

Final Report - ColourU

Mark Abadir and Alice Liang

Word Count: 1997, no penalty

Introduction

Since the beginning of photography, people have always been interested in recolorizing grayscale photos. It is mostly done manually, which is very time consuming and difficult. ColourU is a neural network that seeks to automate that process, with the addition of user inputted colour hints. This is to allow the user to have some control, but also improve performance over networks without colour hints.

ColourU is a generative adversarial network that employs a U-net inspired generator and a convolutional neural net as a discriminator. A 2018 paper [1] by Nazeri et al, details how GANs can be used to solve the colorization problem. The paper demonstrated how GANs could be used to generate better images, determined qualitatively, with less of a sepia effect than a simple U-Net architecture. However, they acknowledge that their model had frequent occurrences of miscolouring, especially on higher resolution images with large amounts of textual detail. This lead us to believe that this would be an appropriate application of colour hints originally proposed in the paper by Zhang, Zhu et al from the University of California, Berkeley [2].

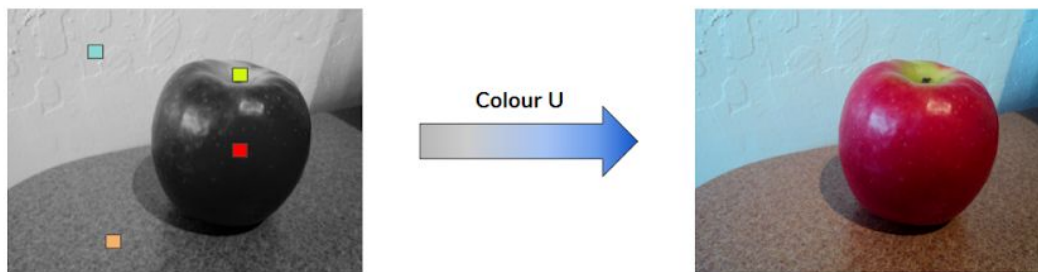


Figure 1: Illustration of ColourU's purpose

Data Processing

The dataset that the model was trained on is the tiny imagenet dataset provided by Stanford University [3]. It contains 200 different classes with 500 64x64 images each. 5 classes were utilized in training.

The images were processed as such:

- The images were converted from RGB to LAB format.
- The AB layers were set to zero to grayscale the image
- A random number, representing the number of coloured squares to add, was generated for each image, via a binomial distribution ($p=1/8$)
- The location of such squares is given by a gaussian distribution centered on the image, with standard deviation a quarter the size of the images

- The size of each sample was determined by a uniform distribution from 1 to 9, representing the length of each square.

A high-pass filter was then applied to the L layer and appended as a fourth channel. This informs the network of edges that may be present, in hopes it learns not to colour over them.

Sourcing colour hints directly from the original image was important as we were using the MSE loss between the original image and the baseline to train the baseline model. Incorrect colour samples from other methods might negatively impact the training of the baseline model. This also applies to the GAN as well due to the fact that the MSE loss was used in the GAN training as well.

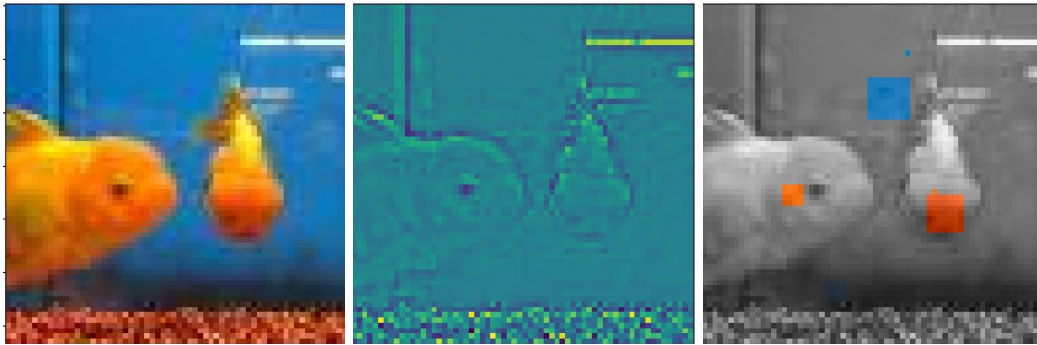


Figure 1: original image, high pass filter image, grayscale with colour samples

In inference mode, the same processing is performed, except the user manually inputs colour squares via the GUI in *figure 2*.

