Individual Final Report: Model Evaluation and Web Application for GAN-Based Image Colourization

By: Naiska Buyandalai

Professor: Amir Jafari

Course: Deep Learning (DATS 6303)

1. Introduction

In this project our team explored the problem of colourization of grayscale images using deep neural networks. We began by replicating the ECCV-2016 classification-based colourization model and then investigated whether generative adversarial training could improve perceptual realism.

My contributions focused on the final stages of the project:

1. Analyzing the quantitative and qualitative performance of our model
2. Building an interactive Streamlit application to demonstrate the models to end-users.

2. Project Overview and Shared Work

The goal of our project was to investigate whether a GAN-based refinement improves colourisation quality over the baseline ECCV-2016 model. We used the ImageNet-50 subset as our dataset, which contains 50 diverse categories of natural images.

The workflow consisted of three main stages:

1. Baseline reproduction and data preprocessing: One teammate implemented data loaders using the Hugging Face Elriggs/imagenet-50-subset dataset and reproduced the ECCV-2016 encoder–decoder network in PyTorch.
2. GAN fine-tuning: Another teammate extended the baseline by adding a PatchGAN discriminator and a VGG-based perceptual loss. The generator was first started from the ECCV-2016 weights and then trained adversarially. Classification-based fine-tuning with a class-rebalancing loss was also explored.
3. Evaluation and deployment: My role was to evaluate both models quantitatively and qualitatively and to create an interactive web app. The evaluation used multiple metrics such as area under the color error curve (AUC), peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) to capture different aspects of quality. I summarized these results in the final report and built a Streamlit application to showcase the models.

3. Individual Contributions

3.1 Model Evaluation Code

To compare the pretrained ECCV-2016 model against our fine-tuned GAN, I wrote a comprehensive evaluation script. The script loads the trained generator weights, iterates over the validation split using a DataLoader, computes multiple loss functions and metrics, and aggregates the results.

The evaluation metrics include:

1. L1 and MSE losses in the ab space: These capture pixel-wise differences between the predicted and ground-truth chrominance channels.

2. Perceptual loss: We reused the VGG-19 feature extractor from the training pipeline. For each generated image we computed the L1 difference between VGG features of the generated and ground-truth color images, thereby assessing high-level perceptual similarity.
3. AUC: The cumulative distribution of per-pixel L2 colour errors (0–150) is integrated to obtain a color error area under curve. Higher AUC indicates a larger fraction of pixels have small color errors.
4. PSNR in RGB space: Measured on [0,1] scaled images, PSNR assesses fidelity in the image domain. We computed PSNR for each image and averaged across the dataset. A higher PSNR implies lower mean squared error.
5. SSIM: We used the skimage.metrics.structural_similarity function to compute SSIM for each image. SSIM measures structural and contrast similarity; values range from 0 to 1, with higher values indicating closer structural resemblance.

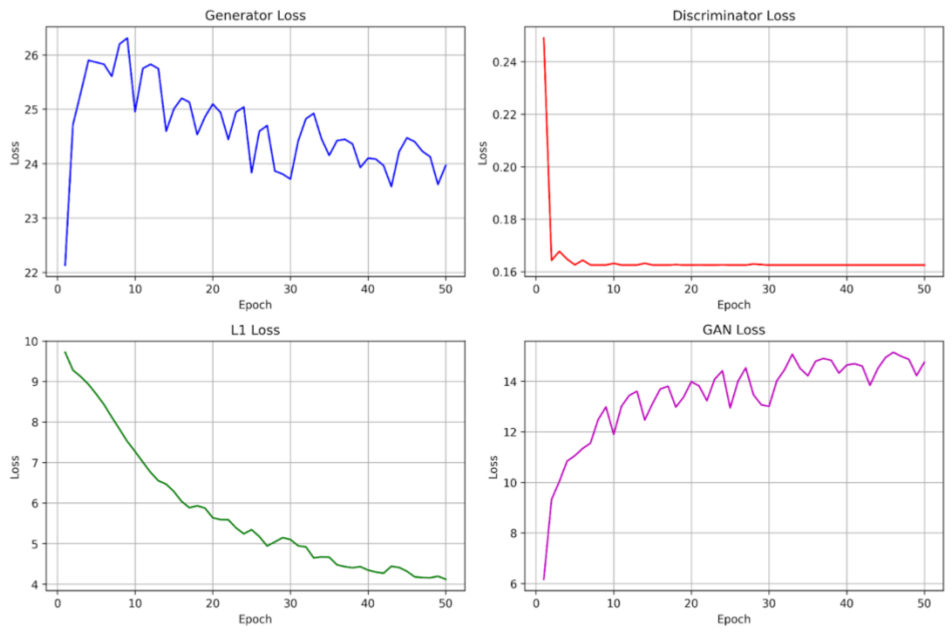
After iterating through the dataset, the script reports the averaged metrics for each model. I ensured the code could evaluate both the fine-tuned GAN and the original ECCV-2016 model in the same run for fair comparison.

