Colorizing Grayscale Paintings using cGANs

Wesley Peisch, Luca Pistor, and Samarpreet Singh Pandher
CS 230
Stanford University

March 19, 2021

Abstract

Many of the world's greatest artworks will never be seen. Too often, artworks are neglected, lost, damaged, stolen, or simply disappear.[1] In some cases, though, we are lucky enough to have black-and-white photographs taken of paintings before their loss. Here, we attempt to use Deep Learning to recolor grayscale images of paintings and discover what such lost or damaged paintings would have looked like. We use a pix2pix implementation of Conditional GANs (cGAN) to colorize grayscale paintings and apply transfer learning in an attempt to improve colorization accuracy, evaluated by using RMSE and VGG-16 classifier accuracy. We found that transfer learning is not a viable strategy for our application, but found a set of hyperparameters that surpass the baseline level of model performance.

1 Introduction

In recent years there has been increasing interest in the area of using neural networks to colorize grayscale images.[2] These attempts have been motivated by the many and varied applications of image colorization, which range from historical image restoration to speeding up the workflow for animation studios.[11]

What sets our work apart from existing image colorization tasks is that we attempted to apply it to images of paintings. Existing models focus on digital photographs, which capture objects as they appear without the detail and texture of a painting. By focusing on painted works, we attempt to create a model that incorporates the nuances of the artist into the recolorization process.

The input to our model is an RGB image converted into the CIELAB color space (lightness, a*, b*). The Conditional GAN (cGAN) model accepts this image array and outputs a $256 \times 256 \times 3$ matrix representing CIELAB values for each pixel of the recolored image. This output image is then converted to RGB and rescaled to a $256 \times 256 \times 3$ matrix to be fed into our classifier.

Our VGG-16 classifier attempts to predict a painting's artist as a 10-class multinomial classification problem. Since our classifier has significantly higher classification accuracy on colored paintings than on grayscale paintings, we use it as an evaluation metric to determine colorization accuracy. We also use the RMSE between a recolored image and its original as a very simple supplementary metric for colorization accuracy.[9]

2 Related work

2.1 Model: cGAN

Here, we build on the work of Isola et al. (2016), who train a Conditional GAN (cGAN) as a general purpose solution for image-to-image translation, for diverse applications ranging from reconstructing objects from edge maps, image colorization, and synthesizing photos from label maps, among others. In general, for non-GAN implementations of these tasks, different hand-engineered loss functions, despite the fact that in each case, the aim is the same: predicting pixels from pixels.[3]

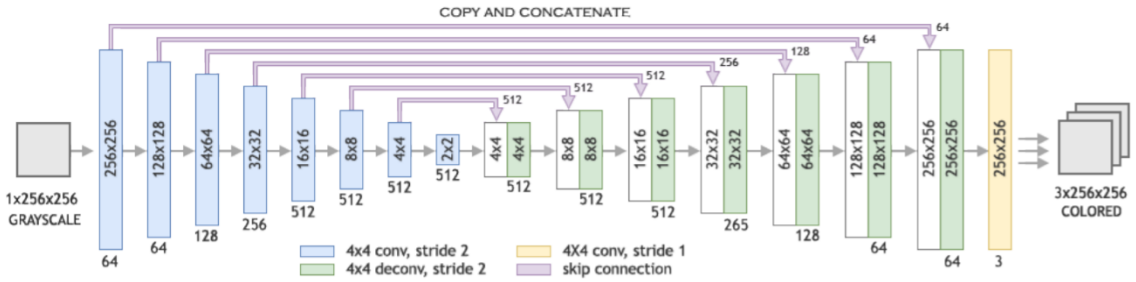
However, Isola et al. (2016) use cGANs to not only learn the mapping from input to output image, but also learn a loss function to train this mapping. For cGANs, we only need to specify a high-level goal, which is to make the output indistinguishable from the ground-truth image. cGANs automatically learn a loss function that tries to classify whether the output image is real or fake,

while simultaneously training a generator to minimize this loss.[3]

For the cGAN model, the condition is the input image, and the goal is to generate a corresponding output image. Isola et al. (2016) mention that traditionally, image-to-image translation tasks treat output as "unstructured" i.e. each output pixel is considered to be conditionally independent from other pixels, for an input image. The use of cGANs instead learns a "structured" loss - which penalizes any structural difference between input and output.[3]

