Introduction

The goal of this project was to develop a deep learning model capable of colorizing grayscale images in a natural way. The process involved experimenting with multiple architectures and post-processing techniques to address challenges like low color fidelity and splotchy color artifacts.

Summary of Different Methods Attempted

Over the course of development, several methods were tested:

- Basic VGG16 Feature Extractor + Simple Decoder:
  - Initially, a basic encoder-decoder structure based on VGG16 was used, with a few transposed convolutions for upsampling.
  - Loss function was Mean Squared Error between predicted and ground truth ab channels.
  - Output quality was blurry and muted.
- Classification-based Colorization (Quantized ab Bins):
  - I used a precomputed file, pts_in_hull.npy, obtained from Richard Zhang's Colorful Image Colorization project, which provides 313 ab cluster centers derived from k-means clustering over ImageNet colors.
    - These bins were used for quantizing and reconstructing the ab color space in my model.
    - https://github.com/richzhang/colorization
  - Instead of regression, the model predicted a class label for each pixel.
  - CrossEntropy loss was used.
  - Gave sharper colors, but often resulted in noisy/splotchy outputs.
- Perceptual Loss Approach:
  - Experimented briefly with VGG-based perceptual losses.
  - Led to richer textures but still had instability in color regions.
- Post-processing with Gaussian Blur:
  - Applied simple Gaussian blur to smooth low-confidence regions.
  - Helped reduce noise but also caused over-smoothing.
- Dynamic Confidence-based Blending:
  - Developed a technique to blend the raw prediction with a blurred version based on confidence scores.
  - Improved results considerably by keeping high-confidence details while smoothing uncertain regions.
- Iterative Color Splotch Suppression:
  - Several versions of splotch removal were tested:
    - Red pixel suppression.
    - General color splotch suppression based on saturation and low confidence.

        - Neighborhood-based repair of low-confidence pixels.
    - Caused holes in the image
- Model tracking
    - Attempted training monitoring using TensorBoard, but it struggled to organize data between many runs.

Final Version

The final version integrates all the best techniques into a single pipeline.

1. Model Architecture:

- Encoder:
    - A pretrained VGG16-BN network.
    - Extracts multiscale features.
    - Layers are frozen to retain rich features without overfitting.
- Decoder:
    - Three transposed convolutional layers for upsampling.
    - Skip connections from encoder layers to decoder layers.
    - Final 1x1 convolution outputs logits over 313 quantized color bins.

2. Training Strategy:

- Dataset:
    - Custom ColorizationDataset that loads images and quantizes ab channels.
- Loss Function:
    - CrossEntropyLoss with class balancing weights (precomputed from training data).
- Optimizer:
    - Adam optimizer with initial learning rate of 1e-3.
- Training Setup:
    - Interrupt-safe saving of model checkpoints.

3. Post-processing Pipeline:

- Softmax Temperature Scaling:
    - Applied temperature sharpening to the softmax output for smoother probability maps.
- Confidence-Weighted Blending:
    - The raw ab prediction is blended with a spatially blurred version based on pixel confidence.
    - Higher-confidence areas retain original sharpness, lower-confidence areas are softened.
- Color Splotch Suppression:

- suppress_color_splotches3 method identifies high-saturation, low-confidence pixels.
- Neighborhood-aware averaging replaces problematic pixels.
- Fallback to Gaussian blur if insufficient confident neighbors.

4. Evaluation:

- Metrics:
    - PSNR (Peak Signal-to-Noise Ratio)
    - MSE (Mean Squared Error)
- Visualization:
    - Grayscale input, raw output, smoothed output, and ground truth color images displayed side-by-side.

5. Output Examples:

- Side-by-side comparisons show visibly reduced splotches.
- PSNR and MSR scores improved after smoothing and splotch suppression.

Comparison and Reasoning

The final method was chosen because it consistently delivered:

- Sharper and more natural colors than the basic regression model.
- Stronger resilience to color noise compared to raw classification outputs.
- A flexible framework to handle uncertain predictions gracefully.
- Significant visual improvement with minimal artifacts after applying smoothing and neighborhood repairs.

Earlier approaches either over-smoothed the entire image or left distracting splotches in uncertain regions. The combined confidence-based methods allowed for fine-grained, adaptive post-processing.

