ColorNet: Convolutional Encoder-Decoders for Image Colorization

Pranav Avva

Princeton University

COS 429: Computer Vision

Dr. Olga Russakovsky

May 9, 2023

ColorNet: Convolutional Encoder-Decoders for Image Colorization

Project Motivation

We can often predict the color of objects given only their grayscale images. Consider Figure 1 and Figure 2.

Figure 1

*A **grayscale** idyllic mountain landscape*

Figure 2

An idyllic mountain landscape

Even from the grayscale image, our experiential knowledge of the outdoors would enable us to presume that the grass in the landscape is green, the clouds are white, the sky is blue, and the snow-capped mountain peaks are white. The true colors could easily be different (in autumn, the tree leaves might instead be red, orange, yellow, or brown), yet our *past experience* with the outdoors would lead us to guess the tree leaves are green.

Machine learning models, on the other hand, have no such past experience. Instead, they rely on a meaningful dataset to provide this "past experience". In this project, we tackle the following problem: **Given a grayscale image, can a model hallucinate a plausible coloring of the image?** Although

we train our models to reproduce the ground truth, the goal of *hallucination* allows for alternate plausible colorings (for example, though the ground truth image may be red, apples in the real world can also be green or yellow). We explore this problem using a variety of convolutional neural network architectures and model learning techniques. Finally, we compare model predictions with ground truth colorings through qualitative and quantitative methods.

Previous Works

Prior attempts in image colorization tasks fall into two categories: regression and classification. Zhang et al. (2016) explores both methods. Their regression network minimizes the Mean Squared Error (MSE) loss. Because MSE loss tended to produce "grayish, desaturated results" due to the underconstrained nature of the problem, they quickly abandoned it in favor of a classification approach (pp. 4-5). They found much more favorable results with quantizing the RGB color space into 313 classes and performing multinomial classification, with class rebalancing to account for more rare colors. Zhang et al. trained both models on 1.3 million ImageNet training images (p. 7).

Iizuka et al. (2016) uses a different approach, constructing a much more complicated model that extracts low, mid, and high-level features of the image to both colorize the grayscale input and also provide object classification (pp. 110:3-4). The colorization sub-network uses MSE loss as in Zhang et al.'s regression approach (p. 110:5).

Isola et al. (2018) applies conditional generative adversarial networks (GANs) for more tasks than just grayscale to color image generation. There are a number of aspects of Isola et al.'s work that are especially relevant to our approach. First, their grayscale to color experiment also uses the ImageNet dataset, just as in Iizuka et al. and Zhang et al. Second, they experimented using both a traditional encoder-decoder and using a U-Net architecture (p. 6). Ronneberger et al. (2015) originally developed the U-Net architecture for biomedical image segmentation. The skip connections in the model, akin to ResNet (He et al., 2015), enables low-level features to "short-circuit" through the model and be preserved in deeper layers. Our approach will build upon the prior work of all of these papers.

Dataset

We use the ImageNet-1K dataset downsampled to 64x64 resolution (Chrabaszcz et al., 2017) (Russakovsky et al., 2015). This gives us access to 1 281 167 training samples and 50 000 validation samples. A sample of the training images is given in Figure 3. Additionally, training on an ImageNet dataset will allow us to compare our results to the aforementioned papers.