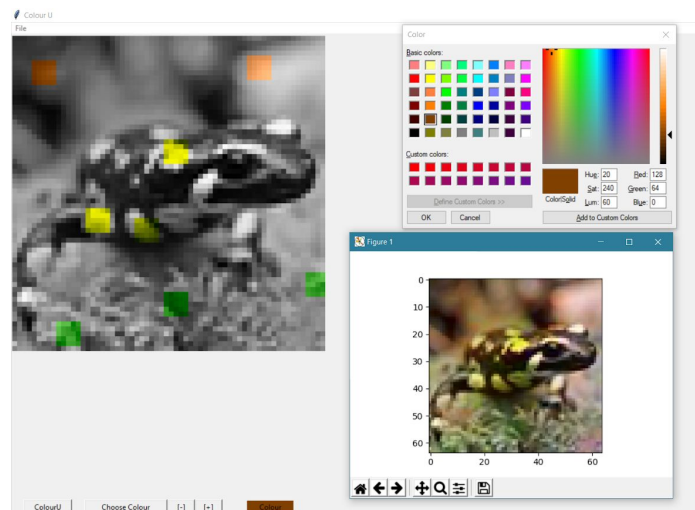


Figure 2: Colour U GUI. User can select their colour via the top right window, and by clicking the image, will place a colour sample at selected locations

