Automatic Image Colorization of Gray Scale Images using Deep Learning.

Presented By:

- Abdelnour FELLAH
- Abderrahmene BENOUNENE
- Adel Abdelkader MOKADEM
- Meriem MEKKI
- Yacine Lazreg BENYAMINA

ESI - SBA

MAY 2024

Introduction

In recent years, the landscape of image processing has undergone a remarkable transformation, propelled by advancements in computer vision and deep learning techniques. This project explores novel approaches in automatic image colorization using Deep Learning.

Challenges of the colorization problem

Ambiguity

Grayscale images lack color information, making it challenging to determine the true colors of objects and scenes.

Realism

Achieving realistic and natural-looking colorizations demands attention to detail and texture preservation.

Semantic Understanding

Interpreting the semantic meaning of objects in grayscale images and assigning appropriate colors requires advanced understanding of context.

Computational Complexity

Implementing efficient algorithms capable of handling large-scale datasets while maintaining reasonable processing times is a significant computational challenge.

Solutions

1. Conditional Generative Adversarial Networks (cGANs)
2. DDColor

Tools

Pytroch for preprocessing, training and evaluation.

Streamlit for Deployement.

Preprocessing steps for both approaches

- Remove images that are already black and white
- Convert the input images from RGB or BGR color space to the LAB color space.
- Extract the L channel (representing lightness) from the LAB color space.
- Use the L channel as input to the model for colorization.
- The model predicts the AB channels (representing color information).
- Add the predicted AB channels to the original L channel to reconstruct the colorized LAB images.
- Convert the colorized LAB images back to the RGB color space for visualization and analysis.

Why LAB isntead of RGB ?

Image colorization approaches in deep learning often use the LAB color space instead of RGB for training because LAB separates color information (AB channels) from brightness (L channel), which better reflects human perception of color. This separation simplifies the learning process, making it easier for models to predict colors independently of luminance. Additionally, LAB is more perceptually uniform, meaning that small changes in color values correspond to similar changes in human perception. This property facilitates more accurate colorization results.

Conditional Generative Adversarial Networks (cGANs)

GANs are generative models that learn a mapping from random noise vector z to output image y, in contrast, conditional GANs learn a mapping from observed image x and random noise vector z, to y. The generator G is trained to produce outputs that cannot be distinguished from “real” images by an adversarially trained discriminator, D, which is trained to do as well as possible at detecting the generator’s “fakes”.

The cGAN follows the following loss function:

$$\mathcal{L}_{cGAN}(G, D) = E_{x,y}[log(D(x, y))] + E_{x,z}[log(1 - D(x, G(x, z))]$$

$$G^* = \arg\min_G \max_D L_{cGAN}(G, D)$$

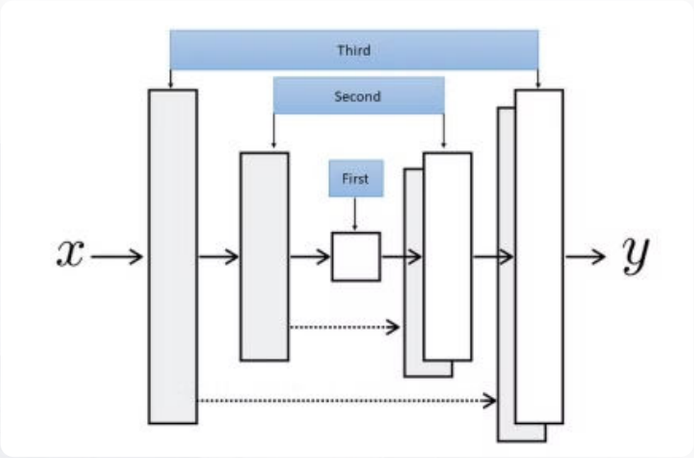
$$\mathcal{L}_{L1}(G) = E_{x,y,z}[\|y - G(x, z)\|]$$

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$$

cGANs Architecture

- **Generator:**

the generator uses a U-Net architecture with skip connections to preserve fine details. It downsamples the input for feature extraction and upsamples for colorization. Techniques like batch normalization and dropout ensure stability, resulting in high-quality colorized images.



- **Discriminator:**

The discriminator follows a PatchGAN architecture, providing localized feedback to the generator. It evaluates multiple patches separately to focus on local details and textures, enhancing its ability to discern real from fake images. Convolutional layers extract features, Batch Normalization stabilizes training, and LeakyReLU activation captures complex patterns.

