Diving into Depth: Image Colorization in Cartoon and Live Action Frames

Enzo Camizan, Sofia Garcia, Valeria Espinoza
Department of Computer Science
University of Engineering and Technology
{enzo.camizan,sofia.garcia,valeria.espinoza}@utec.edu.pe

Abstract

This study looks into cross-domain image colorization, a field that is largely unexplored in current research. We trained and evaluated three neural networks using datasets from both cartoon frames and real movie stills using Generative Adversarial Networks (GANs) (Goodfellow et al., 2014). We wanted to create adaptable models capable of transferring colorization skills between these distinct domains by using GANs, specifically UNet (Zhang et al., 2019) and ResNet34 (Zhang & Schaeffer, 2018) architectures. Our methodology included an in-depth evaluation using metrics such as CIEDE2000 (Sharma et al., 2005) and SSI (Wang et al., 2004), which demonstrate the potential of our approach. This aims to bridge the gap in cross-domain colorization while also shedding light on the benefits of multi-domain training for neural networks in image colorization tasks.

1 Introduction

Color is a key aspect of both human and computer visual processing, with applications in digital imaging, computer vision, and entertainment (Trémeau et al., 2008). This is why **image colorization** is so popular: it allows the enhancement and manipulation of visual content by adding color.

Despite several investigations discussing real-life image colorization (Ironi et al., 2005; Bugeau & Ta, 2012; Cheng et al., 2015) and cartoon colorization (Sýkora et al., 2005; Varga et al., 2017; Chybicki et al., 2019; Chen et al., 2020), none of them explore the development of neural networks capable of colorizing across both domains. Researchers typically use different processing methods for different input images: grayscale images contain more information, while sketch and cartoon images commonly have sparse details (Chen et al., 2022). As a result, there is insufficent data to assess the generalization capabilites of image colorization neural networks from real-life to cartoon domains (and viceversa).

This study aims to address this gap. Our methodology involves training three neural networks: two exclusively in one domain (either cartoon or live action), and a third model using both datasets. The training will involve the implementation of **Generative Adversarial Networks** (GANs), as recent advancements in GANs for image colorization have outperformed other neural networks (Jiang et al., 2022; Zhou & ichiro Kamata, 2022; Sankar et al., 2020; Treneska et al., 2022), particularly in cartoon colorization (Fu & Hsu, 2017; Sun et al., 2021). Subsequently, we will evaluate the performance of the models across both domains using methodologies such as CIEDE2000 (Sharma et al., 2005) and Structural Similarity Index Wang et al. (2004) to assess the accuracy of the colorized images.

Our investigation is expected to validate the efficacy of GAN-based approaches for cross-domain image colorization. We want to study the adaptability and transferability of image colorization neural networks trained in specific domains. Moreover, our research aims to shed light on the potential benefits of using multi-domain training data for neural networks in image colorization.

Final Paper for the ML3061-2023.2 - Machine Learning Class at UTEC. Lima, Peru.

2 Methodology

2.1 Model Architecture

As mentioned earlier, we employed *Generative Adversarial Networks* to build our models. GANs have a dual structure, comprising a **generator** G and a **discriminator** D, both engaged in simultaneous training through a competitive dynamic. The goal of this process is the continual enhancement of the model's generative capabilities. We present a detailed explanation of each component within the implemented GAN structure in our project as presented by the original work of Goodfellow et al. (2014) in Appendix A.1.

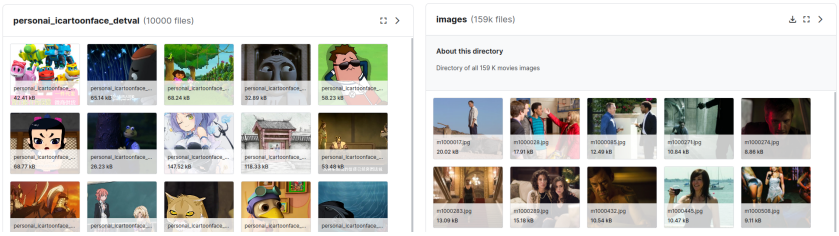
The generator is a pivotal component, tasked with the key mission of synthesizing data that closely resembles real-world data, to the point that they are indistinguishable. Its primary responsibility is creating coherent and visually authentic images. The effectiveness of the generator is determined upon the quality of its architecture, which plays a critical role in the model's ability to learn patterns and essential features from reference images during the training process.