Model

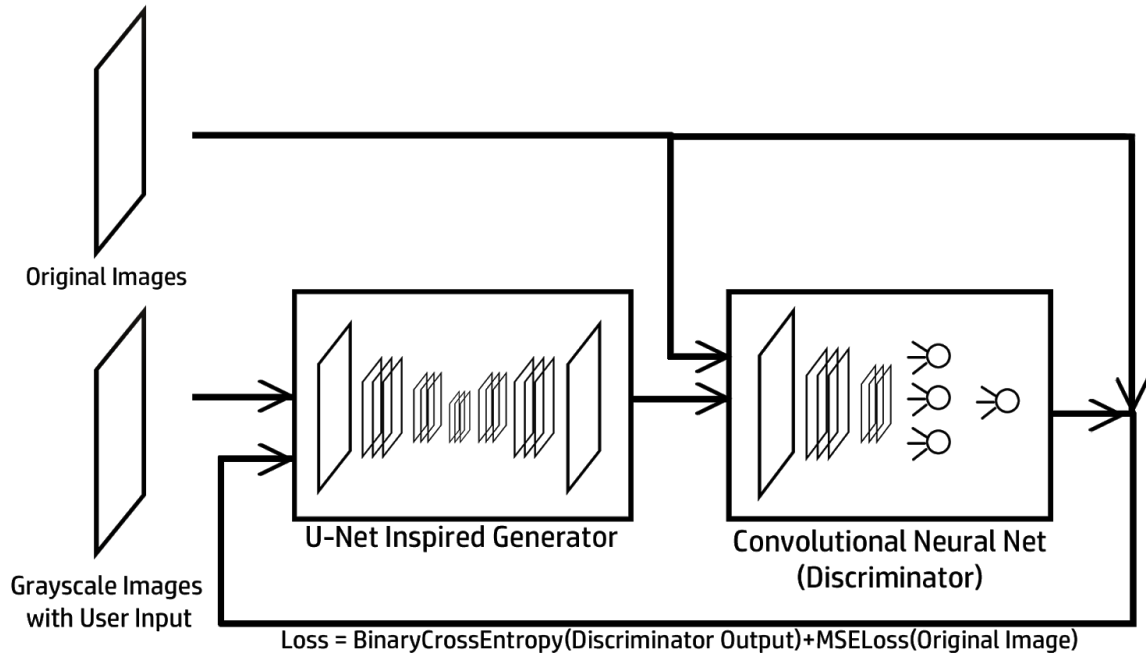


Figure 3: GAN architecture

Generative Adversarial Networks, as originally presented by Goodfellow et al in 2014, are networks consisting of two smaller networks, a generator, which generates fake images, and a discriminator which aims to determine whether an image is an original image or a fake generated by the generator (*figure 3*). ColourU employs a U-Net inspired architecture where feature images from the downsampling convolutional path are combined with corresponding feature images in the transpose convolutional path (*figure 4*). Batch normalization and leaky ReLU output functions were used in all the downsampling convolutional and transpose convolutional layer. The output layers are the AB channels which are then be appended to the lightness channel to generate the final output image.