Training Experimentation

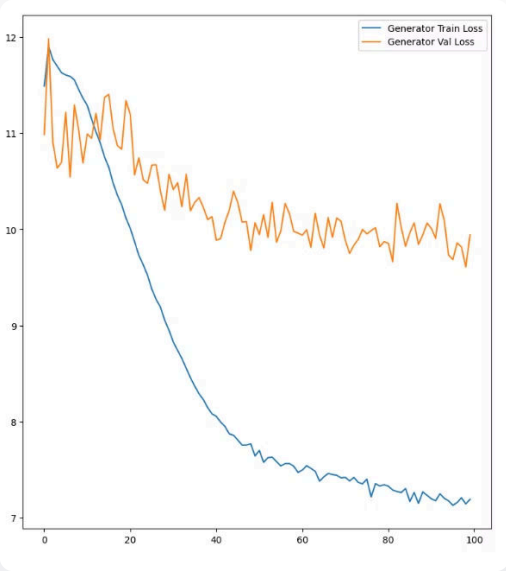
Dataset

The model was trained on a subset of the COCO 2017 dataset which contains 10000 images (8000 were used for training and 2000 for validation), additionally some further pre-processing was involved, since the COCO dataset contains images that are already black and white they were removed, because they are considered as outliers and could elevate the model's error.

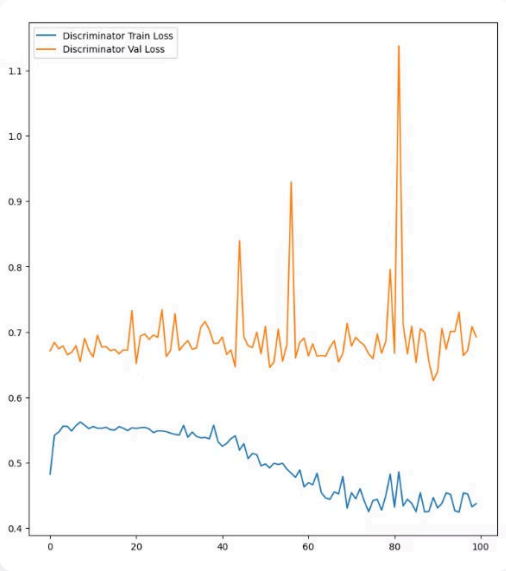
Training details

The model was trained on a GTX 1650 with 4GB vRAM for 100 epochs, with a learning rate of 0.0002, Adam optimizer, a batch size of 16 and Data Augmentation. Each epoch took exactly 7 minutes and 28 seconds (some had a second more or less), so in total the training time took 12hours.

