Image Colorization

CSE 144: Applied Machine Learning: Deep Learning

Aneesh Thippa, Arya Miryala, Matthew Lo, Shaan Mistry

06.11.2024

Professor Yi Zhang

1. Abstract

For our quarter-long project, we have decided to utilize deep learning techniques to implement an image colorizer where we can turn grayscale images into colorized versions, enhancing their visual appeal and quality. Our approach included utilizing Hugging Face to deploy our datasets, using the Stability AI Stable Diffusion model, and fine-tuning the diffusion model with ControlNet by providing the model with an additional control image to condition and fine-tune the Stable Diffusion model. This approach allows us to improve the model's performance and ensure that the colorized images closely align with the desired color schemes. The project demonstrates the effectiveness of combining state-of-the-art diffusion models with fine-tuning techniques to achieve superior results in image colorization tasks.

2. Introduction

Image colorization remains the technology industry's most time-consuming and skill-intensive technique. Colorizing an image enhances the aesthetic appeal of images and aids in various practical applications, such as restoring old photographs, improving medical imaging, and enriching visual content for media and entertainment. The biggest challenge in image colorization lies in generating realistic and contextually accurate colors for each pixel of the grayscale image. Traditional methods often need to catch up, producing results that may appear unnatural or inconsistent. Recent advancements in deep learning, particularly the development of generative models, have shown promising results in overcoming these limitations.

In this project, we tackle the problem of image colorization by leveraging cutting-edge deep learning techniques. Our approach utilizes the Stability AI Stable Diffusion model, a state-of-the-art generative model known for producing high-quality images. To further enhance the performance of this model, we incorporate ControlNet, a fine-tuning technique that conditions the diffusion model with an additional control image. This control image guides the model during the colorization process, ensuring that the generated colors are both realistic and aligned with the desired color schemes.

We deployed our datasets using the Hugging Face platform, which provides a robust and scalable infrastructure for managing and sharing machine learning resources. By combining the power of Stable Diffusion, ControlNet, and Hugging Face, we aim to develop a deep learning model capable of producing visually appealing and contextually accurate colorized images.

This project aims to demonstrate our approach's effectiveness in image colorization, attempt to achieve moderate accuracy in color prediction and quality over various images, and ensure images are visually appealing. Through our project, we wanted to highlight the potential of diffusion models and fine-tuning strategies in achieving adequate results for complex image-processing tasks.

3. Methodology

3.1 Data Preparation & Data Preprocessing

The data our model was trained on included image data from Kaggle and the COCO dataset and the dataset was deployed on the Hugging Face for efficient management. Our dataset was meticulously structured to include a text prompt, the original image, and a conditioning image (a grayscale version). Initially, we trained our model on 5,000 images of flowers to observe preliminary results. Once we were satisfied with our results we expanded our training to include around 50,000 images from the COCO dataset.

Our preprocessing steps involved converting all images to black and white and ensuring they were uniformly resized to 512x512 pixels, a resolution that strikes a balance between detail and computational efficiency. We randomly assigned text prompts to each image and its corresponding conditioning image. The text prompts were something along the lines of "Colorize this image." Additionally, we prepared a JSON file containing the text prompts and paths to the images and conditioning images. To streamline access, we created a script that extracted data from the JSON file and uploaded it to the Hugging Face Hub, which allowed for access to our model.

3.2 Generative Adversarial Networks (GANs) vs Stable Diffusion Model

Originally we had explored using generative adversarial networks(GANs) because of their ability to generate highly realistic images. By using a generator and a discriminator in a competitive framework, GANs effectively learn to create images that are indistinguishable from real ones, leading to high-quality outputs. However, after more literature review and research, we decided on using a Stable Diffusion model because they are inherently more stable and reliable.

They work by iteratively refining an image through a process that gradually adds and removes noise, which allows for a more controlled and consistent generation of high-quality images. This process aids in producing more realistic and contextually accurate colorizations, as it leverages the underlying structure of the grayscale image. Additionally, Stable Diffusion models are less prone to the factors that frequently affect GAN-generated images, leading to smoother and more coherent outputs. Ultimately because of GAN challenges in terms of training stability, we decided to proceed with a pre-trained Stable Diffusion model because of its stability during the training process and higher quality results.

