Comparative Analysis of Regression and Classification-Based CNN Architectures for Image Colorization

Ashad Qureshi

April 13, 2025

Abstract

This report details the exploration of image colorization using Convolutional Neural Networks (CNNs), focusing on the "Horse" category from the CIFAR-10 dataset. Two primary methodologies were investigated: direct regression of RGB values and classification of quantized color clusters. For each methodology, two architectures were implemented and evaluated: a standard encoder-decoder CNN and a UNet-based CNN incorporating skip connections. Performance was quantitatively assessed using Mean Squared Error (MSE), Mean Absolute Error (MAE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM). Qualitative results (including generated samples), activation analyses, and training dynamics are presented for all approaches. The findings consistently demonstrate the significant advantages of the UNet architecture with skip connections across both regression and classification tasks, yielding visually superior and quantitatively improved colorization results. A summary consolidates key findings regarding architectural benefits, and potential improvements to evaluation and model adaptability are discussed.

1 Introduction

Image colorization, the task of assigning plausible colors to grayscale images, remains a challenging problem in computer vision. This report investigates the application of Convolutional Neural Networks (CNNs) to automatically colorize images from the "Horse" category of the CIFAR-10 dataset. Two distinct paradigms for color prediction are explored:

1. **Regression-based Colorization:** Directly predicting continuous RGB color values for each pixel.

2. **Classification-based Colorization:** Quantizing the color space and predicting the most likely color class for each pixel.

Within each paradigm, two architectural variants were implemented: a standard CNN and a UNet CNN with skip connections. The objective is to compare the effectiveness of these methodologies and architectures in generating high-fidelity color images from grayscale inputs.

2 Regression-based Colorization Models

This approach treats colorization as a regression problem, predicting three continuous values (RGB) per pixel.

2.1 Regression CNN Architecture

The first regression model was a standard encoder-decoder CNN without skip connections. This model was trained for 21 epochs (early stopping triggered).

Table 1: Convolutional Layer Configuration for the Regression-based CNN (Summary)

Layer Type	Filter Size	Filters	Stages	Notes
Conv2D Block	3x3	64	Encoder	(2 layers)
Conv2D Block	3x3	128	Encoder	(2 layers)
Conv2D Block	3x3	256	Encoder/Bottleneck	(2 layers)
UpConv Block	3x3	128	Decoder	(2 layers)
UpConv Block	3x3	64	Decoder	(2 layers)
Conv2D Output	1x1	3	Output	(RGB)

2.2 Regression UNet CNN Architecture

The second regression model utilized a UNet architecture with skip connections. This model was trained for 33 epochs (early stopping triggered).

Table 2: Convolutional Layer Configuration for the UNet Regression CNN (Summary)

Layer Type	Filter Size	Filters	Stages	Notes
Conv2D Block	3x3	64	Encoder (E1)	(2 layers)
Conv2D Block	3x3	128	Encoder (E2)	(2 layers)
Conv2D Block	3x3	256	Bottleneck	(2 layers)
UpConv Block	3x3	128	Decoder (D1)	Connects E2, (2 layers)
UpConv Block	3x3	64	Decoder (D2)	Connects E1, (2 layers)
Conv2D Output	1x1	3	Output	(RGB)

2.3 Regression Performance Comparison

The performance of the regression models is summarized in Table 3. The UNet architecture demonstrates significant improvements across all metrics, indicating better pixel-level accuracy and structural similarity compared to the standard CNN baseline for the regression task.

Table 3: Regression Models: Quantitative Performance Comparison

Metric	Regression CNN	UNet Regression CNN	Improvement (%)
MSE	0.0059	0.0032	45.95%
MAE	0.0565	0.0382	32.37%
PSNR (dB)	22.86	26.17	14.52%
SSIM	0.8447	0.9485	12.28%

2.4 Visual Results (Regression Models)

Figure 1 presents a qualitative comparison of the colorization results generated by the two regression-based models. Visual inspection aligns with the quantitative metrics, showing that the UNet Regression CNN produces more visually plausible and detailed colorizations compared to the standard Regression CNN, which tends to yield more muted or blurry outputs.