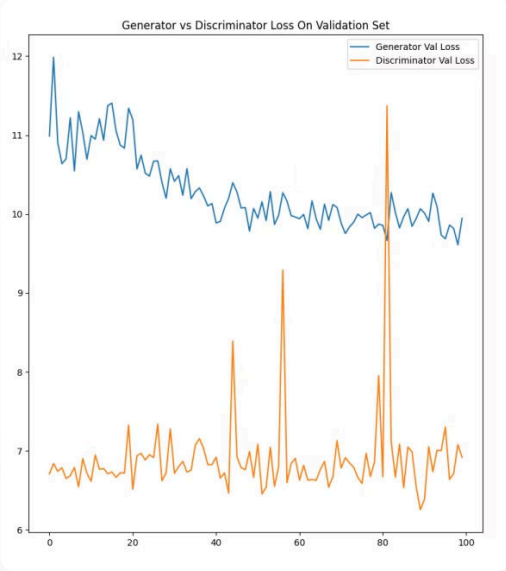
Loss plots



Generator Train and Validation losses

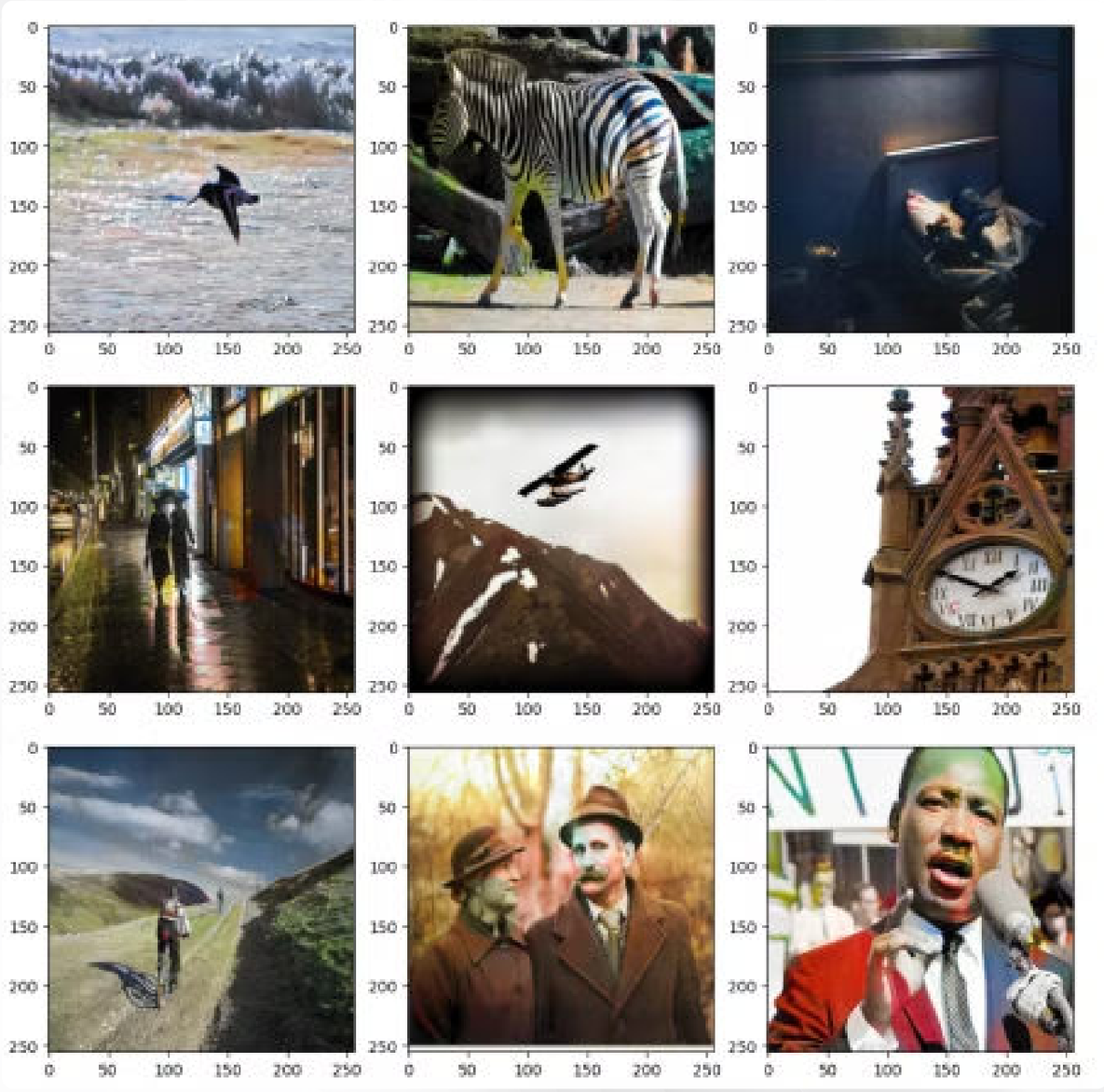


Discriminator Train and Valisation losses



Generator vs Discriminator losses on validation set

