Deep Image Colorization: A ResNet-Based U-Net and PatchGAN Approach with Perceptual Loss

COMP3057 Introduction to Artificial Intelligence and Machine Learning

Constantino Harry Alexander 25206605

Instructor: Dr. Ren Jie, Wan

December 16, 2025

1. Introduction

i. Problem Definition

Image colorization is the process of adding reasonable color information to a grayscale image – is a classic ill-posed problem in computer vision. This is because in a grayscale image which is a matrix of each pixel corresponding to the brightness information, a single grayscale value can correspond to many valid color outputs (e.g., a dark gray pixel could correspond to a value of a red, blue, or green car), there is no. single deterministic solution. Traditional regression-based methods often produced desaturated, "sepia-like" results because they average over possible colors.

ii. Motivation

The primary motivation of this project is to automatically restore and colorize grayscale images, such as those from historical archives and the enhancement of legacy media. By leveraging deep learning, specifically Generative Adversarial Networks (GANs) inspired by the Pix2Pix framework [1] to generate vibrant, I intend to create a system that not only predicts accurate colors but also generate realistic textures, overcoming the "washed-out" or "sepia" look common in standard regression models. The system takes a grayscale (L channel in Lab color space) image as input and predicts the missing chrominance (ab channels). Real-world application includes, restoration of historical black-and-white photographs and films, enhancement of grayscale medical or satellite imagery and creative tools for artistic creation.

iii. Dataset

The model was trained on the publicly available "Image Colorization Dataset" by Shah [2] from Kaggle. This data set was very suitable, it contains approximately 5000 diverse set of nature and landscape images. To address GPU memory constraints on Google Colab while preserving high-resolution details critical for realistic colorization, as advised by my instructor, a subset of 1500 high-quality images was selected, resized to 255×255 pixels using OpenCV bilinear interpolation, and saved to a processed directory for efficient loading. The processing step standardized the input size, reduces loading overhead, and ensures consistency across the pipeline.

Following the methodology introduced by Zhang et al. [3] in their work on colorful image colorization, all images are converted from RGB to the CIELAB (Lab) color space. This decouples luminance (L channel) from chrominance (ab channels), enabling the model to condition solely on the grayscale L input while predicting the missing color components. The

L channel is normalized to [-1, 1] by dividing by 50 and subtracting 1, while ab channels are scaled to [-1, 1] by dividing by 128.

To prepare the data, I created a custom pipeline that splits the images into training, validation, and testing sets (80/10/10 split). A key priority was preventing overfitting; therefore, the training images are subjected to random variations—such as rotation, cropping, and color jitter—before entering the network. However, to ensure that our performance metrics remain comparable across epochs, the validation and test images are processed without these random variations. All images are loaded in batches of 8, ensuring efficient processing while maintaining the synchronization between input images and their target colours.

2. Methodology

i. Overall Architecture

The system employs a conditional Generative Adversarial Network (cGAN) based on the Pix2Pix framework [1], comprising a generator and a discriminator. The generator is tasked with predicting the ab chrominance channels from the input grayscale L channel, while the discriminator evaluates the realism of the generated color images.

The generator adopts a U-Net architecture [5] enhanced with a ResNet-18 backbone pre-trained on ImageNet. This design leverages the ResNet-18's convolutional layers for robust semantic feature extraction in the encoder, enabling the model to recognize high-level concepts (e.g., "sky" or "grass") without training from scratch. The encoder consists of the initial layers of ResNet-18, followed by a bottleneck with 9 residual blocks for deep feature processing. The decoder symmetrically up samples these features using transposed convolutions and incorporates skip connections from the encoder to preserve spatial details and facilitate gradient flow. The output is a 2-channel tensor representing the ab channels, with the final color image reconstructed by concatenating the input L with the predicted ab.

The discriminator is a PatchGAN model with 4 convolutional layers, each followed by batch normalization and Leaky ReLU activations. Unlike a global discriminator, PatchGAN classifies local 70×70 patches as real or fake, producing a 30×30 validity map. This approach emphasizes high-frequency structures and sharpness, mitigating blurriness in generated outputs.

ii. Loss Function

To produce vibrant and perceptually realistic colorizations, the model employs a hybrid loss function that balances pixel-level accuracy with high-level visual quality. The core of this approach is the Adversarial Loss implemented via Binary Cross-Entropy with Logits (BCEWithLogitsLoss), which compels the generator to create images that are indistinguishable from real photographs by attempting to fool the discriminator. To ensure the colours remain true to the original scene, an L1 Reconstruction Loss (L1Loss) measures the absolute pixel-wise differences between the predicted and ground-truth ab channels; while this component is crucial for accuracy, relying on it too heavily can lead to desaturated results. To counteract this "sepia effect" and encourage richer coloration, a Perceptual Loss is