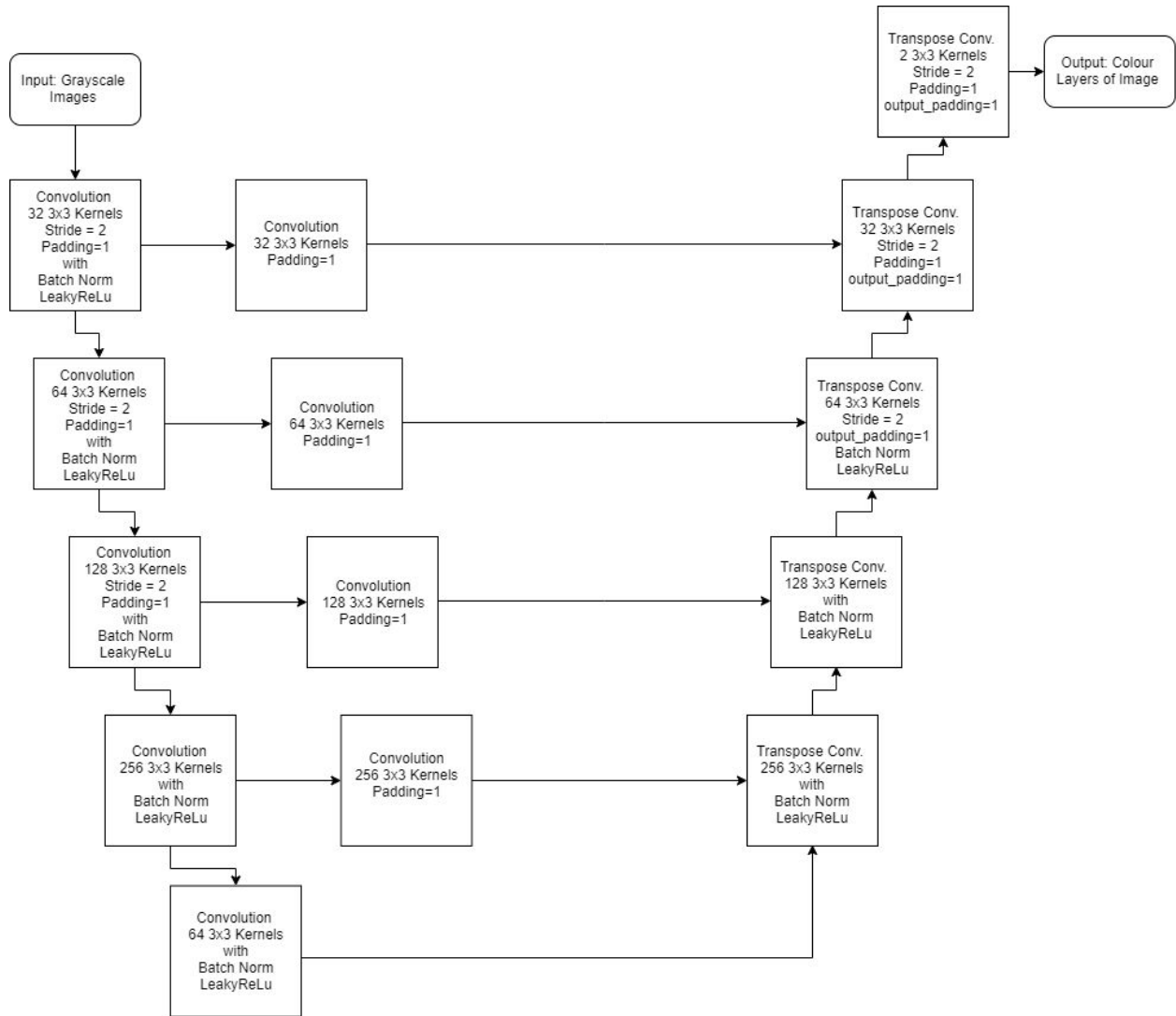


Figure 4: Generator Architecture

The discriminator is a convolutional neural net with three convolutional layers, max-pooling, and two fully connected layers (figure 5). There is batch normalization and leaky ReLU functions on the convolutional layers and leaky ReLU on the fully connected layers. The output function is a sigmoid.

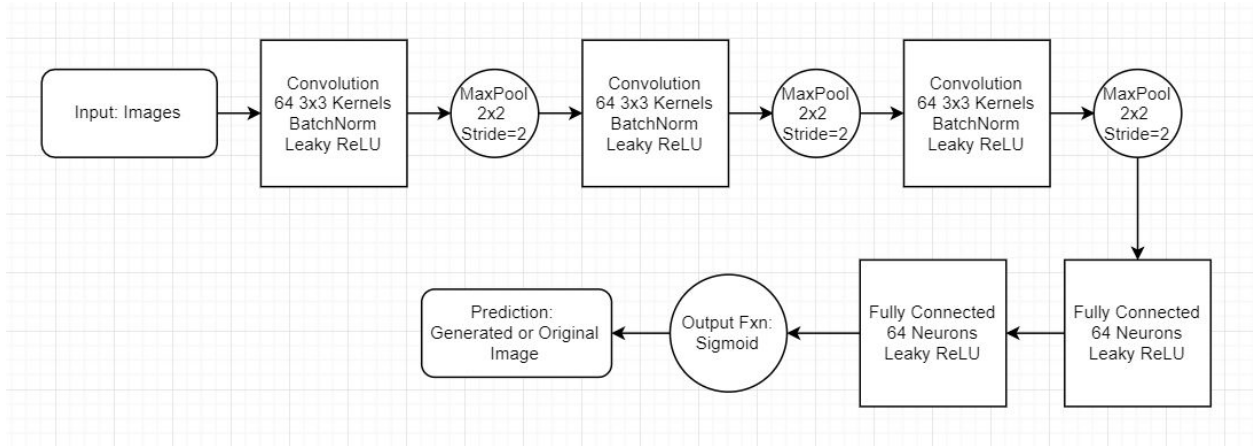


Figure 5: Discriminator Architecture

Both the generator and discriminator were pre-trained to help prevent the model from undergoing mode collapse, which is when the generator repeatedly produces the same image regardless of input. The generator was pre-trained using MSE Loss and the discriminator was pre-trained on the generator. During training, a weighted combination of binary cross entropy loss from the discriminator and MSE Loss was used for the generator. This helped stabilize the GAN during training.

Baseline

The baseline model is a fully convolutional autoencoder with three convolutional and three transpose convolutional layers, *figure 6*. There is a stride of two on both the convolutional and transpose convolutional layers, and a leaky ReLU activation functional on every layer. The output is the colour channels which is appended onto the lightness layer in the same manner as the generator.