Results



Improving the model

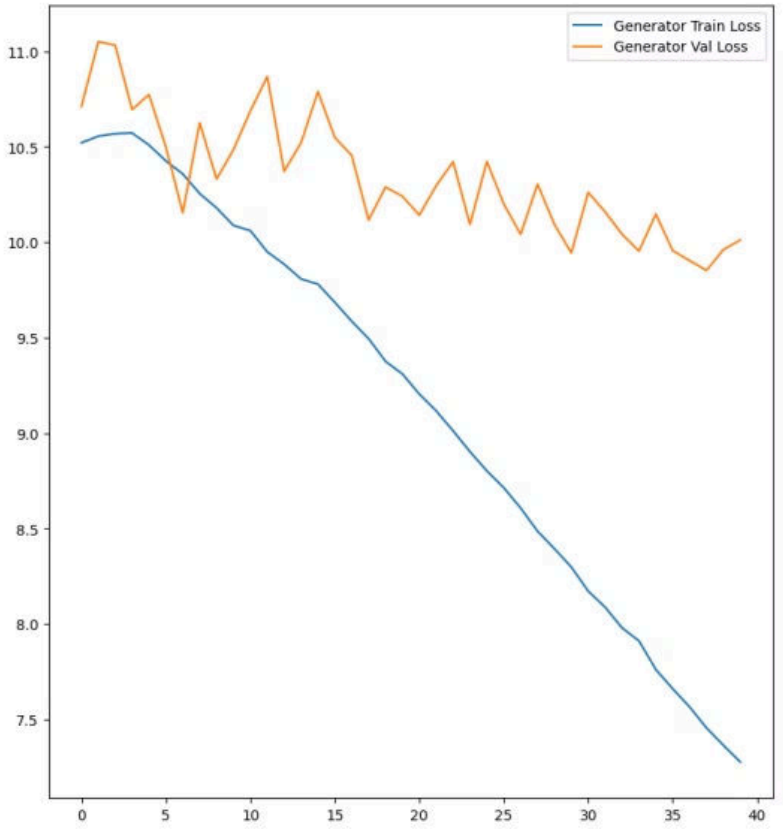
First Stage

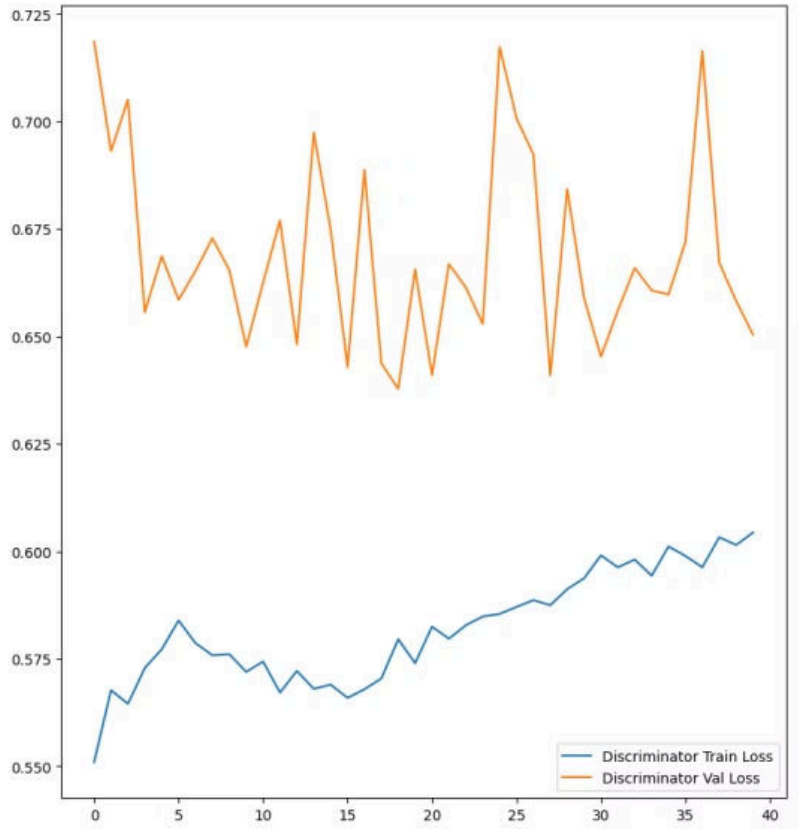
Utilize a pre-trained ResNet18 model as the backbone of the U-Net for basic feature learning.

