DL4CV - Project Report - Team: Error 404
Title: RevivaColor

Member 1: Edoardo Morresi, 3112831, edoardo.morresi@studbocconi.it
Member 2: Riccardo Valdo, 3112743, riccardo.valdo@studbocconi.it
Member 3: Lorenzo d'Imporzano, 3091665, lorenzo.dimporzano@studbocconi.it
Member 4: Benedikt Korbach, 3205523, benedikt.korbach@studbocconi.it

14^{th} December 2023

1 Introduction to the Problem

Have you ever wondered what your great-grandparents looked like on the day of their wedding? Or what your grandmother looked like on her first day of elementary school? Well, we did! This is why the objective of the following project is the colorization of monochromatic images, with a particular emphasis on photographs containing human subjects, through the application of semi/self-supervised learning techniques.

2 Problem Formulation and Relevance

Colorization, the process of changing black-and-white, sepia, or similarly monochromatic images into colored ones, is a task of significant relevance for a span of various domains such as media, entertainment, historical research, museum curation, and public safety. Current state-of-the-art, freely accessible tools, demonstrate satisfactory performance in coloring natural scenes, predominantly landscapes. However, they struggle to colorize images featuring human subjects. This is especially evident in the representation of skin tones, which often appear unnatural, even when using more specific tools that allow to select the category to which the input image belongs (e.g., in IMG2GO the user can select "People and Nature"). This issue is further aggravated in the context of historical photographs, particularly those depicting groups of people.
Therefore, this project specifically targets enhancing the colorization quality of images with people and aims at addressing the current limitations of colorization techniques. The significance of this task of people-centric image colorization is broad. Some potential applications include:

- Restoring family photographs and images of ancestors;
- Reconstructing imagery of historical figures, enhancing their authenticity and educational value;
- Colorizing black-and-white movies and historical footage to make them more appealing to today's audiences;
- Colorizing images taken in low-light conditions or with low resolutions, such as in the enhancement of black-and-white security footage.

3 Overview of the Main Contribution

As stated above, the main contribution of this project to the state-of-the-art is to enhance the performance of image-colorization techniques for human-centric images, as freely available tools seem to be lacking performance for this specific task. In particular, autoencoders and cGANs will be used to address this problem, as described in the following sections of this report.

4 Related Works and Previous Literature

The colorization of monochromatic images is a complex task within the field of image processing. Essentially, it involves the introduction of color to images where it originally does not exist, such as in grayscale pictures. This task becomes particularly challenging in scenarios where the colors to be applied are inherently ambiguous, for example, in images depicting colorful objects like groups of people.

The field of image colorization has developed significantly over the last decades, especially with the advent of digital imaging technology in the 1980s. Initially, colorization methods heavily relied on substantial manual input. These early techniques required users to actively participate in the coloring process, often involving detailed color suggestions and image segmentation, as seen in scribble-based methods. [1]
As the field evolved, there was a shift toward semi-automated techniques which included methods based on the K-Nearest Neighbors (KNN) algorithm [2] or luminance keying [3], which still required some degree of user involvement but represented a step towards greater automation in the process.
In recent years, the emergence of deep learning techniques led to substantial improvements in image colorization. Present-day techniques are increasingly automated, relying on sophisticated self-supervised deep learning algorithms. Common techniques include Autoencoders [4], Conditional Generative Adversarial Networks (cGANs) [5][6], and various other implementations of CNNs. These technologies have significantly enhanced the process of colorization, reducing the need for manual input and opening up new avenues for applications in historical photo restoration, artistic projects, and more. However, little work has focused on improving the colorization of images featuring human subjects. Hence, this project tries to fill this gap present in the literature.

5 Proposed Solutions

As the main solution, an autoencoder was built to perform the aforementioned task. **Autoencoders** are a subset of Neural Networks that can learn efficient representations of input data without the need for labels. In particular, autoencoders can be divided into two parts: the Encoder, which projects the image into a lower dimensional latent representation, and the Decoder, which translates this latent representation back to an image. Ideally, the Autoencoder should be able to disregard noise and capture the most salient features of the input images. To achieve this, Autoencoders are trained to minimize reconstruction errors. For the sake of this project's application, four different error measures were evaluated to find the best one to train the autoencoder (formulas in Appendix 1):

- **Mean Absolute Error**, which measures the average absolute difference between predicted and actual values, providing a single-value representation of the magnitude of prediction errors;
- **Mean Squared Error**, similar to the one above, but measures the squared difference instead of the absolute one;
- **Peak Signal-to-Noise Ratio (PSNR)**, which quantifies the quality of an image by taking the ratio between the maximum possible power of a signal and the noise that affects the fidelity of the representation;
- **Structural Similarity Index Metrics (SSIM)**, which provides a more comprehensive evaluation of perceptual image quality by considering luminance, contrast, and structure.

These four measures were selected so as not to have only pixel-by-pixel measures (MAE and MSE) but also measures based more on the visual perception of the image. Indeed, when considering pixel-by-pixel measures there is always the risk of ending up with images with similar loss but different perceptions (e.g., an image of a person with black squares over the eyes and mouth could possibly display the same pixel-by-pixel loss than a correct but underexposed image), PSNR and SSIM should help not to be falling into this fallacious trap.

Moreover, Autoencoders were employed in this project as these types of architectures are particularly well-suited for deconstructing and reconstructing images based on the most important features. As a matter of fact, the final goal of the project is to colorize the focal part of the input image, namely the human subjects.

On top of the Autoencoders, an attempt was made to build a **Generative Adversarial Network (GAN)** to try and improve the performance. In particular, a conditional GAN (cGAN) was developed. Here, the conditional version of these models is used as the final goal is not to generate a completely new image from scratch but rather to colorize a black-and-white one, resulting in generating a colored image "conditional" on a black-and-white canvas. GANs are characterized by the presence of a Generator and a Discriminator. The scope of the Generator is to learn how to generate new realistic images, and the one of the Discriminator is to learn to distinguish synthetic images from real ones. Ideally, if the Generator performs well, the Discriminator should encounter difficulties in distinguishing between synthetic and real images. By generating the image directly rather than just deconstructing and reconstructing it, as done by Autoencoders, the hope is to maintain a greater fidelity with the target image. The architecture of the cGAN employed in this paper was inspired by the one developed in the pix2pix application [7].
To train, validate, and test the aforementioned architectures, 4863 colored images containing human portraits were used [8].
The final results were then tested both mathematically, by comparing some of the losses above, and visually, namely by plotting a set of test images against each other and against the actual ground truth.

