

Image Colorizer

Aneesh Thippa, Matthew Lo, Shaan Mistry, Arya Miryala



Background

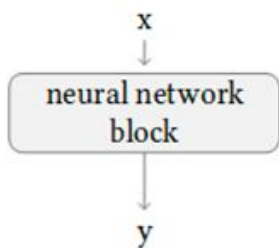
- Traditional manual colorization is time consuming and requires a lot of skill
- Convert grayscale images into realistic and visually appealing color images in a short amount of time

Final Goal:

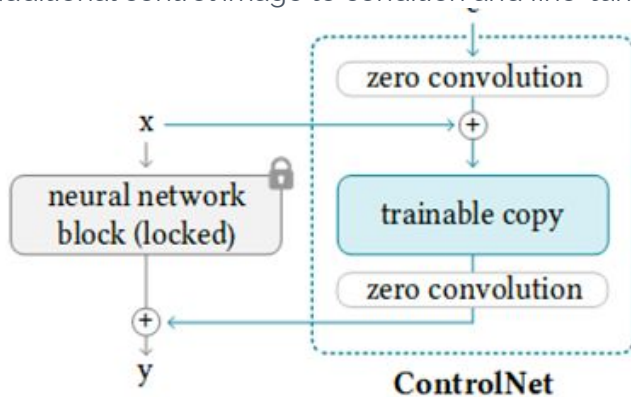
- Achieve moderate accuracy in color prediction and quality over a variety of images and ensure images are visually appealing

Approach

- Stability ai stable diffusion model
 - Initially trained with runwayml/stable-diffusion-v1-5 model but was older compared to stability ai and weren't getting good results
- Fine-tuning with ControlNet
 - With a ControlNet model, you can provide an additional control image to condition and fine-tune Stable Diffusion model.



(a) Before



(b) After

Data Preparation

- Image data found through Kaggle and Coco.
- Dataset was deployed on Hugging Face
- Dataset was structured in a specific format for our model to use: text prompt, original image, conditioning image (greyscale version)
- Initially trained our model on 5k images of flower data to see results
- Then trained model on Coco images - around 50k images

The screenshot shows the Hugging Face interface for a dataset named 'AllFlowerData' by user 'aryamiryal'. The dataset is tagged with 'Croissant' and has 0 likes. The 'Dataset card' tab is selected, showing a 'Dataset Viewer' with a table of data rows. The table has three columns: 'text' (string, classes), 'image' (image, width (px)), and 'conditioning_image' (image, width (px)). The first row shows a text prompt 'Turn the image into color.' with a corresponding color image and a greyscale conditioning image. The second row shows 'Infuse the image with color.' with a color image and a greyscale conditioning image. The third row shows 'Apply color to the image.' with a color image and a greyscale conditioning image. The dataset is split into 'train' with 5.16k rows. The 'Downloads last month' section shows 357 MB for the original dataset files and 357 MB for the auto-converted Parquet files, with 5,160 rows.

Hugging Face Search models, datasets, users...

Models Datasets Spaces Posts Docs Pricing

Datasets: aryamiryal / AllFlowerData like 0

Tags: Croissant

Dataset card Viewer Files and versions Community 2 Settings

Dataset Viewer Auto-converted to Parquet API View in Dataset Viewer

Split (1)
train · 5.16k rows

Search this dataset

text	image	conditioning_image
string · classes	image · width (px)	image · width (px)
10 values	512 512	512 512
Turn the image into color.		
Infuse the image with color.		
Apply color to the image.		

Downloads last month

Use in Datasets library Edit dataset card

Size of downloaded dataset files:
357 MB

Size of the auto-converted Parquet files:
357 MB

Number of rows:
5,160

Split (1)
train · 48.8k rows

Search this dataset

text	image	conditioning_image
string · classes	image · width (px)	image · width (px)
10 values	75 640	75 640
Infuse the image with color.		
Transform the image with colors.		
Add colors to the image.		
Add colors to the image.		