Second Stage

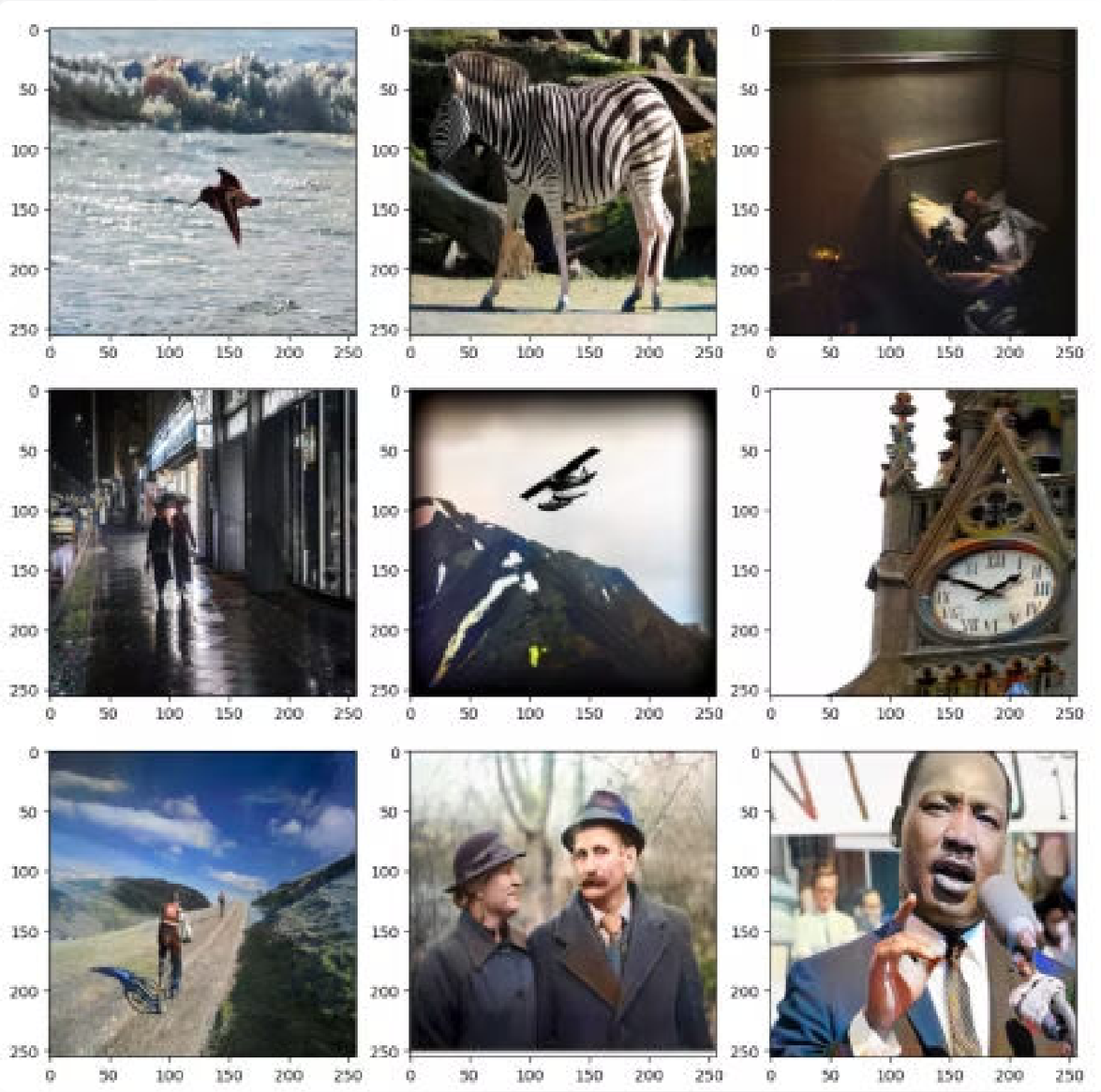
Train the entire generator, including the backbone, specifically for the colorization task using the L1 loss function.

