Enhancing Image Colorization Architectures

COMP411 Term Project

24.01.24

Introduction

- Colorization based on known data
- Dataset availability and diversity
- Model capability and complexity

Recovering high dimensional data from its low-dimensional representation. *Kind of “hallucination”.*

Challenge: Grayscale images only have intensity of tones, hence information loss, while our aim is to create a 3-channel understanding of details and regions.

Currently, CNNs and GANs to learn complex patterns and mappings within the dataset.

Created by DALL-E

Introduction

- Common texture and pattern recognition
- Semantic understanding of objects
- User input

No-ML: histogram matching, pixel correlation

ML: supervised or unsupervised learning, fine tuning, support by transfer learning

Statistical models → U-Nets.

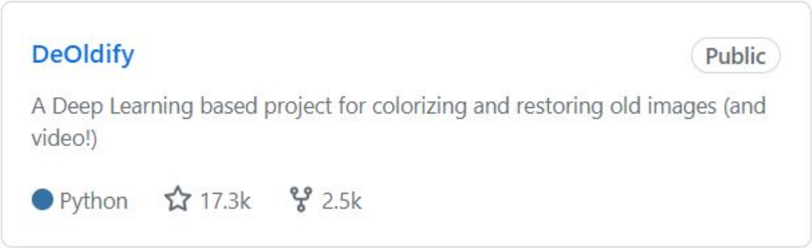
Improving realism and accuracy in colorization.

Combination of art and technology.

Created by DALL-E

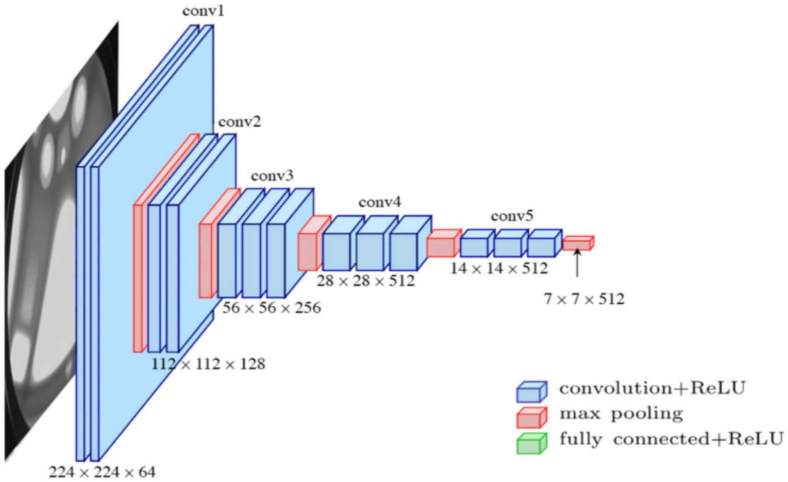
Exploring Models

- Trying one of the most popular colorization model, DeOldify.
- Yet the model outputs were faded, lack of vivid colors, desaturated.



Transfer Learning

- Extracted embeddings using VGG16.