2.5 Training Progress (Regression Models)

The training curves for the regression models are displayed in Figure 2. These plots typically show the evolution of loss (e.g., MSE or MAE) for both the training and validation sets over

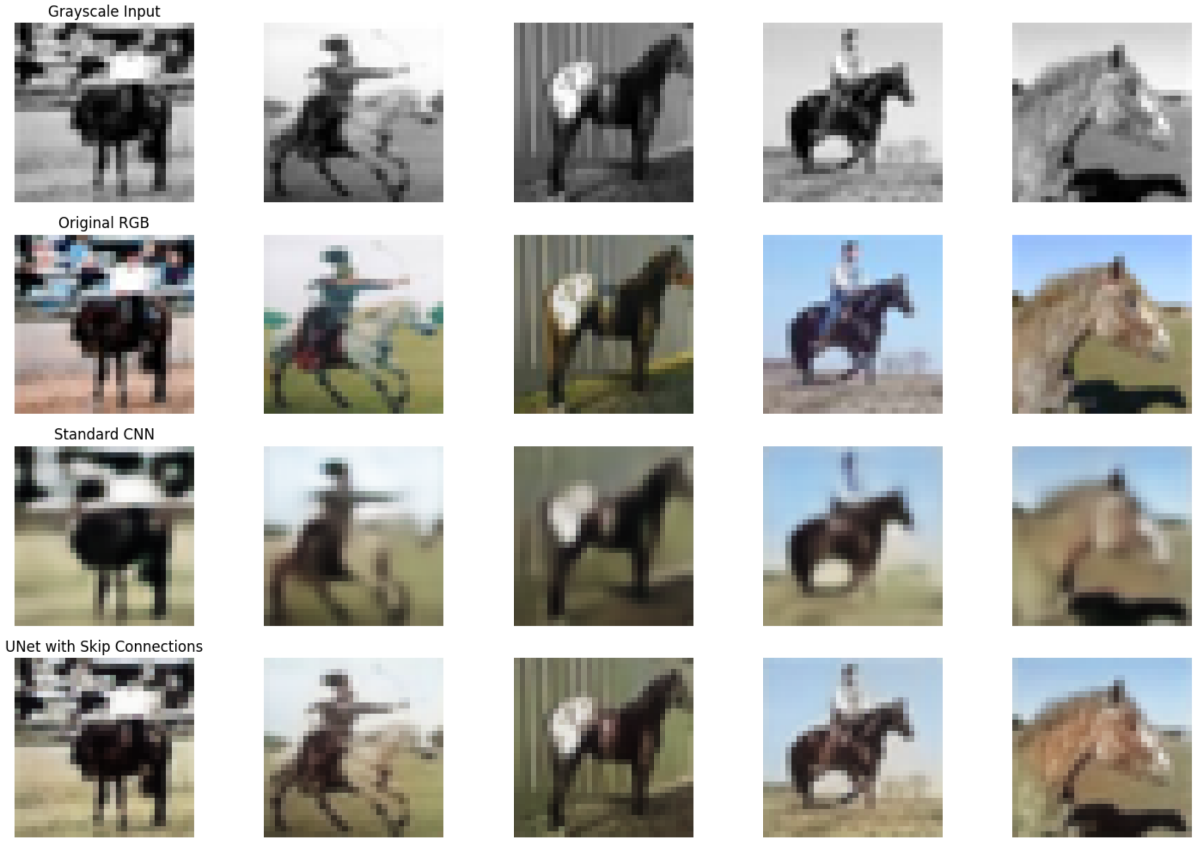

Figure 1: Visual comparison of regression-based colorization results. Rows (top to bottom): Grayscale Input, Ground Truth RGB, Regression CNN Output, UNet Regression CNN Output.

the epochs. The UNet Regression CNN generally shows faster convergence and achieves lower validation loss compared to the standard Regression CNN, correlating with its superior final performance. Early stopping was employed to prevent overfitting, triggered at epoch 21 for the standard CNN and epoch 33 for the UNet.