3.3 Training

In order to train our models we utilized the Google Cloud Platforms virtual machines where we purchased the use of an Nvidia L4 GPU. We trained a total of 3 models for the purpose of this project. The first two models we trained were test models in order to gauge whether our model was beginning to converge. The First model we trained used a learning rate of 1e-5 and trained for 10 epochs on a dataset of around 5,000 images. The pretrained diffusion model we initially used was from Runway ML. After testing on a couple of sample images, the output did not look similar to the input image. We concluded that we would need to make changes to parameters of the model. The learning rate was adjusted to 1e-4 and the pretrained diffusion model we used was changed to one from Stability Ai. This time the results seemed to begin converging, but the output lacked vibrant color. We figured that the model needed a more diverse set of images and needed to be trained for more steps. The final model we trained was on a dataset of around 50,000 images and for 3 epochs. This model resulted in satisfactory colorization for the purpose of this project.

3.4 Cielab Color Space

The output images from the model seemed to lack clarity which highlighted the inaccuracy of the light levels. In order to combat this we opted to use the Cielab color space in order to combine the colors of our image with the luminescence of the input image. This color space is separated into three channels, L*, luminescence channel, and (a* b*), color channels. We combined the L* channel of the input image with the (a* b*) channels of the output image in order to preserve the light levels of the original image in the output image, which in turn improved the clarity of our final image output.

4. Results

We used several metrics to evaluate the performance of our final model. The most important metrics we tested were: PSNR, SSIM, CIEDE2000.

4.1 PSNR

The PSNR or Peak Signal-to-Noise Ratio (PSNR) is an image evaluation metric that assesses the quality of reconstructed images or videos compared to their original versions. It quantifies the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. PSNR is particularly useful in the fields of image processing, video compression, and other applications where it is essential to measure the quality of an approximation to an original. PSNR is expressed in decibels (dB). Higher PSNR values indicate better quality, as it implies that the noise level (or error) is low. Typical PSNR values range between 20 and 50 dB, with higher values representing better quality: 30-40 dB: Acceptable quality for many applications.>40 dB:

High-quality reconstruction, often indistinguishable from the original to the human eye. <20 dB: Poor quality, noticeable differences between original and reconstructed images. In our model, when the prompts were vague and very general, the model often hovered around 15-20 dB for the colorized images. When the model was given more detailed and specific prompts related to the image about to be colorized, the model stayed consistently above 20 dB.

