APS360 PROJECT FINAL REPORT: PICTURE COLOURIZATION

Claire He
Student# 1010255698
claire.he@mail.utoronto.ca

Huda Imran
Student# 1009910606
huda.imran@mail.utoronto.ca

Manha Siddiqua
Student# 1010315375
manha.siddiqua@mail.utoronto.ca

Jihoon You
Student# 1009796801
jihoon.you@mail.utoronto.ca

ABSTRACT

Our project proposes to develop a deep learning system to automatically colourize greyscale images using CNN-based encoder-decoder architectures with generative adversarial training networks (GAN). This method will convert images to LAB colour space, using the L channel (lightness) as input and then predict the a,b channels (colour information). The system will be trained on large datasets to learn the relationship between greyscale patterns and realistic colour distributions. —-Total Pages: 9

1 INTRODUCTION

Colour reigns king when it comes to visual imagery. Whether it be for aesthetics and artistic expression or accurate depictions of historic or nostalgic scenes, the importance of the presence of colour cannot be understated. Historically, digitally colourizing grayscale images required significant manual effort using tools such as Photoshop; however, through the use of machine learning and deep neural networks, we aim to automate this laborious and time-consuming process.

The goal of this project is to design and train a machine learning model that infers reasonable pixel colour values for an image, given its greyscale intensity (lightness), as well as by considering additional geographical and spatial context. Deep learning is an optimal tool for this task, as this kind of inference requires a careful consideration of many independent and overlapping factors. This includes identifying and learning spatial features and patterns, where neural networks have been shown to be very effective.

The importance and implications of this project (and AI image colourization technology as a whole) reach beyond just being able to colour in nostalgia-inducing photographs; between arts, history, forensics, medicine, and more, image colourization can prove to be an important tool for a great variety of fields.

2 ILLUSTRATION

With the dataset found on Kaggle (Sharma), we decided to use a Generative Adversarial Network (GAN) as our primary architecture, with the U-Net architecture incorporated in the generator aspect of the GAN. A GAN is a model where two neural networks (the generator and the discriminator) work in opposition, with the generator creating synthetic data while the discriminator evaluates whether the data is real or fake Varughese (2025). This allows the model to learn patterns from existing training datasets for generating a colour spectrum closely resembling a,b colour spectrum, and allows the discriminator and generator of the GAN to work together in training to create the

model and vibrant outputs. Figure 1 indicates an illustration of the project's structure and process for further explanation.

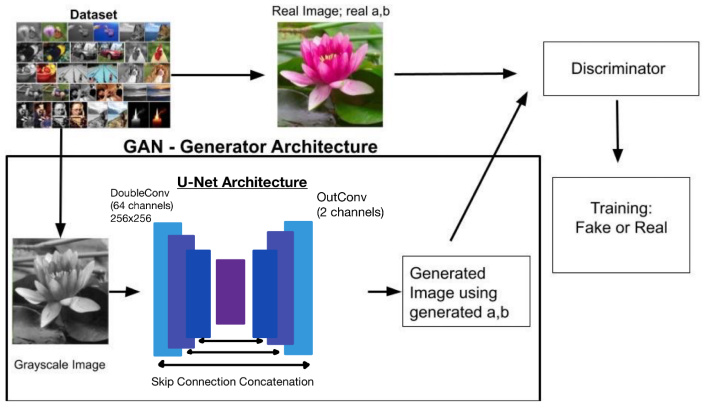

Figure 1: Our overall project structure, which consists of the path from greyscale input to colourized output through our deep learning model

3 Background and Related Work

Image colourization is fundamentally a type of image classification and regression problem, where the goal is to predict possible colour values for each pixel from a black and white image. For some background knowledge, convolutional neural netowrks (CNNs) are usually the standard approach for this task because they are ideal in supervised image classification (Richards, 2022). Expanding on that point, CNNs process images through layers that use different filters (or building blocks, usually called kernels) to analyze the image and automatically detect and learn spatial hierarchies of features. These kernels can be optimized during training to "minimize the difference between outputs and ground truth labels through an optimization algorithm called backpropagation" (Yamashita et al., 2018). The team has also found that the use of Generative Adversarial Networks (GANs) is useful as colorization is a type of image translation problem, and the generator and discriminator are both CNNs.