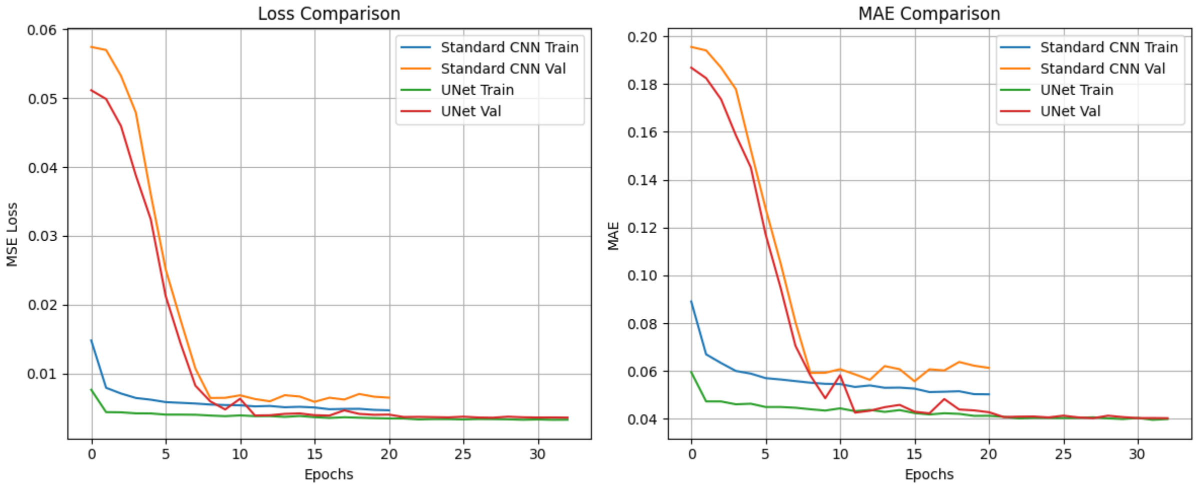

Figure 2: Training dynamics for the Regression models: Regression CNN (left) and UNet Regression CNN (right), showing Training and Validation Loss (or MAE) over epochs.

3 Classification-based Colorization Models

This alternative approach frames colorization as a pixel-wise classification problem.

3.1 Methodology

The RGB color space was quantized into 24 distinct color clusters using a suitable algorithm (e.g., K-Means on the training set colors). The models were then trained to predict the probability distribution over these 24 clusters for each pixel in the input grayscale image. The final color is determined by the cluster center corresponding to the highest probability prediction.

3.2 Standard CNN Architecture (Classification)

A standard encoder-decoder CNN, similar in structure to the regression baseline but adapted for classification output, was implemented. Its detailed architecture is shown in Table 4.

Table 4: Layer-by-Layer Architecture: Standard CNN (Classification)

Layer No.	Layer Type	Parameters	Output Shape
1	Conv2D	64 filters, 3x3, padding='same'	(32, 32, 64)
2	BatchNormalization		(32, 32, 64)
3	ReLU Activation		(32, 32, 64)
4	Conv2D	64 filters, 3x3, padding='same'	(32, 32, 64)
5	BatchNormalization		(32, 32, 64)
6	ReLU Activation		(32, 32, 64)
7	MaxPooling2D	2x2 pool size	(16, 16, 64)
8	Conv2D	128 filters, 3x3, padding='same'	(16, 16, 128)

Table 4: (Continued)

Layer No.	Layer Type	Parameters	Output Shape
9	BatchNormalization		(16, 16, 128)
10	ReLU Activation		(16, 16, 128)
11	Conv2D	128 filters, 3x3, padding='same'	(16, 16, 128)
12	BatchNormalization		(16, 16, 128)
13	ReLU Activation		(16, 16, 128)
14	MaxPooling2D	2x2 pool size	(8, 8, 128)
15	Conv2D	256 filters, 3x3, padding='same'	(8, 8, 256)
16	BatchNormalization		(8, 8, 256)
17	ReLU Activation		(8, 8, 256)
18	Conv2D	256 filters, 3x3, padding='same'	(8, 8, 256)
19	BatchNormalization		(8, 8, 256)
20	ReLU Activation		(8, 8, 256)
21	UpSampling2D	2x2	(16, 16, 256)
22	Conv2D	128 filters, 3x3, padding='same'	(16, 16, 128)
23	BatchNormalization		(16, 16, 128)
24	ReLU Activation		(16, 16, 128)
25	UpSampling2D	2x2	(32, 32, 128)
26	Conv2D	64 filters, 3x3, padding='same'	(32, 32, 64)
27	BatchNormalization		(32, 32, 64)
28	ReLU Activation		(32, 32, 64)
29	Conv2D (Output)	24 filters, 1x1, padding='same'	(32, 32, 24)

This model was trained for 25 epochs before early stopping.

