IMAGE COLOURIZATION VIA CONVOLUTIONAL NEURAL NETWORKS AND DEEP LEARNING

Youssef Fikry
Student# 1005678901
youssef.fikry@mail.utoronto.ca

Harkirpa Kaur
Student# 1011242479
harkirpa.kaur@mail.utoronto.ca

Peter Leong
Student# 1005678901
peter.leong@mail.utoronto.ca

Thulasi Thavarajah
Student# 10115358424
t.thavarajah@mail.utoronto.ca

ABSTRACT

This project addresses the challenge of automated colourization for 256×256 grayscale images using a dataset of 12,600 image pairs, balanced across human subjects, animals, and natural scenery. We frame colourization as a supervised learning problem in the CIELAB colour space, where a model predicts chrominance channels (a^*, b^*) from the luminance channel (L^*). A shallow convolutional neural network (CNN) provides the baseline performance, while our primary solution employs a deeper convolutional encoder-decoder architecture. This design captures high-level semantic features and spatial context, addressing limitations of shallow networks in perceptual realism. All source code, datasets, and results are publicly available here. —-Total Pages: 6

1 BRIEF PROJECT DESCRIPTION

The invention of photography provided the technology to capture a moment in time. However, for most of photographic history, the process of obtaining coloured images eluded photographers (**?**). As a result, much of historic photography is grayscale; lacking the visual richness found in modern photography and is inaccessible to individuals with vision impairments (**?**). This project aims to leverage deep learning to automate the colorization of grayscale images, thereby aiding in the revitalization of grayscale photographs. Furthermore, image colorization technology assists archivists and museums in restoring lost visual information and has applications in the media, medical and geospatial industries.

Traditional image colorization methods involve manual labour that is costly and time-consuming (**?**). On the other hand, deep learning approaches automate the colorization process and supplant traditional techniques (**?**). In machine learning, colorization is defined as a model taking a black-and-white image as an input (see Figure 1), and outputting its coloured counterpart (**?**). From a machine learning perspective, deep convolutional neural networks are best suited for this task as they can extract and learn features such as colors, patterns and shapes in images and affiliate them with object classes (**?**). These characteristics aid CNNs in excelling at object classification tasks, as well as image colorization (**?**).

2 INDIVIDUAL CONTRIBUTIONS & RESPONSIBILITIES

The team is collaborating effectively by dividing the work based on technical strengths and ensuring transparency through regular updates. Thulasi has led the data processing pipeline, handling cleanup, data splitting, and restructuring of the dataset for human, animal, and scenic image categories. Peter developed and trained the baseline model using a shallow CNN architecture and is responsible for model training, as he has access to a better GPU. Youssef and Kirpa are responsible

Figure 1: Example Input, Output and Ground Truth From Primary Model

for model optimization and implementation; however, all modelling decisions are made collaboratively. The current model is producing brown-hued images, and discussion is underway to find possible fixes. The team communicates through a dedicated Discord server, while all code is written on Colab and organized in a shared Google Drive, along with meeting minutes and the Gantt chart. All project deliverables are stored in a public GitHub repository with separate branches for each team member to avoid merge conflicts. The meeting leader's role alternates weekly, and the Gantt chart is updated during each meeting to track progress. A Gantt chart showing the updated project plan and work distribution can be found in Appendix A.

Redundancies are in place to minimize project risks. Currently, Peter runs the model training due to GPU access, but both Youssef and Harkirpa have compatible hardware to take over if needed. All code is documented with extensive comments and stored centrally in the shared drive to ensure that any member can interpret and modify it if someone becomes unavailable.

3 Notable Contribution

3.1 Data Processing

The team compiled the project's image dataset using Kaggle.com, an online platform that provides access to publicly available, real-world datasets and a community for data scientists (**?**). All selected datasets have a public domain license. For the final model to be able to generalize to a vast range of images, the final dataset for this project has three classes: human, animal, and scenic.

3.1.1 Repurposing Online Datasets

The chosen human, animal and scenic datasets were originally compiled to serve specific tasks for machine learning. However, the team repurposed these datasets to train the model in colourising images using deep learning. For instance, the human image dataset was initially intended for human detection tasks; it contains 17,300 images of people in a comprehensive range of environments, including a wide variety of colors (**?**). Additionally, the animal image dataset was developed for image classification and contains 90 different classes of animals equally distributed across 5,400 images (**?**). Moreover, the scenic image dataset has 4,319 images of a variety of scenic landscapes (**?**). For the purpose of this project, these datasets are optimal as these encompass a broad range of colour palettes, possibly impacting the robustness of the final model.