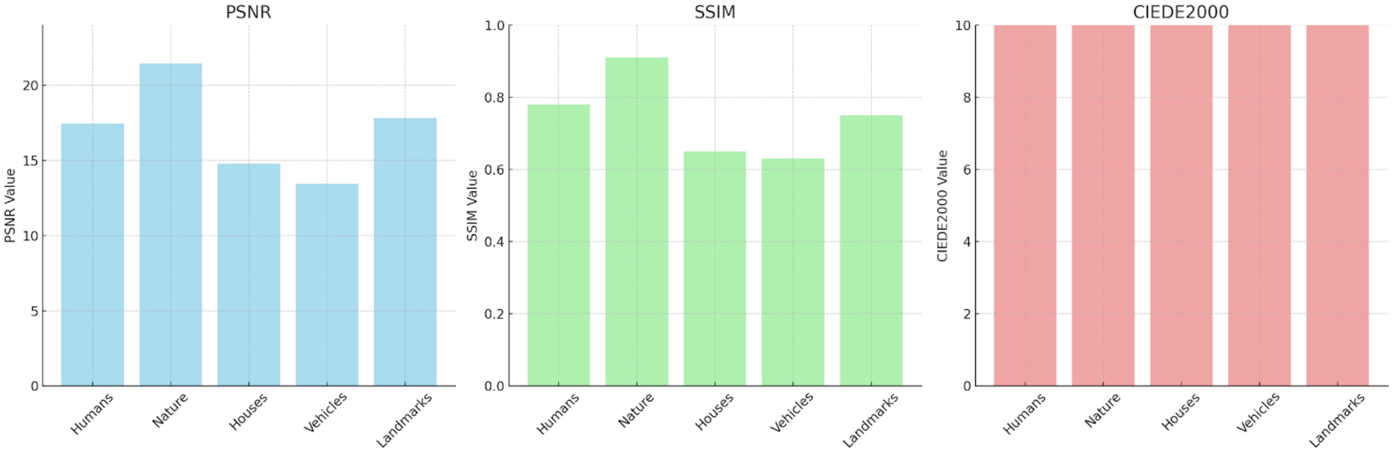
4.2 SSIM

The SSIM or Structural Similarity Index (SSIM) is a perceptual metric that evaluates the quality of images by comparing the structural information between an original and a distorted image. SSIM considers changes in structural information, luminance, and contrast. Luminance measures the similarity of the mean luminance (brightness) between the images. Contrast measures the similarity of the contrast (variance) between the images. Structure measures the similarity of the structure (correlation) between the images. Overall SSIM Index: The overall SSIM index is typically calculated as the mean SSIM over local windows of the image, providing a single quality score ranging from -1 to 1, where: 1 signifies perfect structural similarity (identical images), 0 signifies no structural similarity, and -1 signifies completely dissimilar structures. In our model, when the prompts were vague and very general, the model often hovered below 0.8 and reached lows of 0.6. When the model was given more detailed and specific prompts related to the images about to be colorized, the model stayed consistently above 0.8.

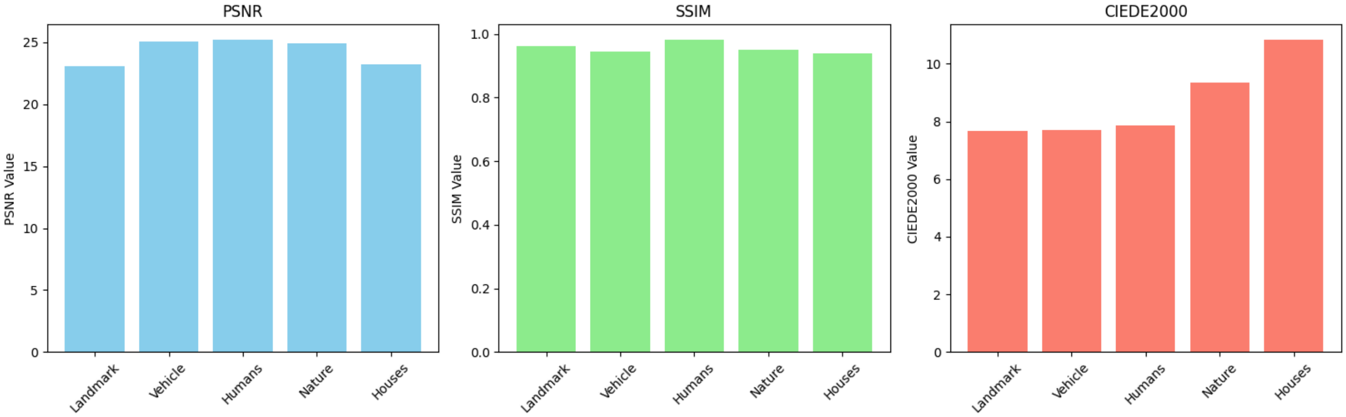
4.3 CIEDE2000

CIEDE2000 is an advanced color-difference evaluation metric developed by the International Commission on Illumination (CIE). It is designed to improve upon earlier color-difference formulas (like CIE76, CIE94) by more accurately reflecting human perception of color differences. This metric is widely used in various industries, including digital imaging, textiles, and printing, to ensure color consistency and quality. CIEDE2000 quantifies the perceived differences between two colors. The colors are represented in the CIELAB color space, which consists of three coordinates: L* (lightness), a* (green-red), and b* (blue-yellow). The metric produces a value, which provides a numerical estimate of the perceived color difference where 0 signifies No color difference, < 1 signifies Color difference generally imperceptible to the human eye, 1-2 signifies color difference perceptible on close inspection, 2-10 signifies Noticeable color difference, and >10 signifies large color difference. In our model, when the prompts were vague and very general, the model exceeded values of 10 and was poorly colored. When the model was given more detailed and specific prompts related to the images about to be colorized, the model stayed below 10 and often hovered around 8.

4.4 Graphs



The above graph details the evaluation metrics described above when the model attempted to colorize five different types of images with a vague and general prompt. As described above, the values were poor on all three evaluations and could be better when given a better prompt.



The above graph details the evaluation metrics described above when the model attempted to colorize five different types of images with a better, more detailed prompt that was specific to a given image about to be colorized. As described above, the values were a lot better on all three evaluations, and our model could colorize better when given a better prompt.

5. Limitations

While our approach to image colorization utilizing the Stability AI Stable Diffusion model and ControlNet demonstrates promising results, several limitations and areas for improvement were identified throughout the project.