6 Experiments and Results

6.1 General Image Preprocessing

As a starting step, images were preprocessed. For both types of architectures, the following steps were carried out:

- Images were resized to a dimension of 256x256 as dealing with bigger images was not feasible due to hardware constraints;
- Images were transformed from the original RGB color space to the Lab color space. In this representation, the L dimension contains information about light intensity (ranging from 0 to 100), while channels a and b contain information about the colors (ranging from -128 to +128): magenta to green and yellow to blue, respectively. This representation was chosen since Lab works much like the human eye, and hence makes it possible to separate very close shades of colors. Moreover, using Lab allows only having to generate two outputs, the a and b components, which can then be concatenated with the lightness channel contained in the black-and-white input to form a proper Lab image which will then be converted to canonical RGB. Therefore, the L channel will be passed to the architectures and the final outputs requested will be the a and b channels;
- Values for the Lab channels were normalized by dividing the L channel by 100 and the a and b channels by 128 to ease the convergence when performing gradient-based optimization.

Moreover, histogram equalization was attempted but seemed to generate poorer results for both architectures. Hence, it was not included in the final algorithms as the equalization rendered the photographs artificial from the start, boosting shadows and over-saturating main colors.
Furthermore, the use of data augmentation techniques was not deemed appropriate as the dataset is already made up of several varied images and, on top of this, the maximum amount of memory was already consumed with the current dataset. On top of this, given the final scope of the project is to colorize portraits, it would be inappropriate, for example, to flip images horizontally as portraits should be oriented properly. Adding these kinds of unrealistic distortions to the data might have led to possible confusion within the architectures.

6.2 Autoencoder

After the model was built (as per displayed in in Figure 1 Appendix 2), the first attempt at training was done by compiling the model with the easiest and most canonical losses: MAE and MSE.

The first insight obtained was that the MAE does not seem to be the most efficient loss function, since the reconstructed pictures appeared blurry and the colorization did not appear entirely saturated (similar reasoning holds also for the MSE). As a first conclusion, it was thought that the problem was due to the low capability of the decoder to reconstruct the initial images, meaning that the model struggles to localize the main subjects of the pictures.

To try to improve the model and allow it to better forecast other shades of the ab spectrum, an attempt was made, as mentioned in section 6.1, to use histogram equalization. As above, this attempt was unsuccessful, prompting a new approach: white balancing. White balancing is the process of adjusting the colors in a picture to guarantee that white objects seem properly white, independent of the lighting conditions at the time the shot was taken. Indeed, varying light sources (such as sunshine, incandescent bulbs, fluorescent lights, and so on) produce different color temperatures, which can shade the overall colors of a photograph. White balance helps to eliminate these color casts, allowing the whites to seem natural and the other colors in the image to appear true, hopefully helping the model in predicting the correct colors for the reconstructions (Figure 2 in Appendix 2).

At this point, the autoencoder was also retrained with the other previously specified loss functions. However, the colorization performance did not reach the perfection in terms of realness; almost all subjects in the reconstructed pictures have been colorized, although generally clothes and background were often miss-predicted (Figure 3 in Appendix 2).

Given the above, a deeper analysis was conducted. Firstly, the value distributions of the RGB pixels of the reconstructed pictures against the originals were plotted. The plots appeared almost identical in terms of peaks, but the distribution of the reconstructed Lab pictures had lower variance, implying that the a and b channels did not investigate all conceivable color shades, resulting in a less accurate reconstruction (Figures 4, 5, 6 in Appendix 2). Secondly, a method to compare the actual and rebuilt ab channels was developed. In contrast to the initial conclusion, the autoencoder was found to extract relevant features from the images, such as faces, clothes, and body parts, albeit in a less precise manner. However, the reconstruction appeared unrealistic because the color channels tended to concentrate more around 0 than around the true ones. Knowing that the pictures are more prominent in reds/magentas and yellows and that the green and blue hues are provided by the negative extreme values of the a and b channels, the reconstructions tend to stick around the more frequent colors and the final images become duller (Figures 7, 8, 9 in Appendix 2).

Being aware that the decoder was reconstructing the original image pretty well in terms of predominant color, the ultimate goal became to improve the performances of the autoencoder for outlier colors (especially blue and green). By

modifying the structure of the autoencoder, more specifically by changing the activation function of the last layers of the decoders to a Leaky ReLU, the decoder can handle negative values during the convolutions more easily. The experiment proved slightly successful, in fact, the colorization now is more realistic, but they still retain some imperfections (Figures 10, 11, 12 in Appendix 2).

As a last attempt, a training with all the losses combined and Leaky ReLU for the last three layers of the decoders was performed. Combining only the PSNR and the SSIM loss did not seem a more accurate strategy since the SSIM is a more grounded metrics for quality reconstruction of images. No theoretical foundation lies behind the choice of combining all three losses, however it is a heuristically robust choice, since the losses may favour the same colors but penalise differently the mistakes. Unfortunately, the performances proved what expected without any surprising result: the colorization is even more precise when it comes to colorizing the faces of the subject but now is much more conservative on other elements, outputting a neutral color.

A completely different path would be to use a pretrained model to exploit larger architecture and training set, but unfortunately many wildly known networks available in Keras are trained on RGB images. An apparently suitable (but suboptimal) solution would be to triplicate the input L channel to match the training input of those NNs, but then the problem of data preprocessing comes up.

After these considerations, the models have been tested on a truly historical and the results were actually surprising (Figures 13, 14, 15, 16 in Appendix 2). Considering the technical limitations that the autoencoders have, combined with the dataset and resources constraints, a different and more powerful model was attempted: Conditional Generative Adversarial Networks.