A significant amount of research has been conducted in the field of image colourization, with various approaches, several contributions include:

1. User-guided colourization methods. Early image colourization models often relied on user interaction to colour an image. Levin et al. (2004) demonstrated how this method allows the user to colourize the image by annotating the greyscale images using coloured strokes or scribbles. The colours are then propagated throughout the image using optimization techniques.

2. Goodfellow et al. (2014) introduced a new way to train generative models, Generative Adversarial Networks (GAN), where two neural networks, a generator and a discriminator, work to train and create realistic data. For example, Pix2Pix was a model developed by Isola et al. (2017) which used GANs for image colourization, from greyscale inputs, where the generator predicts a possible LAB colour scheme, and the discriminator checks for realism, to create the best output. This idea helps us work towards our project and the goal of image colourization, with accurate and vibrant colours.

3. In 2015, Ronneberger et al. (2015) developed a CNN for more precise image segmentation called U-Net, which utilizes an encoder-decoder structure. The encoding layers compress the black-and-white image to capture important features, and the decoding layers reconstruct the image back to its original size to restore and add colour (Nazeri et al., 2018).

U-Net uniquely incorporates skip connections, which help the network retain important details like patterns, making the colourized output more accurate to the original (Nazeri et al., 2018).

4. Another approach is through the use of Stacked Convolutional Auto-Encoders (CAEs), which stacks several Convolutional Auto-Encoders with each layer receiving its input from the layer below. Each CAE learns a different feature of the original input. A CNN can be initialized by a trained CAE stack to have improved performance of the model compared to a standard CNN (Masci et al., 2011).

5. On the other hand, other previous works explored developing fully automatic colourization models that require no human input. For example, Gupta et al. (2012) proposed an example-based method where the user provides a reference colour image similar to the target greyscale image. The model then transfers colours from the reference image to the greyscale image using superpixel-based feature matching. This automatically identifies corresponding regions between greyscale and colour images.

4 Data Processing

The advantage of a project such as image colourization is the abundance of data sources, as any image can be used as data.

We initially planned to use a 50,000 image subset of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset. This was because it had been previously used in some of the most well-known automatic image colourization models, which we thought would provide an interesting point of comparison for our model's performance.

However, due to storage and GPU usage limitations we encountered on Google Colab, we opted to use a smaller dataset of 5739 images found on Kaggle (Sharma).

These images were downloaded and pooled into a single folder, where they were subsequently resized to 256x256 px using the Pillow (Python Imaging) library. The resized images were then converted into the CIELAB colour space using the scikit-image library, where the L value was normalized to [0, 1] and the ab values were normalized to [-127, 128]. Once converted into the LAB colour space, the images could not be saved as typical digital formats, such as .jpg or .png, and had to be stored as .npz (NumPy Array) files. In the final step for data processing, the images were rearranged into train-validate-test subsets using an 60-20-20 split.

Figure 2: Flowchart detailing data processing steps, using examples from our dataset

For final testing of our model, we introduced the testing split of our initial dataset to the model. This 20-percent split of our dataset had never been inputted to the model during training or validation; thus, we would be able to accurately examine the model's generalizability to "never before seen data."

5 ARCHITECTURE

After experimenting with various CNN architectures, our final model implements a conditional Generative Adversarial Network (GAN) specifically designed for image colorization. The Generator is a U-Net with identical structure to our previous CNN model in the progress report. It takes grayscale L-channel images (1×256×256) as input and outputs AB color channels (2×256×256). The encoder follows the same downsampling path (64→128→256→512→1024 channels) using double convolution blocks with batch normalization and ReLU activation, while the decoder mirrors this with four upsampling blocks featuring skip connections. The output layer applies tanh activation to constrain AB values to [-1,1]. For our discriminator, we decided to use a PatchGAN architecture, shown in Figure 3. We thought this would be a better choice because this type of discriminator architecture makes classification decisions on overlapping 70x70 patches of an image rather than classifying the entire image as a single unit. This way, it can deduce which colours might appear more natural in specific regions (like the sky, grass, water, ect.) independently, and then decide whether those boxes are "real" or "fake", instead of the entire photo as a whole. It consists of four convolutional layers with progressively increasing channels (64→128→256→512) using 4×4 kernels, LeakyReLU activation, and batch normalization. Our final architecture enables the model to produce more vibrant and more contextually accurate colourizations compared to the previous regression-based U-Net approach.