integrated. Computed using the Learned Perceptual Image Patch Similarity (LPIPS) metric [6] with a pre-trained VGG network as the backbone [7], this loss assesses similarity based on high-level features rather than just raw pixel values. These components are combined into a total generator loss defined as $\mathscr{L}\_G = \mathscr{L}\_\{GAN\} + \lambda\_\{L1\} \cdot \mathscr{L}\_\{L1\} + \lambda\_\{perc\} \cdot \mathscr{L}\_\{perc\}$, where $\lambda\_\{L1\} = 100$ (adjusted to 50 in later runs for improved vibrancy) and $\lambda\_\{perc\} = 10$, while the discriminator optimizes $\mathscr{L}\_D = (\mathscr{L}\_\{D\_real\} + \mathscr{L}\_\{D\_fake\}) / 2$ using BCEWithLogitsLoss on real and fake classifications.

iii. Training Process

Training was conducted using PyTorch 2.x on Google Colab with the A100 GPU. The Adam optimizer was employed for both networks (learning rate = $1 \times 10^{-4}, \beta_1 = 0.5, \beta_2 = 0.999$), with gradient clipping (max_norm = 1.0) to enhance stability.

A two-phase approach was implemented: a 20-epoch warmup period trains only the generator using L1 loss for initial convergence, followed by full adversarial training over 100–120 epochs (with early stopping based on validation L1 loss). The batch size was set to 8, with the discriminator updated once per generator update (n_critic = 1). Validation occurs after each epoch on unseen data, tracking L1 loss, PSNR, and LPIPS; the best model (lowest validation L1 after warmup) is saved.

To combat early discriminator collapse observed in mid-progress runs, spectral normalization was added to discriminator layers, and perceptual loss provided stronger gradients to the generator. Training duration was approximately 1–1.5 hours, yielding balanced loss convergence in the final runs.

iv. Training Adjustments and Challenges

Initial training runs with a high L1 loss weight ($\lambda\_L1 = 100$) resulted in conservative, desaturated outputs, as the generator prioritized minimizing pixel-wise errors by averaging colors, leading to a "sepia effect" [3]. To address this, the L1 weight was reduced to 50.0, while increasing the perceptual loss weight ($\lambda\_percep = 10.0$) to emphasize high-level features and vibrancy, following recommendations for balancing regression and perceptual objectives in image generation tasks [4]. This adjustment intentionally allowed L1 loss to increase slightly (trading mathematical precision for perceptual realism), as lower L1 penalties encourage the model to make bolder color predictions rather than defaulting to muted tones.

As expected, this led to side effects such as minor color bleeding and artifacts in early epochs, but overall improved color diversity and sharpness. The perceptual loss, computed via LPIPS with VGG features, guided the generator toward outputs that better align with human vision, mitigating the multimodal ambiguity in colorization. Training stability was further enhanced by spectral normalization in the discriminator and gradient clipping, preventing collapse observed in mid-progress runs.

3. Results and Analysis

i. Quantitative Metrics and Insights

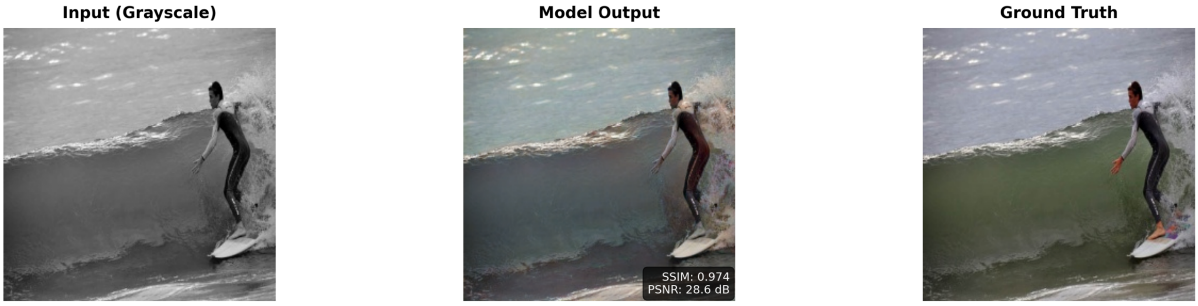
The model was evaluated on a held-out test set of 100 images using standard colorization metrics: Peak Signal-to-Noise Ratio (PSNR) for signal quality and Structural similarity Index (SSIM) for structural Preservation.