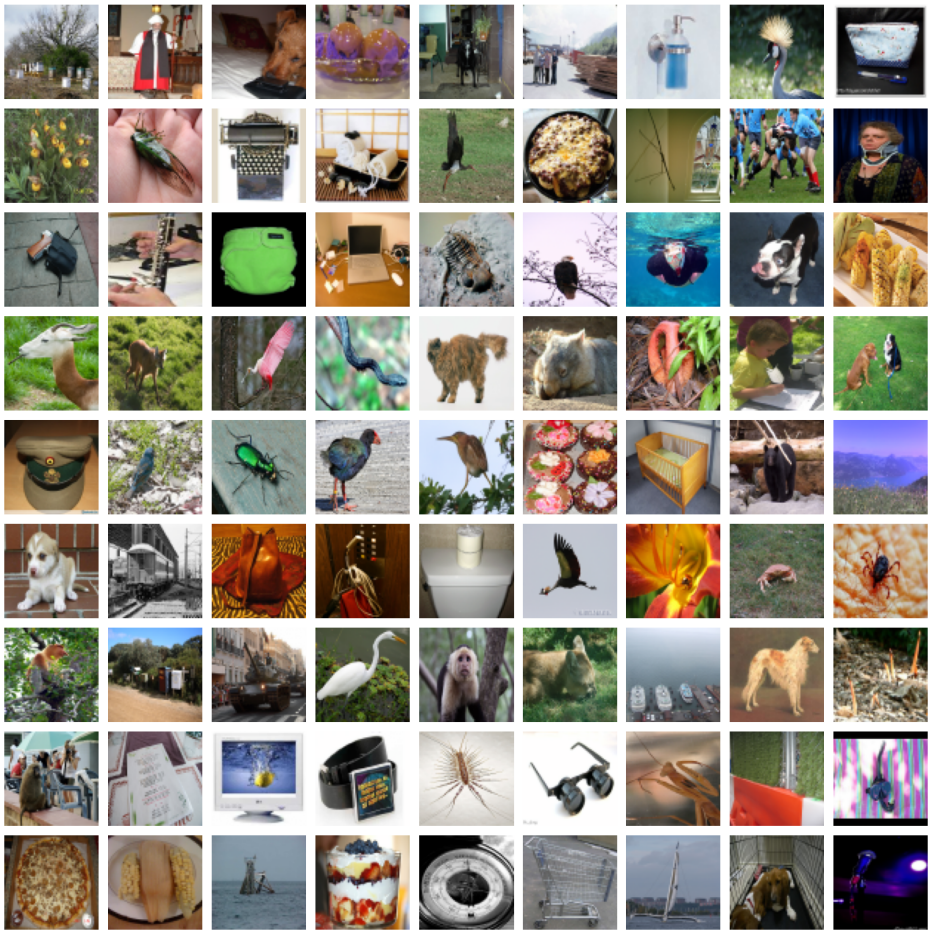


Figure 3

A subset of images from the ImageNet-1K 64x64 training set

Preprocessing

As we are not performing object classification, we can discard the object class data. We use the downsampled dataset to reduce model training time and standardize the model input shape (images in the full ImageNet dataset do not adhere to a standard shape). Additionally, we convert the images from the RGB color space to the CIELAB color space, as is done in Zhang et al. (2016). Using CIELAB provides two improvements over using RGB: (a) the L channel (which encodes the luminance of the image) is equivalent to the grayscale image, and (b) the a and b channels (which together encode the chrominance of the image) reduce the number of channels we need to predict to 2 (as opposed to 3 channels in RGB space). Thus, we only need to store the RGB color images on disk, as they can be converted to CIELAB at train/evaluation time. We use `D50` as our reference white point.

System Design and Implementation

In approaching the problem, we explore two model architectures: a simple convolutional autoencoder and a U-Net. Because we use the CIELAB space, both models will have an input shape of `(batch_size, num_channels=1, width=64, height=64)` and an output shape of `(batch_size, num_channels=2, width=64, height=64)`. Both models are trained as regression tasks, using pixel-wise Euclidean (L2) loss along the a and b channels as the loss function, given in Equation 1, where $\hat{\mathbf{Y}}$ is the model prediction, $\mathbf{Y}$ is the ground truth, and h, w are the height and width of the images. Both models additionally normalize input images in the L channel space and un-normalize output images in the ab channel space.

$$\mathrm{L}_2\left(\hat{\mathbf{Y}}, \mathbf{Y}\right) = \frac{1}{2}\sum_{h,w}\left\|\mathbf{Y}_{h,w} - \hat{\mathbf{Y}}_{h,w}\right\|_2^2 \tag{1}$$

Approach 1: Autoencoder

Our first model uses an autoencoder approach, largely inspired by Zhang et al.'s model. Key differences between our model and theirs include input shape (64x64 in ours compared to 256x256 in theirs), latent space shape (8x8 in ours compared to 32x32 in theirs), and down/up-sampling ratios (we only downsample and upsample by factors of 2, whereas they downsample by a factor of 2 and upsample by a factor of 2 or 4). Each block in our model uses two or three "same" convolutional layers with `kernel_size=3`, each followed by a ReLU layer. Each block is followed by a 2D batch normalization. The output convolutional layer uses `kernel_size=1` with Tanh activation. Downsampling is done in the spatial dimension after the first, second, and third blocks instead of using Pooling layers.