Isola et al. draw from existing work by opting for a "U-Net" architecture for the Generator, and a convolutional "PatchGAN" classifier as the Discriminator. The PatchGAN discriminator is used convolutionally over the image and it penalizes structural difference at the scale of image patches, rather than working on the entire image at a time.[2]

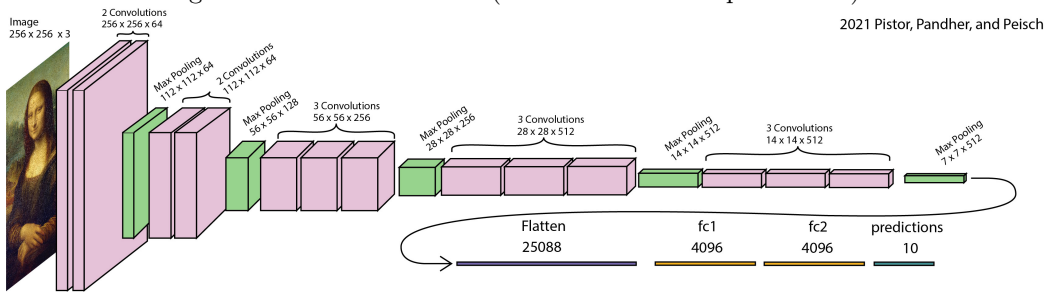
Figure 1: U-NET Architecture: Generator (256 x 256 Input) Nazeri et al. (2018)[2]



2.2 Evaluation Metrics : Classifier Accuracy and RMSE

As our primary evaluation metric, we use transfer learning on a VGG-16 classifier developed by Simonyan et al. (2015) that is trained on the 1000-class ImageNet Dataset.[5] We inherit the pre-trained weights for the first 18 layers of the model and freeze these as we train the last three layers on our data. We then evaluate the performance of this classifier on colorized grayscale paintings produced by our cGAN model. Since the classifier accuracy is far higher for colored paintings than it is for grayscale paintings (60% vs. 24%), we are able to use classifier accuracy as a proxy for accurate colorization.

Figure 2: VGG-16 Classifier(Modified for 10 output classes)[6]



3 Dataset and Features

Our dataset consists of 8000 images taken from a Kaggle project titled, "Best Artworks of All Time." We chose 10 European painters active between the Italian Renaissance and the 1850s who share a broad painting style, with most artists focusing on portraiture or landscape work. They span a subtle variety of styles, but are fundamentally similar in terms of composition and color use.

Based on our preliminary exploration, we noticed that recoloring abstract art was a more difficult task since abstract paintings feature many free-form shapes that don't provide context clues for how they ought to be colored. As such, we chose realistic paintings which feature a set of shared motifs. It is worth noting that none of our painters produced any works featuring nonwhite subjects – this allows our model to better predict accurate skin tones for the paintings that were chosen, but

it underscores the importance of choosing a dataset that is applicable to the problem space under study. If we had wanted our model to be used in a context where it would be expected to recolor *all* realistic paintings, rather than only those colored by European Renaissance and Baroque painters, then this would have been a poor choice of dataset.

We decided to focus the scope of our research to help us achieve more refined results for our specific use case. The artists chosen are listed in Figure 6 in Results. We used a per-artist 80:10:10 training/test/validation split to feed into our classifier as well as our model. Once split, we rescaled every image into a 256×256 square, stretching them with bicubic interpolation where necessary.
As in Isola et al., we converted images from the RGB colorspace into the CIELAB color space [7]. This was done so that the model could receive the L lightness channel, and use this to predict pixel values for the two remaining a* and b* channels. If the model were predicting RGB pixel values, it would need to predict three pixel values, constrained by a less straightforward equation.