3.2 Results and Discussion

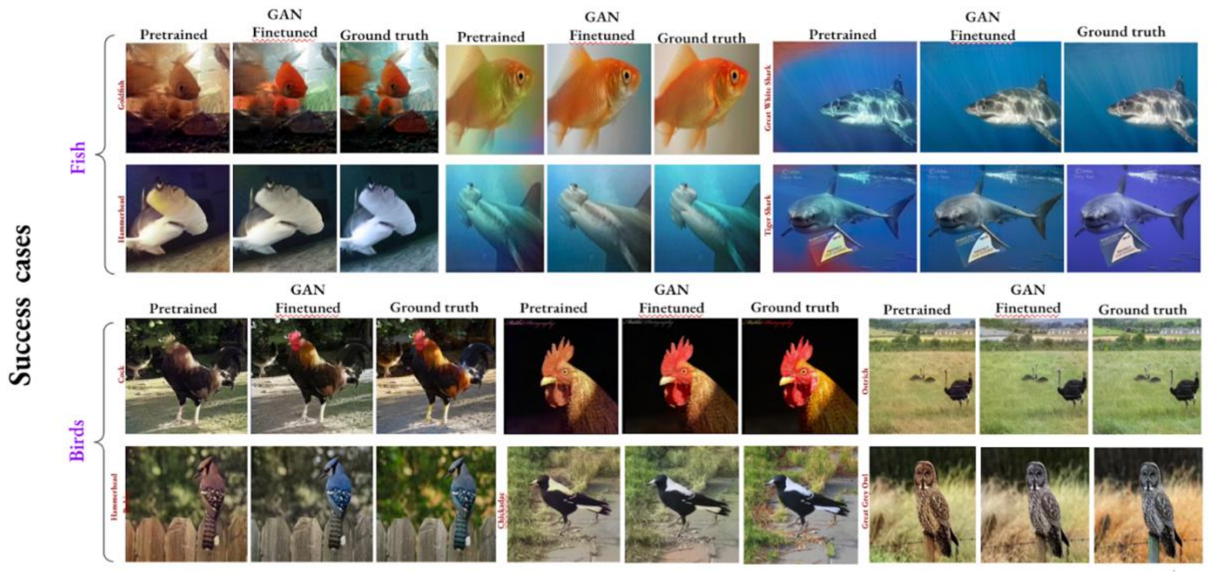
Using the evaluation script, I collected quantitative results that were reported in the group report. The enhanced GAN model outperformed the baseline across all metrics. The table below summarizes the key metrics we observed for the validation set:

Metric	**Pretrained ECCV-16**	**Enhanced GAN**	**Interpretation**
AUC	0.89	**0.91**	Higher AUC means a larger proportion of pixels have small color

