Bringing vibrance to historical paintings through image colorization

Harshit Kumar
Khoury College of Computer Sciences
Northeastern University, Boston
kumar.hars@northeastern.edu

Srilakshmi Kanagala
Khoury College of Computer Sciences
Northeastern University, Boston
kanagala.s@northeastern.edu

November 1, 2023

Abstract

We used Deep Learning techniques to recolor grayscale images of paintings and reconstruct their lost or damaged visual information. We utilize U-Net and Generative Adversarial Networks specifically Conditional Generative Adversarial Networks (cGAN) called pix2pix to colorize grayscale paintings. We use two variations of pix2pix, one with vanilla UNet and the other with UNet having a resnet18 backbone. The ultimate goal of this project is to compare and validate different deep neural networks that can be used to generate plausible and aesthetically pleasing colorizations of grayscale images of paintings to help us rediscover the beauty of the lost artwork.

1 Introduction

Artworks provide insights into history, culture, and aesthetics, but many are lost or damaged over time. Black-and-white photos of paintings taken before loss serve as a resource to reconstruct their appearance. Colorization offers a fresh perspective on historical images and emotional interplay of colors. Lack of color in photos creates detachment from the vivid world they capture.

Recent advances in deep learning techniques have led to significant progress in colorizing grayscale images, including paintings. Before, photoshop was used to colorize grayscale images which required a lot of time and manual effort. With help of deep learning techniques, this can be achieved in a lot less time with reduced manual work and better results. In this project, we propose using deep learning methods to recolor grayscale images of paintings and recover their lost or damaged visual information. Specifically, we will employ U-Net and Conditional Generative Adversarial Networks (cGAN) to colorize grayscale paintings.

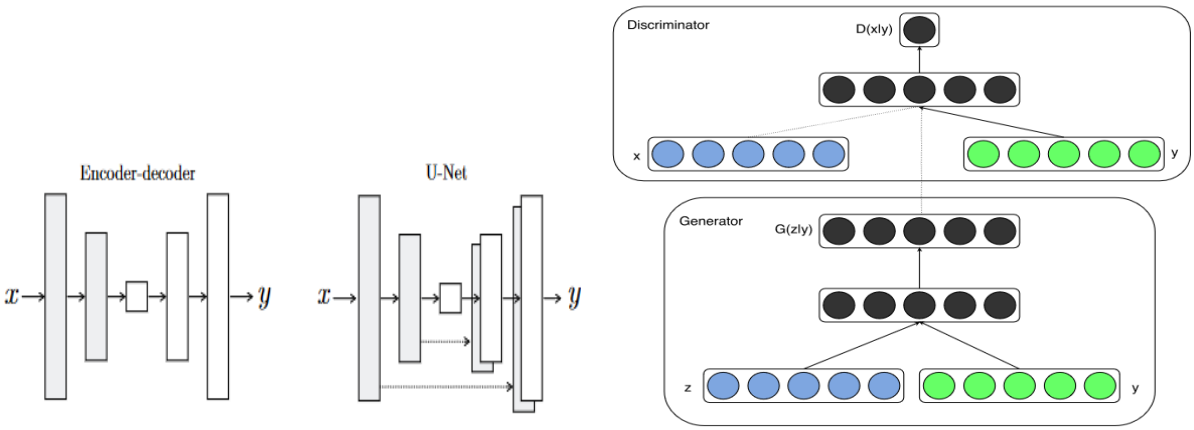

Figure 1: UNet architecture (Isola et al. (2018)) and cGAN architecture (Mirza and Osindero (2014))

The results of this project could have significant implications for the reconstruction of damaged artworks. By creating plausible and aesthetically pleasing colorizations of grayscale images of paintings, we could rediscover the beauty and intended aesthetic qualities of lost or damaged artworks. Additionally, the proposed method could be extended to other applications, such as restoring old photographs, enhancing medical images, etc.