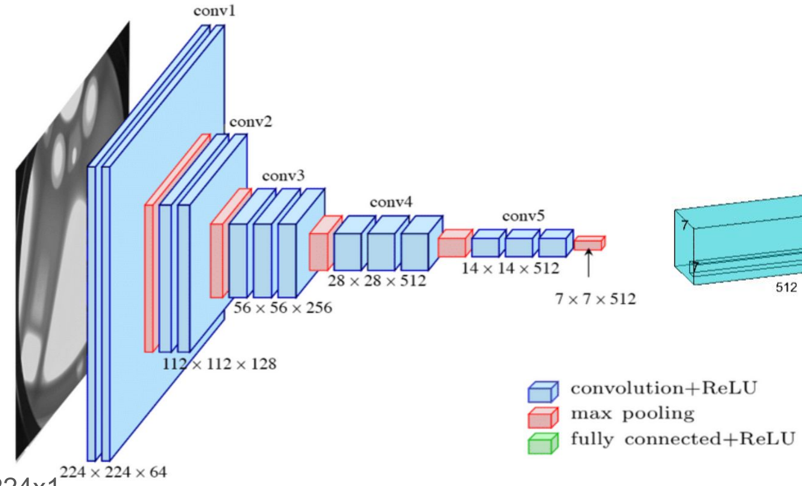
224x224x1

Encoder part of VGG16

Transfer Learning

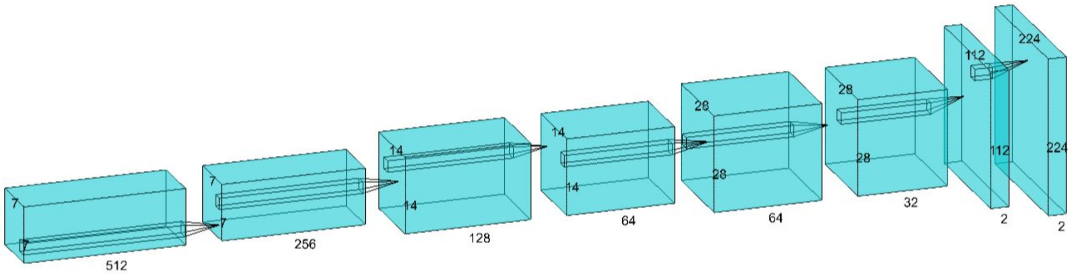
- Extracted embeddings using VGG16.

- Decoded back to LAB color space.



224x224x1

Encoder part of VGG16



Our custom decoder

Decoder Code Block

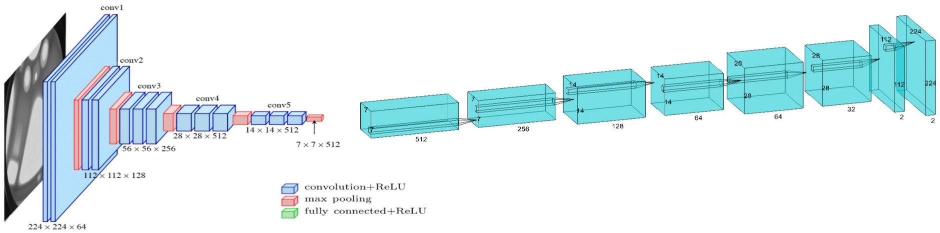
```
#Decoder
model = Sequential()
model.add(Conv2D(256, (3,3), activation='relu', padding='same', input_shape=(7,7,512)))
model.add(Conv2D(128, (3,3), activation='relu', padding='same'))
model.add(UpSampling2D((2, 2)))
model.add(Conv2D(64, (3,3), activation='relu', padding='same'))
model.add(UpSampling2D((2, 2)))
model.add(Conv2D(32, (3,3), activation='relu', padding='same'))
model.add(UpSampling2D((2, 2)))
model.add(Conv2D(16, (3,3), activation='relu', padding='same'))
model.add(UpSampling2D((2, 2)))
model.add(Conv2D(2, (3, 3), activation='tanh', padding='same'))
model.add(UpSampling2D((2, 2)))
model.summary()
```

Transfer Learning

To be sure if our model learns properly, we let the model overfit on a small dataset.

We used **MSE loss function** to realize those experiments.

Our input is L channel of the input image in **LAB color space**.
We predicted AB channels in the output.



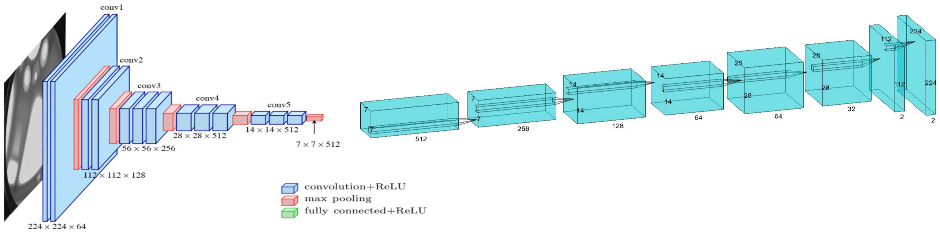
Transfer Learning

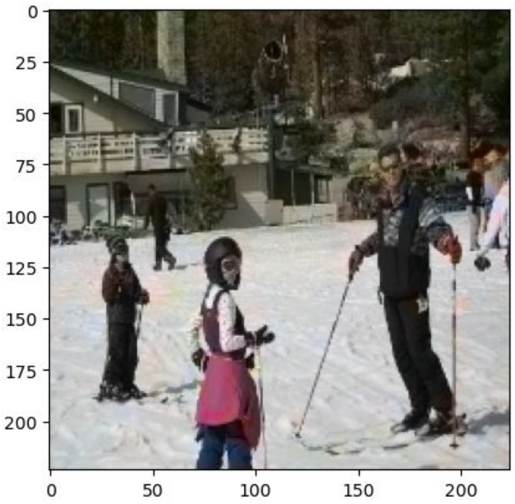
To be sure if our model learns properly, we let the model overfit on a small dataset.

