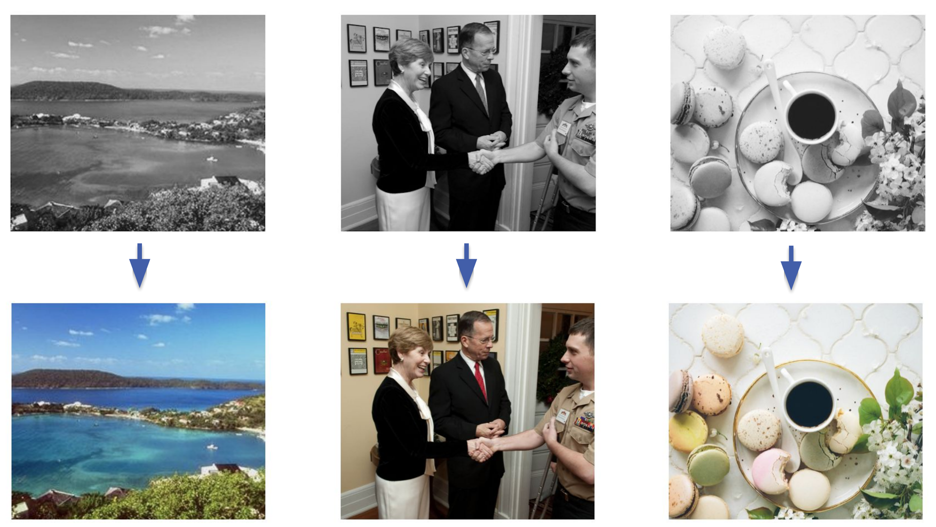
AUTOMATIC IMAGE COLORIZATION

Manwen Li, Tinghao Li, Xueying Wang

DS1008 – Deep Learning 2018 Final Project

INTRODUCTION

Automatic image colorization has been a hot topic since the 2000s. Here we take a statistical-learning approach to perform this task. We design and train a generative adversarial networks (GAN) proposed by Goodfellow in 2014. It accepts a black-and-white image as the input and generates a colorized version of the image as its output. In particular, we will use a slightly modified version of GAN. The innovative parts of this project are not only adapting a GAN based neural network model but also using VGG16 as the basis of the CNNs to extract information for the generator and discriminator.



Natural People Food

APPROACH

In GAN, the generator G is trained to produce outputs that will not be able to be distinguished from the "real" ones by an adversarially trained discriminator, D, whose job is to find the generator's "fakes" as much as it can. In our case, the G will produce color for the grey pictures as "vivid" as it can to fool the D. The initial objective function is defined as follows:

$$\min_{\theta^G} J^{(G)}(\theta_D, \theta_G) = \min_{\theta^G} \mathbb{E}_z \left[\log(1 - D(G(z)))\right],$$

$$\max_{\theta^D} J^{(D)}(\theta_D, \theta_G) = \max_{\theta^D} \left(\mathbb{E}_x \left[\log(D(x))\right] + \mathbb{E}_z \left[\log(1 - D(G(z)))\right]\right)$$

In our project we use a different cost function for the generator based on three points.

- Suppose that the discriminator performs well during the training stages, which means D(G(z)) will be close to 0.
- The original cost function is unbounded to negative infinity and it might cause instability during the optimization process.
- The generator will be optimized to fool the discriminator rather than to reconstruct the color of the original pictures.

The **updated cost function of generator** :