The autoencoder model uses 493 billion multiply-add operations and 31,091,010 parameters. Training was performed for 64 epochs on 1 million train images and 5,000 test images from ImageNet-1K 64x64 using Adam optimization on the MSE loss function with `learning_rate=5e-5`. Training was conducted on the Princeton Adroit HPC cluster using a NVIDIA V100 GPU. The total training time was 37 hours, 3 minutes.

Approach 2: U-Net

We improved on our initial autoencoder model by adding skip connections, drawing inspriation from Ronneberger et al. (2015). Adding skip connections enable lower-level features extracted shallower in the model to "short-circuit" through the model, preserving their effect in deeper layers. The U-Net model architecture is largely the same as the autoencoder model, with a few differences:

1. Downsampling is performed with `stride=2` in the last convolutional layer in the second, third, and fourth blocks, instead of downsampling the block output in the spatial dimension

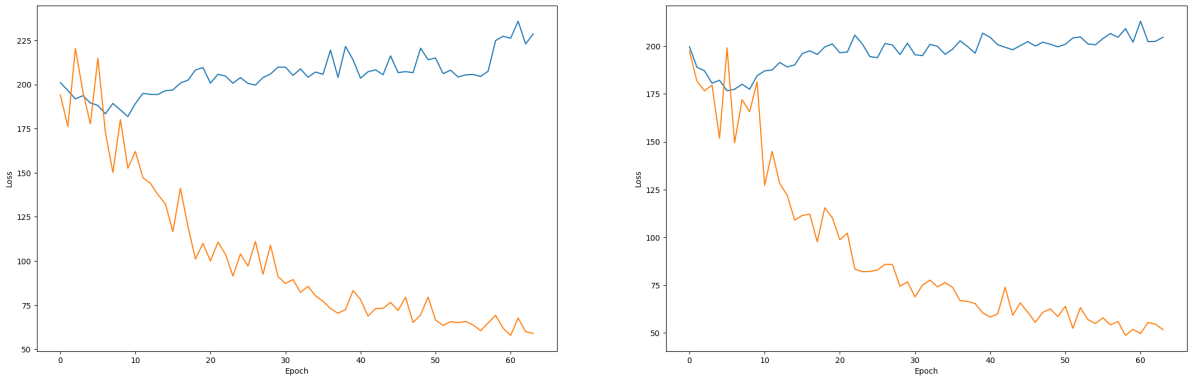
2. Three skip connections are added, short-circuiting the model between blocks 1 and 10, 2 and 9, and 3 and 8. Skip connection layers are "same" convolutional layers with `kernel_size=1`. The output of the skip layer is added to its corresponding (deeper) layer and activated with the ReLU function.

The U-Net model uses 664 billion multiply-add operations and 31,104,450 parameters. It also uses the same epoch count, optimizer, and learning rate settings as the autoencoder model. Training was also conducted on the Princeton Adroit HPC using a NVIDIA V100 GPU. The total training time was 40 hours, 46 minutes.

Results and Analysis

Quantitative Analysis

The MSE train and test loss plots for both models are shown in Figure 4. Both models appear to overfit drastically, as the drop in train loss is accompanied by a slight increase of test loss as the number of epochs increases. The training MSE loss of both models declines at a similar rate. The test MSE loss of the U-Net model is consistently 25 points lower than that of the autoencoder model, though the instability of the loss curves of both models indicates that this may not a significant conclusion.



(a) *Autoencoder* (b) *U-Net*

Figure 4

Epoch-end loss plots of both models. In both loss plots, the orange line is train loss and the blue line is test loss. The y-axis is MSE loss. The x-axis is the zero-indexed epoch count.

Qualitative Analysis

A random sampling of test images and their recolorings as performed by the autoencoder model is presented in Figure 5. This model creates plausible hallucinations for images where large areas of the image have the same color. For example, the ocean scene (5th column, 1st and 2nd rows), the sailboat in the harbor (1st column, 3rd and 4th rows), and beach scene (4th column, 3rd and 4th rows) have

hallucinations that are visually very similar to the ground truths.