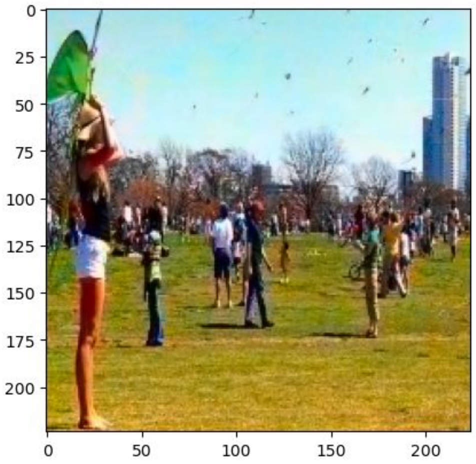
We used MSE loss function to realize those experiments.

Our input is L channel of the input image in LAB color space.
We predicted AB channels in the output.

Then we **concatenated L and AB** to get the outputs you see on the right

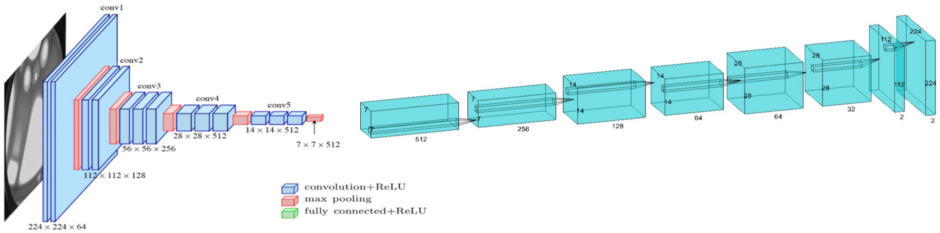


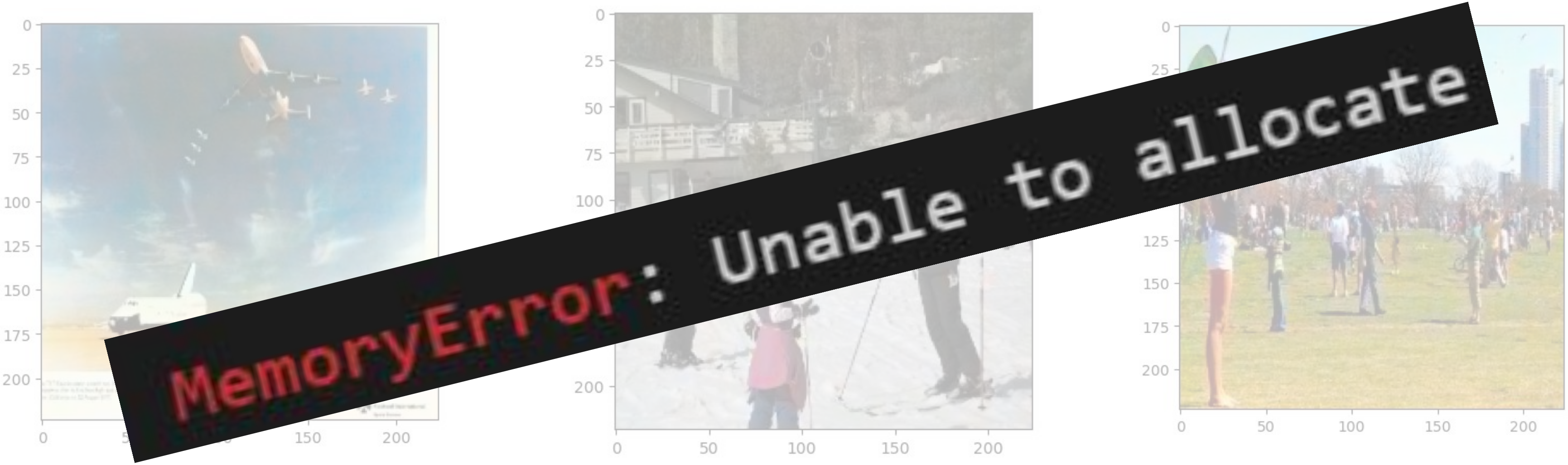




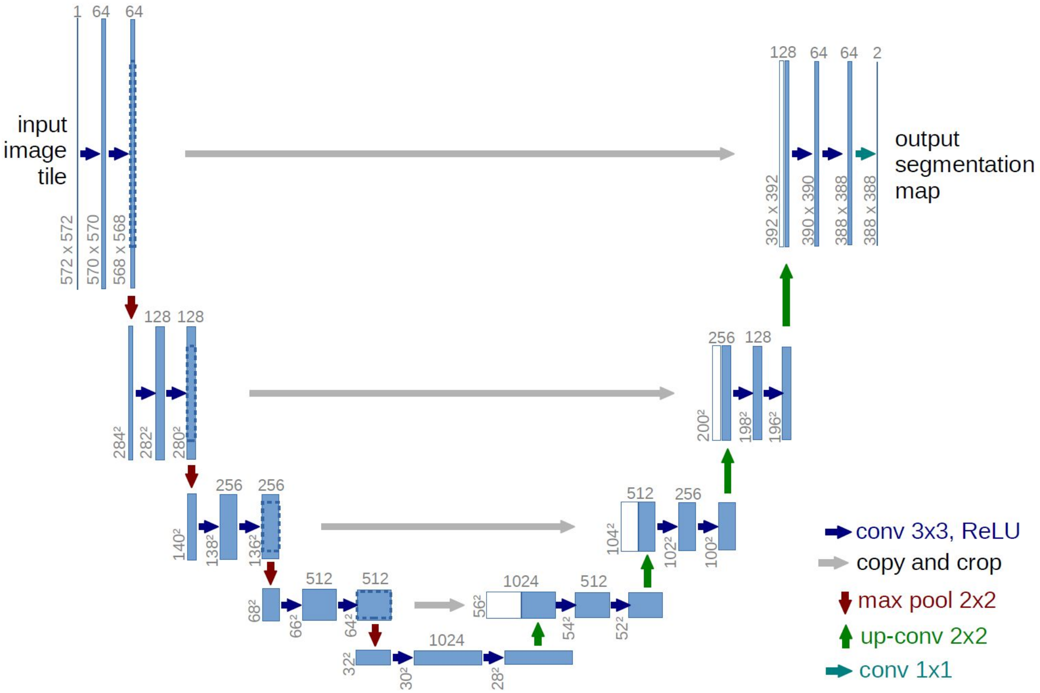
Transfer Learning

Yet we could not train the model on a larger dataset due to computational constraints.