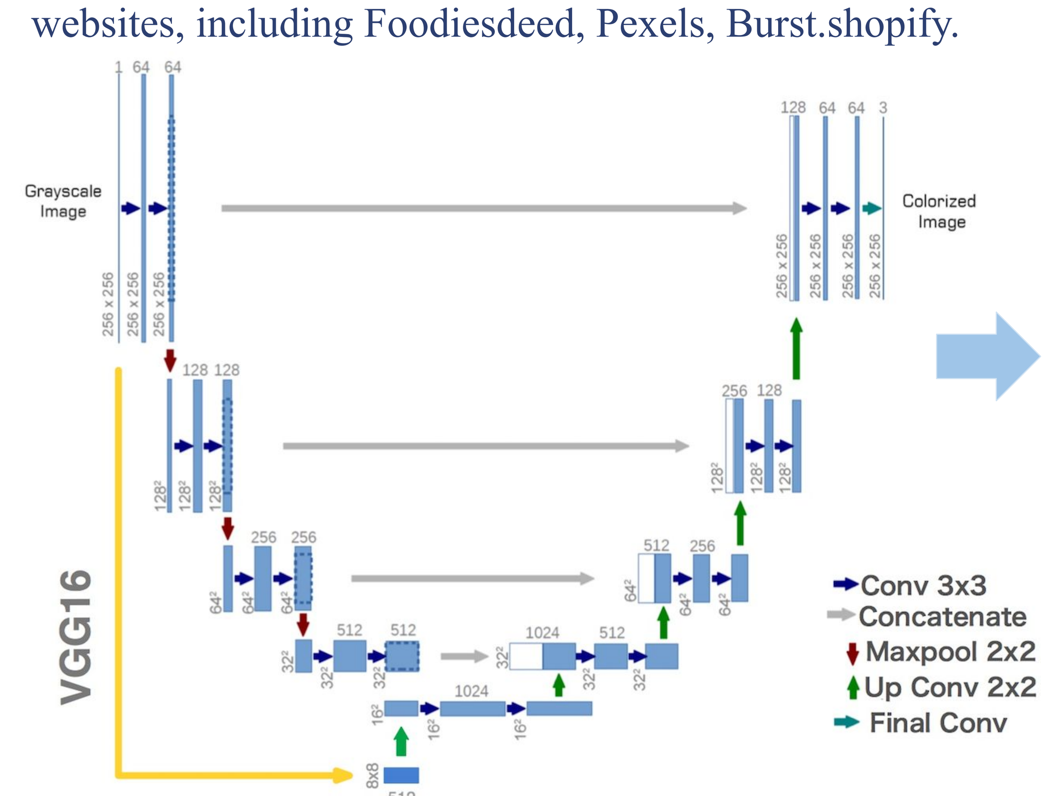
$$\min_{\theta^G} J^{(G)^*}(\theta_D, \theta_G) = \min_{\theta^G} -\mathbb{E}_z \left[\log(D(G(z)))\right] + \lambda \|G(z) - y\|_1$$

Creation:

We fix the weights of VGG16 instance that has been trained on the huge ImageNet dataset and use it to extract information from training images. Even though the VGG16 is mostly used for classification purposes rather than colorization, we reason that to classify images with high accuracies, a neural network must effectively capture the details of images such as colors, patterns, and shapes/edges. Thus, we use VGG16 as a basis for our model.

MODEL ARCHITECTURE

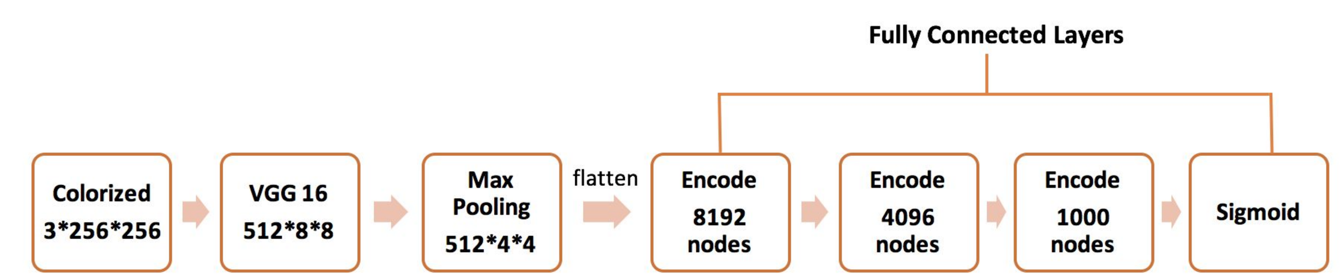
Data To gauge how well our model generalizes the colorization process, we test our models on several datasets including natural urban images, people images as well as food images. The urban natural images database from a SuperParsing paper contains 2,688 images.. We also experiment with larger datasets including real life people images: People In Photo Albums(PIPA) dataset consisting of over 5000 instances collected from public Flickr photo albums, and a thousand of food images from different all-free-download websites, including Foodiesdeed, Pexels, Burst.shopify.