In this context, we opted to implement the *UNet model* (Zhang et al., 2019), renowned for its high efficiency in data manipulation and the generation of convincing visual representations. Additionally, we integrated a *Backbone ResNet34* (Zhang & Schaeffer, 2018) into the process. For subsequent experiments, more robust backbones should be considered to explore and enhance the system's performance.

The discriminator holds a crucial role within GANs, serving as a central component in discerning between generated and real data. Given a specific neural architecture, the discriminator looks to refine its ability to differentiate, continuously learning patterns that help in providing increasingly accurate responses.

2.2 Datasets

We gathered two different datasets. The first set has 10,000 cartoon frames with a wide range of cartoon styles (Car, 2023). Thus, our program can learn how to add colors to cartoons in various ways. The second set, called *Movie Stills 2000-2020* (Mov, 2020), has 159,000 images from several movies. These frames come from various movie genres, giving our models a chance to learn about the colors used in different scenes. The selection and composition of these datasets is due to our commitment to train models that excel in both realism and stylized representation.



(a) Cartoon Dataset (b) Movie Dataset

Figure 1: Screenshots of Kaggle datasets with metadata and sample image previews.

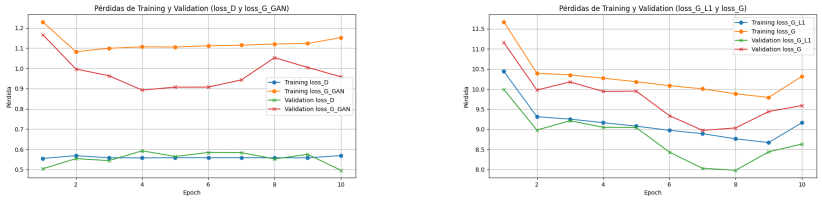
Our training datasets consisted of a total of 8000 images for the training set and 2000 images for the testing set. This substantial dataset size aimed to expose the models to diverse scenarios and enhance their generalization capabilities across both cartoon and live-action domains. For the third model, the dataset was evenly divided between the training and testing sets, with an equal number of images to ensure balanced representation in both datasets.

2.3 Experimentation

The project was executed on the Kaggle platform, leveraging the computational power of an NVIDIA GPU T4 with 14.8 GB of dedicated memory. This hardware configuration, combined with 29 GB of RAM, provided the necessary resources for training our neural networks efficiently. For the training

process, we employed two loss functions: **L1 loss** and **Generative Adversarial Network (GAN)** loss using BCEWithLogitsLoss. L1 loss ensures that the generated images are visually close to the ground truth, while GAN loss adds an adversarial component to the training, enabling the network to create more realistic and visually appealing colorizations (Mahadik et al., 2020). Another values we took into account where D_Loss and G_Loss respectively, which are the loss functions for the discriminator and generator respectively. The optimization of our neural networks was carried out using the **Adam optimizer**, a popular choice in machine learning due to its efficiency in handling non-stationary objectives and noisy gradients (Kingma & Ba, 2014).

Throughout the experimentation phase, we monitored key performance metrics, including the convergence of loss functions and the visual quality of generated images. After analyzing these metrics, we chose the model with the least overall loss and that presented visually appealing results. In the case of the model trained with both datasets, the models at epoch 8 and 9 were considered (Figure 2), with the final selection being model 9.



(a) D_Loss and G_GAN Loss (b) G_Loss and G_L1_Loss

Figure 2: Comparison of the validation and training loss metrics for the final model.

The utilization of a GAN-based approach, coupled with a carefully designed architecture incorporating UNet and ResNet34 models, reflects our commitment to pushing the boundaries of image colorization (V et al., 2023). By introducing a third model trained on both cartoon and live-action datasets, we aim to explore the adaptability and transferability of neural networks in the context of image colorization, contributing valuable insights to the field (Yang et al., 2018). The formalized experimentation process ensures the robustness and reliability of our results, laying the foundation for advancements in cross-domain image colorization techniques.

(a) Epoch 1 (b) Epoch 9

Figure 3: Comparison of the initial vs best epoch while training the final model.