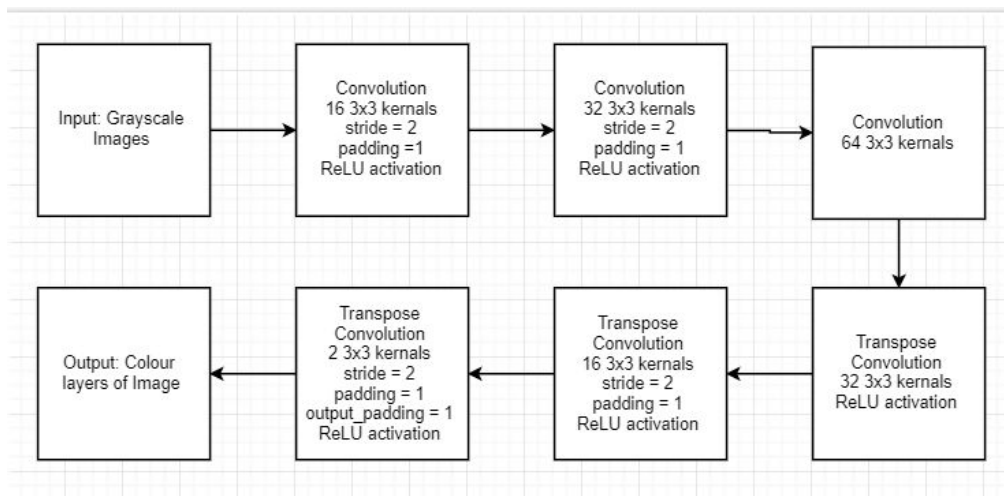


Figure 6: Baseline Autoencode Architecture

Results

Metrics

With generators and autoencoders, accuracy cannot reasonably be measured. Instead, we used normalized loss

$$NormalizedLoss = \frac{MSE(recolorization,original)}{MSE(grayscale,original)}$$

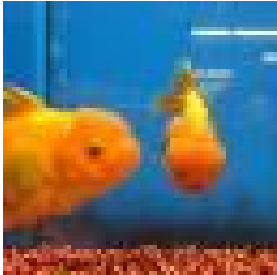
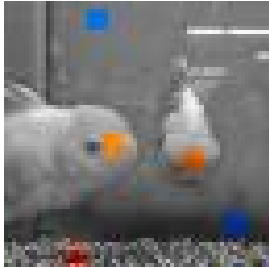
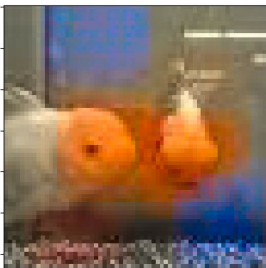

Ideally, normalized loss should be closer to 0. And if normalized loss is greater than 1, then the recolourization is “worse” than the grayscale. It’s a means of approximating how well it performs on individual images. However, in order for normalized loss to be an accurate measurement of the model’s performance, the given colour samples must similar to the colours present in the original image. Otherwise, the colour hints could guide the models towards an incorrect colouring.







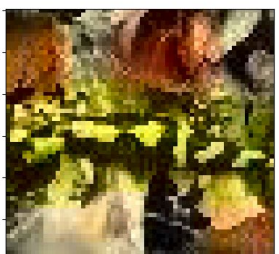




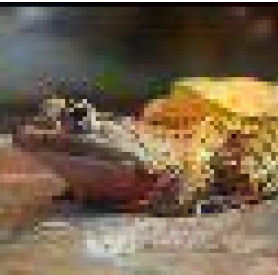

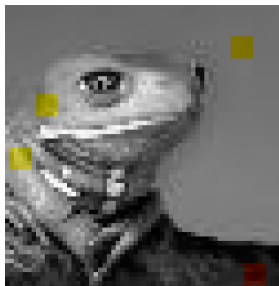
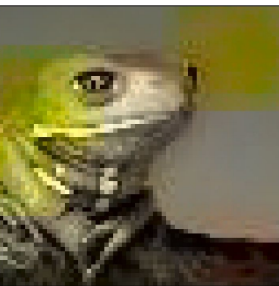

Qualitatively, we can also examine for certain features in the image

- Gray patches
- Preservation of colour samples
- Similarity to original image
- How convincing the generated image is independent of an original image