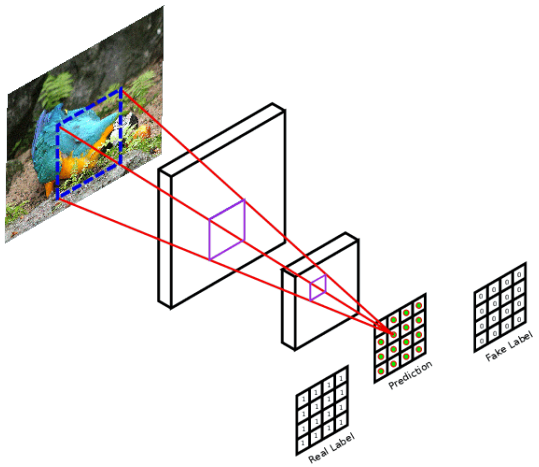

Figure 3: Diagram of a PatchGAN (Demir & Unal, 2018)

6 BASELINE MODEL

The baseline model incorporates a simple CNN-based encoder and decoder as our benchmark for image colorization, because CNNs are well-suited for image data. They take advantage of two key principles: locality, where nearby pixels are often related, and translation invariance, which allows the network to recognize features regardless of their position it is in Kamtziridis (2022). We included an encoder and decoder architecture because autoencoders are a generative learning model that can efficiently learn compressed representations of data (encodings) and then reconstruct the original input Kamoutsis (2020). These properties make CNNs ideal for our task of image colourization. For our model, we will take a greyscale L channel (from the LAB colour space) as the input and predict the missing a and b colour channels, giving us the output. As shown in Figure , the encoder and decoder uses three symmetrical conventional layers to extract features from the greyscale image. The decoder then uses upsampling to reconstruct the colored data. Specifically, the encoder applies three convolutional layers with ReLU activations and max-pooling to progressively extract features from the greyscale image while reducing its dimen- sions. The three encoder layers expand the channel depth (1 → 64 → 128 → 256) and downsamples the input (256×256 to 112×112 to 56×56 to 28×28). This compressed representation captures com- plex features such as edges and textures. The decoder then upsamples these features using three transposed convolutional layers: the layers upsamples (28×28 to 56×56 to 112×112 to 256×256) and reduces channels (256 → 128 → 64 →

2) to output 2 ab channels at 224×224 resolution. A final tanh activation maps the output to the normalized [-1,1] range of Lab colour space.

Due to there being too many images in the Kaggle Sharma, we decided to run the CNN on a smaller dataset of 1000 images with their colored and greyscale pairings. The data set was divided into 60-20-20 splits for training, validation, and testing. Accuracy was measured using a deltaE-based threshold that computes how close the predicted ab channels were to the ground truth values, offering a perceptually meaningful measure of color prediction accuracy.