3.1.2 Data Collection Challenges and Evaluation

A challenge faced during data processing was finding a ready-made dataset online intended for image colourization. The datasets available on Kaggle.com for image colourization contained around 5,000 images, which the team determined was insufficient to split into training, validation and test sets. Furthermore, while these datasets had grayscale and coloured pairs, the files were named inconsistently. This introduced the possibility of mismatched image pairs upon download, leading to skewed results when training and incorrect losses/errors. As a result, the team decided to compile a

unique dataset for this project using multiple different datasets from Kaggle.com, and creating the input grayscale counterparts of the colored images found in these sets.

3.1.3 CLEANING UP THE DATASETS

All three datasets were uploaded to Google Drive. Using Google Colab, the images were relocated into folders according to their category (human, animal or scenic). To equally distribute the dataset among the three categories, each dataset was then reduced to 4200 images using Python's `random.sample` function with the seed set to 42. Furthermore, the image files were renamed with respect to their category (ex. human_0001.jpg).

3.1.4 FORMATTING THE DATA AND CREATING A UNIQUE DATASET

This project requires a unique dataset with black-and-white images paired with their coloured counterparts as the ground truths. To format the cleaned up data and create this dataset, PyTorch's `torchvision.transforms` library was utiltized. The team first converted the original images into 256 x 256 pixel ground truth images using the `Resize` transform and sorted them according to their category. Following this, `Grayscale` transform with a output channel of 1 was used to convert the 256 x 256 pixel colour images to 256 x 256 pixel grayscale images to feed into the model.

human_0004.jpg bw_human_0004.jpg

Figure 2: Sample Data Pair from Training Set

3.1.5 SPLITTING DATASET INTO TRAINING, VALIDATION AND TEST SETS

To create the training, validation and test sets, a ratio of 70:15:15 was chosen. The first 2940 images of each of dataset categories (human/animal/scene) were selected for the training set. The remaining images were split evenly to create the validation and testing datasets.

3.1.6 THE FINAL DATASET

The final dataset is composed of 12,600 pairs of colour and their corresponding grayscale images. Each category (human, animal, and scenic) contains 4,200 image pairs and an even distribution of each category was maintained in the training, validation and test sets.

3.2 BASELINE MODEL

To establish a performance benchmark for our colorization task, we implemented a shallow convolutional neural network called `ShallowColorizationNet`. This model uses a simple encoder-bottleneck-decoder architecture and does not incorporate adversarial training. Its purpose is to test the feasibility of learning a direct mapping from grayscale luminance (L) input to the chrominance (ab) channels in the CIELAB color space.

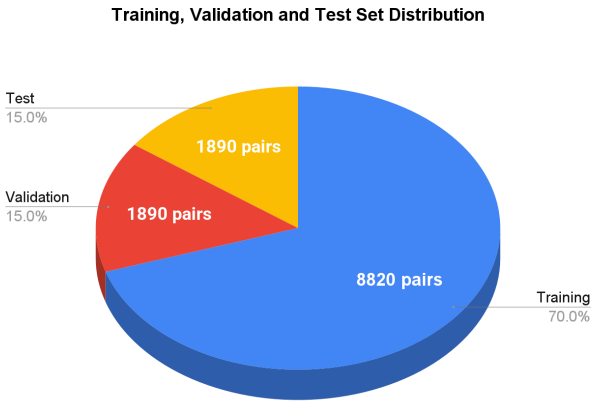

Figure 3: Training, Validation and Test Set Distribution of 12600 Image Pairs

3.2.1 Model Architecture

The architecture of the baseline model is as follows:

- **Encoder:** Two convolutional layers with ReLU activation followed by max pooling, reducing spatial dimensions while increasing channel depth.
- **Bottleneck:** A single convolutional layer that expands the feature representation.
- **Decoder:** A transposed convolution to restore spatial resolution and a final convolutional layer outputting two channels (for a and b), with a $\tanh$ activation to constrain values between $[-1, 1]$.

3.2.2 Training and Comparison Approach

The baseline model was trained using the L1 loss between predicted and ground truth *ab* channels. This model was compared to our primary neural network—a conditional GAN model that includes a discriminator to encourage more realistic colorizations—based on both quantitative and qualitative results.

3.2.3 Quantitative and Qualitative Results

Quantitative: The baseline model was trained for 5 epochs using L1 loss. The generator's loss remained stable around **1.5647**, while the discriminator's loss converged to approximately **0.7074**. These consistent values suggest that the baseline network was able to learn a basic mapping from grayscale to chrominance, but likely lacked the capacity or incentive to produce high-fidelity or diverse outputs. These values serve as reference points for evaluating improvements made by our primary GAN-based model.

Qualitative: The baseline model produced smooth but somewhat desaturated colorizations. It tended to predict average colors in ambiguous regions, resulting in low color diversity. See Figure 5 for representative examples.