Figure 3: Image pairs (input to classifier and model)

4 Methods

As in other GANs, the objective function of a cGAN can be expressed with the following formula:

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

The generator G tries to minimize this loss, while the discriminator D tries to maximize it.[3]
By adding an L1 loss term to the loss function of the generator, Isola et al. (2016) attempt to ensure that the generated image is similar to the ground-truth image, in addition to being able to fool the discriminator. This gives us an objective function that can be expressed as:

$$G^* = \underset{G}{\operatorname{argmin}} \underset{D}{\operatorname{argmax}} L_{cGAN}(G, D) + \lambda L_{L1}(G)$$

In this expression, λ is a coefficient that is used to balance the two terms in the loss equation. We explore the performance of our model under different values for λ[3]. We also explore the use of L2, rather than L1 loss, because of how L2 loss has been found to yield good results when applied to problem spaces with few outliers[13].
Similar to Isola et al. (2016), we use Conv-BatchNorm-ReLu layers for both our generator and discriminator, along with dropout at both training *and* test time, which they find to be an effective way to introduce the randomness that is necessary to generate realistic images.
As mentioned earlier, Isola et al.(2016) propose using a "U-Net" architecture for their generator. Since the conditional GAN is supposed to preserve structural information between the input and output images, the skip connections present in a U-Net architecture are intended to shuttle this information across the net.[3]

The PatchGAN discriminator tries to penalise structural differences at the patch-scale. That is to say, it tries to classify whether an $N \times N$ pixel area in the image is real or fake. This discriminator is run convolutionally over the entire image, and averages all responses over the image to provide final output D. We can consider PatchGAN as a form of texture/style loss.[3]

Our model learns using minibatch stochastic gradient descent, with a learning rate that we tune by trying different hyperparameter values. We use the Adam optimization algorithm, with the same momentum parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$ that were used in Isola et al. (2016). We train for 40 epochs in all tests, though when we explore transfer learning, the second training phase involves only a specific artist's works. The ratio between these two epochs is a hyperparameter that we tune experimentally.

Finally, the question of how to evaluate the performance of our model emerged. Zhang et al. (2016) propose that per-pixel evaluation metrics are not an ideal evaluation metric for colorization tasks, where one should look at the plausibility of a coloring rather than the accuracy of predicting the ground truth color. They propose evaluating the accuracy of a recoloring by using a pretrained VGG-16 classifier to classify objects, and then observing the difference in classification accuracy between ground-truth images and their recolorings.[7]

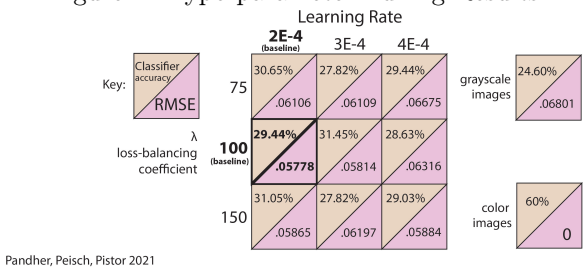
Building on this approach, we used transfer learning to have a VGG-16 classifier (pre-trained on the 'ImageNet' dataset [5]) predict a painting's artist from the 10 possible artist labels. By comparing the classifier's accuracy on recolored paintings to its accuracy on the ground-truth coloring and grayscale version of those paintings, we are able to arrive at an indicator of colorization accuracy.

To train our classifier, we replace the 1000-class output layer of the pre-trained VGG-16 network with a 10-class output layer to match the 10 artists present in our dataset. We trained the output layer along with the last 2 fully-connected layers for our prediction task, while freezing the weights for the other layers to the pre-trained weights of the 'ImageNet' dataset. Lastly, we also use RMSE as a standard quantitative measure of pixel-to-pixel colorization accuracy for our model.[9][10] Lastly, we also use RMSE as a standard quantitative measure of pixel-to-pixel colorization accuracy.[9][10]

5 Experiments/Results/Discussion

Testing different hyperparameter values, we found that $\lambda = 100$ and a learning rate of 3×10^{-4} resulted in a 6.8% higher classifier accuracy than the baseline levels of 100 and 2×10^{-4} for λ and learning rate. As discussed earlier, we consider classifier accuracy to be a better metric than RMSE or other pixel-to-pixel evaluation metrics, and so it was our primary target during development. Given the relative closeness between the classifier accuracy values across our tests, it is not certain that this pair of hyperparameters is the best. Similarly, since there is no trend as a function of learning rate or λ, it is unclear whether the variances are simply random chance.