2 Related Work

In recent years, generative adversarial networks (GANs) (Mirza and Osindero (2014)) have emerged as a powerful tool for image colorization. GANs are a type of deep learning model that can generate realistic images by training a generator network to produce images that are indistinguishable from real images. The use of GANs for image colorization was first introduced by Isola et al. (2018) who proposed a method called pix2pix that uses a cGAN to generate realistic colorized images from grayscale images.

Another important deep learning model used for image colorization is the UNet architecture. The UNet is a fully convolutional neural network that was originally developed for biomedical image segmentation tasks. Ronneberger et al. (2015) proposed the UNet architecture, which consists of an encoder and a decoder network, that captures both local and global information in the image. UNet has been successfully applied to a wide range of image processing tasks, including image colorization.

In this paper, we propose a method for bringing vibrance to historical paintings through image colorization using UNet and GANs. Our approach combines the strengths of these two architectures by using a UNet as the generator network and a GAN as the discriminator network. Our method is inspired by the work of Isola et al. (2018) and uses a similar pix2pix framework. However, instead of using a simple CNN as the generator network, we use the UNet architecture, which has been shown to produce better results for image colorization tasks.

3 Methodology & Experiment

The proposed project solves an Image-to-image translation problem. We've solved it as a regression and generation task.

3.1 Dataset and Preprocessing

Our dataset has 519 images sourced from a Kaggle project titled "Best Artworks of All Time" (kag). We focused our study on four painters who share a broad painting style with a focus on portraiture or landscape work. Although they demonstrate a subtle variety of styles, their compositions and use of color are fundamentally similar. To achieve more refined results, we narrowed the scope of our research to these artists. We divided the dataset into a 70:10:20 split for training, testing, and validation, and we rescaled each image to a 256 x 256 square using bicubic interpolation where necessary. We use random horizontal flip data augmentation to increase the size of our training dataset.

To enable the model to predict pixel values for the a* and b* channels, we converted images from the RGB color space to the CIELAB color space (Wikipedia contributors (2023)), following the approach of Isola et al. Specifically, we used the L lightness channel as input for our model, which allowed for a more straightforward prediction equation than if we had used RGB pixel values.

The dataset link is provided in the References section.

3.2 Models

We will use UNet and Conditional Generative Adversarial Network (cGAN) called pix2pix to colorize grayscale images because they are a powerful class of deep learning models that can learn to generate new data that is similar to the training data.

3.2.1 UNet

UNet follows an encoder-decoder architecture. The encoder will take a grayscale image as input and generate an intermediate representation which is then fed to the decoder that generates the colored image. The UNet is trained with the L1 loss in a supervised manner. We use resnet18 (He et al. (2015)) as the backbone for the UNet and use ImageNet (Deng et al. (2009)) pretrained weights of ResNet.

3.2.2 pix2pix cGAN

GANs consist of two distinct models, a generator and a discriminator, competing with each other. We will use adversarial loss, which is based on the idea of a minimax game. The generator network aims to generate colored images, from grayscale images, that can fool the discriminator network, while the discriminator network aims to distinguish real (ground-truth) colored images from generated ones.

Let us consider x to be the grayscale image, z to be the input noise for the generator, and y to be the desired 2-channel output from the generator. In the case of a real image, y can represent the 2 color channels. We use G to refer to the generator model and D to the discriminator. Then the loss, L for our conditional GAN will be as follows.

$$L_{cGAN(G,D)} = \min_G \max_D \mathbb{E}_{x,y}\big[logD(x,y)\big] + \mathbb{E}_{x,z}\big[log(1 - D(G(x,z)))\big] + \lambda L_1(G) \quad (1)$$