6.3 cGAN

cGANs are undoubtedly more powerful models for addressing generative tasks and are commonly used even for more complex image translation problems. Given the difficulty of the task of implementing cGANs from scratch, this paper builds over the results obtained in the pix2pix paper [7]. Considering that RevivaColor's goal was more confined, some modifications were made compared to the pix2pix structure. The aim was to assess, with the limited computational power and dataset available, the degree of improvement that such an architecture would bring compared to this present paper's main solution: the aforementioned Autoencoders. Modifications made to the architecture involved lowering the Dropout, reducing the depth of the U-Net, and adjusting the receptive field (from 70x70 to 16x16) used in the PatchGAN. These changes were made since the problem at hand implies a more limited and less generative task than the one addressed by the original authors since the aim of this paper is to colorize images and not to fully generate them from scratch. In particular, the reduced receptive field was useful for providing a more accurate loss on smaller portions of the image to account for faces and other smaller body parts. Additionally, a pretrained approach using VGG16 was attempted for the "encoder" section of the U-Net to overcome the limitations due to the size of the dataset. However, this approach was not successful since, as specified also above, all the ImageNet pretrained models required three channels for input, while black-and-white images only contain one. Several attempts were made to adapt the input (e.g., replicating the Lightness channel three times for the input picture to keep it in the RGB color space) but none caused acceptable results and hence was not included in the final model. This could be due to the limitations posed by using a pretrained model who is suited for RGB pictures, such as the ones available on Keras, and that moreover has its specific pre-processing of images that might not match the one of this analysis.

Still, the overall results obtained with the cGAN architecture were satisfactory, as will be later discussed in the results' section.

The model structure and the results obtained with the approach explained above are available in Appendix 2 (Figures 17-23).

7 Results and Conclusions

Overall, the results obtained were satisfactory. Indeed, both architectures, Autoencoders and cGAN, provided good colorization performance and also the tests run on real historical black-and-white images provided visually pleasant results. Clearly, these results are not yet optimal, likely due to time and computational power limitations, as will be later explored in the following section.

More in-depth, as can be seen from the metrics computed on the test set, displayed in Figure 24 of Appendix 2, the Autoencoders seem to perform slightly better than the cGAN when it comes to all metrics used (remembering that PSNR and SSIM indicate the similarity between pictures and hence the higher the better). This could be due to the fact that cGAN, due to its structure, probably needs more input images to be performing at its best potential. Moreover, the cGAN, following the pix2pix paper, was trained on a mix of (100x) L1 and (1x) Adversarial Loss, and not specifically considering PNSR or SSIM.

Still, visually, results for both architectures' types seem satisfactory and provide an improvement with respect to the current literature about freely available colorization tools.

8 Limitations and the Way Forward

Overall, this project represents a satisfactory first approach towards the enhancement of colorization techniques for human portraits. Still, significant work and further improvements are possible. Surely, hardware constraints played a key role in limiting the possible upsides of the approaches applied. Indeed, constrained by GPU and RAM limitations, it was not feasible to use a larger dataset or images with a higher resolution than 256x256. Hence, a possible step forward would be to try and train the architectures proposed in this paper with more powerful computational tools.

Moreover, further studies and optimization of the architectures, especially the cGAN, or building different architectures using, for instance, Stable Diffusion, can surely help improve the results achieved in this paper.

On top of this, it would also be interesting to train the cGAN based on SSIM or PSNR (in place of L1 loss) to see whether the performance improves.

Unfortunately, due to time and hardware constraints, further investigation in these directions was not viable.

9 References

1. Levin, A., Lischinski, D., & Weiss, Y. (2004). Colorization using optimization. ACM Transactions on Graphics, 23(3), 689-694.

2. Irony, R., Cohen-Or, D., & Lischinski, D. (2005). Colorization by example. In Proceedings of the Eurographics Symposium on Rendering (pp. 201-210).

3. Sýkora, D., Buriánek, J., & Zára, J. (2004). Unsupervised colorization of black-and-white cartoons. International Symposium on Non-Photorealistic Animation and Rendering.

4. Zaware, S., Pathak, D., Patil, V., Sangale, G., & Gupta, V. (2021). Gray Scale Image Colorization for Human Faces. 2021 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), 107-110.

5. Jolly, V., Dwivedi, M., Patel, Y.J., Yadav, T., & Dhage, S. (2023). Bringing Monochrome to Life: A GAN-based Approach to Colorizing Black and White Images. 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), 1-6.

6. Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).

7. pix2pix: Image-to-image translation with a conditional GAN. (n.d.). TensorFlow. https://www.tensorflow.org/tutorials/generative/pix2pix?hl=en

8. Cornell University, The images of groups dataset. (n.d.). http://chenlab.ece.cornell.edu/people/Andy/ImagesOfGroups.html

