Enhancing 'Image Colorization' With Conditional Generative Adversarial Network

Avanith Kanamarlapudi *Rami Huu Nguyen* *Lakshmi Pranathi Vutla*

A.Kanamarlapudi001@umb.edu rami@mpsych.org L.Vutla001@umb.edu

University of Massachusetts in Boston

Abstract

Our study focuses on adding realistic colors to black and white images using a deep learning model, which is built on this Isola et al. paper. We utilized a Conditional Generative Adversarial Network (cGAN); this model allows us to turn grayscale images into beautiful, colorful images. In the generator part, our model integrates a ResNet18 encoder, which is used to understand the input, and a U-Net decoder, which helps in generating the color image. For the discriminator, we employ a 70×70 PatchGAN, which helps make the images sharper and more realistic. Our training strategies involve two stages; we first pretrained the ResNet18 generator with L1 loss to make the image structure clear and then fine-tuned it using both L1 loss and adversarial loss, helping us improve the color quality. In our study, we trained the model on 5,000 images and tested it on 100 images, all randomly extracted from the ImageNet-1K dataset. On the test set, the model achieved a Peak Signal-to-Noise Ratio (PSNR) score of 22. We also highlight that the model performs well on images with clear shapes, simple textures, and familiar objects like dogs, birds, and snow, with high PSNR values. However, the model struggles with complex backgrounds, reflective surfaces like water, and rare objects such as helmets, where stronger semantic understanding would be needed for better color prediction.

Keywords — Adversarial Generative Networks, Conditional GAN, Deep Learning, Image Colorization

I. INTRODUCTION

Recently, computer vision has included tasks that involve turning one kind of image into another, such as translating grayscale images to colored ones, generating a semantic label map, or creating an edge map. These tasks share a similar purpose—predicting new pixels based on existing ones. However, they have traditionally required special tools or extensive manual labor, such as drawing color hints or manually selecting similar reference images [1]. In traditional colorization, people often draw color scribbles on the picture, use another colored image as a reference, and/or segment the image into regions [2] [3] [5].

Adding color to images helps AI understand and recognize objects in a way that mimics human perception. With millions of colored photos available online, convolutional neural networks (CNNs) can automatically learn which colors belong to certain scenes or shapes [3].

Colorizing grayscale images is not a straightforward task. For instance, the same gray pixel might originate from red, blue, or green—it all depends on what the object is. This color ambiguity is a major challenge for machines trying to determine the correct color without prior information. To address this, recent research has turned to deep learning to automate the image colorization process. One of the most successful approaches called Generative Adversarial Networks (GANs), which includes two components: a Generator and a Discriminator [6]. The generator adds color to gray images, while the discriminator evaluates whether the output looks realistic or generated. Through adversarial training, both components try to outsmart each other. Over time, this process helps the generator improve its ability to produce colored images that closely resemble the ground truth. Victoria et al. also introduced ChromaGAN, a colorization model that uses adversarial learning along with shape, visual, and object information to create realistic colored images.

Moreover, Conditional GANs (cGANs) are built on top of the basic GAN architecture, where cGANs include external information—such as a grayscale image—into both the generator and the discriminator. The special design of cGANs is to learn how to generate outputs that are directly informed by the input. This design is particularly well-suited for image colorization tasks, where the predicted color images need to align closely with the structure and content of the grayscale image. To enhance model performance, researchers also incorporate ResNet and U-Net architectures into the networks. ResNet is built using skip connections that mitigate vanishing gradient problems and support deep learning. These skip connections allow the model to capture complex color patterns more effectively. On the other hand, U-Net, with its encoder-decoder design and skip connections, is not only good at capturing the overall structure of the image but also at retaining low-level image details [1].

In our paper, we built upon the work of Isola et al. for developing cGANs for the task of automatic image colorization. We also keep the same 70×70 PatchGAN

discriminator used in their paper [1]. We also follow the L1 loss and the cGAN loss during our training. What we improved in this paper is that we only customized the generator, where we include the ResNet18 encoder and Dynamic U-Net. Our paper aims to create sharper and more realistic colors in the final colorized image. We also use a subset of the ImageNet dataset, which we augmented by randomly rotating images 180 degrees to increase dataset diversity during training. Our paper evaluates the models using Peak Signal-to-Noise Ratio (PSNR) to measure the quality of the colored images [7].