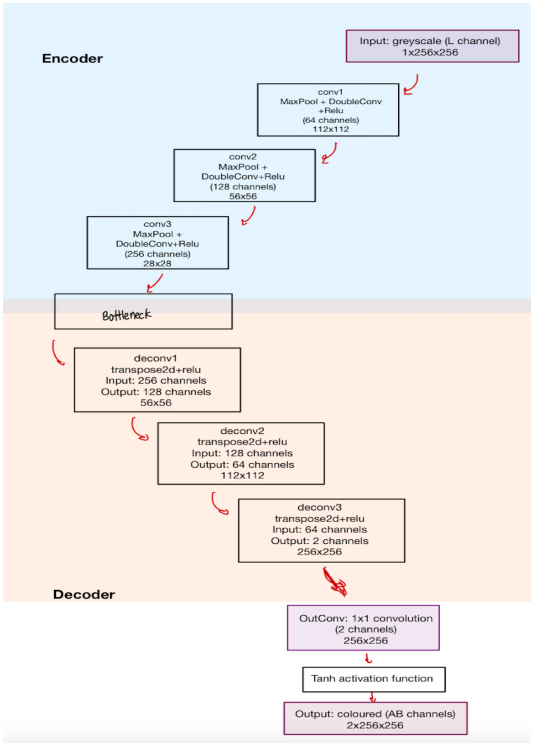

Figure 4: A visual representation of a CNN encoder-decoder for image decomposition

7 Quantitative Results

Due to the nature of our neural network, where we are inferring and comparing specific pixel colour values, it is difficult to obtain a reasonable quantitative metric to evaluate our model's performance.

Our team opted to evaluate our model using MAE (mean absolute error) instead of MSE (mean squared error). This is due to the fact that MSE tends to "overcompensate" for variances and outliers, resulting in the model learning to generate "average" results, lacking in realism.

As observed in the graphs in Figure 5, while the generator and discriminator losses do not converge, they follow a downwards trend towards each other. This suggests that, despite the fact that the losses do not converge, the model seems to be learning. In addition, we can observe the adversarial relationship of the discriminator and generator in the right graph, as the troughs and peaks of the discriminator and generator losses coincide, and vice-versa.

After 150 epochs, we obtained a final MAE loss of 0.0431. This seems to be quite a small value, which implies that the output images should be nearly identical to the ground-truth. However, as slight variances in pixel colour values could result in a significantly different colour visually, we interpret these results to suggest that our model generally provides output images similar to the ground-truth image, while not perfect.

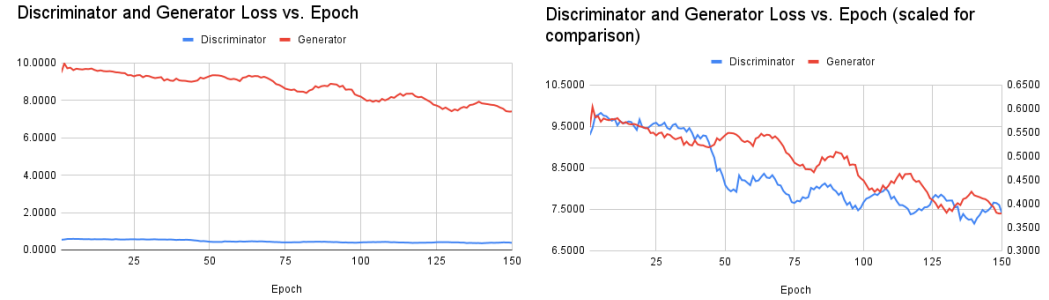

Figure 5: The generator and discriminator training losses. The right graph has scaled vertical axes for trend comparison between the generator and discriminator losses.

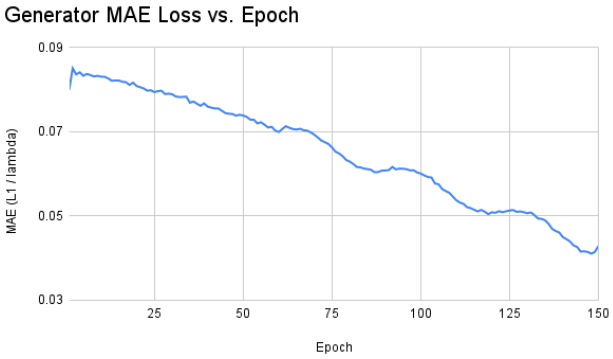

Figure 6: The mean absolute error loss from training.