Generator

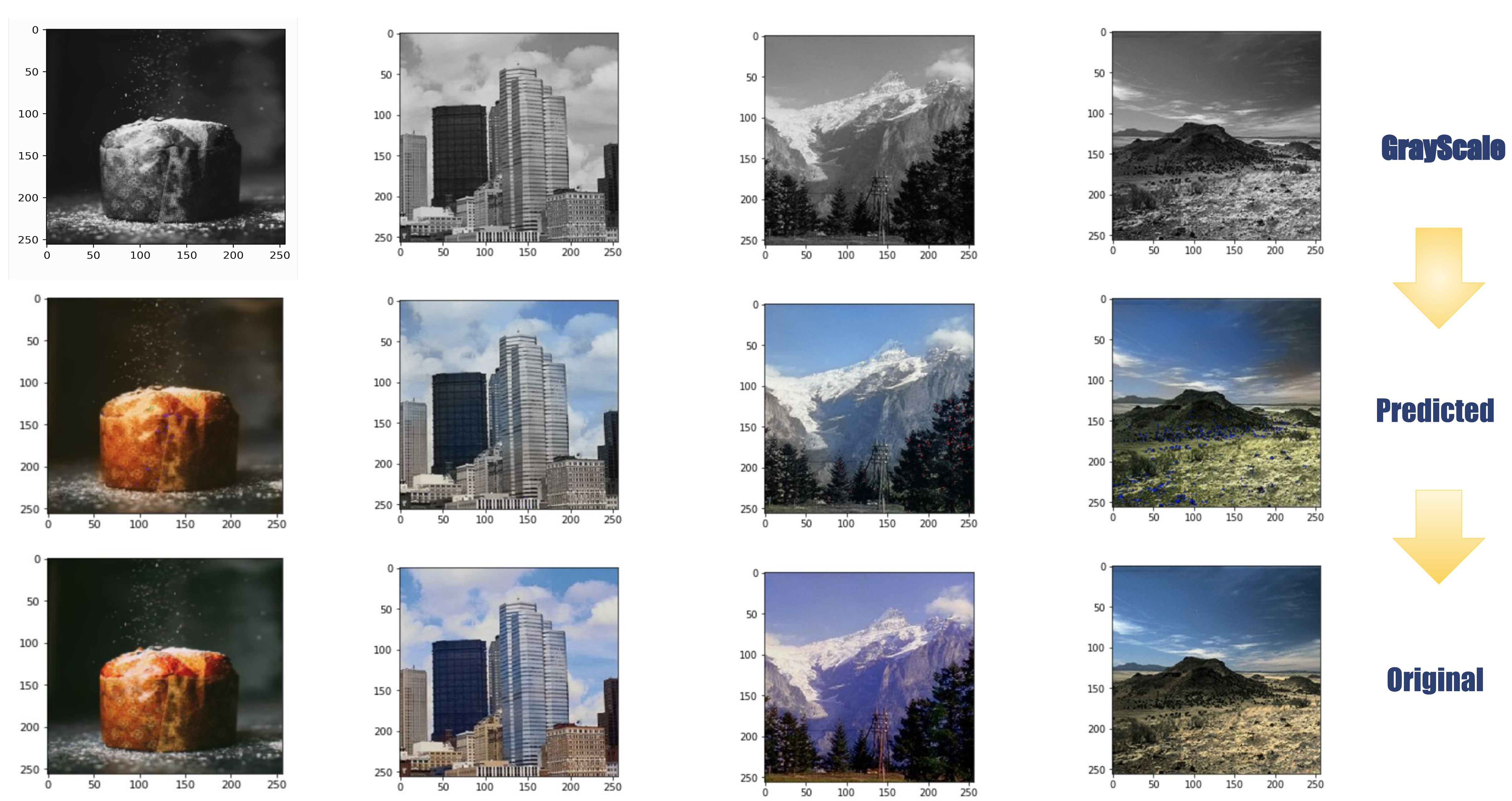
For the **GENERATOR**, we employ standard convolutional networks, the inputs of which are conditioned on gray or edge-only images. We also adopt VGG16 as the initial layers of the neural networks to extract information. Each layer in the contractive path is followed by batch normalization and leaky rectified linear unit. In particular, on each convolutional layer, we perform 3 * 3 convolution twice, each followed by a batch normalization, a (leaky) ReLU activation function and a 2 * 2 max pooling operation for downsampling. The dimension of inputs and outputs of the generator will be the same. We visualize the architecture as the figure on the left.