Design and Implementation



https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

Design and Implementation

Simple UNet Architecture

Encoder 1: 1x150x150 -> 64x146x146

Encoder 2: 64x146x146 -> 128x69x69

Middle: 128x69x69 -> 64x48x48

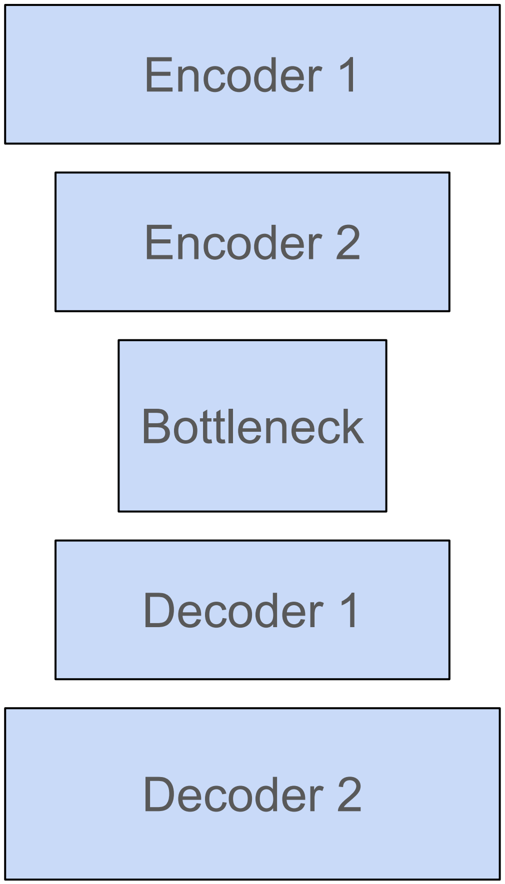
Decoder 1: 64x48x48 -> 32x88x88

Decoder 2: 32x88x88 -> 32x168x168

Final: 32x168x168 -> 3x150x150