Sample Outputs

Small Images: 64x64 pixel













Original	Grayscale (Colour Hints Manually Added)	Baseline	GAN
 (a)	 5 colour hints	 Norm. Loss: 1.125	 Norm. Loss: 0.798

			
(b)	5 colour hints	Norm. Loss: 0.846	Norm. Loss: 0.564
			
(c)	8 colour hints	Norm. Loss: 2.879	Norm. Loss: 0.971
			
(d)	2 colour hints	Norm. Loss: 0.473	Norm. Loss: 0.436
			
(e)	4 colour hints	Norm. Loss: 0.394	Norm. Loss: 0.398





 <p>(f)</p>	 <p>5 colour hints</p>	 <p>Norm. Loss: 1.550</p>	 <p>Norm. Loss: 0.791</p>
 <p>(g)</p>	 <p>5 colour hints</p>	 <p>Norm. Loss: 0.618</p>	 <p>Norm. Loss: 0.765</p>
 <p>(h)</p>	 <p>0 colour hints</p>	 <p>Norm. Loss: 0.522</p>	 <p>Norm. Loss: 0.489</p>
 <p>(i)</p>	 <p>0 colour hints</p>	 <p>Norm. Loss: 0.602</p>	 <p>Norm. Loss: 0.924</p>

Note: Images (a) through (e) and (h) are test images from the tiny imagenet dataset. Other images were randomly sourced from the internet. Images (a) to (e) were selected from each of the 5 categories the model was trained on.

Large Images: 256x256 pixels

Original	Grayscale (Colour Hints Manually Added)	Baseline	GAN
 <p data-bbox="326 695 363 730">(j)</p>	 <p data-bbox="565 695 748 730">6 colour hints</p>	 <p data-bbox="841 695 1089 730">Norm. Loss: 0.844</p>	 <p data-bbox="1149 695 1398 730">Norm. Loss: 1.068</p>
 <p data-bbox="326 1058 363 1094">(k)</p>	 <p data-bbox="565 1058 748 1094">0 colour hints</p>	 <p data-bbox="841 1058 1089 1094">Norm. Loss: 0.687</p>	 <p data-bbox="1149 1058 1398 1094">Norm. Loss: 1.161</p>
 <p data-bbox="326 1430 363 1465">(l)</p>	 <p data-bbox="565 1430 748 1465">0 colour hints</p>	 <p data-bbox="841 1430 1089 1465">Norm. Loss: 0.520</p>	 <p data-bbox="1149 1430 1398 1465">Norm. Loss: 0.651</p>

Very Large Images: 512x512 pixels

Original	Grayscale (Colour Hints Manually Added)	Baseline	GAN
 <p data-bbox="321 695 370 730">(m)</p>	 <p data-bbox="548 688 766 724">8 colour samples</p>	 <p data-bbox="847 695 1091 730">Norm. Loss: 0.759</p>	 <p data-bbox="1156 695 1399 730">Norm. Loss: 1.220</p>

Discussion

Interpretation

In terms of the defined normalized loss, the models perform similarly with the baseline performing better with some images and the GAN performing better with others. While the qualitative evaluation of an image is usually in agreement with the quantitative measurements, that is not always the case. Image (j) in particular is an example where the qualitative and quantitative metrics do not align. In this case the baseline model has a lower normalized loss yet based on the qualitative metrics the GAN performs better. While quantitative analysis is important, the end objective of the model is to generate realistic looking coloured images. Therefore, qualitative analysis will take greater importance than quantitative.

The images where the GAN performed worst were images with flat solid colours including (a), (e), (g), and (i). In these images, the GAN introduced new colours and textures into areas that were originally solid colours. Images with better GAN performance were images with texture and lightness variation, such as (b), (d), and (h).

The GAN is able to avoid some of the key pitfalls of the baseline model. The baseline model is very dependent on colour hints. This can be seen in examples (b), (f), (g), (j), and (m). There the baseline model only adds large rectangles of colour surrounding the colour hints on top of a slight hue. Additionally, in (h) and (l), when there are no colour hints, the baseline only adds a slight “sepia effect”, while the GAN generates colours.