Data Preprocessing

Preprocessing Steps

- Convert all images to black and white
- Make sure all images were same resolution (512x512). Most common chosen resolution for Diffusion model because it balances detail and computational efficiency
- Randomly assign text prompts to each image & conditioning image, i.e. something along the lines of "Colorize this image"
- Prepare a JSON file with the text prompt, and paths to the image and conditioning image
- Make a script that took data from JSON file and pushes the data to Hugging Face Hub for easy access for our model

```
{  
  "text": "Make the image colorful.", "image": "images/image_44421.jpg", "conditioning_image": "conditioning_images/image_44421.jpg"}  
{  
  "text": "Transform the image with colors.", "image": "images/image_44422.jpg", "conditioning_image": "conditioning_images/image_44422.jpg"}  
{  
  "text": "Convert the image to color.", "image": "images/image_44423.jpg", "conditioning_image": "conditioning_images/image_44423.jpg"}  
{  
  "text": "Transform the image with colors.", "image": "images/image_44424.jpg", "conditioning_image": "conditioning_images/image_44424.jpg"}  
{  
  "text": "Infuse the image with color.", "image": "images/image_44425.jpg", "conditioning_image": "conditioning_images/image_44425.jpg"}  
{  
  "text": "Turn the image into color.", "image": "images/image_44426.jpg", "conditioning_image": "conditioning_images/image_44426.jpg"}  
{  
  "text": "Apply color to the image.", "image": "images/image_44427.jpg", "conditioning_image": "conditioning_images/image_44427.jpg"}  
{  
  "text": "Enhance the image with color.", "image": "images/image_44428.jpg", "conditioning_image": "conditioning_images/image_44428.jpg"}  
{  
  "text": "Infuse the image with color.", "image": "images/image_44429.jpg", "conditioning_image": "conditioning_images/image_44429.jpg"}  
{  
  "text": "Enhance the image with color.", "image": "images/image_4443.jpg", "conditioning_image": "conditioning_images/image_4443.jpg"}  
{  
  "text": "Enhance the image with color.", "image": "images/image_44430.jpg", "conditioning_image": "conditioning_images/image_44430.jpg"}  
{  
  "text": "Add colors to the image.", "image": "images/image_44431.jpg", "conditioning_image": "conditioning_images/image_44431.jpg"}  
{  
  "text": "Enhance the image with color.", "image": "images/image_44432.jpg", "conditioning_image": "conditioning_images/image_44432.jpg"}  
{  
  "text": "Transform the image with colors.", "image": "images/image_44433.jpg", "conditioning_image": "conditioning_images/image_44433.jpg"}  
{  
  "text": "Turn the image into color.", "image": "images/image_44435.jpg", "conditioning_image": "conditioning_images/image_44435.jpg"}  
{  
  "text": "Bring color to the image.", "image": "images/image_44436.jpg", "conditioning_image": "conditioning_images/image_44436.jpg"}  
{  
  "text": "Make the image colorful.", "image": "images/image_44437.jpg", "conditioning_image": "conditioning_images/image_44437.jpg"}  
{  
  "text": "Transform the image with colors.", "image": "images/image_44438.jpg", "conditioning_image": "conditioning_images/image_44438.jpg"}  
{  
  "text": "Apply color to the image.", "image": "images/image_44439.jpg", "conditioning_image": "conditioning_images/image_44439.jpg"}  
{  
  "text": "Apply color to the image.", "image": "images/image_4444.jpg", "conditioning_image": "conditioning_images/image_4444.jpg"}  
{  
  "text": "Infuse the image with color.", "image": "images/image_44440.jpg", "conditioning_image": "conditioning_images/image_44440.jpg"}  
{  
  "text": "Convert the image to color.", "image": "images/image_44441.jpg", "conditioning_image": "conditioning_images/image_44441.jpg"}  
{  
  "text": "Enhance the image with color.", "image": "images/image_44442.jpg", "conditioning_image": "conditioning_images/image_44442.jpg"}  
{  
  "text": "Bring color to the image.", "image": "images/image_44443.jpg", "conditioning_image": "conditioning_images/image_44443.jpg"}  
{  
  "text": "Convert the image to color.", "image": "images/image_44444.jpg", "conditioning_image": "conditioning_images/image_44444.jpg"}  
{  
  "text": "Convert the image to color.", "image": "images/image_44445.jpg", "conditioning_image": "conditioning_images/image_44445.jpg"}  
{  
  "text": "Colorize the image.", "image": "images/image_44446.jpg", "conditioning_image": "conditioning_images/image_44446.jpg"}  
{  
  "text": "Bring color to the image.", "image": "images/image_44447.jpg", "conditioning_image": "conditioning_images/image_44447.jpg"}  
}
```