II. Related Work

*a) **Image-to-Image Translation with Conditional Adversarial Networks***

Image-to-image translation covers a broad range of problems where it aims to convert one visual representation into another, such as adding color to grey scale images or transforming edge maps into detailed photographs. Historically, these tasks were managed with specialized networks and manually created loss functions. Isola et al. provided a general-purpose solution utilizing conditional GAN (cGANs), which is capable of learning both input-output mapping and automatically learning a suitable loss function through the adversarial training. Their findings demonstrated that the combination of U-Net generator, and Patch GAN discriminator produced strong outcomes on many different tasks without requiring intensive manual adjustment. Building on this project, our project uses the existing Pix2Pix framework specially for colorization, highlighting how integrating the L1 loss and adversarial loos can boost the quality of generated color images.

*b) **Enhanced Image Colorization Using ResNet-34 and U-Net in a Conditional GAN Framework***

This paper presents an enhanced image colorization method using a Conditional GANs framework, where the generator integrates the ResNet-34 encoder into the U-Net architecture to better extract and preserve image features. The discriminator is a standard CNN, and the model is trained using a combination of L1 loss for structural accuracy and adversarial loss for realism. The paper experimented with a diverse dataset of 9,000 grayscale images, the model generates vibrant, realistic outputs with good texture and detail preservation. As compared to traditional cGAN models, this approach achieves sharper results but requires more computational resources due to its deeper architecture and training complexity.

Our research chose this paper [1] as it includes open-source code and clearly shows the visual effects of different loss functions, allowing us to understand how L1 loss and adversarial loss each influence the colorization quality. This research provided a practical way to compare models side-by-side and see how adversarial training sharpens textures while the L1 loss alone often leads to smoother but blurrier results. Although the second paper [2] uses a deeper ResNet-34 encoder, we chose to use ResNet-18 to make the model faster and lighter while keeping the results realistic.

III. Methods

We used the existing CGANs model for this image colorization project. Our generator implements an encoder-decoder structure, with the Resnet18 encoder to extract feature maps from grayscale images and the Dynamic U-Net decoder to recreate the colorized output while retaining fine detail features. In the discriminator section, we trained with a PatchGAN discriminator using a 70x70 field size were focusing on small regions of the image to refine texture realism and sharpness. Our project employs a small segment of ImageNet-1K Web Dataset (timm/imagenet-1k-words), taken from the ILSVRC 2012 dataset. We randomly extracted 5000 images for training, 1000 images for validation, and 100 for testing. Please note that we only use validation images once we trained Resnet18. To support generalization, we applied data augmentation techniques by applying random rotations between 0 and 180 degrees. We also added those augmented dataset into current training dataset, totaling 10,000 images.

IV. Objective

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$

Our final objective was inspired by Isola et al. integrating the adversarial loss and L1 loss to guide the generator. The adversarial loss enables the model to create images that look realistic by trying to fool the discriminator, while the L1 loss ensures the generated image stays close to the ground truth in pixel values. By minimizing both losses together, the generator learns to produce images that are not only sharp and vivid but also structurally accurate.

V. Training and Evaluation Approach

Our model was built into two-phase training progresses. We began by pretraining the Resnet18 model for 20 epochs using the L1 loss to improve its ability to extract features. After the first step, we loaded the weights into the cGANs framework and trained the model using a combination of adversarial and L1 losses. This approach enhances early learning stability, and results in a sharper colorized image during adversarial training. After training, we assess our model by using the Peak Signal-to-Noise Ratio (PSNR), a standard metric used to gauge similarity between outputs and ground truth images [7]. Higher PSNR values signifies higher image reconstruction and lower distortion.