However, the autoencoder model fails to create visually plausible colorings for images with rapid shifts in chrominance in local regions. For example, this is seen in the "splotches" of color that bleed beyond object boundaries, such as in the buckets (5th column, 5th and 6th rows), the bus (5th column, 7th and 8th rows), the four standing people (8th column, 1st and 2nd rows), and the bagels (8th column, 3rd and 4th rows). These splotches of color are likely due to the architecture of the model itself. Encoding the input image into a latent space likely destroys low-level features of the image, including semantic separation of objects in the image. Resolving this issue is where we believe the skip connections in the U-Net model will produce the greatest improvement.

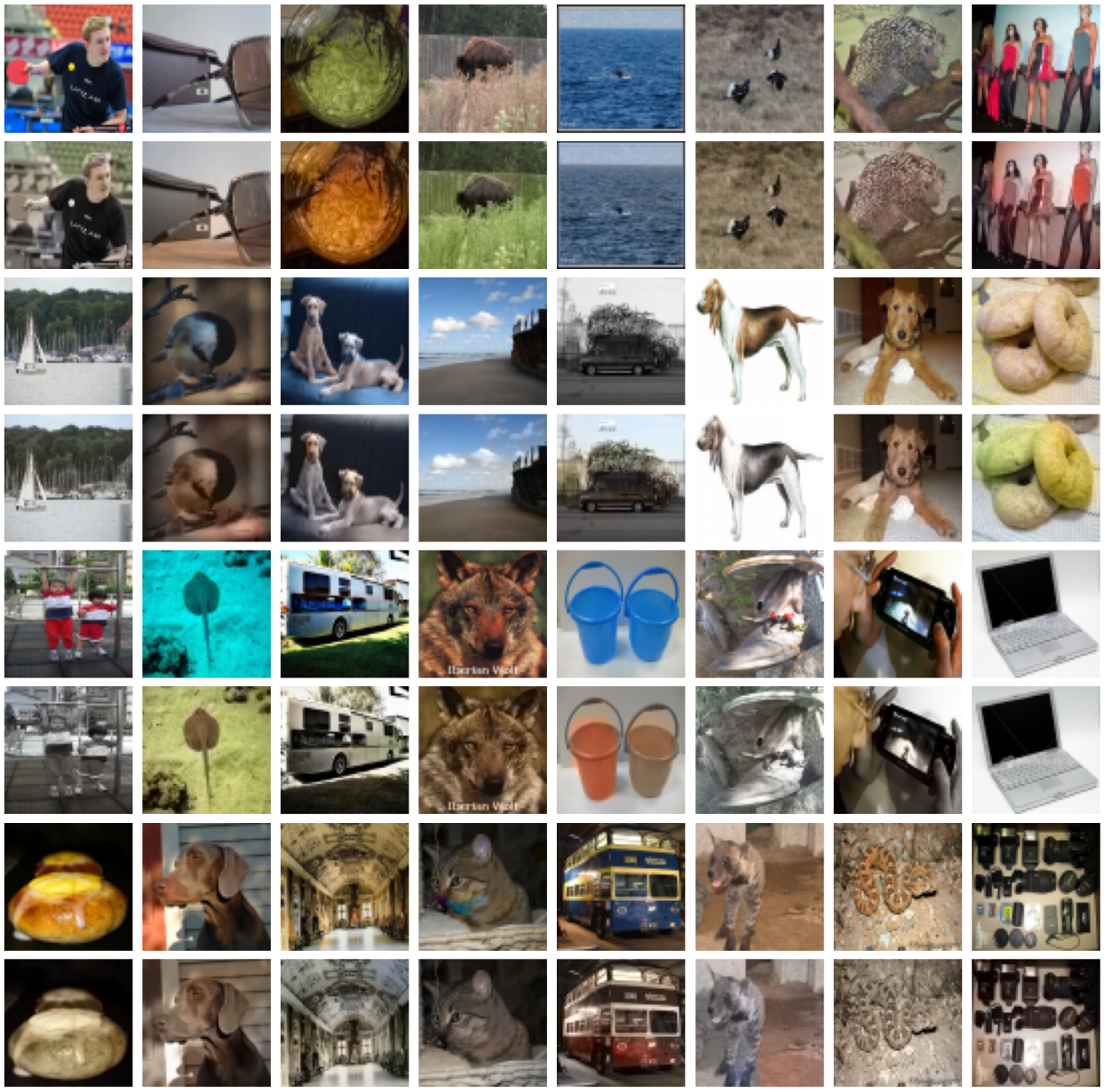


Figure 5

Random samples of test set images and their recolorings by the ***autoencoder*** *model. Recolorings appear below ground truth images.*

A random sampling of test images and their recolorings as performed by the U-Net model is presented in Figure 6. This model seems to produce much more plausible hallucinations as compared to the autoencoder. None of these random samples show color bleeding past object boundaries, indicating that adding skip connections did improve the model's understanding of semantic separation.