3.2.4 Challenges

The primary challenge encountered during development was the tendency of the baseline model to learn overly conservative color predictions. Without a discriminator or explicit diversity loss, the network favored colorizations close to the dataset mean. Additionally, selecting the appropriate normalization strategy for LAB space and constraining output to valid chrominance ranges required careful tuning.

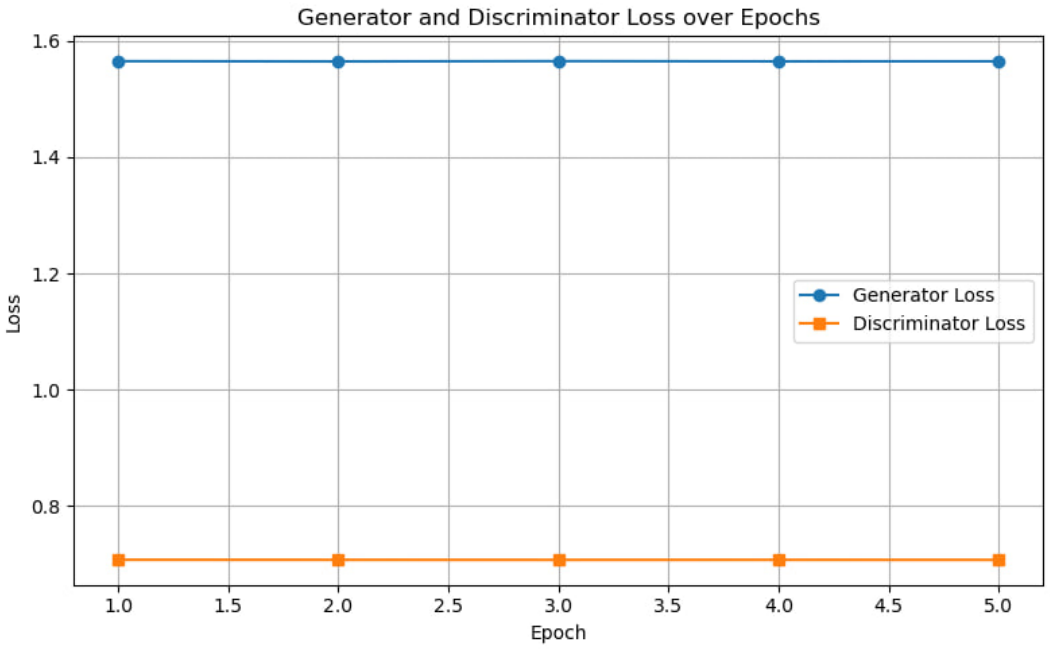

Figure 4: Generator and discriminator loss values over 5 epochs of training. The architecture will be investigated further during the final phase of the project to determine why the loss values fluctuate by such small amounts.

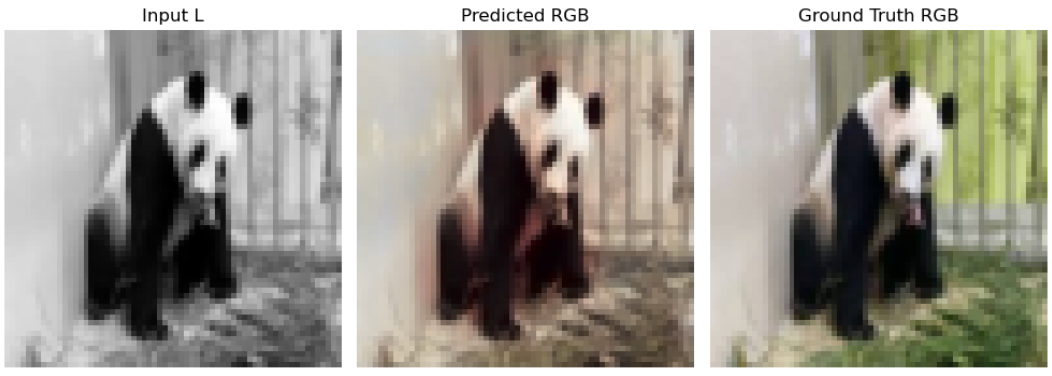

Figure 5: Sample output from the baseline model on grayscale input images.

3.2.5 Feasibility Assessment

Despite its simplicity, the baseline model demonstrated that the colorization task is learnable with a low-capacity network, although the outputs lacked the vividness and fidelity achieved by our primary GAN-based model. The baseline thus served its role in confirming the viability of end-to-end colorization from grayscale input and establishing a benchmark for model improvement.

3.3 Primary Model

Our colouriser combines a lightweight ResNet-18 encoder with a U-Net decoder. This design keeps global scene context (from the pretrained encoder) while preserving fine edges through skip connections, all within a 15 M-parameter budget suitable for Google Colab.