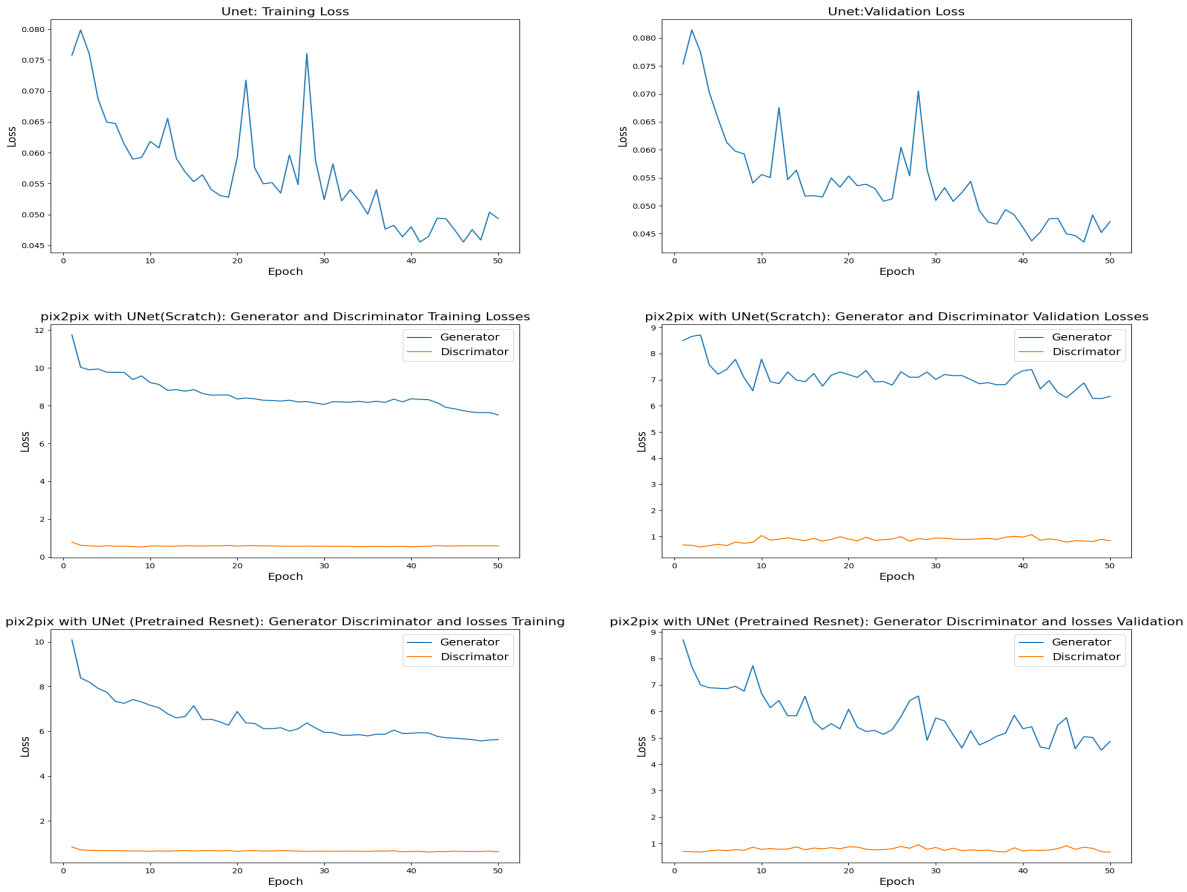

Figure 2: Training and Validation losses

Here, λ is the weighing parameter for the L_1 loss, $L_1(G)$, which is calculated only for the generator.

The pix2pix cGAN implements a U-Net to be used as the generator of our GAN. The discriminator is made up of strided convolution layers, batch norm layers, and LeakyReLU activations without max-pooling layers i.e. convolution $>$ batch norm $>$ leaky ReLU. We've specifically used Patch discriminator. It's is a type of discriminator that is used to distinguish between real and fake images at a local level. Specifically, instead of looking at the entire image and deciding whether it is real or fake, the patch discriminator divides the image into smaller patches and evaluates the authenticity of each patch individually. This approach allows the discriminator to capture more fine-grained details and textures in the image, which can help prevent the generator from producing blurry or low-quality images.

We've trained two types of cGANs.