8 Qualitative Results

Using the CNN and then the GAN architecture with the U-Net architecture as the generator, we were able to have many different qualitative data that were progressively improving with the new features we were including in our baseline and primary model. With the baseline model where we used a CNN, we saw outputs as seen in Figure 4. The model was able to distinguish between different regions of the image and apply color in a spatially coherent manner. For example, it correctly identified the body of the giraffes compared to the sky and grass, applying some color variations. However, the model struggled to create vibrant colors, particularly in the blue and green regions. This tells us that although the model learned the general placement and shading of colors, it had difficulty predicting saturation. However, the result indicates that we are on the right track and that the baseline model can generate meaningful colorization, even if it lacks vividness and fine detail.

Our primary model performed adequately using the U-Net architecture, as seen in Figure 5, however, the right example also shows the model's difficulty with more artificial, man-created environments, particularly the tennis court scene where it failed to restore the blue court's surface. This suggests that our model performs better on natural scenes, likely because these types of images were more prevalent in the training dataset, while it struggles with scenes containing artificial materials that require more precise color prediction.

In our current primary model where we use a GAN to improve our model, we can see that not only does the model clearly learn where the colors are specified in different regions, it is also able to create vibrant colors for man-created environments that we could not get previously. The images are able to distinguish different complex features such as shadows and different levels of saturation. Through these qualitative results, we have noticed that with more number of epochs the visual outputs of our images are the best, with our best being 150 epochs, as seen in Figure 6. As seen in the colorized

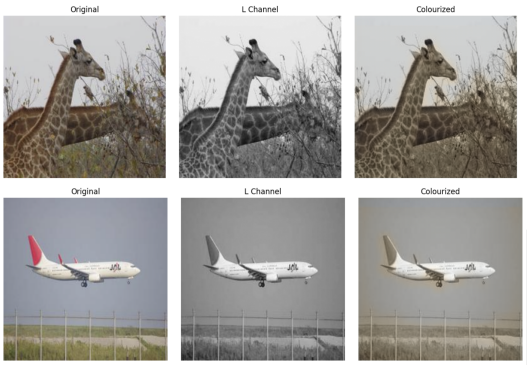

Figure 7: Output of Baseline Model run on 30 epochs

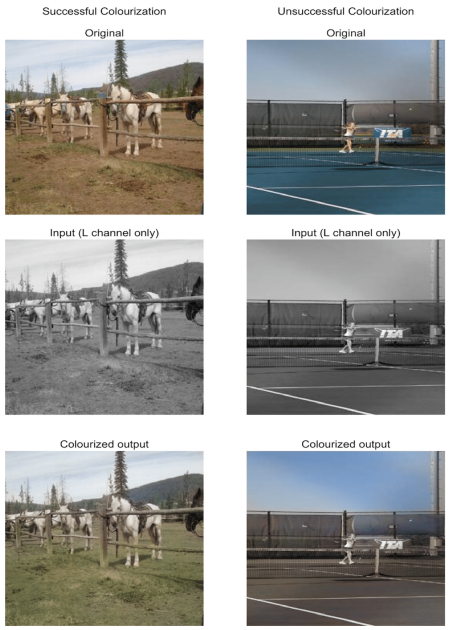

Figure 8: Qualitative comparison of successful and unsuccessful colourizations

image we can see the color of the baseball bat and the matching uniform resemble closely to the ground truth label.

To compare the qualitative results we had from the beginning to now, we can see that while our baseline model can distinguish between different color gradients, it can not visualize those colors the best unless we add a GAN, with the U-Net architecture as the generator. The output of the GAN model allows us to see much more vivid and a wide variety of images included in the testing data, meaning our training has worked on a variety of images.

9 Discussion

A specific trend we noticed throughout the quantitative data that influenced our qualitative results was the L1 loss and its steady decrease by the end of training indicating that the generator was continuously improving, getting closer to the ground truth. However, as a result the discriminator loss decreases significantly less indicating that while the generated images do become more accurate the discriminator is still able to tell the difference in some of the images created. This could be due

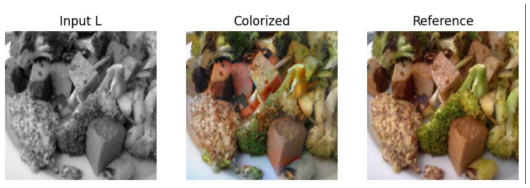

Figure 9: Final Image of Primary Model using GAN

to the fact that the images in the dataset have more colors of blue and green and brown, while vivid colors such as yellow and sometimes red do not show up as properly. Even though we use the adversarial loss to generate realistic colors and combat the discriminator, the L1 was weighted heavily, therefore allowing for more safer and averaged out colors, thus explaining some of the hues we had as outputs where red sometimes looked like brown.