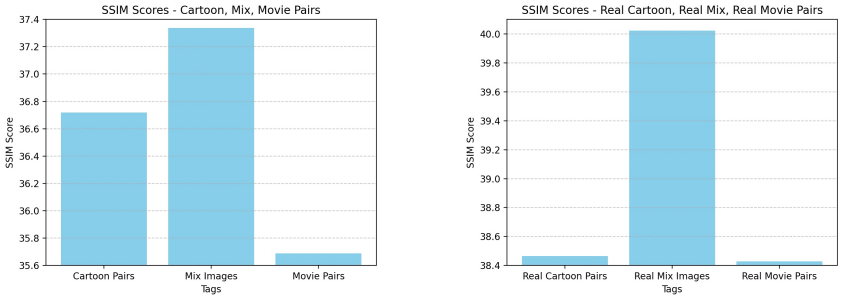
3 Results

In the context of colorization, it is essential to ensure that colors are consistent and meaningful, as they may appear visually correct but not accurately represent the intended colors. In the colorization process, the CIEDE2000 (Sharma et al., 2005) metric plays a fundamental role due to its ability to accurately quantify differences in color perceived by the human eye. It is particularly valuable when aiming for a faithful reproduction of real colors, enabling an objective evaluation of colorization quality. We have an in depth explanation in Appendix A.2.

For the Figure 4a, the average score for the dataset trained with mixed images is higher compared to other scores, with it consistently yielding better results than the other two datasets. Additionally, in instances where a cartoon dataset was present, it demonstrated superior performance compared to

the movie dataset. Consequently, we can infer that mixed datasets are likely to exhibit higher scores compared to other datasets.

Regarding Figure 4b, it is apparent that it follows the same trend: the mixed dataset exhibits significantly higher scores compared to others, reflecting outstanding performance. The metric employed compares the given features in space, aligning well with the provided datasets.

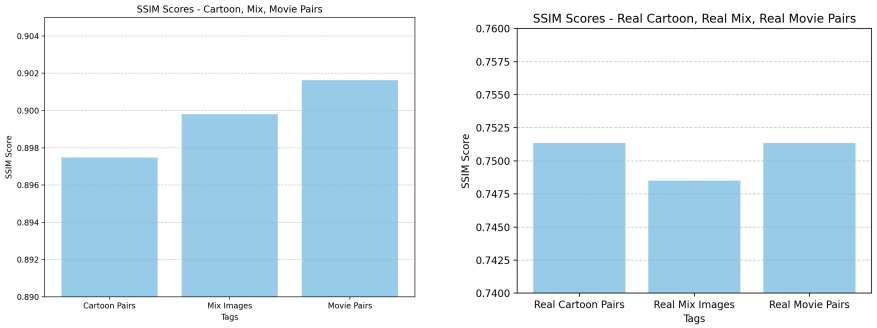


(a) CIEDE2000 Score using Cartoon images (b) CIEDE2000 Score using Real images

Figure 4: Comparison of CIEDE2000 Score using Cartoon vs Real images.

Besides ensuring accurate colors, evaluating the structural integrity of colorized images is equally crucial. The **Structural Similarity Index (SSIM)** Wang et al. (2004) is a metric widely adopted for this purpose. Unlike traditional metrics that only focus on pixel-wise differences, SSIM considers the luminance, contrast and structure of images. For a deeper understanding review Appendix A.3.

As we can see in the image below, when using Cartoon images as input, our model using the Movie has a higher SSIM Score. In the other hand, using real images as input makes our Mix model the worst overall. Given that the index structure changes, the color placing is worst compare to CIEDE2000 score. Nevertheless, the differences among the scores is fairly minimum, in contrast to the said score.



(a) SSIM Score using Cartoon images (b) SSIM Score using Real images

Figure 5: Comparison of SSIM Score using Cartoon vs Real images.

3.1 Discussion

While a higher CIEDE2000 score implies more color accuracy, a lower SSIM suggests potential issues in the coloring process, possibly attributed to nuances within the GAN architecture itself. The minor differences in SSIM scores across models indicate subtle variations in image structure, prompting further investigation into the underlying mechanisms affecting colorization quality.