3.3.1 Architecture

The encoder copies the first four blocks of ResNet-18, pretrained on ImageNet, but the opening convolution is modified to ingest a single-channel L image. Feature maps are therefore produced at 256×256, 128×128, 64×64 and 32×32. A symmetric decoder upsamples in three steps (transpose-conv $\rightarrow$ 3 × 3 conv × 2, BatchNorm, ReLU). Skip links inject the corresponding encoder features to keep edges crisp. A final 3 × 3 layer with tanh activation outputs the a and b chroma channels. Total capacity is 15 M parameters, comfortably below Colab's memory ceiling. The whole network sits under 25 MB on disk, so checkpoints save quickly.

3.3.2 Training Progress

We train with AdamW (lr = 3×10^{-4}, batch = 32) on the cleaned 12.6 k-image set, and the learning rate decays each epoch via cosine annealing. Five epochs on a pro A100 GPU took just under 20 minutes. Validation L1 dropped from 0.07 to 0.0064, and SSIM climbed from 0.58 to 0.74. Qualitatively the model recovers plausible sky blues and fur textures that the two-layer baseline could not capture. The model also learns to colorize human skin tones, but it struggles with complex scenes like forests and crowds, where it tends to produce muddy colors.

3.3.3 Challenges

On some photos the network over-uses warm hues, giving a slight brown cast. Early inspection shows that brown-dominant scenes are over-represented in the training set and the loss does not penalise colour bias. We plan to rebalance LAB histograms per batch and add a small perceptual term to discourage monotone outputs.

3.3.4 Feasibility Assessment

After five epochs the model already lifts SSIM from 0.58 (baseline) to 0.74, showing it can learn credible colour cues. Remaining errors are mostly a warm "brown" bias. Because these mistakes are consistent, modest adjustments should yield clear gains. We plan to (i) rebalance training batches so cool and warm tones appear equally, and (ii) add a light perceptual loss to discourage over-use of one hue. These changes require no extra parameters or memory. With a 20-epoch run and a cosine learning-rate schedule, we expect further improvement toward an SSIM of 0.80 while staying within course limits.

4 APPENDIX A: GANTT CHART

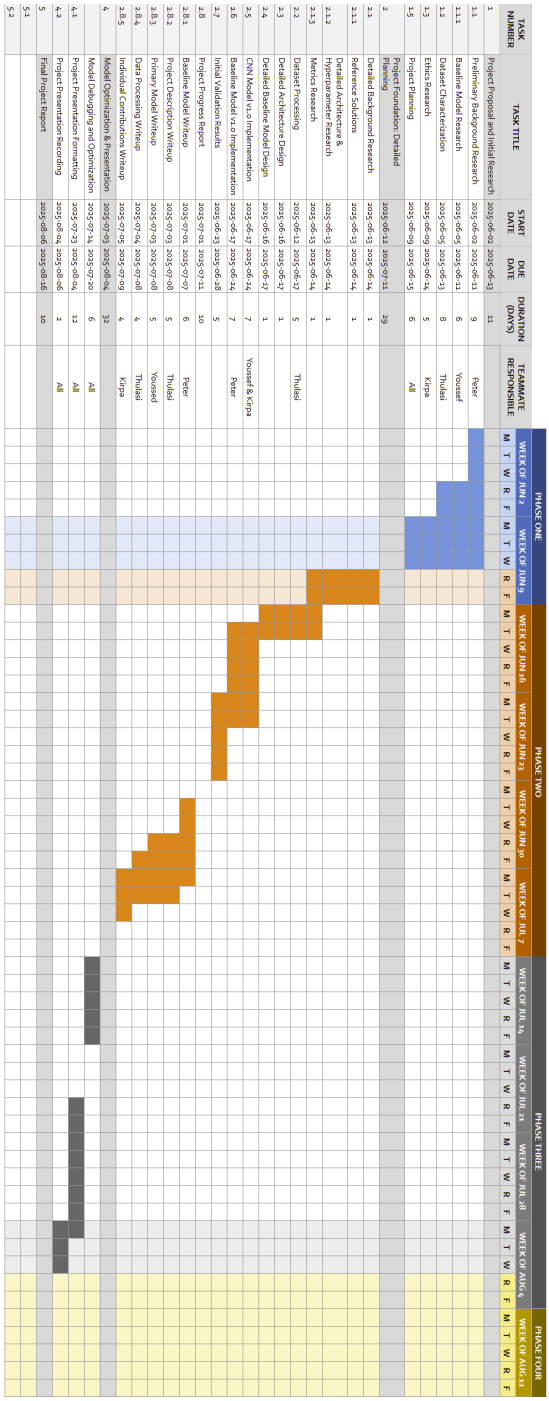

Figure 6: Figure A1. Team 40 Gantt chart, updated July 10, 2024