One of the most unexpected patterns we observed across all model iterations was that the outputs consistently fell within a certain colour range during early training. Across each phase of the project, greens, blues, greys, and browns appeared most frequently and accurately in our colourized images, while vivid colours like reds and yellows were more difficult to reproduce. As a result, images of natural scenes tended to yield the best results, likely because natural landscapes were made up with these easier-to-predict colours, such as sky, grass, and water. This pattern suggests that the model may have been learning colour distributions directly from the training data rather than truly understanding the relationships between objects and their expected colours.

One of our biggest challenges came up when we tried to improve our CNN architecture with a range of different modifications such as adding more complex layers, using attention gates, fine-tuning hyperparameters during the training process, and introducing more skip connections to better preserve spatial details. However, the outputs were very muted and desaturated, with a narrow range of colours mostly in greens and blues. This pushed us to shift our approach entirely, and implement a GAN instead. The biggest takeaway from this was that effective colourization goes beyond simply matching pixels to pixels, it requires an understanding of the whole scene, its context, and the typical colours associated with different objects, something standard CNNs have a hard time reproducing. The strong results from our GAN model showed that adding an adversarial loss, rather than making the network deeper or more complex, was a pivotal decision.

In conclusion, our model has many strengths, being able to generate images closely resembling the original image, with few anomalies due to the limited variety in the training dataset. The training was done with a positive outcome, the losses decrease with each epoch, following expected trends. We were able to develop a model from the CNN to the GAN with U-Net generator leading to better outcomes. Some of the weaknesses we can continue to work on, is creating an adversarial loss that is also heavily weighted, as well as improving the generator loss and improving the discriminator, such as by using more advanced or multi-scale designs, might give the generator more precise feedback. This could be done by improving the adversarial loss, and also expanding the dataset with more variety in images and training for longer could help lead to results that are more realistic and less biased.

10 Ethical Considerations

Many ethical concerns need to be considered for image colourization. One major issue is the probability of false or incorrect colourization of historical photos, resulting in the distortion of historical events and the perpetuation of discrimination. If the training data we feed into the neural network contains biases, the model might reproduce those biases, creating discriminatory or insensitive outcomes (Carleton University, 2024). For example, if our model was trained on a specific dataset that may not represent global diversity in skin tones, cultural contexts, or geographic regions, this could potentially lead to biased colorization of people from underrepresented groups. There is also the ethical issue of the corruption of artists' creative vision, as many artists may not want their images to be turned to colour, as it may ruin their initial intent with the original form (Liu et al., 2022).

11 Evaluating Model on New Data

During the data processing stage, we reserved 20 percent of the overall data accumulated for just testing in a 60-20-20 split. This was done to ensure our results accurately represent the model's performance on unseen data. These samples were never used in training or for tuning hyperparameters, ensuring an unbiased evaluation.

Before feeding images into the model, all samples were converted from the RGB to the LAB colour space. Only the L channel (lightness) was provided as model input, while the model's task was to predict the a and b channels (colour information).

For the final inference step with our test data, we loaded grayscale L-channel images from the test dataset and passed them through the trained generator to predict the a and b channels. The predicted channels were then converted back to real LAB values, combined with the original L channel, and converted to RGB for visualization. This allowed direct comparison with the reference RGB images. This process is outlined in Figure 7

Figure 10: Model predictions on the test data

As seen in Figure 8, the test results showed that most of the colourized outputs closely matched the original reference images, with minor saturation errors. The tones of each of the images were well matched and were able to be picked up very well. As the model was trained with more epochs, the results improved. This indicates the model generalized well to new, unseen data, meeting our expectations for the problem being solved.