Our findings highlight the need of further research in order to test more robust neural networks used recently in colorization, such as Deep Convolutional Neural Networks, to observe if results are applicable to DCNNs as well. Furthermore, we observed that a general flaw of the colorized images produced by our models was a lack of saturation in Appendix A.4. This is something we would like to improve in the future in order to be able to analyze our results with all three models at their best.

References

Movie stills 2000-2020 images, 2020. URL https://www.kaggle.com/datasets/kleinertee/cartoon.

cartoon, 2023. URL https://www.kaggle.com/datasets/kleinertee/cartoon.

Aurélie Bugeau and Vinh-Thong Ta. Patch-based image colorization. *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 3058–3061, 2012. URL https://api.semanticscholar.org/CorpusID:7253996.

Shu-Yu Chen, Jia-Qi Zhang, Lin Gao, Yue He, Shi hong Xia, Min Shi, and Fang-Lue Zhang. Active colorization for cartoon line drawings. *IEEE Transactions on Visualization and Computer Graphics*, 28:1198–1208, 2020. URL https://api.semanticscholar.org/CorpusID:225533218.

Shu-Yu Chen, Jia-Qi Zhang, You-You Zhao, Paul L. Rosin, Yu-Kun Lai, and Lin Gao. A review of image and video colorization: From analogies to deep learning. *Vis. Informatics*, 6:51–68, 2022. URL https://api.semanticscholar.org/CorpusID:253422472.

Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 415–423, 2015. URL https://api.semanticscholar.org/CorpusID:64884.

Mariusz Chybicki, Wiktor Kozakiewicz, Dawid Sielski, and Anna Fabijańska. Deep cartoon colorizer: An automatic approach for colorization of vintage cartoons. *Eng. Appl. Artif. Intell.*, 81:37–46, 2019. URL https://api.semanticscholar.org/CorpusID:86712423.

Qiwen Fu and Wei-Ting Hsu. Colorization using convnet and gan. 2017. URL https://api.semanticscholar.org/CorpusID:49476592.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems*, 2014. URL https://api.semanticscholar.org/CorpusID:261560300.

Revital Ironi, Daniel Cohen-Or, and Dani Lischinski. Colorization by example. In *Eurographics Symposium on Rendering*, 2005. URL https://api.semanticscholar.org/CorpusID:818126.

Shuyang Jiang, Qiuyi Luo, and Huaitian Bi. Image colorization : Gan and autoencoder models' performance comparison. *2022 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE)*, pp. 367–372, 2022. URL https://api.semanticscholar.org/CorpusID:253555737.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL https://api.semanticscholar.org/CorpusID:6628106.

Ranjit Mahadik, Rashi Ryapak, Yash Y Panchal, and Ankita Korde. Colorization of black and white images using generative adversarial networks. *International Journal of Advance Research, Ideas and Innovations in Technology*, 6:484–487, 2020. URL https://api.semanticscholar.org/CorpusID:216251279.

Rahul Sankar, Ashwin Nair, Prince Abhinav, Siva Krishna P. Mothukuri, and Shashidhar G. Koolagudi. Image colorization using gans and perceptual loss. *2020 International Conference on Artificial Intelligence and Signal Processing (AISP)*, pp. 1–4, 2020. URL https://api.semanticscholar.org/CorpusID:216105663.

Gaurav Sharma, Wencheng Wu, and Edul N. Dalal. The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research and Application*, 30:21–30, 2005. URL https://api.semanticscholar.org/CorpusID:29119937.

Qian Sun, Yanfang Chen, Wenyuan Tao, Han Jiang, Mu Zhang, Kan Chen, and Marius Erdt. A gan-based approach toward architectural line drawing colorization prototyping. *The Visual Computer*, 38:1283 – 1300, 2021. URL https://api.semanticscholar.org/CorpusID:237642803.

Daniel Sýkora, Jan Buriánek, and Jirí Zára. Colorization of black-and-white cartoons. *Image Vis. Comput.*, 23:767–782, 2005. URL https://api.semanticscholar.org/CorpusID:14185386.

Alain Trémeau, Shoji Tominaga, and Konstantinos N. Plataniotis. Color in image and video processing. *EURASIP Journal on Image and Video Processing*, 2008:1–3, 2008. URL https://api.semanticscholar.org/CorpusID:1046373.