10 Appendix

10.1 Appendix 1 - Loss formulas

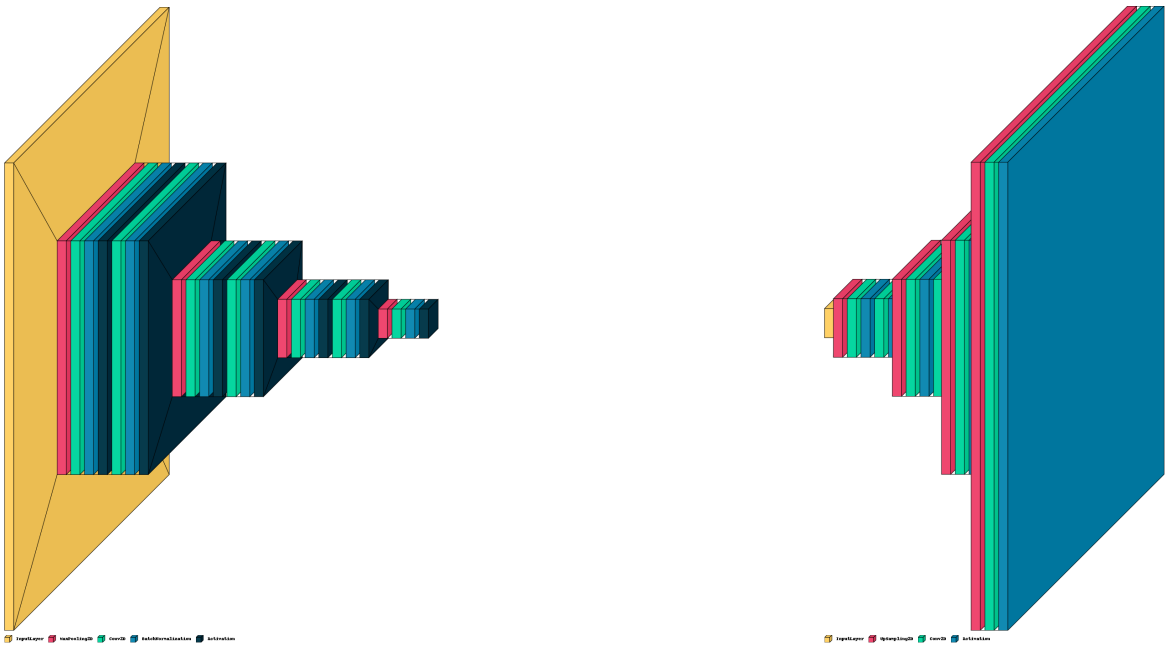
- MAE$(x, y)= \frac{1}{N}\sum_{i=1}^{N}|x_i - y_i|$
- MSE$(x, y)= \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2$
- PSNR$(x, y)= 10 \cdot log_{10}\left(\frac{MAX(I)^2}{MSE}\right)$

  Where $MAX(I)$ are the maximum values of the pixels that the images under analysis can take.

- SSIM$(x, y) = \frac{(2\mu_x\mu_y+C_1)+(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)}$

  Where μ_x and μ_y are the pixel sample means respectively of the windows x and y (of common size NxN), σ_x^2 is the variance of x, σ_y^2 is the variance of y, and σ_{xy} is the covariance between x and y. Moreover, C_1 and C_2 are two variables to stabilize the division with weak denominator. More in depth, $C_1 = (k_1L)^2$ and $C_2 = (k_2L)^2$ where L is the dynamic range of the pixel-values (typically this is $2^{number\ of\ bits\ per\ pixel} - 1$) and $k_1 = 0.01$ and $k_2 = 0.03$ by default

10.2 Appendix 2 - Models and Results



(a) Encoder model.

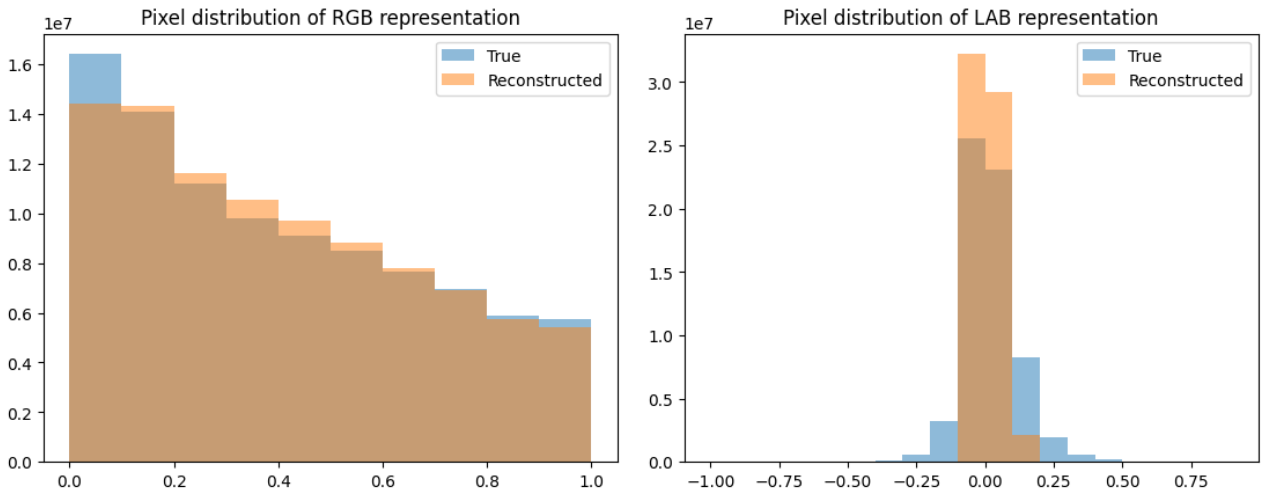
(b) Decoder model.

Figure 1: VisualKeras representation of the autoencoder

Figure 2: Comparison between original and balanced images.

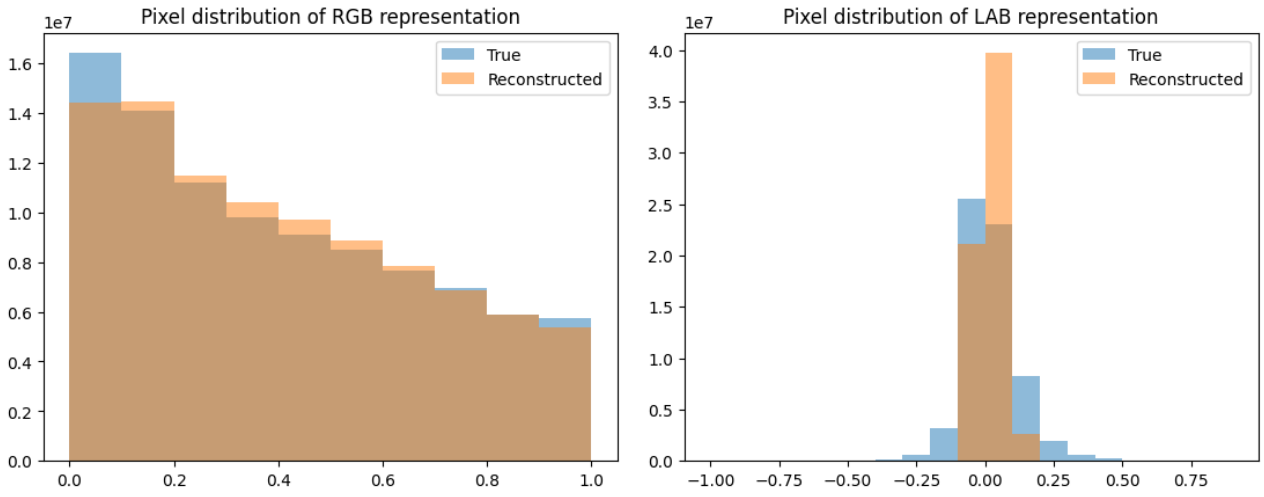
(a) Reconstructions under MSE loss. (b) Reconstructions under PSNR loss. (c) Reconstructions under SSIM loss.