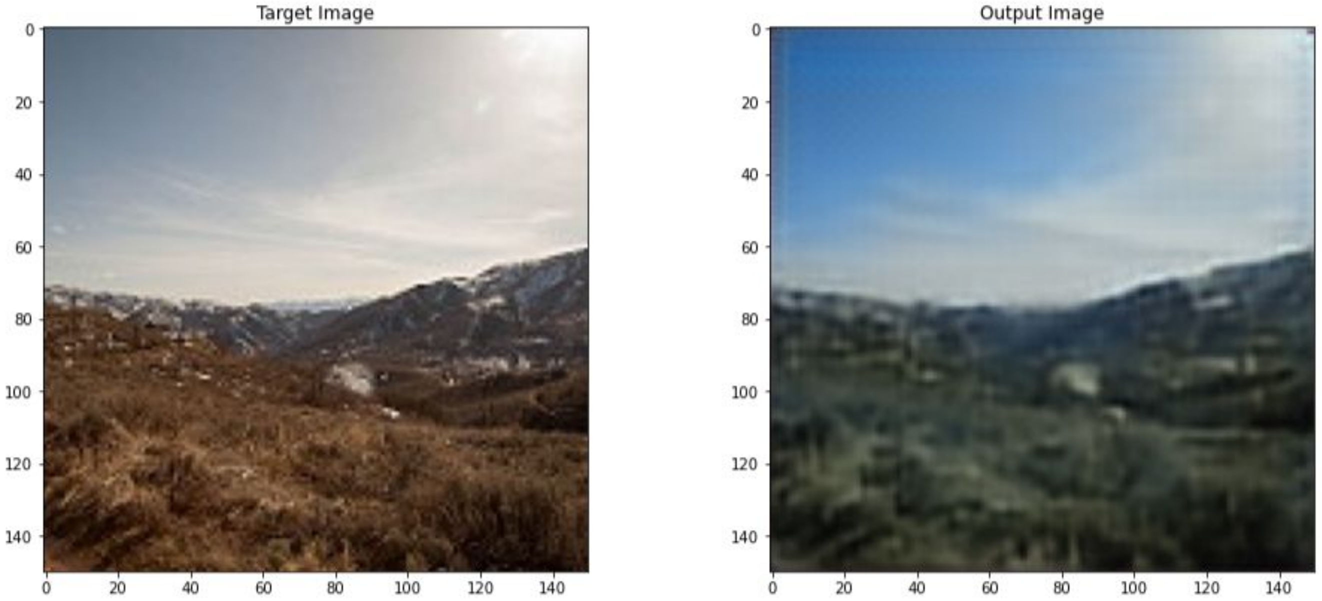
No skip connections.

Datasets: [5,6] combination of 12k images


Encoder 1
Encoder 2
Bottleneck
Decoder 1
Decoder 2

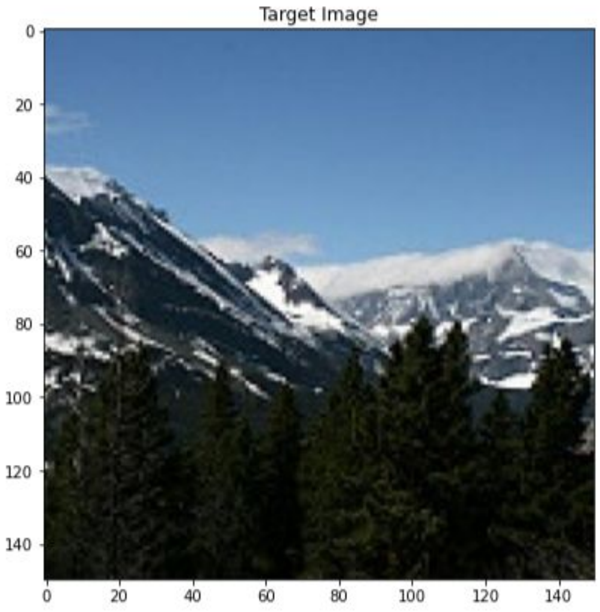
Results and Problems: Color Bias

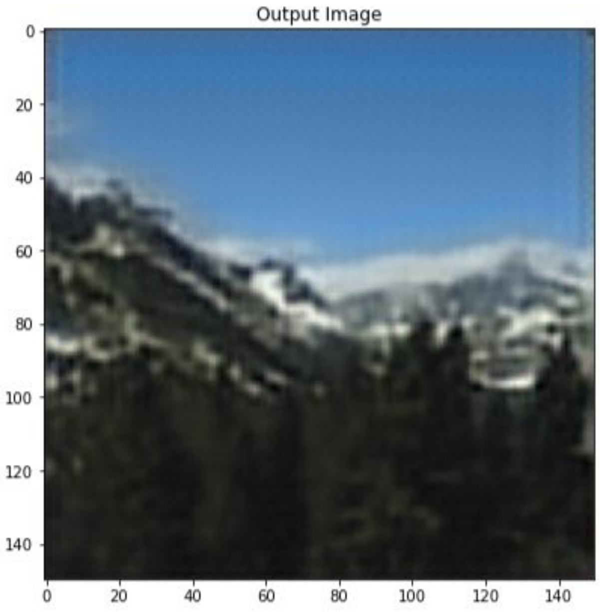
Due to massive amount of landscape images on dataset, output images contains more of blue color.