Figure 4: Hyperparameter Tuning Results



However, upon visually inspecting the recolored photographs, we noticed that when using a learning rate of 2×10^{-4}, images were desaturated, and when using a learning rate of 4×10^{-4}, images were oversaturated; sometimes to the point of adding stray splotches of color to the recolored image. This guided our decision to conduct our additional tests with a learning rate of 3×10^{-4}.

When testing different values of λ, we found that higher values of λ produced more desaturated images with blurrier lines, while lower values would produce images with crisper lines but stray splotches of color. This makes sense when we consider the purpose of the λ parameter, which is to act as a balancing coefficient between L1 loss and cGAN loss. Examples of images produced by each of these models are included in the appendix.

We further trained this model on specific artists' works, intending that the model could learn general principles of painting recoloring in the first training phase, and then refine to a specific artist's style in training. Surprisingly, however, the model actually performed worse after further training on

Figure 5: With a learning rate of 4×10^{-4}, the recolored images are often miscolored and splotchy

a specific artist's works. Studying the output images from these 'artist-specific models' showed us that they tended to learn color distributions, rather than coloring techniques – an example included in the appendix shows how a model trained on Raphael would tend to place white blotches at the center of each painting, because this pattern was present in some of the training data. This suggests that the artist-specific datasets were too small to prevent this type of overfitting, and our data-augmentation strategies were insufficient to combat it. Thus, we've determined that artist-specific models are not a viable use case for transfer learning. It is possible that larger categories, such as *genres*, may be a better fit, but it is clear that most artists do not have enough works to properly retrain the model in the second training phase.

Figure 6: Classifier accuracy by artist with 10:30 epoch ratio & overall epoch ratio comparison

Artist	Caravaggio	El Greco	Delacroix	Courbet	Michelangelo
Classifier Accuracy	.168	082	.241	.167	.121

Rubens	Raphael	Rembrandt	Botticelli	Titian
.517	.362	.632	.541	.219

Pretraining epochs / Artist epochs	Classifier Accuracy
40 : 0	.3145
30 : 10	.3050
20 : 20	.2901
10 : 30	.2970

6 Conclusion/Future Work

In all, we tested two models – one proposed by Zhang et al. (2016), as well as a cGAN proposed in Isola et al. We began with the model proposed in Zhang et al., and decided to move to the Isola et al. architecture when a preliminary investigation revealed that the Zhang et al. architecture performed worse after we reduced our dataset to 10 artists, down from 80. We tuned three hyperparameters – learning rate, the λ term in the generator's loss function, and the ratio between both learning phases when transfer learning. We were able to find some pairs of learning rate and λ that improve the performance of the baseline model on our evaluation metrics, and we found that using transfer learning to tune our model to recolor a specific artist's paintings was not a successful strategy. We also experimented with L1 and L2 loss functions, and learned that an L2 loss function tends to normalize pixel colors, resulting in desaturated image outputs.

With more time to explore, further areas to explore include training our model on larger images as well as experimenting with other models. In the resizing process, a lot of data is lost. For object identification this is generally not a problem, but it is helpful to have high-resolution images of artwork to pick up on the nuances of a painter's brushstrokes. One way to do this is by tiling patches of a more detailed image, which would also help us detect how much an image patch's context (i.e. the pixels surrounding the patch) influences the coloration of that patch. Additionally, we would like to explore image inpainting–in many cases, paintings (or images of lost paintings) are damaged, and it is important for art historians to determine not just the colors of a painting, but an idea of what a missing section of a painting may have looked like.

Finally, unfreezing earlier layers of the classifier could make it more accurate; the frozen layers were trained on object detection, not artist identification, which is a slightly different task. By making the classifier more accurate, we might be able to see more robust trends in our hyperparameter comparisons. In all, we believe that we've made strong progress on the problem of painting recoloring.

7 Contributions

Designing the architecture, doing research, writing the reports, and discussing the project biweekly was important for us to all do together as a group. We made all engineering decisions together, but we implemented different parts of our strategy individually. Luca, who had the AWS credits, did the model training and evaluation, Samarpreet did much of coding the classifier, and Wesley did the manual painting selection and some data augmentation.