Training

- Google Colab
- Google Cloud Deep Learning Virtual Machines
 - L4 GPU



Deep Learning VM

[Deep Learning VM \(Google Click to Deploy\)](#)

Estimated costs: \$294.45/month

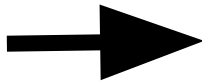
GPU-ready machine learning frameworks optimized for Google Cloud

[LAUNCH ON COMPUTE ENGINE](#)

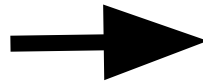
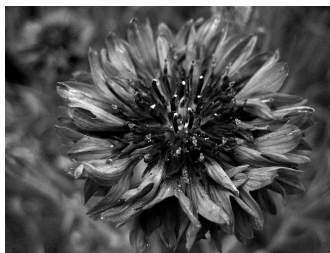
Model 1

- Learning Rate: $1e-5$
- Dataset: ~5k images
- Train epochs: 10
- Diffusion model: runwayml/stable-diffusion-v1-5

Original



Input



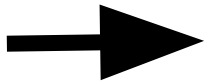
Output



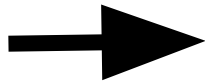
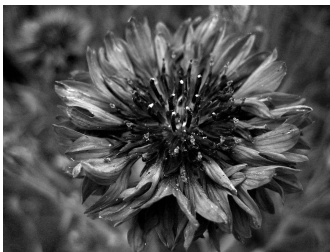
Model 2

- Learning Rate: $1e-4$
- Dataset: ~5k images
- Train epochs: 10
- Diffusion model: `stabilityai/stable-diffusion-v2-1`

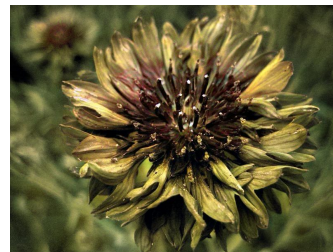
Original



Input



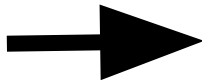
Output



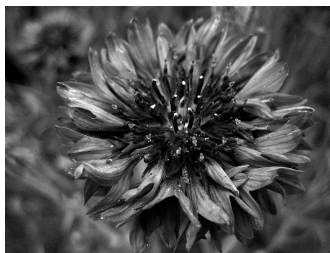
Model 3

- Learning Rate: $1e-4$
- Dataset: ~50k images
- Train epochs: 3 (18 hours)
- Diffusion model: stabilityai/stable-diffusion

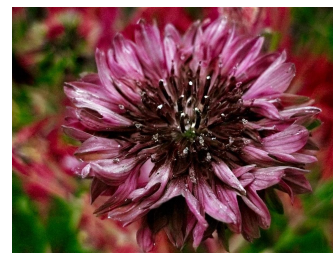
Original



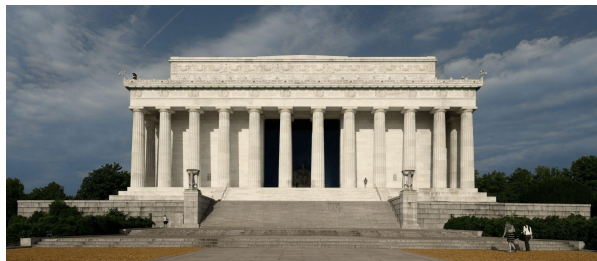
Input



Output



Results (with descriptive prompts)



Cielab color space

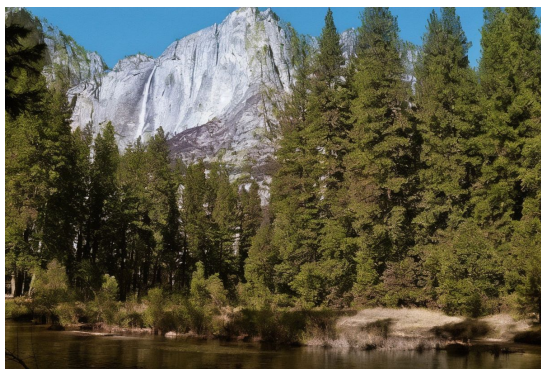
Image quality enhancement

- Cielab has one channel for lightness L^* and two for color (a^* and b^*)
- Combine L^* channel of input image with color channels of output image