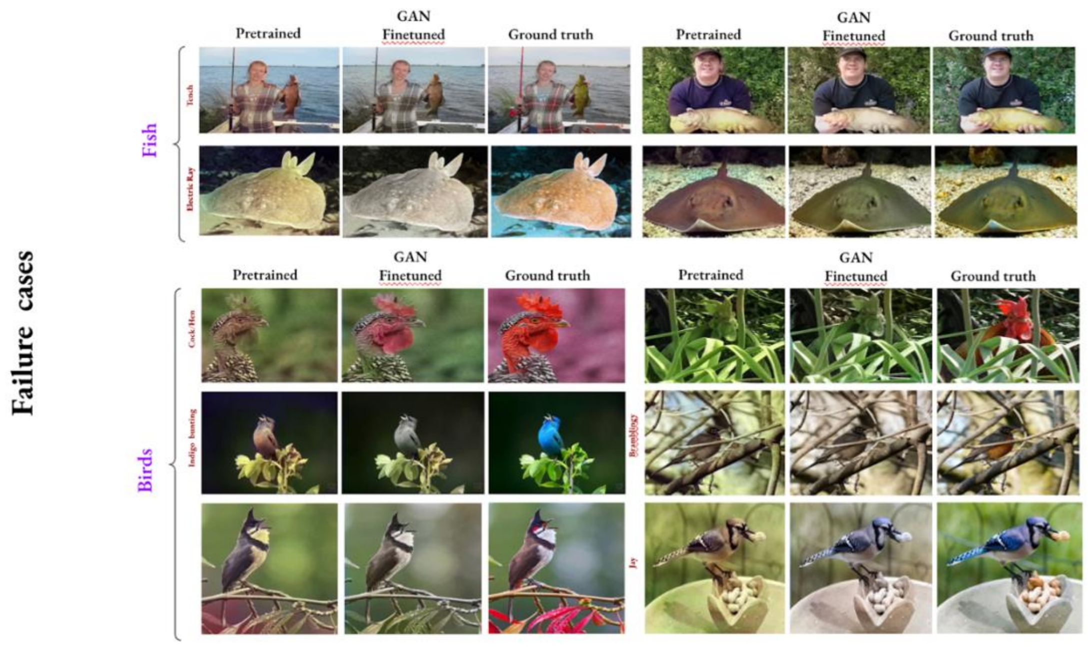
			errors. The GAN model shifts the error distribution toward lower values, indicating more accurate colorization.
PSNR (dB)	22.27	**23.76**	Higher PSNR reflects lower MSE in the RGB domain. The GAN's PSNR increase demonstrates improved fidelity despite adversarial training.
SSIM	0.91	**0.92**	SSIM improvement shows the GAN better preserves structure and contrast of the original images.
Perceptual loss	–	–	While not tabulated here, the GAN consistently achieved lower VGG perceptual error, supporting its improved realism.



I also inspected training curves (generator loss, discriminator loss, L1 loss and GAN loss) to understand model dynamics. The generator loss decreased gradually after an initial spike, the discriminator loss stabilized quickly, and the L1 loss steadily decreased. These trends confirmed that our hyperparameters provided stable adversarial training.



Visual inspection complements these metrics. The figure above shows example images comparing the pretrained baseline, our fine-tuned GAN, and the ground truth. Across diverse classes (animals, vehicles, household objects), the GAN model produces more vibrant and plausible colours. For instance, aquatic animals such as goldfish and sharks are colourised with appropriate orange or grey hues, while the baseline often produces desaturated or off-hue results.



Despite overall improvements, the GAN model exhibits notable failure outputs. Colourization remains ambiguous for certain objects. For example, complex scenes with multiple objects confuse the generator, such as background color may bleed into foreground objects, or colours may spill across edges. Failure cases are highlighted in the figure above.