The GAN is able to accomplish some edge detection, possibly as a result of the high pass filter layer. In example (c), the colour samples were placed on the salamander spots and don’t bleed into the surrounding area much, as opposed to the baseline, where the surrounding area is yellow.

It can also be seen in example (k), an unsampled image of a fish. In that instance, the GAN does fail at recognizing that the fish should be orange, and instead colours the surrounding water orange; however, what's notable is that the colouring does not pass the edge between the water and the fish, and recognizes it as a separate object.

One flaw in the models is that the colours that were generated tended to be a little less vibrant than the ones in the original image, as seen in (a) primarily. Both models tended to lighten the intensity of the colour samples. The GAN in particular doesn't always preserve the colour of the colour hints very well. This can be seen in the red sample at the bottom of (a) being subdued and spread out in the bottom of the GAN image.

Overall quality

While the GAN may have had some good results, it may not necessarily be a good quality recolorization algorithm. The scope of the network was limited to only 5 classes of images and it was only trained on images of a very small size. The GAN did demonstrate the ability to generalize to images of new objects as well as larger, higher resolution images. However, the generated images still falls short of the quality needed for this model to have practical applications. As seen with image (m), the model fails to generate a high quality recolouring with image resolutions higher than 256x256 pixels, which limits its use as most images have resolution in the high hundreds or even thousands. However, the model itself is promising as it is possible that more training on higher resolution images could improve its performance.

The application of a GAN in this problem is justified, especially when compared to the baseline model. The GAN is able to produce better quality colourings overall, and avoids some of the major flaws of the baseline model.

Future Projects

One key change that could be made in future iterations is to train on higher resolution images. The larger dimensions would allow for more layers in the model without the feature images getting to small. Another change would be to augment the data with instances meant to train the GAN to generate solid colours in areas where the lightness is consistent.

Ethical Framework

The main ethical issue with image colourization is the potential for biases to be trained into the network. The principal stakeholders in this problem is the owner of the image and the people/objects depicted in the image. These images can have large personal significance for the individual stakeholders. The colour selection process allows for greater autonomy in the colourization process. This allows for the stakeholders to colour the image in a manner that best suits them. In addition to these individual stakeholders, the general public is a stakeholder as

well. Legacy black and white photos usually have large historical significance as a piece of historical evidence. The colour hints also allow for historical research and context to be inputted and considered, which the neural net would not be able to do on its own. Having historically accurate photo evidence is very important for the public interest and as a result it offers non-maleficence for most people.

References

[1] R. Zhang et al., "Real-time user-guided image colorization with learned deep priors", ACM Trans. Graph., vol. 36, no. 4, 2017.

[2] K. Nazeri, E. Ng, and M. Ebrahimi, "Image colorization using generative adversarial networks," in Int'l Conf. on Articulated Motion and Deformable Objects, 2018.

Dataset

[3] "Tiny ImageNet Visual Recognition Challenge," Tiny ImageNet Visual Recognition Challenge. [Online]. Available: <https://tiny-imagenet.herokuapp.com/>. [Accessed: 04-Dec-2019].

Images

Goldfish: Getty/Cultura RF. Sourced from:

<https://www.theguardian.com/environment/shortcuts/2019/may/28/carping-on-how-did-a-small-pond-become-home-to-10000-goldfish>

Salamander: D. S. Wikimedia Commons. Sourced from:

<https://www.sciencemag.org/news/2017/04/deadly-salamander-disease-just-got-lot-scarier>

Rhino: Shutterstock. Sourced from:

<https://nypost.com/2019/01/31/toddler-miraculously-survives-fall-into-rhino-exhibit>

Tiger: Shutterstock. Sourced from:

<https://www.livescience.com/62863-more-tigers-pets-than-wild-worldwide.html>

We, Mark and Alice, grant permission for the instructors to post our presentation video, final report, and source code on the website.