3.3 UNet CNN Architecture (Classification)

A UNet architecture with skip connections was also adapted for the classification task. Its detailed architecture is shown in Table 5.

Table 5: Layer-by-Layer Architecture: UNet CNN (Classification)

Layer No.	Layer Type	Parameters	Output Shape
1	Conv2D	64 filters, 3x3, padding='same'	(32, 32, 64)
2	BatchNormalization		(32, 32, 64)
3	ReLU Activation		(32, 32, 64)
4	Conv2D	64 filters, 3x3, padding='same'	(32, 32, 64)
5	BatchNormalization		(32, 32, 64)
6	ReLU Activation		(32, 32, 64)
7	MaxPooling2D	2x2 pool size	(16, 16, 64)
8	Conv2D	128 filters, 3x3, padding='same'	(16, 16, 128)
9	BatchNormalization		(16, 16, 128)
10	ReLU Activation		(16, 16, 128)
11	Conv2D	128 filters, 3x3, padding='same'	(16, 16, 128)
12	BatchNormalization		(16, 16, 128)
13	ReLU Activation		(16, 16, 128)
14	MaxPooling2D	2x2 pool size	(8, 8, 128)

Table 5: (Continued)

Layer No.	Layer Type	Parameters	Output Shape
15	Conv2D	256 filters, 3x3, padding='same'	(8, 8, 256)
16	BatchNormalization		(8, 8, 256)
17	ReLU Activation		(8, 8, 256)
18	Conv2D	256 filters, 3x3, padding='same'	(8, 8, 256)
19	BatchNormalization		(8, 8, 256)
20	ReLU Activation		(8, 8, 256)
21	UpSampling2D	2x2	(16, 16, 256)
22	Concatenate	Skip with Layer 13 output	(16, 16, 384)
23	Conv2D	128 filters, 3x3, padding='same'	(16, 16, 128)
24	BatchNormalization		(16, 16, 128)
25	ReLU Activation		(16, 16, 128)
26	Conv2D	128 filters, 3x3, padding='same'	(16, 16, 128)
27	BatchNormalization		(16, 16, 128)
28	ReLU Activation		(16, 16, 128)
29	UpSampling2D	2x2	(32, 32, 128)
30	Concatenate	Skip with Layer 6 output	(32, 32, 192)
31	Conv2D	64 filters, 3x3, padding='same'	(32, 32, 64)
32	BatchNormalization		(32, 32, 64)
33	ReLU Activation		(32, 32, 64)
34	Conv2D	64 filters, 3x3, padding='same'	(32, 32, 64)
35	BatchNormalization		(32, 32, 64)
36	ReLU Activation		(32, 32, 64)
37	Conv2D (Output)	24 filters, 1x1, padding='same'	(32, 32, 24)

This model was trained for 22 epochs before early stopping.

3.4 Classification Performance Metrics

The classification-based models were evaluated using the same metrics (MSE, MAE, PSNR, SSIM) applied to the final colorized images (after mapping predicted cluster indices back to cluster center colors). The results are summarized in Table 6.

Table 6: Classification Models: Quantitative Performance Comparison

Metric	Standard CNN (Class.)	UNet CNN (Class.)	Improvement (%)
MSE	0.0069	0.0050	27.66%
MAE	0.0618	0.0500	19.15%
PSNR (dB)	21.98	23.83	8.43%
SSIM	0.7770	0.8578	10.41%

While the UNet again outperforms the standard CNN within the classification paradigm, the absolute performance metrics are generally slightly worse than their regression counterparts, potentially due to the information loss from color quantization.

3.5 Activation Analysis (Classification Models)

Visualizing activations from selected layers provided insights similar to the regression models:

- Early layers in both standard and UNet classification models focused on detecting basic features like edges and textures.

- The UNet's skip connections proved crucial for maintaining spatial resolution and feature integrity through the network. Activations in its decoder path showed richer spatial detail compared to the standard CNN.

- The standard CNN's bottleneck layers exhibited significant loss of spatial information.

- UNet decoder activations demonstrated a clearer reconstruction process, leveraging the skipped features for higher fidelity.