1. pix2pix with (scratch) UNet: We train both the models i.e. generator (UNet) and discriminator (PatchDiscriminator) from scratch.

2. pix2pix with pre-trained resnet18 backbone UNet: We use earlier trained UNet in section 3.2.1, which had pretrained resnet18 backbone, and then train the generator and discriminator networks.

Table 1: Quantitative comparison with learning rate of 1e-3 for Unet and 2e-4 for GANs

Datset	Model	MSE	Visual Comparison
Training	UNet (resnet18 backbone)	0.006	
	pix2pix (scratch UNet)	0.0045	
	pix2pix (pretrained resnet18 backbone UNet)	0.0054	
Validation	UNet (resnet18 backbone)	0.0062	
	pix2pix (scratch UNet)	0.0047	
	pix2pix (pretrained resnet18 backbone UNet)	0.0055	
Testing	UNet (resnet18 backbone)	0.0133	Good
	pix2pix (scratch UNet)	0.0132	Best
	pix2pix (pretrained resent18 backbone UNet)	0.0135	Good

Table 2: Quantitative comparison for learning rate of 2e-3 for both UNet and GANs

Datset	Model	MSE	Visual Comparison
Training	UNet (resnet18 backbone)	0.0061	
	pix2pix (scratch UNet)	0.0043	
	pix2pix (pretrained resnet18 backbone UNet)	0.0051	
Validation	UNet (resnet18 backbone)	0.0062	
	pix2pix (scratch UNet)	0.0046	
	pix2pix (pretrained resnet18 backbone UNet)	0.0051	
Testing	UNet (resnet18 backbone)	0.0134	Good
	pix2pix (scratch UNet)	0.0130	Best
	pix2pix (pretrained resent18 backbone UNet)	0.0134	Good

3.3 Experiment

We use a batch size of 16. We perform three training experiments as discussed above. We train UNet for 50 epochs with Adam optimizer with a learning rate of 1e-4. Then, we train both versions (scratch UNet and resnet18 backbone UNet) of pix2pix for 50 epochs with a learning rate of 2e-4, beta1 as 0.5, beta2 as 0.999, and λ in loss as 100. We also trained GANs for a learning rate of 2e-3. Increasing the learning rate increased the learning rate of the models. We use the same configuration for training both the generator and discriminator. Training each GAN took 1.5hr for 50 epochs on Tesla T4 on Colab.

4 Results

The best way to compare the colorization results is by manually looking at the output images. We observed that the pix2pix trained from scratch performs the best among the three models. We also notice that the output colors in the images from Unet and pix2pix (pretrained resnet18 backbone Unet) are similar. This visual similarity is because we're using the pretrained Unet in that pix2pix model. For quantitative comparison, we use the MSE (mean squared error) for pixel-to-pixel colorization accuracy on the testing set. We've summarized the results in Table 2. We observe that the MSE values are not quite apart for the models, unlike the visual comparison where the colors are quite different. Nevertheless, our models are performing fairly good and giving the correct colors for the black and white paintings.