Figure 3: Comparison of reconstructions under different losses.



(a) Distribution of RGB pixels. (b) Std Distribution of LAB pixels.

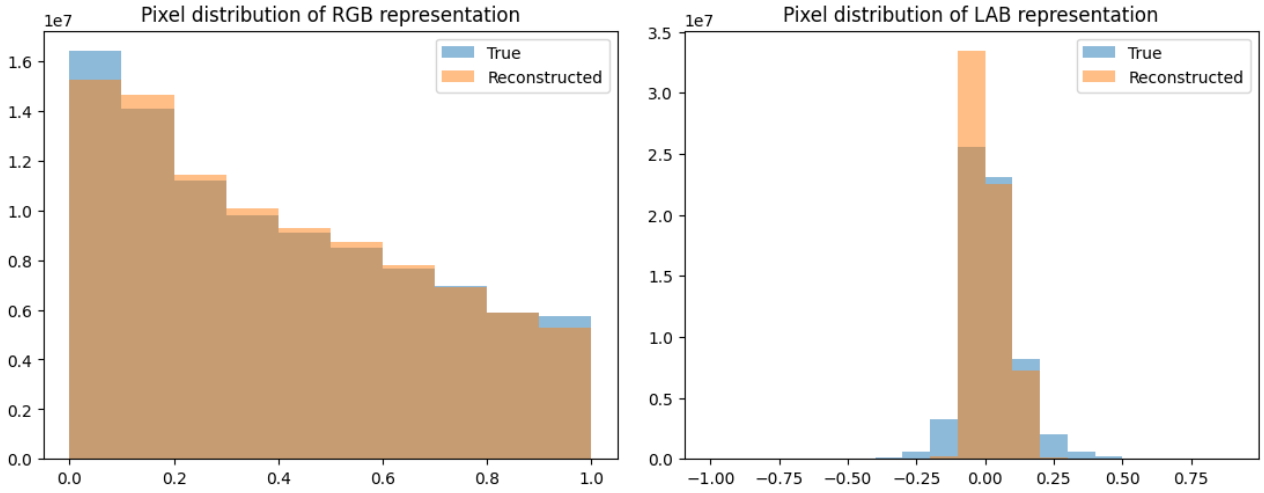
Figure 4: Recostructions under MAE.



(a) Distribution of RGB pixels.

(b) Std Distribution of LAB pixels.

Figure 5: Reconstructions under PSNR.



(a) Distribution of RGB pixels.

(b) Std Distribution of LAB pixels.

Figure 6: Recostructions under SSIM.

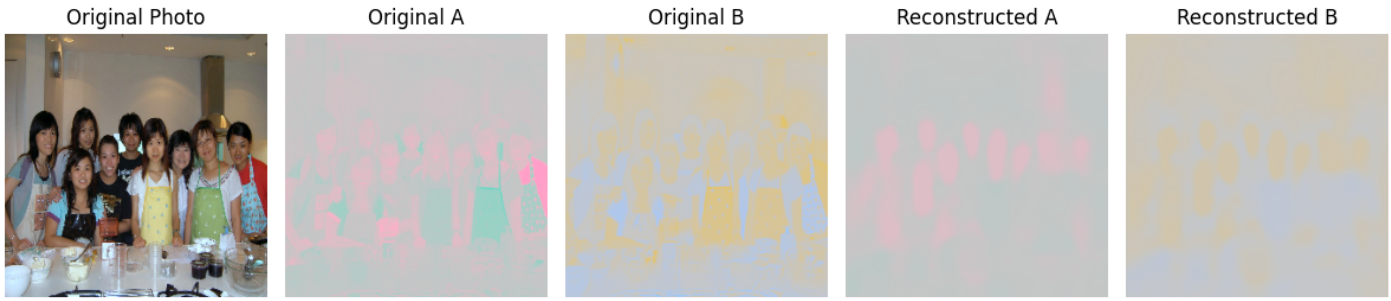


Figure 7: Channel-wise reconstruction under MAE.

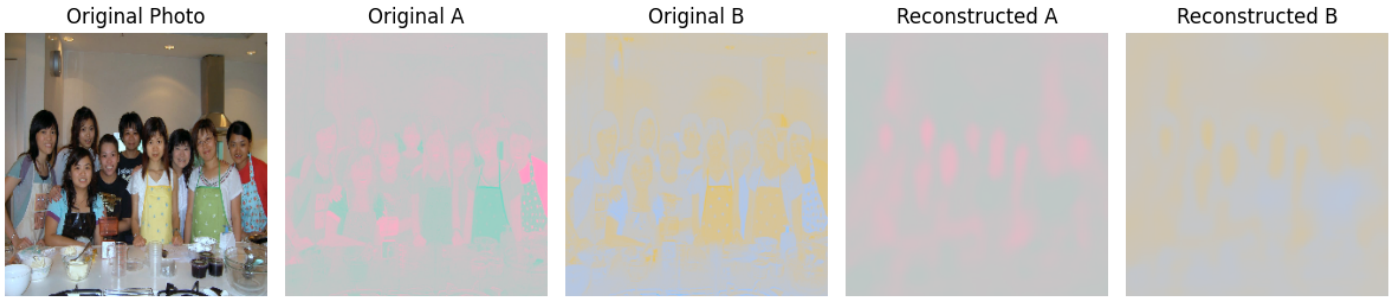


Figure 8: Channel-wise reconstruction under PSNR.

