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 proposal explores automatic image colourization of 256x256 grayscale images using a dataset of 12,600 image pairs balanced across human subjects, animals, and natural scenery. The task is approached as a supervised learning problem in the CIELAB colour space, where the model predicts the chrominance channels given the luminance input. A shallow convolutional neural network (CNN) serves as the baseline architecture to establish initial performance. The primary proposed solution is a deeper convolutional encoder-decoder network designed to capture higher-level semantic features and spatial context, enabling more accurate and perceptually realistic colourizations. Our Google Drive containing all source code and datasets can be found here. —-Total Pages: 6

1 INTRODUCTION

While colour photography processes first emerged in the 1890s, colour photography did not become widely accessible until the 1970s (Science & Museum, 2020). As a result, most historic photographs are black and white and lack the visual richness that modern viewers are accustomed to. In addition, individuals who have their cataracts removed as a part of vision restoration processes have shown to struggle with identifying objects in grayscale images Vogelsang et al. (2024), making most historic photographs inaccessible to them. This project aims to use deep learning to automatically colourize black and white images, with the goal of restoring visual information and making historical imagery more accessible for all audiences. Traditional, non deep-learning based colourization methods often produce desaturated results and rely heavily on human guidance, making them non-scalable (Cheng et al., 2016). Conversely, deep networks such as CNNs effectively capture spatial and semantic features and produce realistic colourized images without user interaction, making deep learning an ideal approach for image colourization (Zhang et al., 2016).

1.1 BACKGROUND & RELATED WORK

The challenge of image colourization has been approached in a wide variety of ways. Even within solutions involving deep learning techniques, there are numerous unique design choices. One grouping method results in five categories: simple colourization neural networks, user-guided colourization neural networks, diverse colourization neural networks, multi-path colourization neural networks, and exemplar-based neural networks (Zĕger et al., 2021).

Simple colourization neural networks use feedforward CNNs to map grayscale inputs to colour outputs. One of the foremost solutions proposed by Zhang et al. (2016) used a fully convolutional network to predict the a and b channels of the CIELAB colour space from grayscale images. Their architecture is composed of several convulational layers, each followed by a ReLU activation function and a batch normalization layer.

User-guided colourization neural networks utilize user input to guide the colourization process. One such solution uses a fully convolutional network to predict the a and b channels of the CIELAB colour space from grayscale images, but also takes user input in the form of user-provided colour scribbles (Zhang et al., 2017). The network is trained to minimize the difference between the predicted and user-provided colours, allowing it to learn to colourize images in a way that is consistent with the user's input.

Diverse colourization networks produce multiple colourization outputs for a given grayscale input. One such solution uses a generative adversarial network (GAN) to produce multiple colourization outputs for a given grayscale input (Vitoria et al., 2020). The GAN is trained to minimize the difference between the predicted and ground truth colours, allowing it to learn to produce diverse colourization outputs.

Multi-path colourization neural networks differentiate features at different scales. Iizuka et al. (2016) proposed a multi-path colourization neural network that uses multiple convolutional layers to extract features at different scales. The network is trained to minimize the difference between the predicted and ground truth colours, allowing it to learn to produce colourization outputs that are consistent with the features at different scales.

Exemplar-based neural networks use a set of exemplar images to guide the colourization process. In Su et al. (2020), example images are used to transfer the colour to the target image. Each instance is output to two different colourization networks which fuse to yield the final result. This group of solutions is easier to implement, as learning to colourize instances is significantly easier than learning to colourize an entire image.

2 METHODOLOGY

This section outlines our methodology for training and evaluating models on a dataset of 256×256 black-and-white images paired with their colour counterparts, comprising an even distribution of humans, animals, and natural scenery. We first detail the data preprocessing steps, followed by a description of our proposed encoder-decoder convolutional neural network architecture, which draws inspiration from a simplified U-Net design. A shallower CNN is introduced as the baseline model to enable comparative analysis of performance and colourization fidelity.

2.1 DATA PROCESSING

The project's image dataset was compiled using publicly available datasets on Kaggle.com, an online platform that provides access to real-world datasets and a community for data scientists (Kaggle). To train a model that can generalize to a broad range of images, the final dataset for this project includes three categories: human, animal, and scenic. All selected datasets are licensed for public domain use.

2.1.1 REPURPOSING ONLINE DATASETS

The human image dataset, originally intended for human detection, contains a diverse range of 17,300 images of people in different environments (Elmenshawii, 2023). Furthermore, the animal image dataset, initially developed for image classification contains 5,400 images of 90 different animals (Banerjee, 2022). For this project's purpose, this dataset is ideal as it encompasses a diverse set of images with an equal distribution of each animal. Additionally, the scenic image dataset from Rougetet (2020) contains 4,319 images of a variety of landscapes spanning a large breadth of colour palettes, potentially influencing the robustness of the final model.

