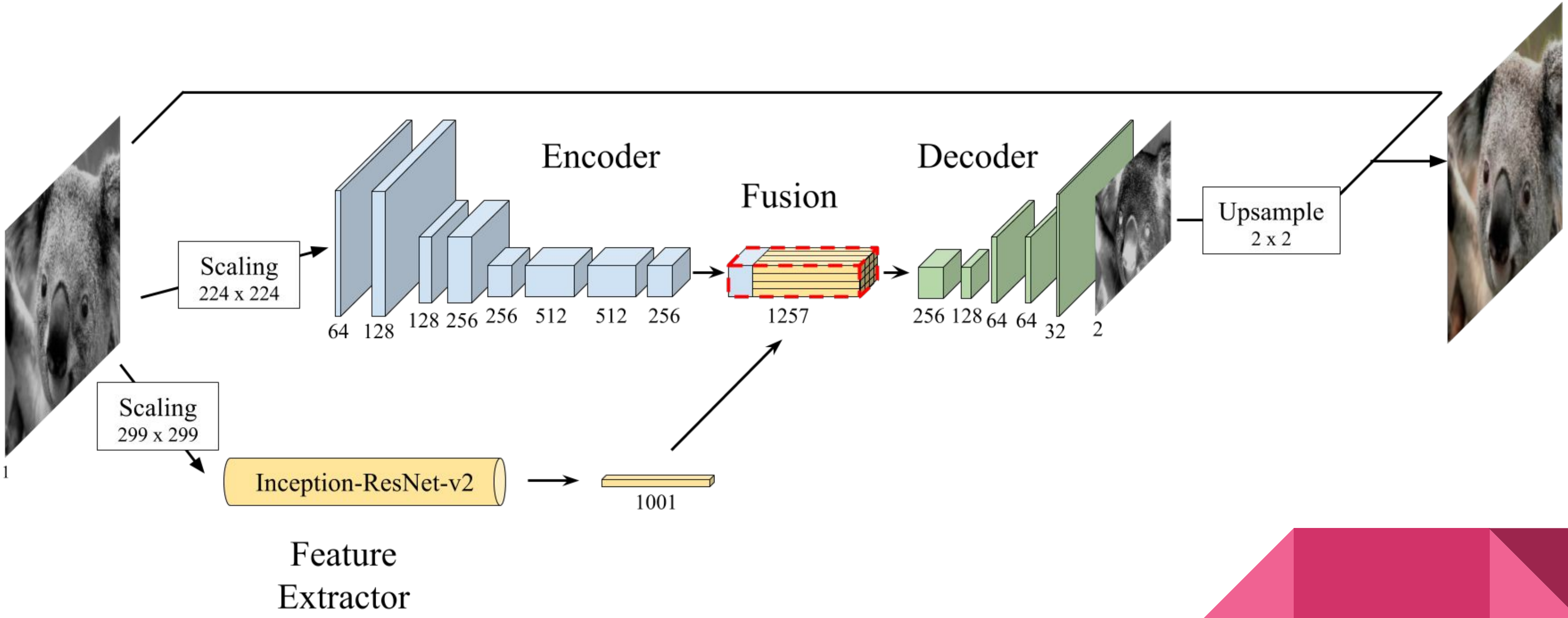
RGB → Grayscale

Zero centering

Normalizing

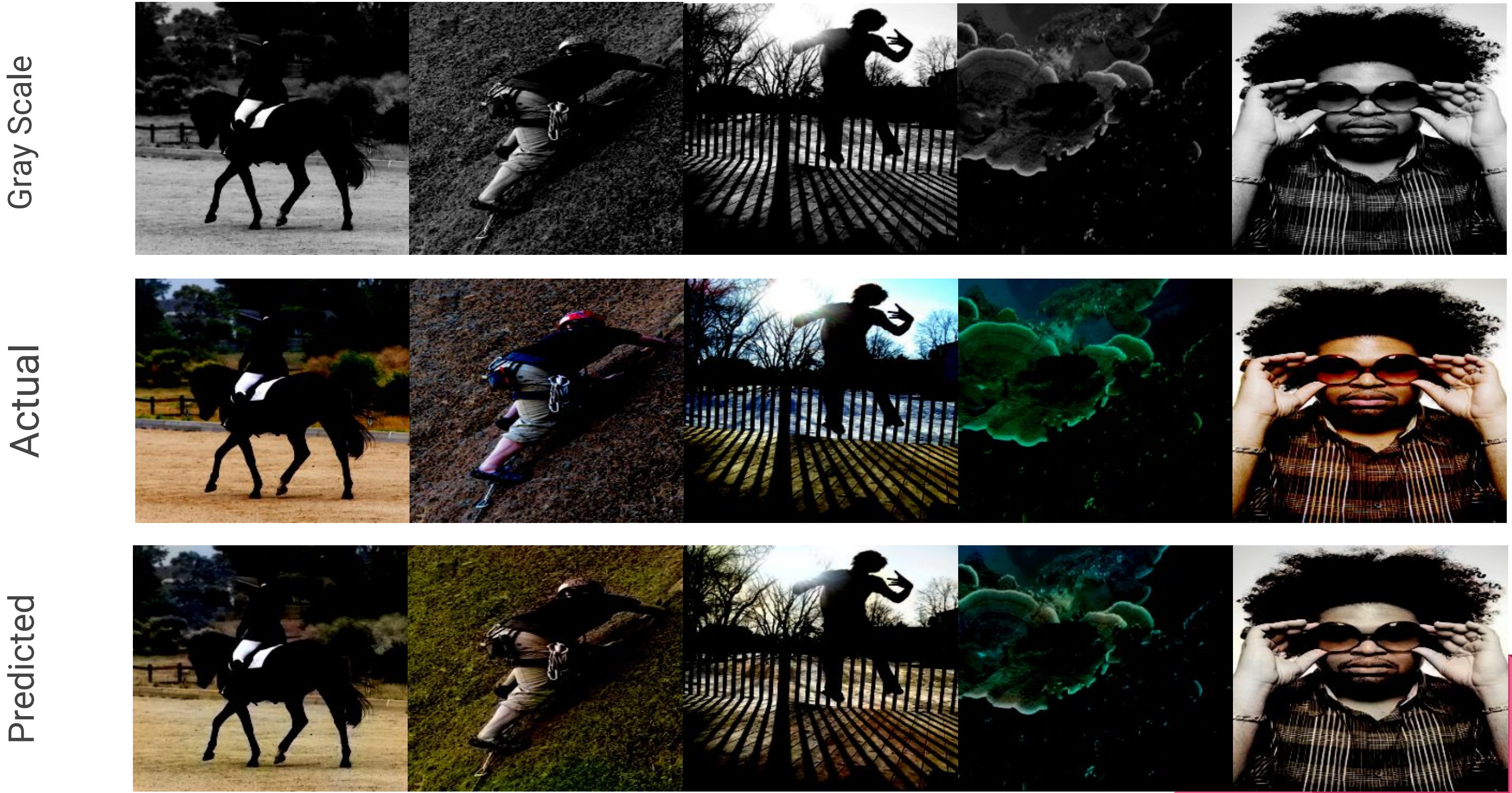
Train/Test/Validation split

Model Architecture



Jupyter Notebook

Results



Hiccups

Time/Resource constraints made the input dataset quite small, but with lot of variations

Having a huge model, we had to use pre trained network.

Conversion between color space representations is tricky

Future Work

- Generalize network input size so as to enable it to train on multiple resolution images
- Train on larger dataset to increase accuracy and generalization
- Giving context in terms of either object or texture while coloring
- Experiment using other pretrained models
- Can be extended to videos

Reference

- *Iizuka, Satoshi*, Edgar Simo-Serra, and Ishikawa Hiroshi. **Let there be Color!** *SIGGRAPH 2016*
- *Larsson, Maire, and Shakhnarovich.* **Learning Representations for Automatic Colorization.** *ECCV 2016*
- *Zhang, Richard, Phillip Isola, and Alexei A. Efros.* **Colorful Image Colorization.** *ECCV 2016*
- *Baldassarre, Morin and Lucas Rodes-Guirao.* **Deep Koalarization: Image Colorization using CNNs and Inception-Resnet-v2.** https://github.com/baldassarreFe/deep-koalarization/blob/master/paper.pdf
- http://www.whatimade.today/our-frst-reddit-bot-coloring-b-2/

Questions