3.3 Streamlit Web Application

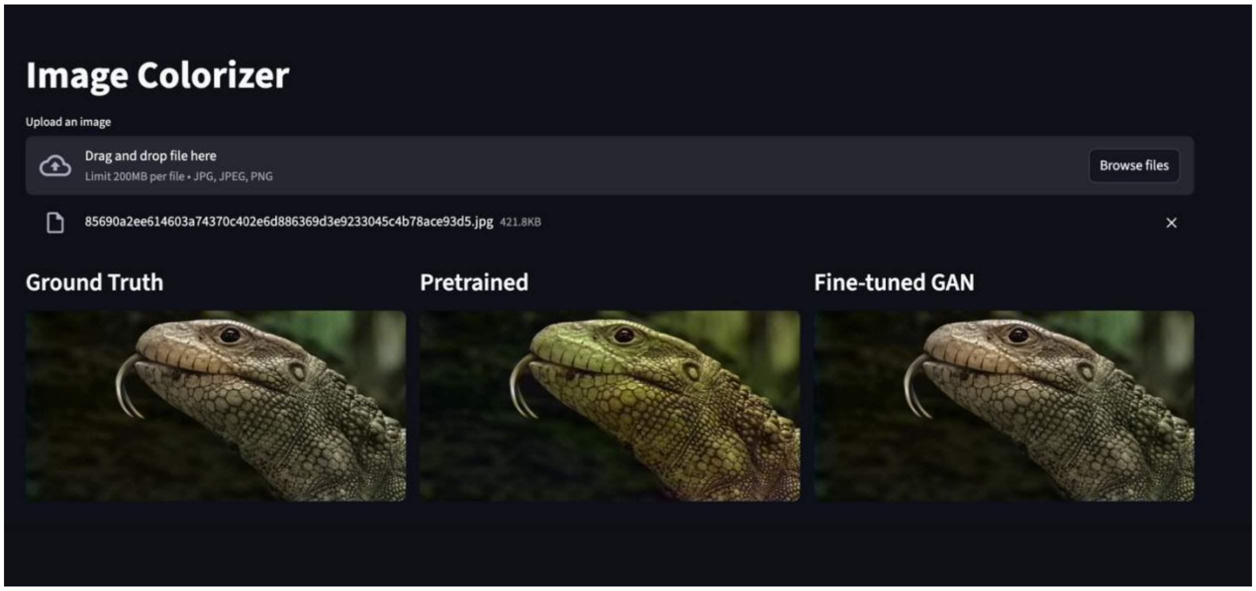


Image Colorizer
Upload an image
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG
Browse files
85690a2ee614603a74370c402e6d886369d3e9233045c4b78ace93d5.jpg 421.8KB
Ground Truth
Pretrained
Fine-tuned GAN

To make our model accessible to users, I developed a Streamlit application. The app performs the following functions:

1. Model loading: Using st.cache_resource, the app loads both the pretrained ECCV-2016 generator and the fine-tuned GAN into GPU memory. We download the baseline weights from the official release and load our trained GAN weights from the local checkpoint.
2. Image upload: Users can upload a grayscale image in JPEG or PNG format. The app converts it to RGB and extracts the L channel using our preprocess_img function.
3. Inference: Both models process the resized L channel (256×256). Their predicted ab channels are combined with the original L channel using postprocess_tens to produce colorized images.
4. Visualization: The app displays three columns: the original grayscale image, the baseline colorization and the GAN colorization. This allows users to qualitatively compare outputs side-by-side.

User experience: The layout is responsive and includes descriptive headings. By running streamlit run streamlit_app.py after installing dependencies (PyTorch, scikit-image, Streamlit), anyone can interact with our colorizer. We deployed the app at https://dl-group1-colorization-app.streamlit.app/ for convenient access.

4. Discussion of Results

The quantitative results show that adversarial fine-tuning with perceptual loss yields noticeable improvements over the baseline. The AUC increase from 0.89 to 0.91 means more pixels have accurate chrominance predictions. The rise in PSNR and SSIM implies better overall fidelity and structural preservation, which aligns with visual inspection. The slight improvements may appear modest, yet small increases in these metrics correspond to perceptually significant improvements. During evaluation we observed that the GAN often produced richer and more realistic colors, particularly for objects with ambiguous coloring, whereas the baseline sometimes generated desaturated tones.

However, adversarial training introduces variability: some classes (man-made objects) saw larger gains than others (natural landscapes). This echoes the challenge mentioned in the group report that ambiguity in correct colors remains. Multiple plausible colorizations may exist for a grayscale image, and a single deterministic model cannot capture the full distribution. The metrics we used (PSNR, SSIM) are imperfect proxies for human perception and sometimes reward oversmoothed outputs. Future work could incorporate learned perceptual metrics or user-guided colorization to resolve ambiguous regions.

5. Summary and Conclusion

This individual report summarizes my contributions to the evaluation of the models and the development of an interactive demo. By creating a comprehensive evaluation script I quantified the differences between the pretrained ECCV-2016 model and our fine-tuned GAN. The metrics AUC, PSNR, SSIM and perceptual loss consistently favor the GAN model, demonstrating that adversarial learning and perceptual loss improve colorization quality. I distilled these findings into the final report and explained their implications for perceptual realism.

Furthermore, I built a Streamlit web app that allows users to upload grayscale images and view colorizations from both models side-by-side. The app provides a practical demonstration of our research and invites user feedback. It is available online and can be run locally with minimal setup. Through this work I gained experience in evaluating generative models, interpreting quantitative metrics, and deploying machine learning models to interactive web interfaces.

While the gains are promising, there is still room for improvement. The dataset size (ImageNet-50) limits generalization, and color ambiguity remains a challenge. Future work should explore larger and more diverse datasets, incorporate conditional inputs (e.g., scene labels or semantic maps), and investigate multimodal or interactive colorization to handle ambiguous cases. Nonetheless, this project demonstrates that combining GANs with perceptual losses provides a viable path to more realistic image colorization.

References

[1] Zhang, R., Isola, P., & Efros, A. (2016), "Colorful Image Colorization," *arXiv preprint arXiv:1603.08511*. https://arxiv.org/abs/1603.08511

[2] Qiu, C., Cao, H., Ren, Q., Li, R., & Qiu, Y. (2025). Automatic Image Colorization with Convolutional Neural Networks and Generative Adversarial Networks. arXiv preprint arXiv:2508.05068. https://arxiv.org/abs/2508.05068