2.1.2 CLEANING UP THE DATASETS

The team extracted all the images from each dataset and relocated them into folders corresponding to their category (human/animal/scenic). Due to the disparity in the size of the three datasets, each dataset was reduced to exactly 4200 images using Python's `random.sample` function with the seed set to 42. The images were then renamed in accordance to their respective categories (ex. human_0001.jpg).

2.1.3   FORMATTING THE DATA

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.

2.1.4   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.

2.1.5   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.
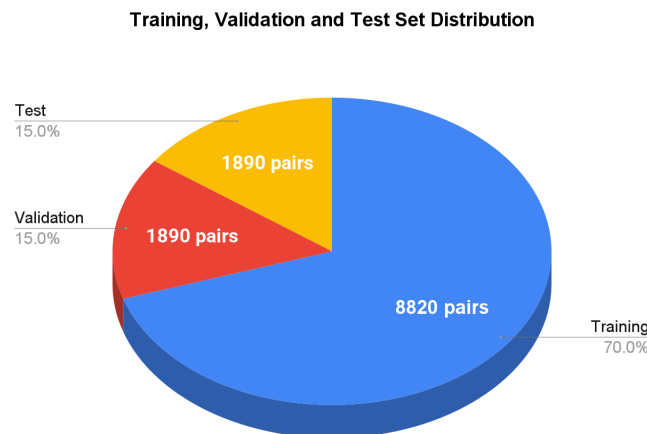

Figure 1: Training, Validation and Test Set Distribution of 12600 Image Pairs

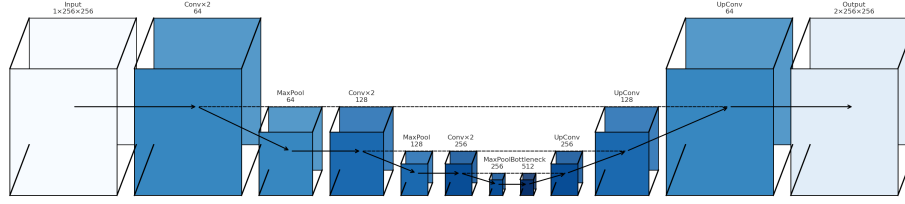Figure 2: Sample Data Pair from Training Set

2.2 ILLUSTRATION

Figure 3: Encoder–decoder colourization network with skip connections.

The top row is the encoder (Conv ×2 → MaxPool blocks) that downsamples the grayscale L channel from 256×256 to 32×32 while increasing feature depth (64 → 512). The bottom row is the decoder (UpConv blocks) that upsamples back to full resolution and predicts the two chroma channels (a, b). Dashed arrows show the three skip connections that pass fine-grained features from encoder to decoder stages.

2.3 ARCHITECTURE

We propose a convolutional encoder–decoder network for the colourization task, which is a common and effective choice for image-to-image translation problems (Leatvanich, 2025). The model will operate in the CIE Lab colour space: the input is a grayscale image (the L channel), and the network is trained to predict the two chrominance channels (a and b) as output. After prediction, the input L channel and the output a,b channels are combined and converted back to an RGB image for visualization. Using the Lab space decouples intensity from colour, which aligns with human vision and generally produces more realistic results than direct RGB prediction (Leatvanich, 2025).

The architecture itself is an encoder–decoder CNN resembling a simplified U-Net. The encoder consists of several convolutional layers (with ReLU activations and batch normalization) that gradually downsample the image, extracting higher-level features as the spatial size reduces. For example, we might start with a 3x3 conv layer with 64 filters, followed by another conv layer with 128 filters, each followed by a pooling (or stride-2 conv) to halve the spatial dimensions. After a few such layers, a bottleneck layer (e.g. a 3x3 conv with 256 filters) will capture abstract features of the grayscale input. The decoder then mirrors this process: it uses upsampling layers (e.g. Conv2DTranspose or nearest-neighbor upsampling) and additional conv layers to upsample the feature maps back to the original image resolution while predicting the a,b colour channels. We plan to include skip connections between matching encoder and decoder levels (as in U-Net) so that fine-grained details from early layers (edges, textures) are directly passed to the corresponding decoder layers. These skip connections should help the network recover details and prevent blurry outputs by combining low-level and high-level features Leatvanich (2025).