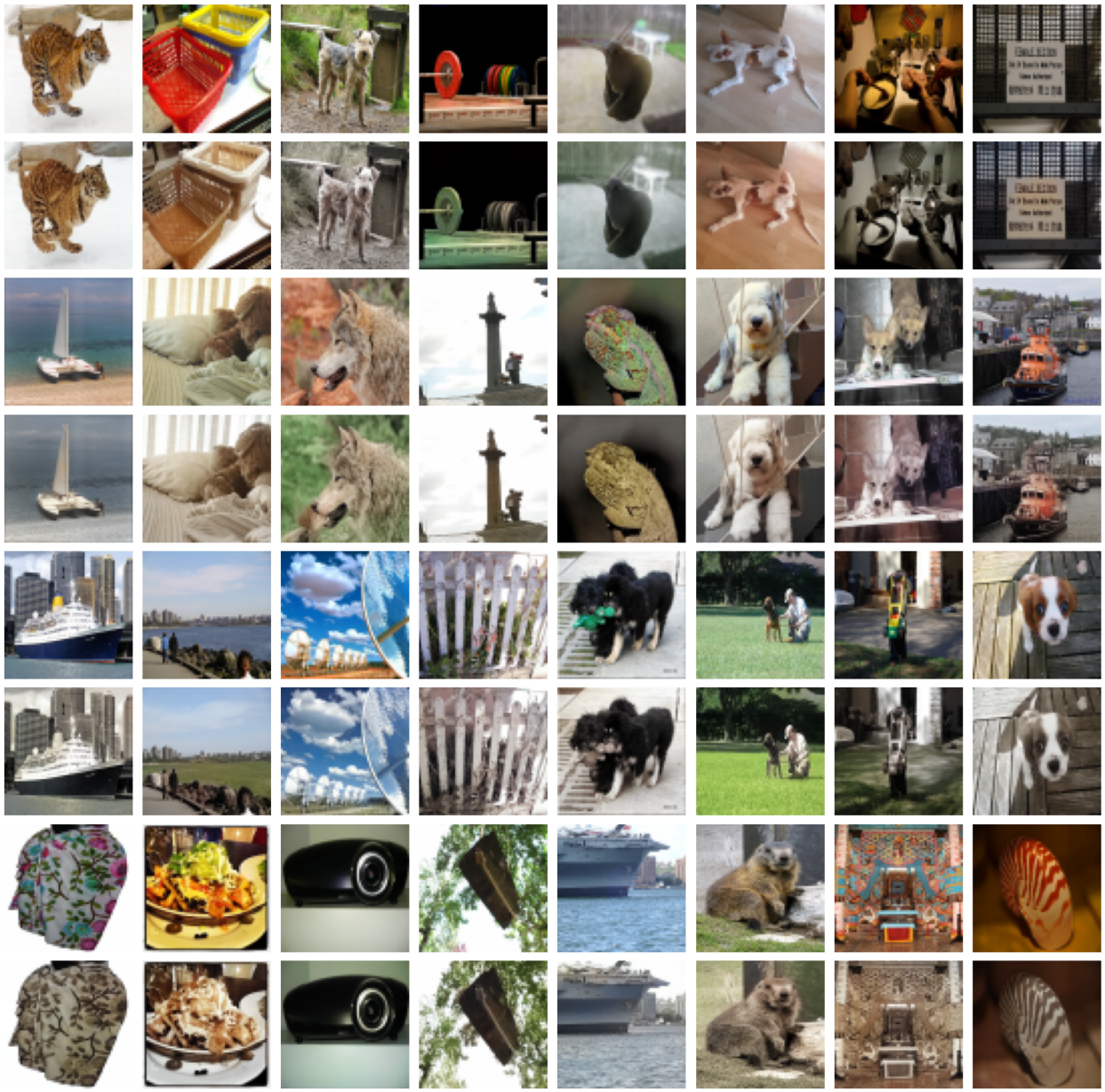


Figure 6

Random samples of test set images and their recolorings by the ***U-Net*** *model. Recolorings appear below ground truth images.*

More model hallucinations appear in Appendix C.

Conclusions and Future Steps

Image colorization is an underconstrained computer vision task. Not enough information exists in a grayscale image to correctly reproduce its ground truth coloring. However, training a model to hallucinate perceptually realistic colorings is certainly possible. This task would rely on a very large

training dataset and require a model with millions of paramters that can capture both low-level and high-level semantics in the image.

There are many avenues for futher work in this area of research. We've already shown that the U-Net model, originally developed for biomedical image segmentation (Ronneberger et al., 2015) can be effectively adapted to colorize images as a regression task. A possible improvement is combining our U-Net colorizer with an image classfier to determine if image class is correlated with hallucination realism.

The classifier concept can also be applied in a different manner. Instead of building a classifier directly into the colorizer model, the colorizer can be used as an intermediary step in a grayscale image classification pipeline. Instead of training a classifier on grayscale images, perhaps those images can first be colorized and then passed through a pre-trained color image classifier, such as an ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winner.

References

Chrabaszcz, P., Loshchilov, I., & Hutter, F. (2017, August 23). A downsampled variant of ImageNet as an alternative to the CIFAR datasets. Retrieved May 8, 2023, from http://arxiv.org/abs/1707.08819

He, K., Zhang, X., Ren, S., & Sun, J. (2015, December 10). Deep residual learning for image recognition. Retrieved May 8, 2023, from http://arxiv.org/abs/1512.03385

Iizuka, S., Simo-Serra, E., & Ishikawa, H. (2016). 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*, *35*(4), 1–11. https://doi.org/10.1145/2897824.2925974