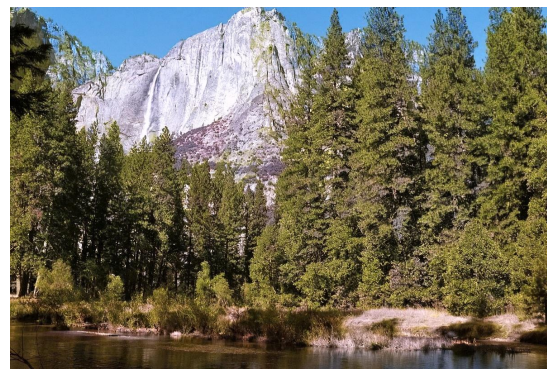
Input



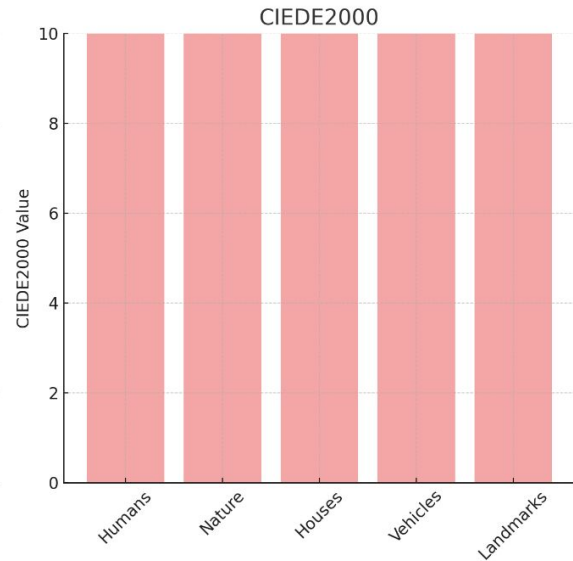
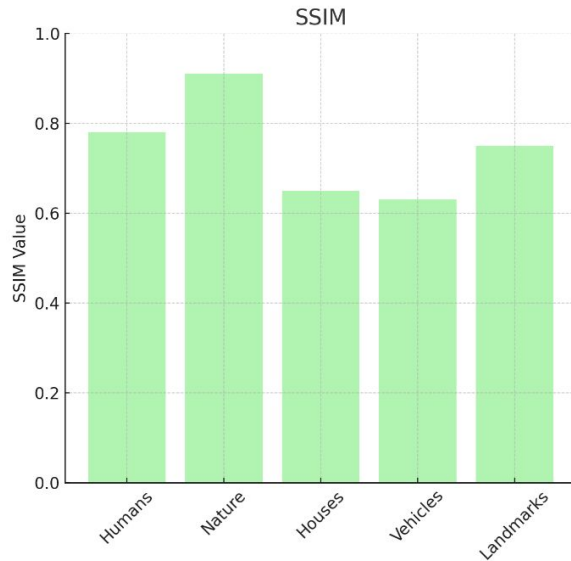
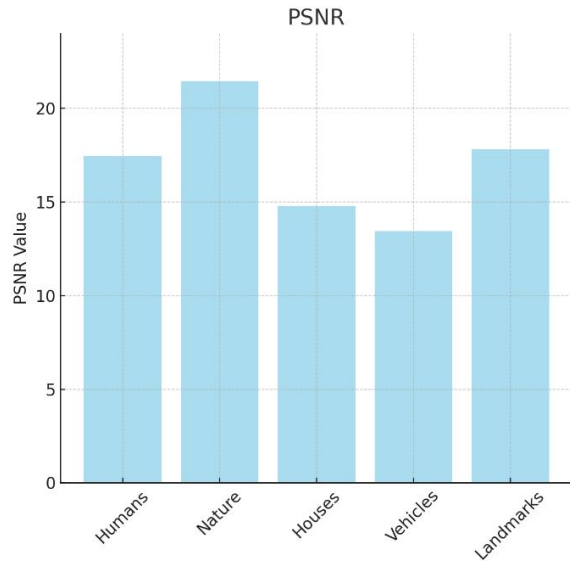
Output