To balance performance and simplicity, we will leverage transfer learning for the encoder. In particular, we are considering initializing the encoder with a pretrained backbone like the first few layers of ResNet-18 or VGG16 (trained on ImageNet). This gives the model a head-start in recognizing semantic features—useful given our dataset of human and animal images, where recognizing objects like faces, fur, or backgrounds can inform colour choices (Olah & Yang, 2022). Using a pre-trained encoder should enable the network to learn meaningful colourization with fewer training examples and epochs (Olah & Yang, 2022). The decoder and any additional layers will be initialized randomly and learned from scratch. We will train the network end-to-end to minimize a pixel-wise loss between the predicted and ground-truth colour channels (e.g. mean squared error in Lab space).

It's worth noting that some prior works frame colourization as a classification problem over discrete colour bins to better capture the ambiguous nature of predicting colour. For instance, Zhang et al. (2016) predict a probability distribution over possible a,b values for each pixel instead of regressing exact values (Olah & Yang, 2022). While such techniques (often coupled with class re-balancing or perceptual losses) can produce vibrant results, they add complexity beyond the scope of a course project. In our architecture, we favor a straightforward regression approach – the network directly

outputs continuous a and b values – which is simpler to implement and sufficient for plausible colourization.

2.4 BASELINE MODEL

As a baseline, we will start with a very simple colourization approach against which to compare our full model. One trivial baseline is to output the input grayscale image as-is in colour (i.e. replicate the L channel into RGB), which yields a degenerate colourization. This sets a minimum benchmark – any learning model should outperform simply producing a monochrome image.

For a more meaningful baseline, we will implement a shallow convolutional network inspired by basic encoder–decoder examples (Leatvanich, 2025). This baseline model will take the grayscale L channel as input and produce a,b chrominance channels, but it will have only a small fraction of the capacity of our main architecture. For instance, the baseline could use just two convolutional layers for encoding (e.g. 64 filters and 128 filters) followed by a single upsampling decoder to reconstruct the output colours. Concretely, the image might pass through Conv2D(64) and Conv2D(128) layers (with ReLU), then a pooling layer to downsample; at that point a minimal "bottleneck" conv layer can be applied, and finally an upsampling (Conv2DTranspose) step generates the 2-channel output map (Leatvanich, 2025). This simplistic model does not incorporate any skip connections or pretrained knowledge, and it has far fewer parameters than the proposed full model. We expect its predictions will capture only basic correlations (for example, mapping lightness to a bland average colour) and often look desaturated or unrealistic (Rosebrock, 2019).

Such a baseline provides a useful point of comparison: it represents what a rudimentary learning method can achieve without much capacity or context. By evaluating our advanced model against this baseline, we can quantify the gains from using a deeper architecture and more sophisticated design. If the baseline's output is dull or mostly grayish (as often seen in naive colourization attempts), whereas our full model produces more vibrant and context-appropriate colours, it will demonstrate the effectiveness of the chosen architecture (Rosebrock, 2019).

3 ETHICAL CONSIDERATIONS

The dataset being used is public, so there are no copyright or consent issues. However, the dataset may contain racial or demographic imbalances, which could cause the model to generalize poorly or be biased towards specific skin tones. This may result in racially inaccurate or culturally insensitive outputs. A similar behaviour may be observed with animals, where a lack of diversity in breeds or fur colours in the dataset can result in misleading results. If the outputs produced by the model are used in educational contexts or in breed identification, they can contribute to misinformation. Furthermore, since the model results are plausible but cannot be verified, there is a risk that users may overtrust the outputs in sensitive contexts.

4 PROJECT PLAN

Our high-level project planning has been completed using a Gantt chart (see Appendix A). It contains many of the detailed tasks we have completed and will complete in the future. This easily shows who is responsible for each task, the time it will take to complete, and the dependencies between tasks. Generally, our internal deadlines have been set at least a few days prior to the official deadlines to ensure ample time to review our work and make necessary changes. In the event that a new member joins the team, they can easily refer to this chart to see what responsibilites they can take on and what tasks are still available.

4.1 TEAM COMMUNICATION & COORDINATION

We intend to use Discord as our primary communication platorm, Google Drive for file sharing (Colab notebooks, datasets, etc.), and GitHub for version control. To ensure we do not overwrite each other's work, we will use branches for each feature or task, and merge them into the main branch once they are complete. We will also have weekly meetings to discuss our progress, any issues we are facing, and plan for any action items for the subsequent meeting. Currently, we intend

to meet on Saturday mornings at 8:00 AM EST as some of our team members are in different time zones.

5 RISK REGISTER

This section outlines potential risks that may impact the progress or quality of the project, along with corresponding mitigation strategies. Identifying these risks early allows the team to plan contingencies and maintain project momentum even in the face of unforeseen challenges.