Results and Problems: Corner Artifacts

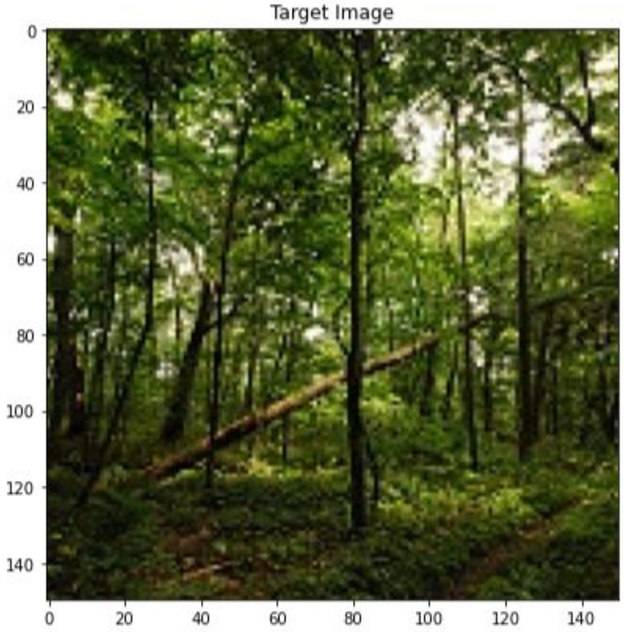
Since there are no padding during convolution we see lower resolution on edges and corners.

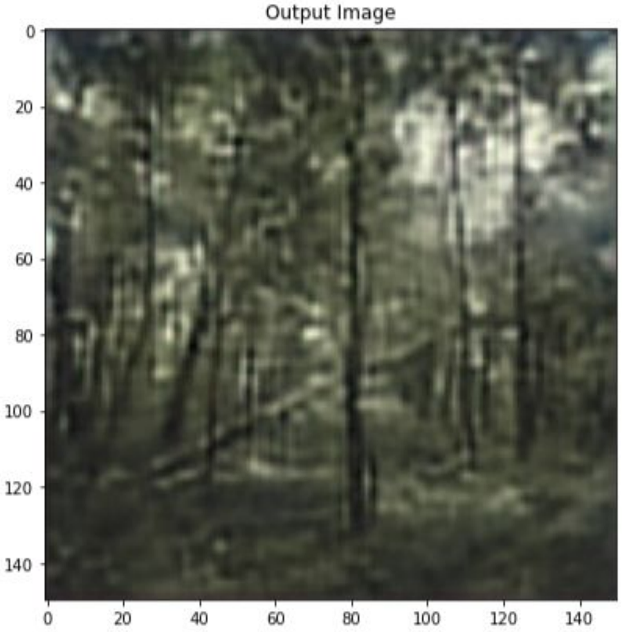




Results and Problems: Reduced Resolution

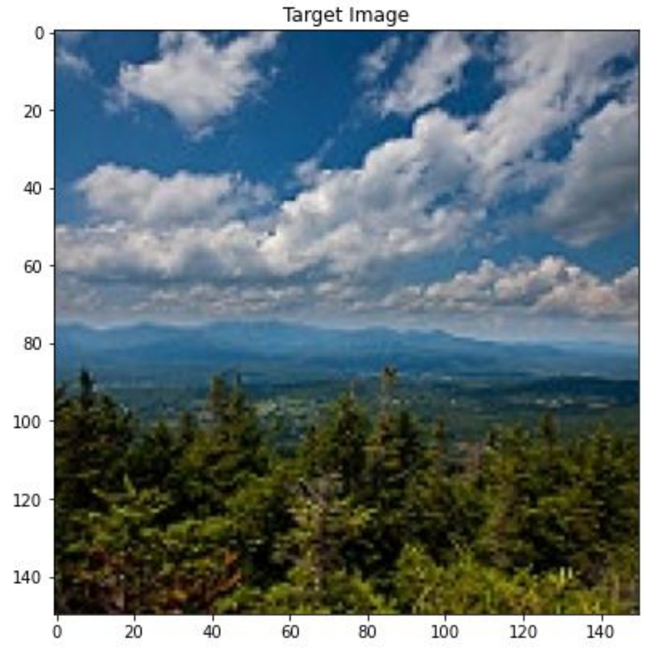
Due to wrong model hyperparameters we see low resolution outputs.