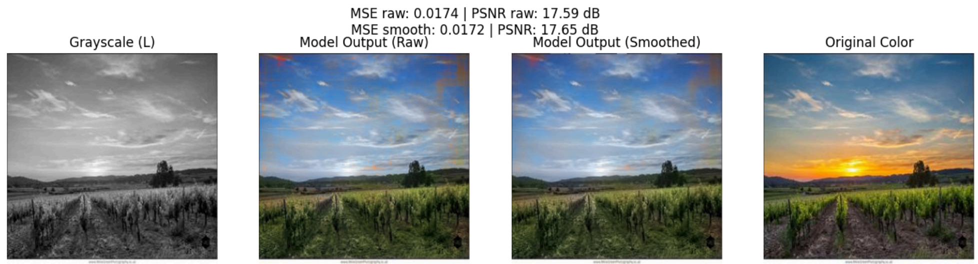
Conclusion

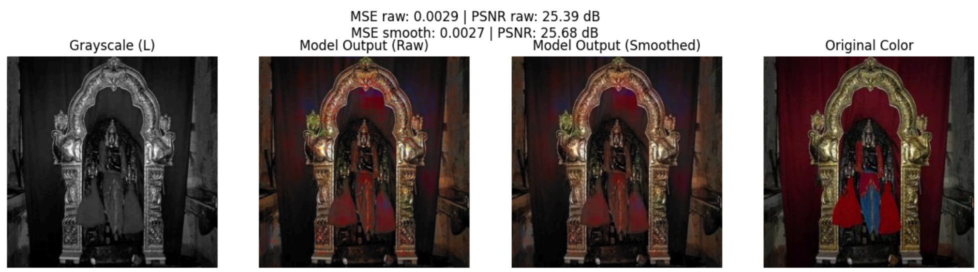
This project evolved significantly from a basic colorization model to a sophisticated confidence-aware system. The final version leverages pretrained feature extraction, quantized classification, confidence-weighted blending, and targeted splotch suppression to produce high-quality, stable colorizations. There are still issues with unnatural color splotches. While there's still room for further refinement, the current system strikes a strong balance between complexity and visual quality.

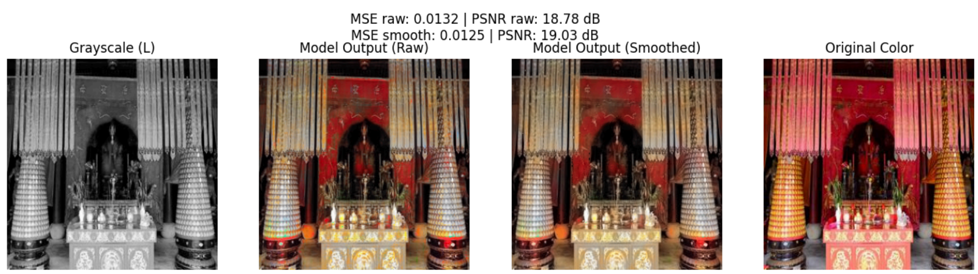
Results

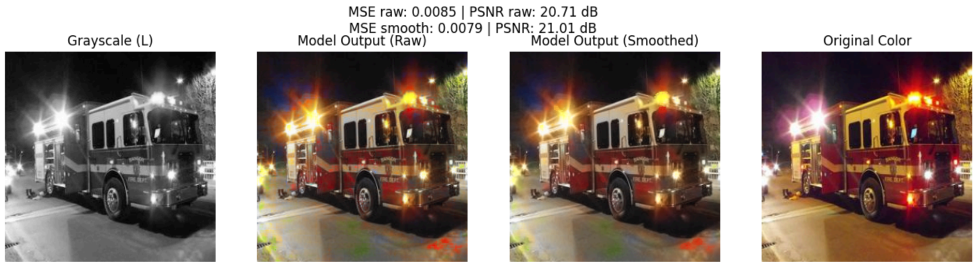
Furthest left: Grayscale version of the image
Second from the left: Raw model output
Second from the right: Smoothed version of the raw model output
Furthest right: Original image

Good/Decent Outputs:









Poor Outputs/Needs Improvement:

Didn’t understand skin tone

Mistook the ceiling for sky