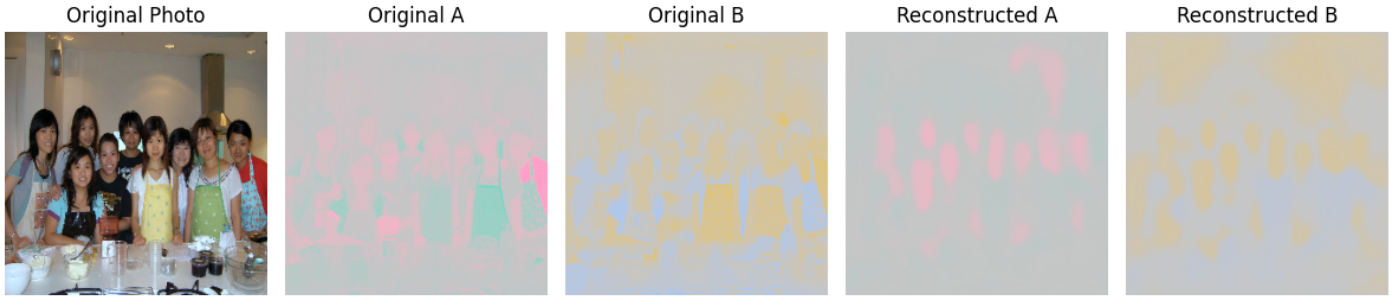
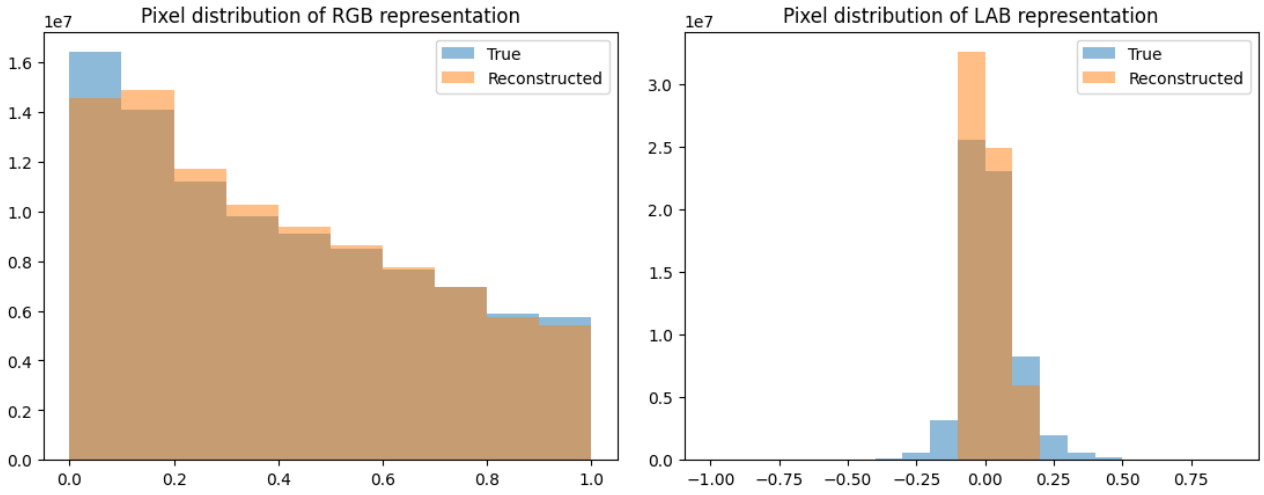


Figure 9: Channel-wise reconstruction under SSIM.

Figure 10: Reconstructions under SSIM loss with LeakyReLU.



(a) Distribution of RGB pixels.

(b) Std Distribution of LAB pixels.

Figure 11: Recostructions under SSIM with LeakyReLU.

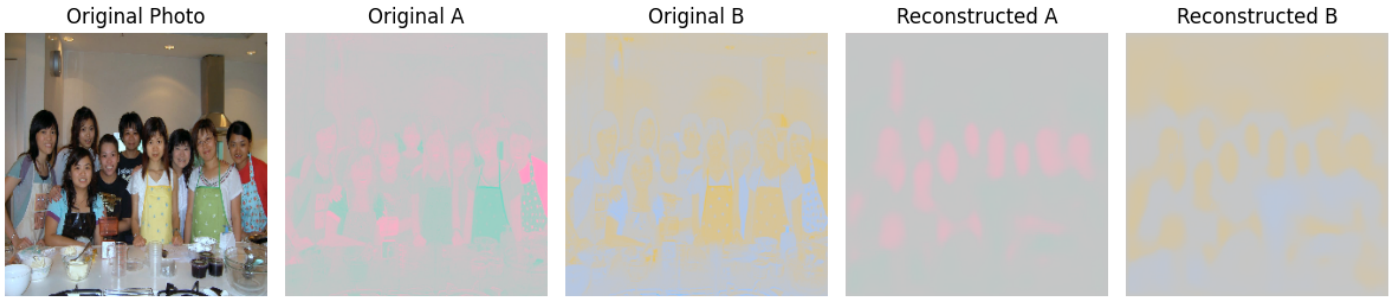


Figure 12: Channel-wise reconstruction under SSIM.

Figure 13: Test on a real image.

Figure 14: Channel-wise reconstruction under MAE of the real image.

Figure 15: Channel-wise reconstruction under PSNR of the real image.

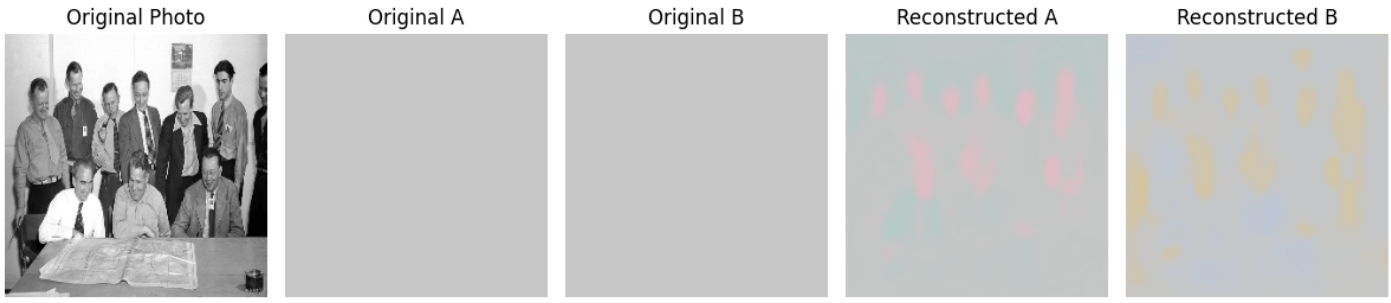


Figure 16: Channel-wise reconstruction under SSIM of the real image.