3.6 Visual Results (Classification Models)

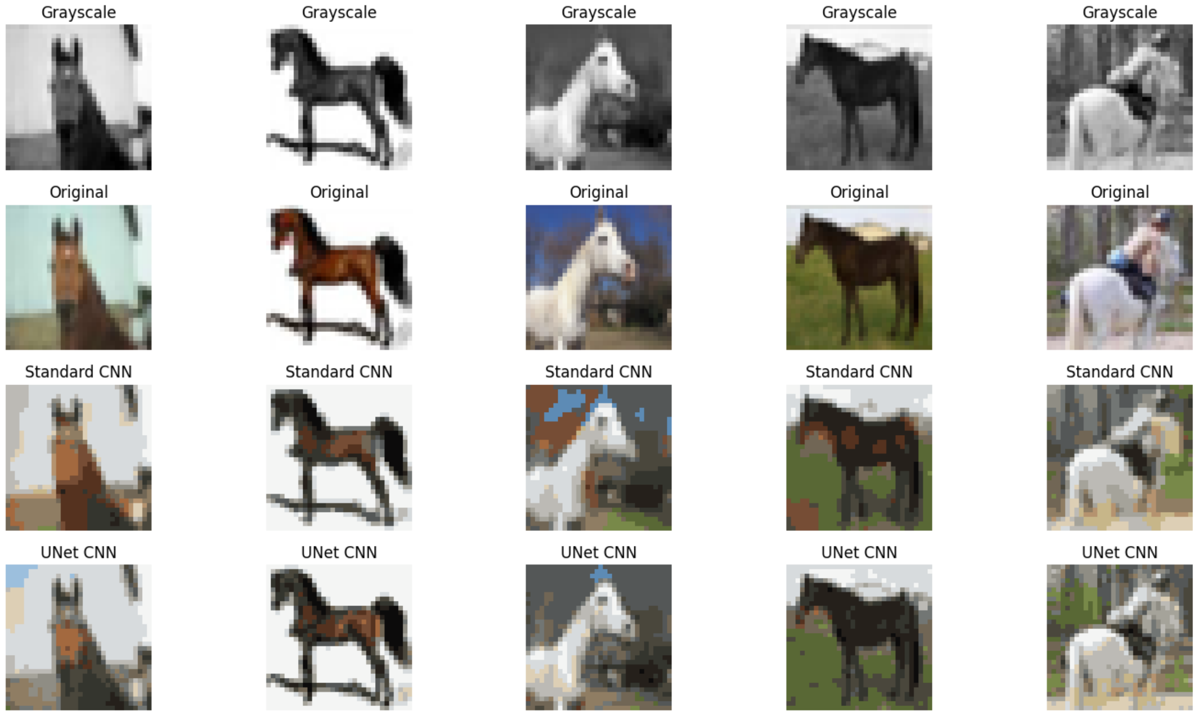

Figure 3: Visual comparison of classification-based colorization results. Rows (top to bottom): Grayscale Input, Ground Truth RGB, Standard CNN Output (Class.), UNet CNN Output (Class.).

As shown in Figure 3, the UNet model adapted for classification produced more vibrant and perceptually accurate colors than the standard classification CNN. The latter often resulted in more muted or patchy colorations, likely due to the combined effects of quantization and architectural limitations. The UNet's ability to preserve detail via skip connections was visually evident.

3.7 Training Progress (Classification Models)

The training progress for the classification models (Figure 4) typically involves monitoring classification loss (e.g., Categorical Cross-Entropy) and accuracy. Similar to the regression case, the UNet classification model generally exhibited better convergence and validation performance compared to the standard CNN. (Note: The caption assumes typical classification metrics; adjust if different metrics were plotted).