References

[1] https://www.dailyartmagazine.com/10-important-masterpieces-lost-ii-world-war/

[2] Nazeri, Kamyar, Ng, Eric, & Ebrahimi, Mehran *Image Colorization using Generative Adversarial Networks.* 2018, https://arxiv.org/pdf/1803.05400.pdf

[3] Isola et al. *Image-to-Image Translation with Conditional Adversarial Networks.* 2016, arXiv:1611.07004 2016

[4] Ziaee, A., Dehbozorgi, R., Doller, M. *A Novel Adaptive Deep Network for Building Footprint Segmentation.* 2021, arXiv:2103.00286v1

[5] Simonyan, Karen & Andrew Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition.* 2015, arXiv:1409.1556v6

[6] https://www.cs.toronto.edu/ frossard/post/vgg16/

[7] Zhang, R., Isola, P., & Efros, A. A. *Colorful Image Colorization.* ECCV, 2016

[8] Zhang, R., Zhu, J-Y., Isola, P., Geng, X., Lin, A. S., Yu, T. & Efros, A. A. *Real-Time User-Guided Image Colorization with Learned Deep Priors.* ACM Transactions on Graphics (TOG), 9.4, 2017, ACM

[9]An, Jiancheng, Kpeyiton, Koffi Gagnon, Shi, Qingnan, *Grayscale images colorization with convolutional neural networks*, 2020, Springer Soft Computing (2020) 24:4751–4758

[10] Ouni, Sonia, Zagrouba, Ezzeddine, Chambah, Majed, *A New No-reference Method for Color Image Quality Assessment.*2012. International Journal of Computer Applications (0975 – 8887) Volume 40– No.17

[11] F. Zhu, Z. Yan, J. Bu, & Y. Yu. *Exemplar-based image and video stylization using fully convolutional semantic features.* 2017, IEEE Transactions on Image Processing, VOL. 26, NO. 7

[12] Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-net: Convolutional networks for biomedical image segmentation.* In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

[13] Janocha, K., & Czarnecki, W. M. *On Loss Functions for Deep Neural Networks in Classification.* TFML, 2017

Appendix

Figure 7: With an L2 loss term, the recolored images end up overly desaturated

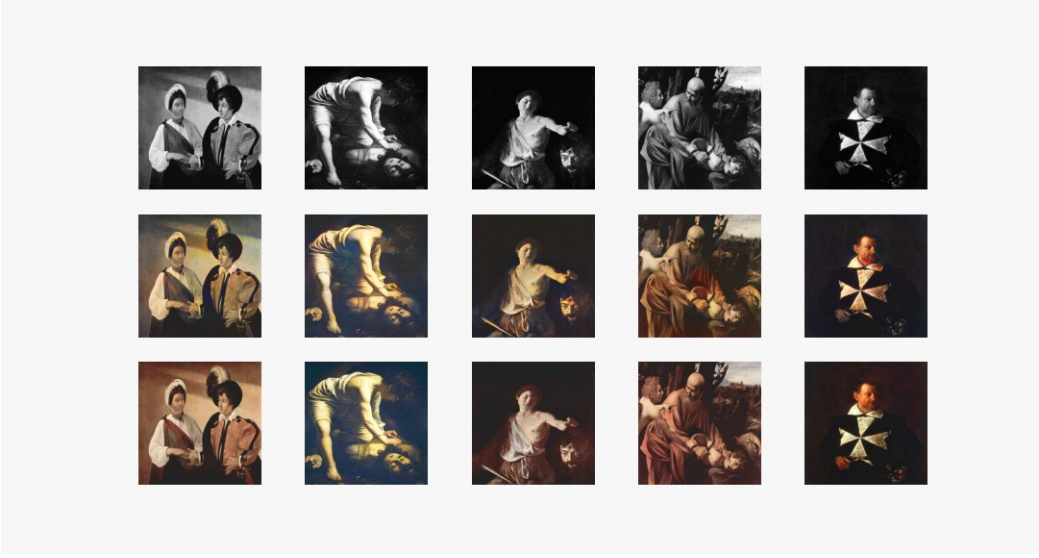


Figure 8: Artist-specific models generated through transfer learning tend to learn color distributions, rather than coloring techniques. In this example, a model trained on Raphael tends to place white blotches at the center of each painting, because this pattern was present in some of the training data.