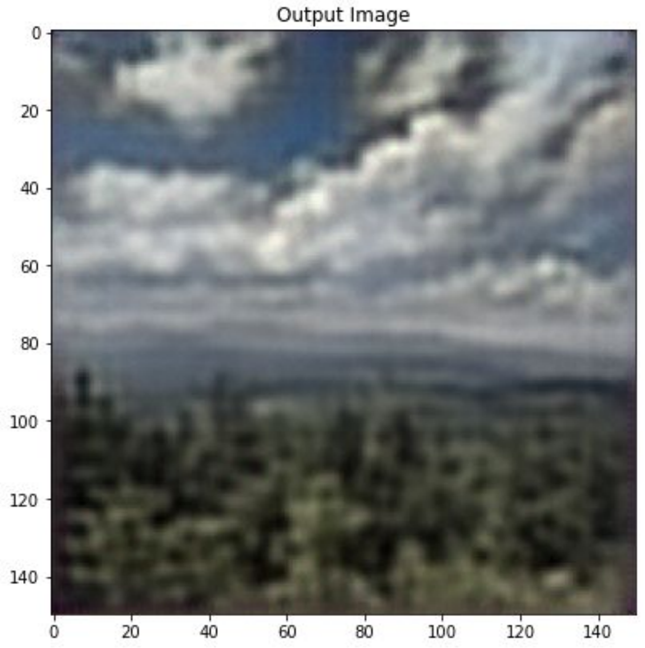




Results and Problems

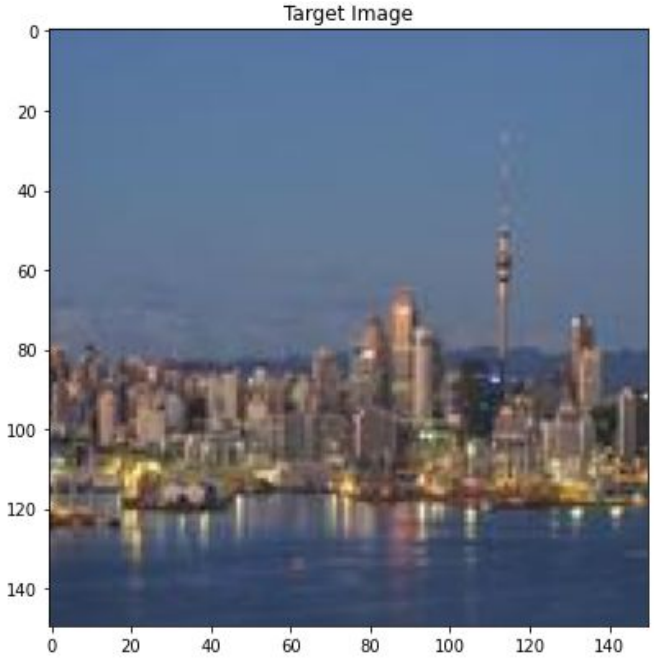
Here you can see more examples.