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2018, November 26). Image-to-image translation with conditional adversarial networks. Retrieved May 8, 2023, from http://arxiv.org/abs/1611.07004

Ronneberger, O., Fischer, P., & Brox, T. (2015, May 18). U-net: Convolutional networks for biomedical image segmentation. Retrieved May 8, 2023, from http://arxiv.org/abs/1505.04597

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015, January 29). ImageNet large scale visual recognition challenge. Retrieved May 8, 2023, from http://arxiv.org/abs/1409.0575

Zhang, R., Isola, P., & Efros, A. A. (2016, October 5). Colorful image colorization. Retrieved May 8, 2023, from http://arxiv.org/abs/1603.08511

Acknowledgements

The author would like to thank Professor Olga Russakovsky, TA Maxine Perroni-Scharf, and the rest of the COS 429 instructional staff for their guidance in completing this project.

The author would also like to thank his fellow members of the Princeton Quadrangle Club for providing feedback on the realism and plausibility of the model colorization predictions.

The author is pleased to acknowledge that the work reported on in this paper was substantially performed using the Princeton Research Computing resources at Princeton University which is a consortium of groups led by the Princeton Institute for Computational Science and Engineering (PICSciE) and Office of Information Technology's Research Computing.

Appendix A

Honor Code Statement

This paper represents my own work in accordance with University regulations.

/s/ Pranav Avva

May 9, 2023

Appendix B

Codebase

All code written for this project is made open-source on GitHub: https://github.com/pranavavva/image-colorizer. Refer to the `README.md` file in the repository for information on how to run the models.

Appendix C

Selected U-Net Model Hallucinations

We showcase in Figure C1 further hallucinations from the U-Net model, as this model's hallucinations are visually more plausible than that of the autoencoder model. Some of these images may have been included in user studies, as the studies were generated uniformly at random.