Matric	Value	Notes
Average PSNR	22.53 dB	Moderate fidelity: higher than mid-progress (18.4dB) due to perceptual loss integration.
Average SSIM	0.9162	Strong structural Preservation.
Best PSNR	30.96 dB	Visually excellent cases
Best SSIM	0.933	

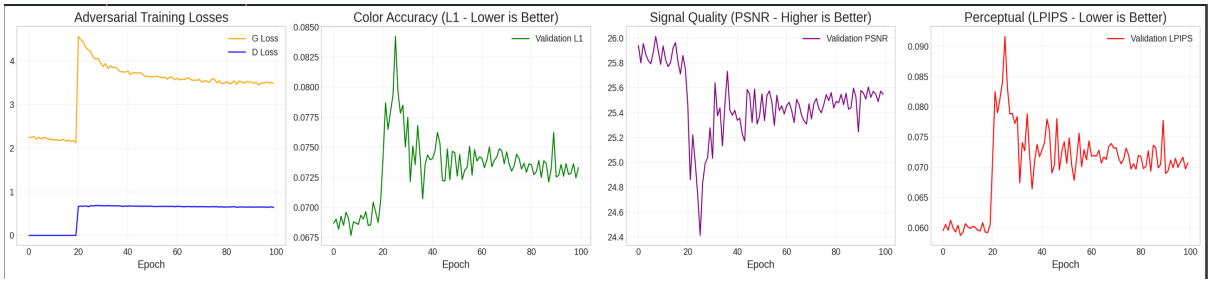
These metrics indicate improved fidelity and structure compared to earlier iterations, with the hybrid loss contributing to better perceptual quality despite the multimodal challenges of colorization. PSNR measures reconstruction quality (higher is better), while SSIM assesses similarity in luminance, contrast, and structure (closer to 1 is better). The averages reflect solid performance on natural scenes, though lower scores in ambiguous cases highlight data bias limitations.

ii. Qualitative Results

The model produces vibrant, semantically consistent colorizations (e.g., blue skies, realistic skin tones), with perceptual loss successfully mitigating desaturation. While high-SSIM examples achieve near-photorealism in landscapes and portraits, the model occasionally reverts to conservative priors or exhibits color bleeding in ambiguous, low-SSIM scenes.



iii. Training Dynamics and Loss Analysis



Diagnostic plots reveal a clear transition between the warmup phase (epochs 1–20) and adversarial training. During warmup, generator loss steadily decreases as the model learns basic color mappings. Upon introducing the discriminator at epoch 21, discriminator loss drops rapidly while generator loss spikes briefly, reflecting the model's adaptation to adversarial pressure.

Post-warmup, validation L1 loss increases slightly (from ~0.070 to ~0.073), representing a deliberate trade-off where pixel-perfect accuracy is sacrificed for improved vibrancy. Similarly, LPIPS spikes initially as the generator shifts from conservative regression to perceptually richer outputs, mitigating the "sepia effect." By epoch 100, losses stabilize with a healthy G/D ratio, avoiding mode collapse. Ultimately, the hybrid loss strategy successfully balances stability and realism, improving final PSNR to 22.53 dB (up from 18.4 dB mid-progress).

4. Conclusion and Future Work

i. Summary

This project successfully developed a deep learning framework for automatic image colorization by integrating a ResNet-based U-Net generator with a PatchGAN discriminator. By moving beyond simple pixel-wise regression and employing a hybrid loss function—combining Adversarial, L1, and Perceptual (LPIPS) losses—the system effectively overcame the "sepia effect" common in traditional approaches. The implementation of a two-phase training strategy (warmup followed by adversarial training) proved crucial for stability, allowing the model to learn structural features before refining color vibrancy. Quantitative evaluation yielded an average PSNR of 22.53 dB and an SSIM of 0.9162, while qualitative results demonstrated the model's ability to generate semantically consistent and visually compelling colorizations for natural landscapes.

ii. Limitations

Despite the promising results, several limitations persist. First, the model occasionally exhibits color bleeding across distinct boundaries and introduces artifacts in highly complex scenes, a side effect of the aggressive perceptual loss weight used to boost vibrancy. Second, the dataset bias toward nature and landscape images limits the model’s generalization capabilities; it may struggle to accurately colorize specific indoor objects, man-made structures, or historical artifacts not represented in the training distribution. Finally, the input resolution was constrained to 255×255 due to computational limits, resulting in a loss of high-frequency detail in the final output compared to high-definition source material.

iii. Future Improvements

Future iterations of this work could significantly enhance performance and versatility through several key technical advancements. To address color bleeding and improve global context understanding, the architecture could incorporate Self-Attention mechanisms or Vision Transformers (ViT), which are better suited for capturing long-range dependencies than standard convolutions. Furthermore, to resolve the inherent ambiguity of the ill-posed colorization problem, the system could be extended to accept user hints—such as color scribbles—giving users creative control over specific object colors. To overcome resolution constraints, a post-processing Super-Resolution GAN (SRGAN) could be integrated to upscale the 255×255 outputs to high definition, restoring lost texture details. Finally, extending the pipeline to enforce temporal consistency would allow for the restoration of archival film footage, broadening the application to video colorization.

5. References

[1] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. CVPR.

[2] Shah, A. (2021). Image colorization dataset. Kaggle.

[3] Zhang, R., Isola, P., & Efros, A. A. (2016). Colorful image colorization. ECCV.

[4] Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. ECCV.

[5] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. MICCAI.

[6] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. CVPR.

[7] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. In International Conference on Learning Representations.