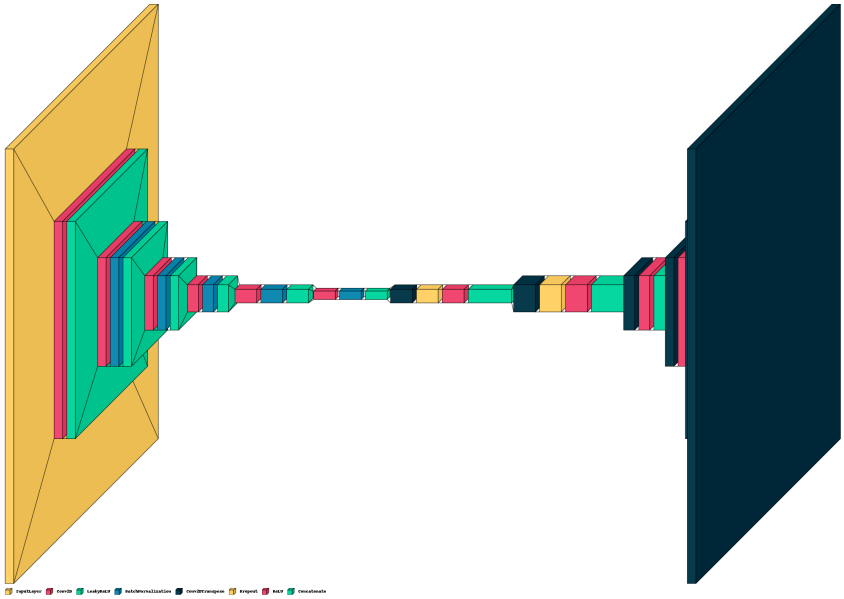


Figure 17: UNet-like generator.

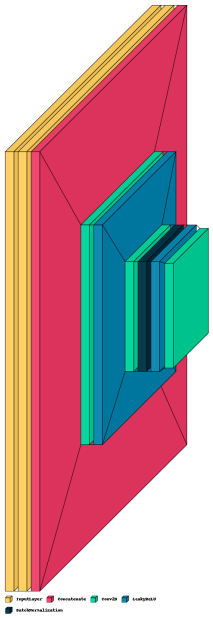
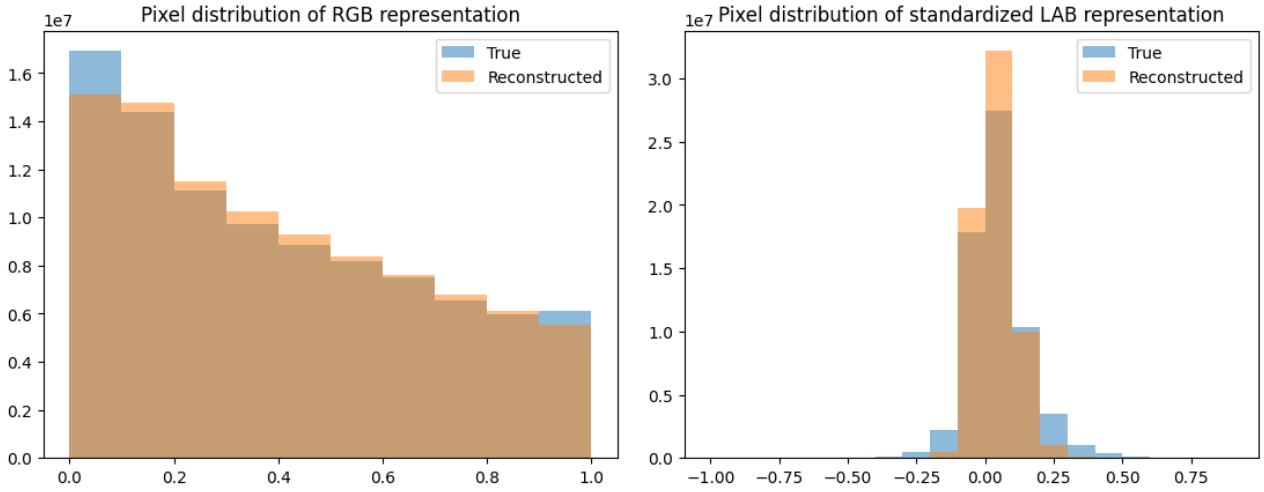


Figure 18: Discriminator with 16x16 Receptive Field.

Figure 19: Reconstruction using cGAN.



(a) Std distribution of RGB pixels.

(b) Std Distribution of LAB pixels.

Figure 20: Reconstructions using cGAN.

Figure 21: Channel-wise reconstruction using cGAN.