5.1 Storage Limitations

One significant limitation we faced was storage capacity. Due to these constraints, we could not train the model on a larger dataset. As a result, we had to resort to training for multiple epochs to maximize the use of the available data. Although this helped improve the model to some extent, a larger dataset would have potentially yielded better results. Due to storage limitations and the need to manage computational resources effectively, we had to employ checkpoint training. While this approach allows for incremental improvements and recovery from potential disruptions, it can introduce complexities in model management and may not be as efficient as training on a more robust system with ample storage.

5.2 Computational Power

Another major limitation was the need for more access to powerful GPUs. Training deep learning models, especially those as complex as the Stable Diffusion model, requires substantial computational resources. Our access was primarily limited to L4 GPUs, which, although helpful, could have been better for our needs. The limited computational power hindered our ability to train the model more efficiently and effectively.

5.3 Time Constraints

Training the model with large amounts of data and for multiple epochs is inherently time-consuming. Each training session took many hours, which slowed down our overall progress. This extended training time also meant that iterations and improvements upon existing models were significantly delayed. Consequently, refining and enhancing the model to achieve optimal performance took a lot of work. Access to more powerful GPUs like the Nvidia A100 would save more time.

6. Conclusion

Ultimately, our project successfully developed image colorization using Stability AI and ControlNet. This achievement highlights the potential of deep learning techniques to emphasize automation, providing valuable tools for applications such as restoring old digital images, and improving visual data in many industries. This project contributes to the field of deep learning by underscoring the significance of fine-tuning pre-existing models to create robust and practical solutions. By building upon established architectures like Stability AI and ControlNet, we demonstrated that leveraging and refining these powerful models can yield highly effective and efficient outcomes. This approach not only maximizes the utility of existing technologies but also highlights a pathway for developing specialized applications with reduced training times and resource requirements. In the future, given more resources, we can extend our work by first training the model further to enhance its output quality. Additionally, we can incorporate another model capable of identifying objects within images to achieve more accurate colorizations of specific elements. This dual approach will not only refine the overall colorization process but also ensure that individual objects are rendered with higher precision and realism.

7. Team Member Contributions

Aneesh Thippa and Shaan Mistry: We mainly worked on training and developing the models that we tested. We trained multiple models each taking multiple hours to train and judged the results in order to tune the parameters better. Through conducting a literature review in order to find out what models were being used today we transitioned from initially training a GAN based model to using a pretrained model. In order to fine tune the existing model we also learned how to use controlnet alongside pre-trained diffusion models. Lastly we were also responsible for the inference portion of the model as well allowing us to produce colorized images from the models that we trained.

Arya Miryala: I primarily worked on creating our datasets and gathering relevant data from sources like Coco and Kaggle. I worked on exploring and learning the Hugging Face hub to deploy our dataset so our model can easily access our dataset and did all the preprocessing that was necessary so our model could be trained on relevant and ideal data. I also helped with training the model before we switched over to the Google Cloud VM instances and discovered how unsuccessful the initial model, Runway ML, was. After conducting some research, I discovered the Stability AI model and recommended we switch to it for training because it was newer and more advanced than the initial model we were using.

Matthew Lo: I primarily worked on creating the algorithms that would help evaluate our model in a way that would provide important feedback and information on what we could do to improve our model. I researched the best possible evaluation metrics that would cater to our model's needs and provide good information for us and the users who would be using our model. I also initially worked on helping preprocess our datasets so we could deploy our datasets to get early renditions of our model running.

8. References

“ControlNet Documentation.” *ControlNet*, huggingface.co/docs/diffusers/en/training/controlnet. Accessed 8 June 2024.

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2018, November 26). *Image-to-image translation with conditional adversarial networks*. arXiv.org. https://arxiv.org/abs/1611.07004

Shu-Yu Chen a, a, b, c, d, e, & AbstractImage colorization is a classic and important topic in computer graphics. (2022, June 8). *A review of image and video colorization: From analogies to Deep Learning*. Visual Informatics. https://www.sciencedirect.com/science/article/pii/S2468502X22000389

Sortino, Renato. “ColorizeNet: Stable Diffusion for Image Colorization.” *Medium*, Medium, 7 Oct. 2023, medium.com/@rensortino/colorizenet-stable-diffusion-for-image-colorization-bdc9c35121fa.

Zhang, R., Isola, P., & Efros, A. A. (2016, October 5). *Colorful image colorization*. arXiv.org. https://arxiv.org/abs/1603.08511