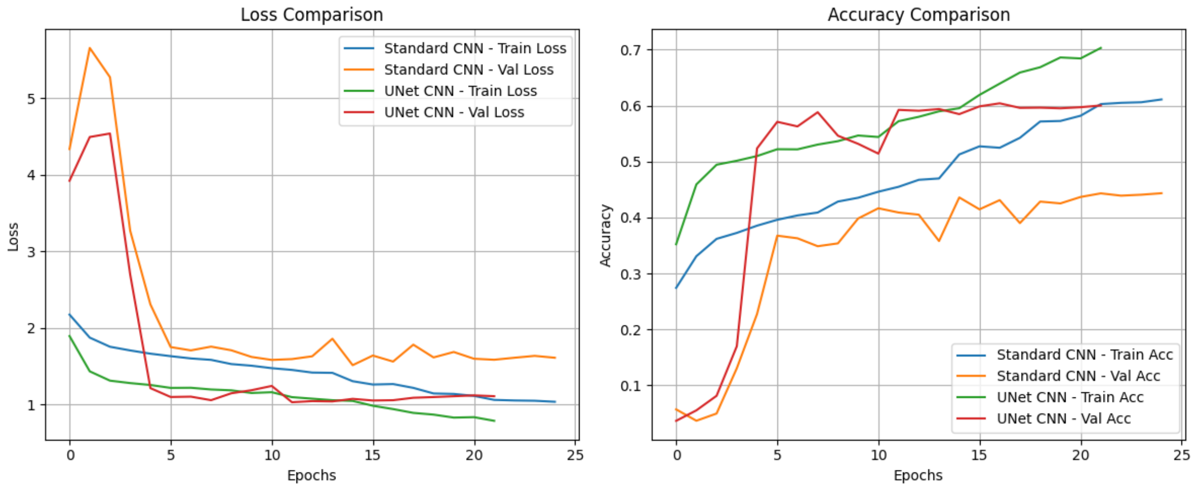

Figure 4: Training dynamics for the Classification models: Standard CNN (left) and UNet CNN (right), showing Training and Validation Loss/Accuracy (or other relevant metric like MAE if used) over epochs.

4 Summary of Colorization Findings

Across both regression and classification approaches, several key observations emerged:

- **Benefit of Skip Connections (UNet Architecture)**:
  - Consistently improved pixel-level accuracy metrics (lower MSE/MAE, higher PSNR/SSIM) by providing pathways for high-resolution features from the encoder to the decoder.
  - Enabled superior preservation of structural information and fine details in the output images.
  - Likely contributed to more stable training by mitigating potential gradient vanishing issues in deeper networks.
- **Visual Quality Comparison**:
  - UNet models (both regression and classification) produced more vibrant, sharp, and perceptually plausible colorizations.
  - Standard CNN models tended towards producing more muted, averaged, or blurry colors, especially noticeable in the regression approach.
  - The UNet's skip connections were critical for reconstructing finer details effectively.
- **Intermediate Activations**:
  - Activation patterns confirmed the theoretical benefits: early layers captured low-level features in all models.
  - UNet activations demonstrated better spatial information retention throughout the network, particularly across skip connections and into the decoder layers.
  - Standard CNN activations showed clear information loss at the bottleneck, hindering detailed reconstruction.
- **Regression vs. Classification**:
  - Regression models achieved slightly better quantitative scores (MSE, MAE, PSNR, SSIM) in this study.

  - Classification can sometimes produce more vibrant results by avoiding the averaging effect inherent in regression losses, but is limited by the quantization level. The choice may depend on the specific dataset and desired output characteristics.

5 Discussion

5.1 Improving Evaluation Metrics

Pixel-level metrics like MSE and PSNR often correlate poorly with human perception of colorization quality. Plausibility and visual appeal are subjective. To improve evaluation:

- **Perceptual Loss Functions:** Integrate losses based on high-level feature differences in pre-trained networks [1].
- **User Studies:** Conduct evaluations where human raters assess the naturalness and quality of results.
- **Distributional Metrics:** Use metrics like FID (Fréchet Inception Distance) to compare the statistical distribution of generated image features against real image features.

5.2 Adapting Models for Larger Images

The fully convolutional nature of the implemented CNNs (Standard and UNet) allows them to process inputs of varying sizes without architectural modification. When applied to test images larger than the 32x32 training size:

- The models will produce correspondingly larger output maps.
- Performance depends on the generalization capability of features learned on small images. While basic colorization might work, fine details or object-specific coloring learned from CIFAR-10 might not translate perfectly to significantly different, larger images without fine-tuning or domain adaptation.

6 Conclusion

This report compared standard CNN and UNet architectures for image colorization using both regression and classification paradigms on CIFAR-10 horse images. The UNet architecture consistently outperformed the standard CNN in both quantitative metrics and qualitative visual results, regardless of whether regression or classification was used. The skip connections inherent to the UNet design proved crucial for preserving spatial detail and enabling higher-fidelity color reconstruction. While regression yielded slightly better standard metrics in this instance, both approaches benefited significantly from the UNet structure. These findings underscore the importance of architectural design, particularly the use of skip connections, for generative image tasks requiring detailed spatial understanding like colorization.

References

[1] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.