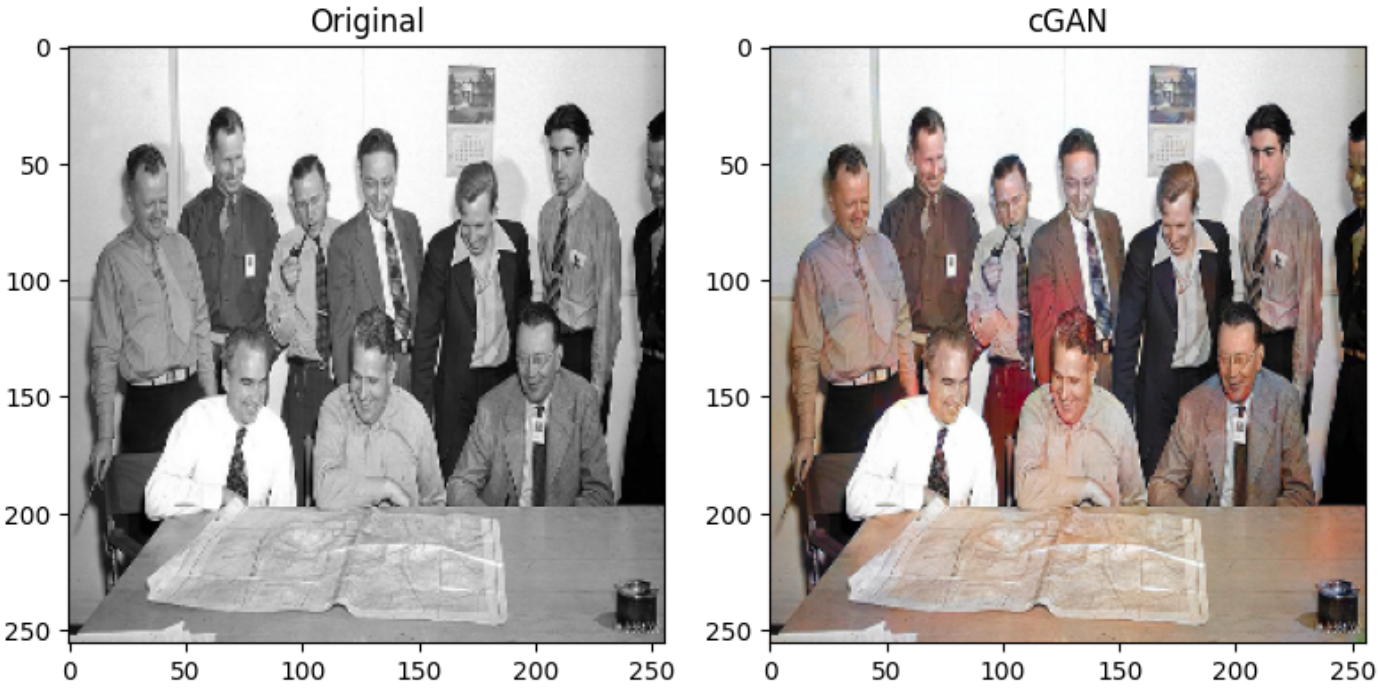


Figure 22: Test on a real image using cGAN.

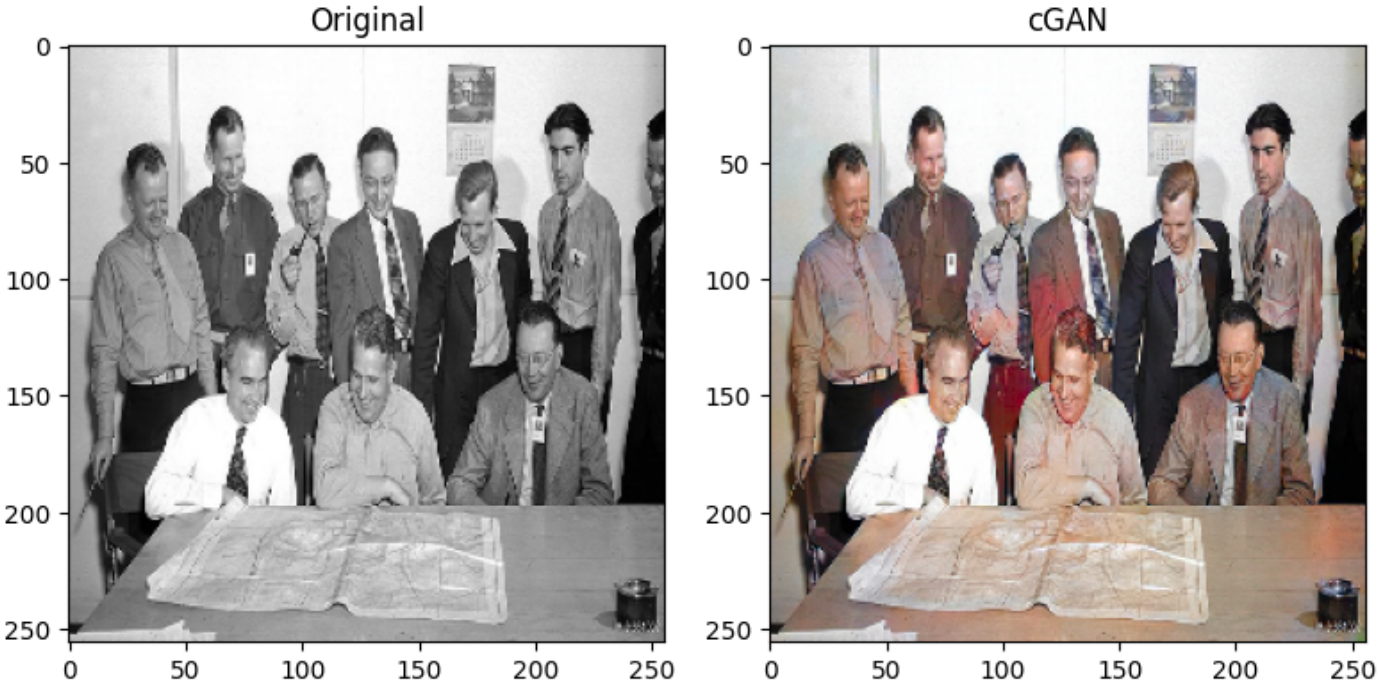


Figure 23: Channel-wise reconstruction of the real image using cGAN.

Architecture	Metric	Value
Autoencoder_trained_on_MAE	MAE	0.058
	SSIM	0.423
	PSNR	27.908
Autoencoder_trained_on_SSIM	MAE	0.060
	SSIM	0.427
	PSNR	27.57
Autoencoder_trained_on_PSNR	MAE	0.059
	SSIM	0.395
	PSNR	27.874
Autoencoder_trained_on_SSIM+Leaky_ReLu	MAE	0.059
	SSIM	0.435
	PSNR	27.684
cGAN	MAE	0.069
	SSIM	0.411
	PSNR	26.477

Figure 24: Losses computed on test set for different models