Loss plots after Enhancement





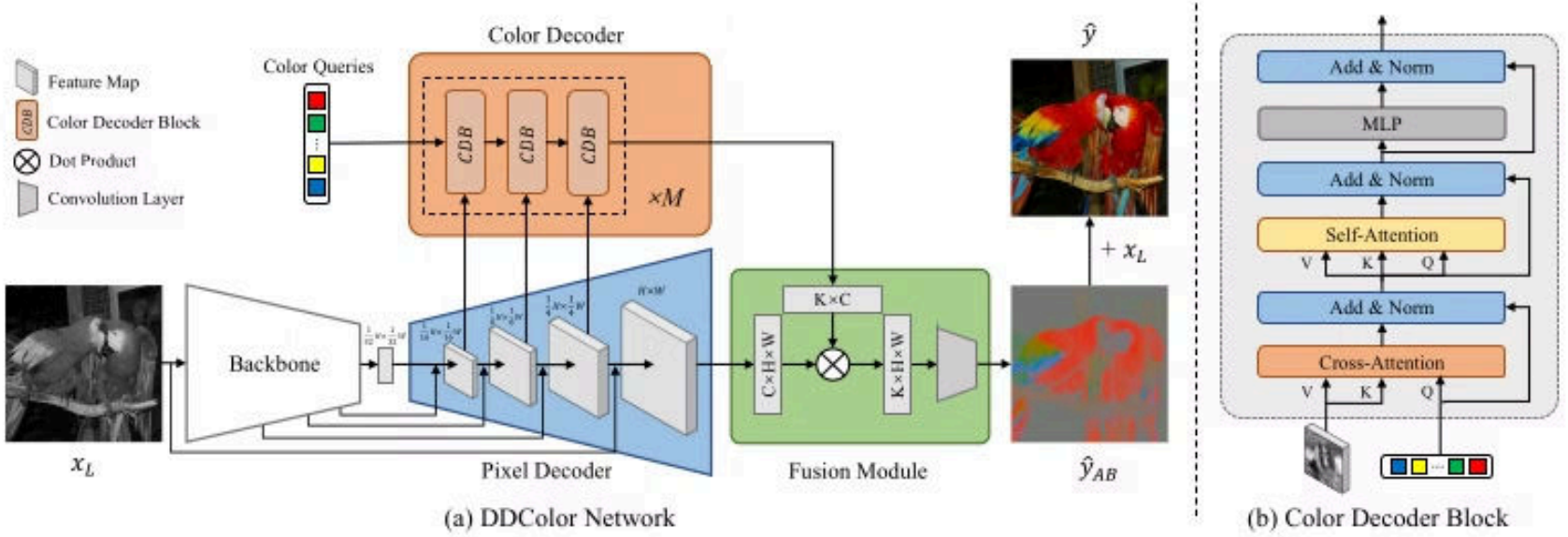
Results After Enhancement



DDColor

DDColor is a work from the ICCV 2023 Paper ”DDColor: Towards Photo-Realistic Image Colorization via Dual De- coders”. It is an end-to-end method with dual decoders for image colorization which achieves superior performance to existing state-of-the-art works. It alleviates color bleeding and improves the colorization of complex contexts and small objects significantly.

DDColor Architecture



Loss Functions

Perceptual loss

To ensure that the generated image $\hat{y}$ is semantically reasonable, a perceptual loss is used to minimize the semantic difference between it and the real image y . This is accomplished using a pre-trained VGG16 to extract features from both images

Pixel loss (L1 loss)

The pixel loss is the distance between the colorized image $\hat{y}$ and the ground truth image y, which provides pixel-level supervision and encourages the generator to produce outputs that are similar to the real image.

Adversarial loss

A PatchGAN discriminator is added to tell apart predicted results and real images, pushing the generator to generate indistinguishable images.

Colorfulness loss

A new colorfulness loss was introduced, inspired by the colorfulness score. This loss encourages the model to generate more colorful and visually pleasing images.

Training Experimentation

The original paper employed the AdamW optimizer with specific parameters, including beta values of 0.9 and 0.99, and weight decay set to 0.01, with an initial learning rate of 1e-4. Loss term weights were assigned as pix=0.1, per=5.0, adv=1.0, and col=0.5, with ConvNeXt-L as the backbone network. Training proceeded for 400,000 iterations with a batch size of 16. Learning rate decayed by a factor of 0.5 at 80,000 iterations, and further decays followed every 40,000 iterations. Adapting to hardware limitations, we utilized Kaggle's GPU resources, notably the Tesla P100, though memory constraints persisted, diverging from the original setup's 4 Tesla V100 GPUs.

Our Attempt to addapt the approach to our available hardware

- Reduction of input and output image dimensions from 3x256x256 to 3x32x32.
- Using Convnext-T as the backbone instead of Convnext-L.
- Decrease in the number of heads within both multi-head self-attention and multi-head cross-attention, from 8 to 2.
- Reduction of embedding size from 100 to 50.
- Reduction of batch size from 16 to 1.
- Adoption of a single multiscale color decoder group in place of three.
- Adoption of a single singlescale color decoder group in place of three.

Conclusion

This project explored automatic image colorization methods such as Pix2Pix and DDColor, navigating resource limitations and showcasing a commitment to innovation. Despite challenges, adapting cGAN methodology demonstrated dedication to advancement. Through extensive experimentation, a deeper understanding of image colorization emerged. These methods promise significant applications in digital art, historical image restoration, and visual content enhancement

Thank You for your Attention

Any questions ?