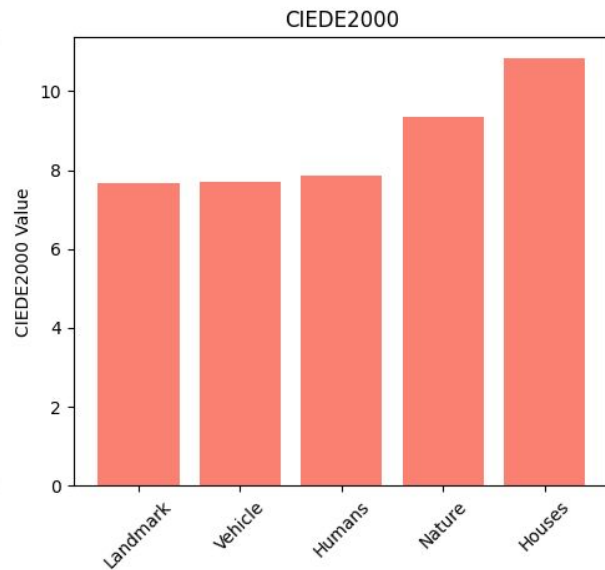
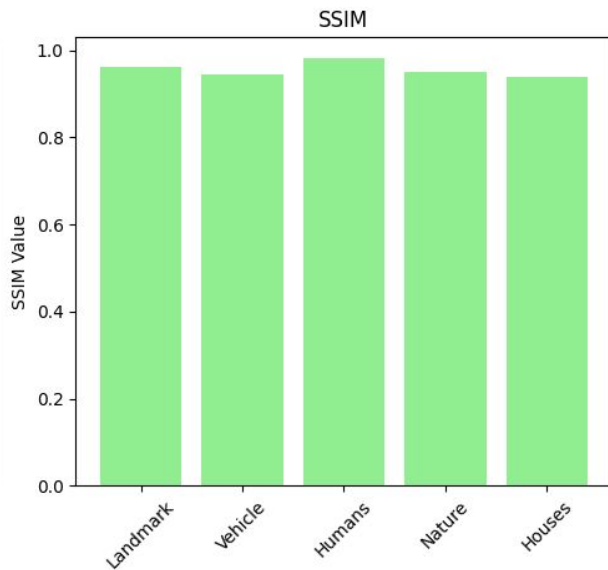
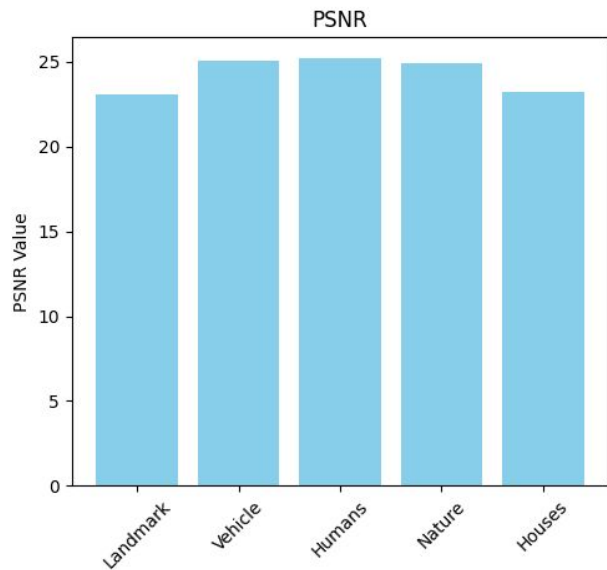
Enhanced output



Evaluation



Evaluation after hand making prompts



Limitations

- Couldn't train on more images because of storage limitations -
 - resorted to training for multiple epochs because of this
 - Resorted to checkpoint training
- Lack of computational power
 - Did not have much access to powerful GPUs to train our model
 - Had to utilize L4's for the most part which was not that great for our needs
- Time
 - Training the model with large amounts of data and for multiple epochs would take many hours at a time
 - Would take a while to improve upon our existing models significantly because of this

Conclusion + Future Goals

- Overall we achieved our goal of getting mediocre results given the resources that we had
- Learned the importance of having the right resources for training such a robust model on a large dataset
- In the future we would like to train for longer on a larger dataset if we are able to get the resources to do so

Thank You!