VI. Results and Discussions

Our final model obtained a PSNR value of 22 on the test set, reflecting a reasonable level of reconstruction quality. To further evaluate performance, we picked top 10 images with the best PSNR scores to study the best colorization results, and the 10 images with worst PSNR scores to investigate the model's poorest predictions.

a) Figure 1: Top 10 good result

Our Figure 1 clearly shows us that the model performed strongly when grayscale images had well-defined object boundaries, simple texture, and realistic color tone. For instance, the shopping cart, dog on snow, bird near water and goats highlight smooth textures, where backgrounds such as snow, water, and rock were consistent and lacked complex details, making prediction easier. Looking closer for the image is the dog on snow, where only two colors—black fur and white snow—are present, without much internal variation. Additionally, this colorization works very well on 8th columns with the image of a dog where the dog's fur, tongue, and nose have common colors that the model has seen during training, making it easier to predict realistic tones. The clear difference between the dog and the background also supports the model colorizing correctly. As a result, the final colorized output for this image looks natural and achieved a high PSNR score of 27.8 dB.

The main reason arises why those top models predicted well were due to training data that have many similar examples. During training, the model captured clear patterns from the training dataset, particularly for animals, outdoor scenes, and simple objects. Dogs, birds, and goats are commonly found in the small portion of ImageNet dataset, helping the model recognize their standard fur colors, forms, and textures. Snow, wood, and sky have simple and straightforward color patterns, making them easier for models to learn. Consequently, these test images reached high PSNR levels and maintained a natural and vivid appearance.

b) Figure 2: Top 10 worst results

Figure 2 highlights the worst example of colorization based on the PSNR score. In general, the models are difficult to vivid colors, and once the background gets more complex. Looking closer at the first example of a duck on reflective water, the model might be confused as reflective water might change based on light (sunset and noon), weather (clear vs. cloudy), and background (trees, mountains, buildings), even if the dataset likely contains some images of water. The model also failed on the Hermes image (5th column), possibly because the dataset did not contain enough similar helmets during training. This suggests the model lacks semantic understanding—for example, recognizing it as the plastic helmet that might typically be light blue.

VII. Future Work

First, we should improve the size of dataset. At present, the dataset was trained only on a small portion of the ImageNet dataset. To make it better, we can add unique and difficult images — like shining surfaces, clear details, or rare objects. Adding images like this can help the model learn more effectively and be able to handle harder cases and color them realistically. The next step that we can be doing is to enhance the model architecture. Sometimes, the model makes poor color predictions as it can't properly understand the grayscale image yet. For example, it might color the water or the helmet incorrectly because the grayscale doesn't tell it enough. We can fix these problems by incorporating these into our generator architecture:

First one is to add semantic segmentation; this will help the model in figuring out what type of object it is seeing — like water, sky, animal, or tree. Once it knows that, it will be able to pick better colors. For example, it will be able to color the sky blue or color the trees green even if the grayscale image is confusing. Second, we can add an Attention Mechanism; this will help the model focus on the important parts of the image (like a guinea pig or helmet) instead of getting distracted by background details like street textures or shadows. It will improve the model's ability to generate meaningful colors in the correct places.

Finally, we need to have a better evaluation. We are using PSNR (a technical score), which is not enough to see how good the colors are. We can do better by adding:

- **Human Feedback Ratings**: Asking people around to judge which image looks better, or using tools like Amazon Mechanical Turk, which will help in evaluating the image more effectively. This way would help us reflect the actual visual quality.

- **Using Other Models**: Testing by incorporating pretrained models (like image classification or segmentation networks) to check if the colorized still makes sense. For example, can a machine recognize a dog or a tree from the image? If yes, then the colorization is not just visually appealing but also semantically correct.

VIII. Conclusion

In this study, we tried to change black and white images into colorful images using a deep learning model called Conditional GANs. We chose this method as various tasks

in computer vision, like coloring, edge detection, and labeling, all involve turning one kind of image into another, and deep learning has shown remarkable success. To accomplish this, we utilized a cGAN model and built on top two papers [1] and [2]. The generator has the ResNet-18 encoder and the U-Net decoder, whereas the discriminator uses the PatchGAN 70x70 architecture. We first trained the generator using L1 loss. After that, we fine-tuned the generator using a mix of the L1 loss and the adversarial loss. Our model was able to achieve a PSNR score of 22, which shows how close the colorized images are to the real or original images.