Figure 3: UNet results comparison

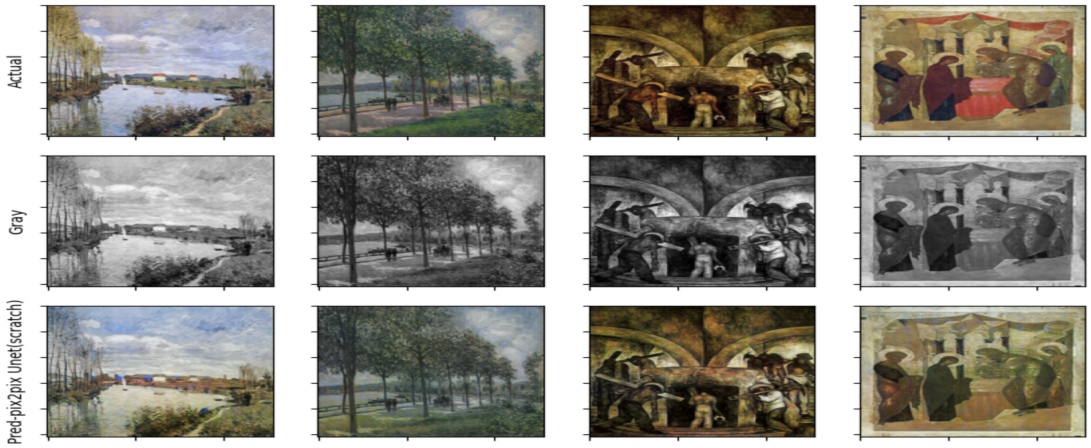

Figure 4: pix2pix (scratch UNet) results comparison

5 Conclusion and Future Work

We trained three models - 1) Unet, 2) pix2pix with UNet as generator trained from scratch, and 3) pix2pix with UNet having ResNet18 as backbone trained using transfer learning. We observed the visual similarity in the output colors of the first and 3rd models due to the usage of pretrained UNet. In conclusion, the models especially pix2pix with UNet trained from scratch are fairly good for colorizing the black and white paintings. These models can give better results with larger training datasets. Future work may involve testing different configurations of the models and hyperparameters to arrive at the best model for painting colorization.

6 Contributions

Most of the work was done in collaboration with bi-weekly meetings. We used Google Colab to train the three models.

Srilakshmi'work: She worked on researching the dataset and selecting images of paintings based on the author's styles and preprocessing the data for training. She also worked on the UNet model.

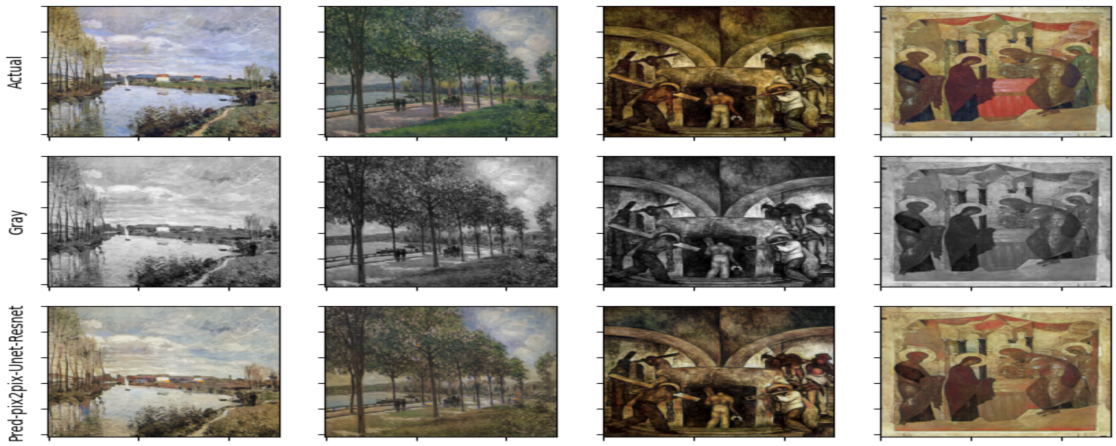

Figure 5: pix2pix (pretrained resent18 backbone UNet) results comparison

Harshit's work: He worked on researching the models to be used to solve the task and setting up the training pipeline. He also worked on the pix2pix GANs.

We both trained multiple versions of all three models to compare performances and various hyperparameters. We both worked on the project report.

References

Best artworks of all time dataset (google drive). URL `https://drive.google.com/drive/folders/1tBuEFQiOFp2P1hOxp5HMMui46jd-E-8H?usp=share_link`.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks, 2018.

M. Mirza and S. Osindero. Conditional generative adversarial nets, 2014.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

Wikipedia contributors. Cielab color space, 2023. URL `https://en.wikipedia.org/wiki/CIELAB_color_space`. [Online; accessed 27-April-2023].