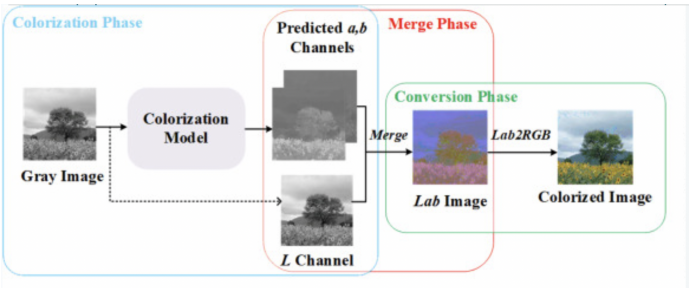

Figure 11: Inference and visualization process of test data

12 Link to Repository

Colab Notebook Link:

Click here to view Colab page

REFERENCES

Carleton University. Reflection on the ethics of colorizing with ai. https://hh2024w.amason.sites.carleton.edu/week-4-spatial-humanities/reflection-on-the-ethics-of-colorizing-with-ai/, 2024. Accessed: 2025-06-13.

Uğur Demir and Gozde Unal. Patch-based image inpainting with generative adversarial networks. 03 2018. doi: 10.48550/arXiv.1803.07422.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2672–2680, 2014. URL https://papers.nips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf.

Raj Kumar Gupta, Alex Yong-Sang Chia, Deepu Rajan, Ee Sin Ng, and Huang Zhiyong. Image colorization using similar images. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pp. 369–378, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450310895. doi: 10.1145/2393347.2393402. URL https://doi.org/10.1145/2393347.2393402.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5975, 2017.

George Kamoutsis. Building an image colorization neural network - part 1: Generative models and autoencoders. https://medium.com/@geokam/building-an-image-colorization-neural-network-part-1-generative-models-and-autoenc 2020. Accessed: 2025-07-11.

George Kamtziridis. Building an image colorization neural network — part 3: Convolutional neural networks. Medium, 2022. URL https://medium.com/@geokam/building-an-image-colorization-neural-network-part-3-convolutional-neural-networks

Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. *ACM Trans. Graph.*, 23(3):689–694, August 2004. ISSN 0730-0301. doi: 10.1145/1015706.1015780. URL https://doi.org/10.1145/1015706.1015780.

Siyang Liu, Arjun Taluja, and David Huang. Image colorization using generative adversarial networks and transformers. Technical Report CS231n Final Report, Stanford University, 2022. URL https://cs231n.stanford.edu/reports/2022/pdfs/109.pdf. Accessed: 2025-06-13.

Jonathan Masci, Ueli Meier, Dan Cireşan, and Jurgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Artificial Neural Networks and Machine Learning – ICANN 2011*, Lecture Notes in Computer Science. Springer, 2011. URL https://people.idsia.ch/~ciresan/data/icann2011.pdf.

Kamyar Nazeri, Eric Ng, and Mehran Ebrahimi. Image colorization using generative adversarial networks. In *International Conference on Articulated Motion and Deformable Objects (AMDO)*, pp. 85–94. Springer, 2018. URL https://www.imaginglab.ca/data/NaNgEb2018AMDO.pdf.

John A. Richards. *Supervised Classification Techniques*, pp. 263–367. Springer International Publishing, Cham, 2022. ISBN 978-3-030-82327-6. doi: 10.1007/978-3-030-82327-6_8. URL https://doi.org/10.1007/978-3-030-82327-6_8.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015*, 2015. doi: 10.48550/arXiv.1505.04597.

Aayush Sharma. Image colorization dataset. URL https://www.kaggle.com/datasets/aayush9753/image-colorization-dataset.

Jobit Varughese. Generative adversarial networks (gans). https://www.ibm.com/think/topics/generative-adversarial-networks, 2025. IBM, Accessed: 2025-08-15.

Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9(4): 611–629, 2018. doi: 10.1007/s13244-018-0639-9. URL https://doi.org/10.1007/s13244-018-0639-9.