When comparing the best and worst results, we found that the model performed well with simple images—those with clear edges and textures. However, the model had trouble with complex or confusing scenes, especially when it couldn't understand what the object was from the grayscale alone. To improve performance in the future, our study will include a wider variety of images so the model can learn from more diverse cases, including shiny objects, animals, and unusual textures. We also plan to help the model better understand objects in the image (like water, trees, or animals) by using semantic segmentation.

In addition, our research will focus on the most important parts of the image using an attention mechanism, which can reduce distraction from the background. To make the colorized images look more natural, we may also use improved loss functions such as perceptual loss and colorfulness regularization. These loss functions can support the model in learning more realistic colors. Finally, instead of relying only on PSNR to evaluate quality, our future work may include both human judgment and semantic evaluations to better understand how good the colorized images look and what meaning they convey.

REFERENCES

[1] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," *arXiv preprint arXiv:1611.07004*, 2016. [Online]. Available: http://arxiv.org/abs/1611.07004

[2] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification," *ACM Transactions on Graphics (Proc. of SIGGRAPH)*, vol. 35, no. 4, pp. 110:1–110:11, 2016.

[3] P. Vitoria, L. Raad, and C. Ballester, "ChromaGAN: An Adversarial Approach for Picture Colorization," *arXiv preprint arXiv:1907.09837*, 2019. [Online]. Available: http://arxiv.org/abs/1907.09837

[4] H. Shafiq and B. Lee, "Transforming Color: A Novel Image Colorization Method," unpublished, 2024.

[5] S. M. Li, Q. F. Liu, and H. Y. Yuan, "Overview of Scribbled-Based Colorization," *Art and Design Review*, vol. 6, pp. 169–184, 2018. [Online]. Available: https://doi.org/10.4236/adr.2018.64017

[6] S. Kuchana, "Enhanced Image Colorization Using ResNet-34 and U-Net in a Conditional GAN Framework," unpublished, Dec. 2024. [Online]. Available: https://doi.org/10.13140/RG.2.2.14523.73768

[7] O. Ieremeiev, V. V. Lukin, K. Okarma, and K. O. Egiazarian, "Full-Reference Quality Metric Based on Neural Network to Assess the Visual Quality of Remote Sensing Images," *Remote Sensing*, vol. 12, no. 15, p. 2349, 2020. [Online]. Available: https://doi.org/10.3390/rs12152349

Figure 1: Top 10 good result

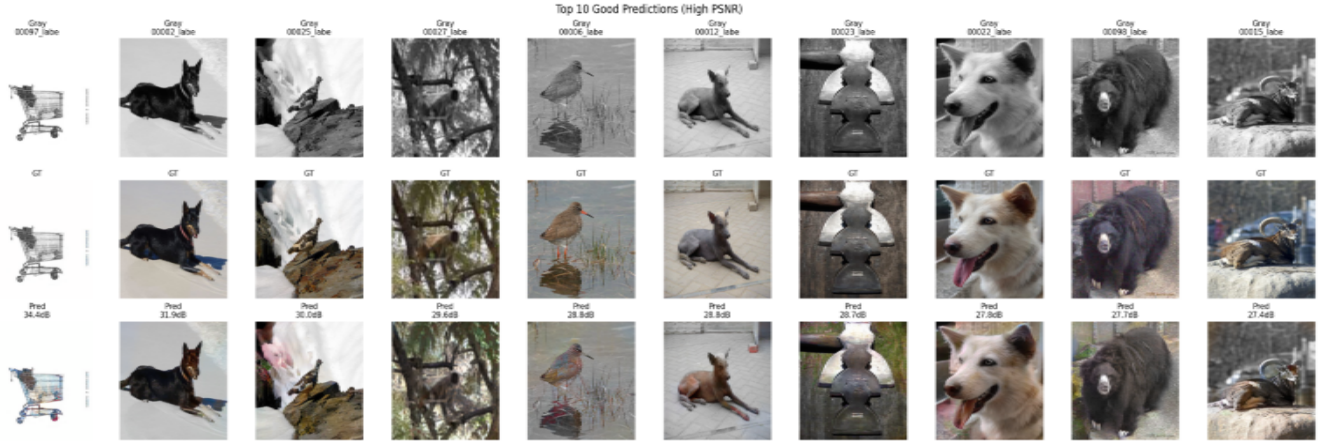

Figure 2: Top 10 worst results