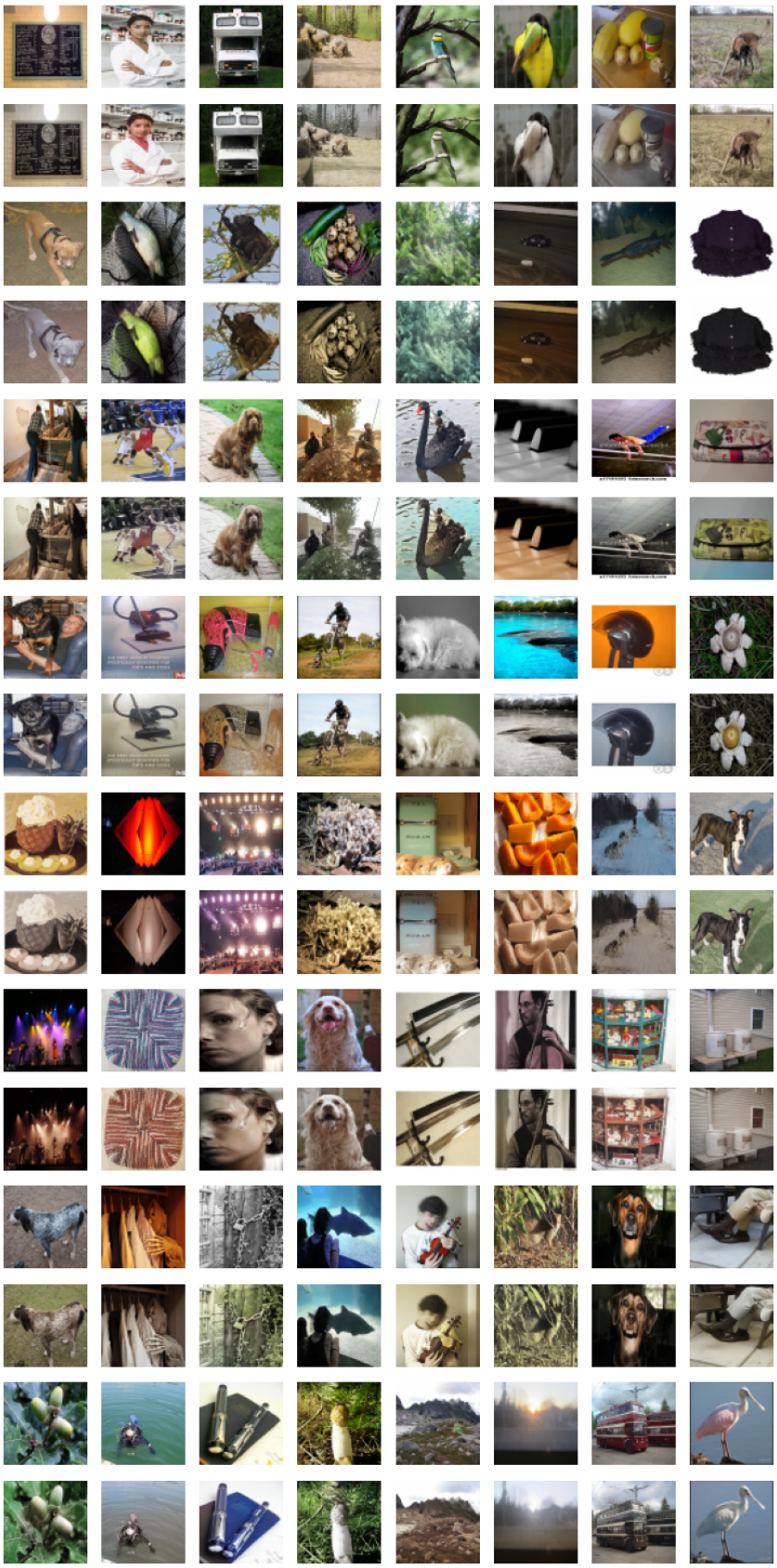


Figure C1

Further hallucinations from the U-Net model. As in Figure 6, hallucinations appear below their corresponding ground truth image.