For the **DISCRIMINATOR**, we directly use VGG16 to extract the information from the image to get 512 * 8 * 8 with a max pooling following by to further shrink the dimension by half. We flatten the results for the full connected layers parts, where each Encode unit denotes FullyConnected-BatchNorm-LeakyReLU-Dropout. Finally, we use sigmoid function to get the probability.



Discriminator

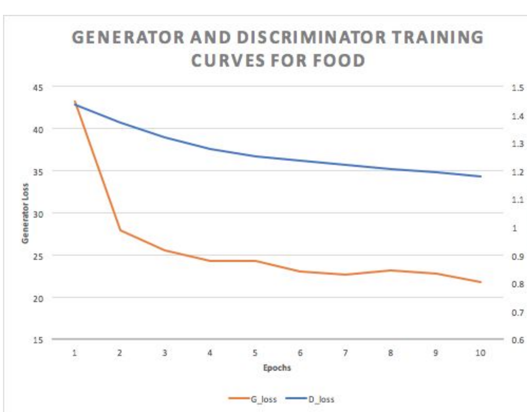
AUTOMATIC COLORIZATION RESULTS

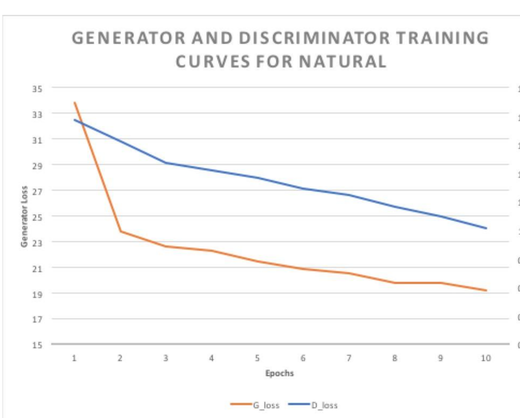


TRAINING PROCEDURES

Parameter optimization settings and training results:

Dataset	D_loss	G_loss	Learning Rate	Adam Optimize Parameter Beta1	MAE
Natural	0.99	19.70	0.0002	0.50	0.19
Natural	0.69	23.90	0.0020	0.50	0.22
Natural	1.04	20.30	0.0002	0.05	0.19
Natural	1.01	19.20	0.0002	0.90	0.18
Food	1.14	20.90	0.0002	0.50	0.70
Food	0.59	26.50	0.0020	0.50	0.86
Food	1.18	21.80	0.0002	0.05	0.66
Food	1.16	21.60	0.0002	0.90	0.69





CONCLUSIONS & FUTURE WORKS

We have creatively combined the Deep Convolution GAN with VGG16, and made the two giant architecture work together successfully on the automatic colorization task.

There are several future works:

- Increase number of training images and training epochs to allow generators and discriminators to further decrease training loss.
- Perform trainings on a more balanced dataset.
- Optimize model architectures to reduce computational costs.
- A comparison study among our proposed architecture and other successful ones.

Reference

[FHY17] Qiwen Fu, Wei-Ting Hsu, and Mu-Heng Yang. Colorization using convnet and gan. 2017.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, and David Warde-Farley. it generative adversarial nets. *In Advances in neural information processing systems*, 2014.

[HZ16] Jeff Hwang and You Zhou. Image colorization with deep convolutional neural networks. 2016.

[IZZE17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[LLW04] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *In ACM Transactions on Graphics (ToG), volume 23, pages 689–694. ACM,*, 2004.

[MHN13] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

[MO14] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[NN18] Kamyar Nazeri and Eric Ng. Image colorization with generative adversarial networks. 2018.

[RDS+15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, and Z. Huang. it imagenet large scale visual recognition challenge. *International Journal of Computer Vision 115(3):211–252*, 2015.

[TL10] Joseph Tighe and Svetlana Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *European conference on computer vision*, pages 352–365. Springer, 2010.

[TMK02] T.Welsh, M.Ashikhmin, and K.Mueller. it transferring color to greyscale images. *In ACM Transactions on Graphics (TOG), volume 21, pages 277–280. ACM,*, 2002.

[ZPT+15] Ning Zhang, Manohar Paluri, Yaniv Taigman, Rob Fergus, and Lubomir Bourdev. Beyond frontal faces: Improving person recognition using multiple cues. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4804–4813, 2015.