Sandra Treneska, Eftim Zdravevski, I. Pires, Petre Lameski, and Sonja Gievska. Gan-based image colorization for self-supervised visual feature learning. *Sensors (Basel, Switzerland)*, 22, 2022. URL https://api.semanticscholar.org/CorpusID:247028933.

Ravindhar N. V, K Akshaya, Pokala Ramadevi, and Madithati Yuvateja Reddy. An effective architecture for image colorization using adaptive gan. *2023 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–5, 2023. URL https://api.semanticscholar.org/CorpusID:258869488.

Domonkos Varga, Csaba A. Szabó, and Tamás Szirányi. Automatic cartoon colorization based on convolutional neural network. *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, 2017. URL https://api.semanticscholar.org/CorpusID:20066337.

Zhou Wang, Alan Conrad Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004. URL https://api.semanticscholar.org/CorpusID:207761262.

Zhenpei Yang, Lihang Liu, and Qi-Xing Huang. Learning generative neural networks for 3d colorization. In *AAAI Conference on Artificial Intelligence*, 2018. URL https://api.semanticscholar.org/CorpusID:19102893.

Jiayuan Zhang, Ao Li, Yu Liu, and Minghui Wang. Adversarially regularized u-net-based gans for facial attribute modification and generation. *IEEE Access*, 7:86453–86462, 2019. URL https://api.semanticscholar.org/CorpusID:198145997.

Linan Zhang and Hayden Schaeffer. Forward stability of resnet and its variants. *Journal of Mathematical Imaging and Vision*, 62:328 – 351, 2018. URL https://api.semanticscholar.org/CorpusID:53765912.

Sicong Zhou and Sei ichiro Kamata. Near-infrared image colorization with weighted unet++ and auxiliary color enhancement gan. *2022 7th International Conference on Image, Vision and Computing (ICIVC)*, pp. 507–512, 2022. URL https://api.semanticscholar.org/CorpusID:252396503.

A Appendix

A.1 Generative Adversarial Networks

In the adversarial setup, using multilayer perceptrons for both models works best. To teach the generator how to create data x, we set up a rule for input noise variables $pz(z)$, and then create a map to data space called $G(z;\theta_g)$. Here, G is a smooth function created by a multilayer perceptron with parameters θ_g.

We also set up another multilayer perceptron, $D(x;\theta_d)$, which gives us a single number. $D(x)$ shows how likely it is for x to be from the real data instead of from pg. We train D to increase the chance of correctly labeling both real data and samples from G. At the same time, we train G to decrease $log(1 - D(G(z)))$.

The function that models the dynamic of *D* and *G* is the following:

$$\min_G \max_D V(D,G) = E_{x\sim p_{man}}(x)[\log_0 D(x)] + E_{z\sim pz}(z)[\log_{\frac{1}{2}}(1 - D(G(z)))] \tag{1}$$

A.2 CIEDE2000

Given a pair of color values in this space, denoted by:

$$\Delta E_{00}(L_1^*, a_1^*, b_1^*, L_2^*, a_2^*, b_2^*) = \Delta E_{12}^{00} = \Delta E_{00} \tag{2}$$

It is used to calculate the color difference between two colors in the L_b^* and L_a^* color space. It transforms color components and provides results in terms of color, luminosity, and chromaticity. It is symmetric with respect to the order of colors and performs calculations in polar coordinates to address luminosity, chromaticity, and hue difference between colors. This algorithm is useful where precision in color difference evaluation is required.

A.3 Structural Similarity Index

By considering not only color accuracy but also the spatial relationships between pixels, SSIM provides a more comprehensive evaluation of the structural fidelity in the colorization process. The SSIM index is computed across different image windows. The measure between two windows, denoted as x and y, both of size $N \times N$, is given by:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3}$$

In Equation 3, μ_x and μ_y represent the pixel sample mean of x and y, respectively. Moreover, σ_x^2 and σ_y^2 are the variance of x and y, respectively, while the covariance of x and y is denotated as σ_{xy}. The variables to stabilize the division with weak denominator are c_1 and c_2.

A.4 Final Results of the Model (Movie and Cartoon Dataset)

(a) Cartoon Frame (b) Movie Frame

Figure 6: Results of the final model (cartoon and movie trained).