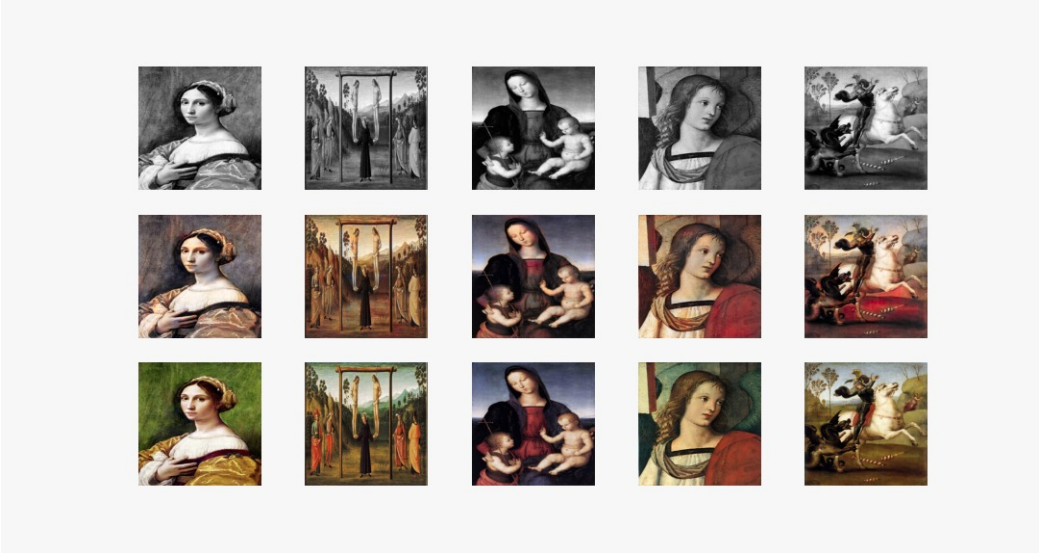


Figure 9: With a loss function of L2 loss, the recolored images end up overly desaturated

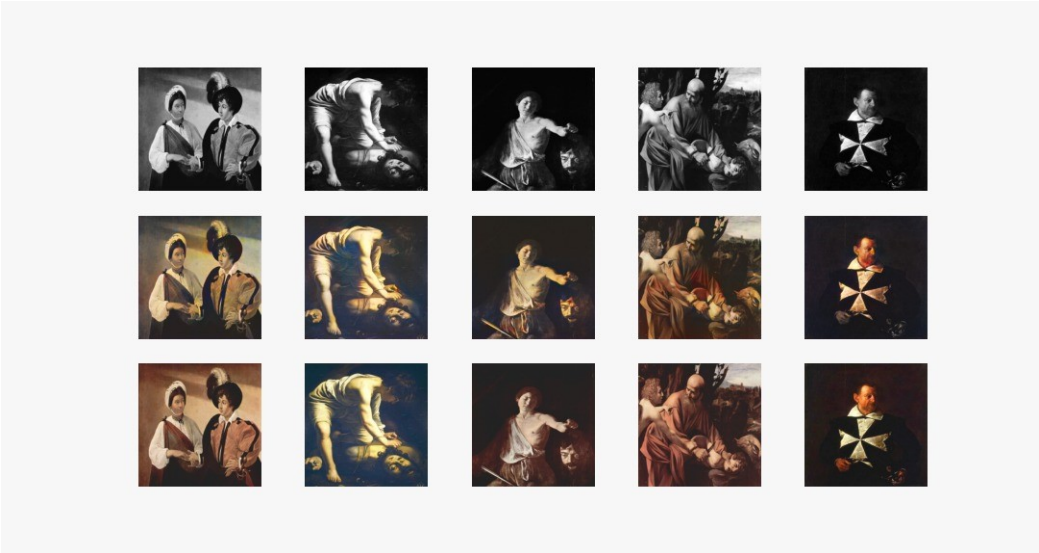


Figure 10: PatchGAN Discriminator : Ziaee et al. (2021)[4]