5.1 TEAM MEMBER(S) DROP COURSE

If a team member drops the course or is unable to contribute due to illness or other unforeseen circumstances—especially close to a major deadline—we will revisit the Gantt chart to reallocate their tasks evenly among the remaining team members based on individual availability and strengths. Where necessary, we will reduce the scope of non-critical tasks to ensure that core deliverables are still completed on time. Regular check-ins and shared documentation will help ensure continuity.

5.2 MODEL TRAINING TAKING LONGER THAN EXPECTED

Training time may exceed expectations due to hardware limitations, unexpected model complexity, or hyperparameter tuning. To mitigate this, we will build buffer time into our schedule to accommodate longer training runs. We will also regularly checkpoint models and keep logs for easy resumption. If training remains infeasible, we will simplify the architecture or reduce the dataset size without compromising class diversity, or shift to using cloud resources like Google Colab Pro if needed.

5.3 MODEL OVERFITS TO STEREOTYPICAL COLOURS

Our model may learn biases from the dataset, such as assuming all skies are blue or all humans have certain skin tones. To mitigate this, we will include a validation set with diverse examples, use data augmentation (e.g., colour jitter, flipping), and regularly evaluate qualitative outputs—especially for edge cases. We will also explore loss terms or regularization strategies that discourage overconfident predictions. If overfitting persists, we may rebalance the dataset or adjust the loss function to reduce dependence on prior colour distributions.

REFERENCES

Sourav Banerjee. Animal image dataset (90 different animals), 2022. URL `https://www.kaggle.com/datasets/iamsouravbanerjee/animal-image-dataset-90-different-animals`.

Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization, 2016. URL `https://arxiv.org/abs/1605.00075`.

Fares Elmenshawii. Human dataset, 2023. URL `https://www.kaggle.com/datasets/fareselmenshawii/human-dataset/data`.

Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (TOG)*, 35(4):110:1–110:11, July 2016. ISSN 0730-0301. doi: 10.1145/2897824.2925974. URL `https://doi.org/10.1145/2897824.2925974`.

Inc. Kaggle. Level up with the largest ai and ml community. URL `https://www.kaggle.com/`.

Crystal Leatvanich. Image colorization using neural networks, May 2025. URL `https://medium.com/@crystal.leatvanich/image-colorization-using-neural-networks-aeafd10e6ef5`.

Justin Olah and Jenny Yang. Let there be color: Deep learning image colorization, 2022. URL `https://cs231n.stanford.edu/reports/2022/pdfs/109.pdf`. CS231n Course Project Report.

Adrian Rosebrock. Black and white image colorization with opencv and deep learning, Feb 2019. URL `https://pyimagesearch.com/2019/02/25/black-and-white-image-colorization-with-opencv-and-deep-learning/`.

Arnaud Rougetet. Landscape pictures, 2020. URL `https://www.kaggle.com/datasets/arnaud58/landscape-pictures`.

National Science and Media Museum. A short history of colour photography, Jul 2020. URL `https://www.scienceandmediamuseum.org.uk/objects-and-stories/history-colour-photography`.

Jheng-Wei Su, Hung-Kuo Chu, and Jia-Bin Huang. Instance-aware image colorization, 2020. URL `https://arxiv.org/abs/2005.10825`.

Patricia Vitoria, Lara Raad, and Coloma Ballester. ChromaGAN: Adversarial Picture Colorization with Semantic Class Distribution. In *Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2434–2443, Los Alamitos, CA, USA, March 2020. IEEE Computer Society. doi: 10.1109/WACV45572.2020.9093389. URL `https://doi.ieeecomputersociety.org/10.1109/WACV45572.2020.9093389`.

Marin Vogelsang, Lukas Vogelsang, Priti Gupta, Tapan K. Gandhi, Pragya Shah, Piyush Swami, Sharon Gilad-Gutnick, Shlomit Ben-Ami, Sidney Diamond, Suma Ganesh, and Pawan Sinha. Impact of early visual experience on later usage of color cues. *Science*, 384(6698):907–912, 2024. doi: 10.1126/science.adk9587. URL `https://www.science.org/doi/abs/10.1126/science.adk9587`.

Ivana Zĕger, Sonja Grgic, Josip Vuković, and Gordan Šišul. Grayscale image colorization methods: Overview and evaluation. *IEEE Access*, 9:113326–113346, 2021. doi: 10.1109/ACCESS.2021.3104515.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. *European conference on computer vision*, 2016.

Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S. Lin, Tianhe Yu, and Alexei A. Efros. Real-time user-guided image colorization with learned deep priors, 2017. URL `https://arxiv.org/abs/1705.02999`.

6 APPENDIX A: GANTT CHART