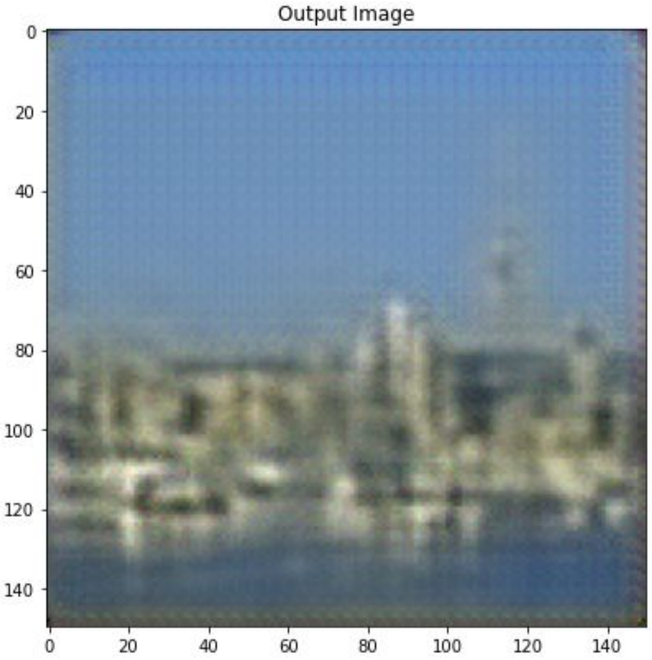




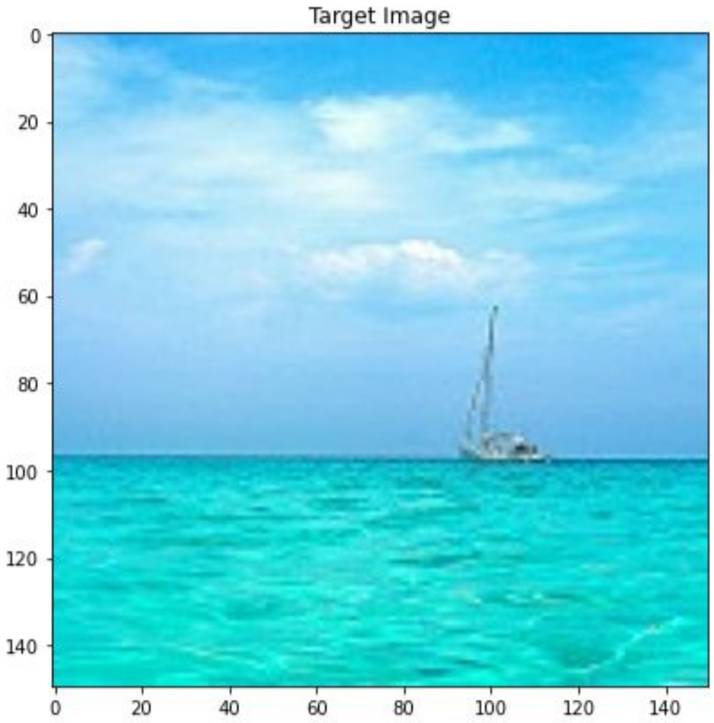
Results and Problems: Checkerboard Artifacts

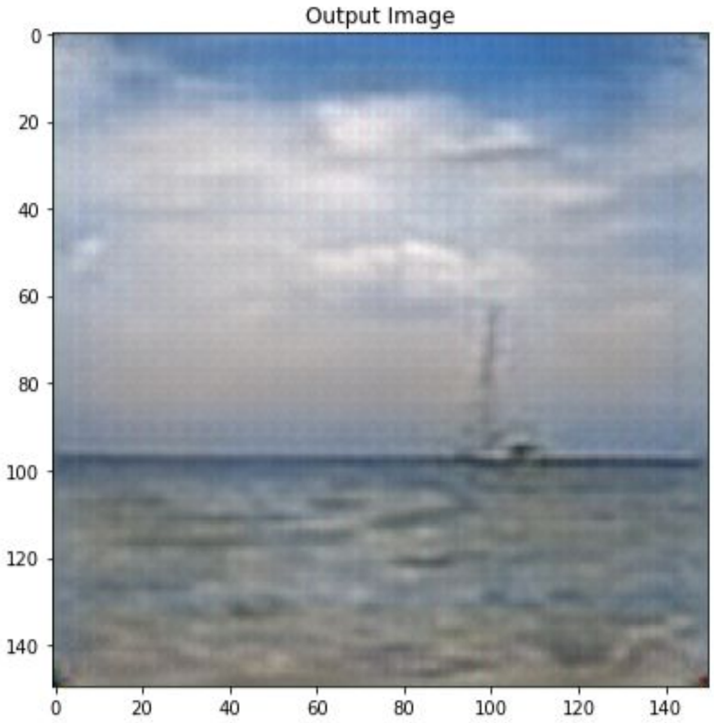
Due to transpose convolution we see a checkerboard like pattern on the image.



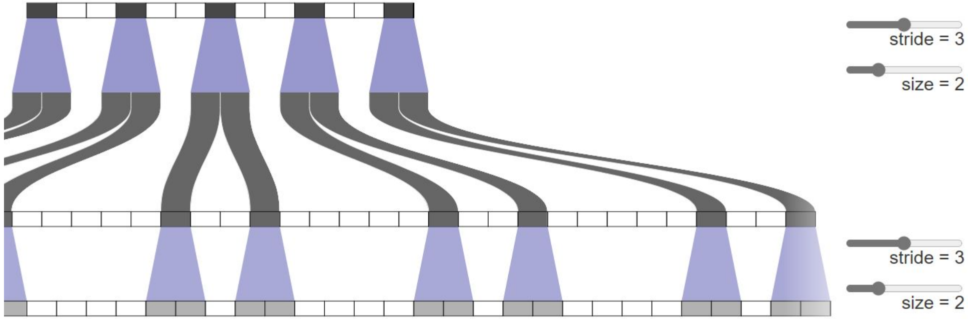


Results and Problems: Checkerboard Artifacts



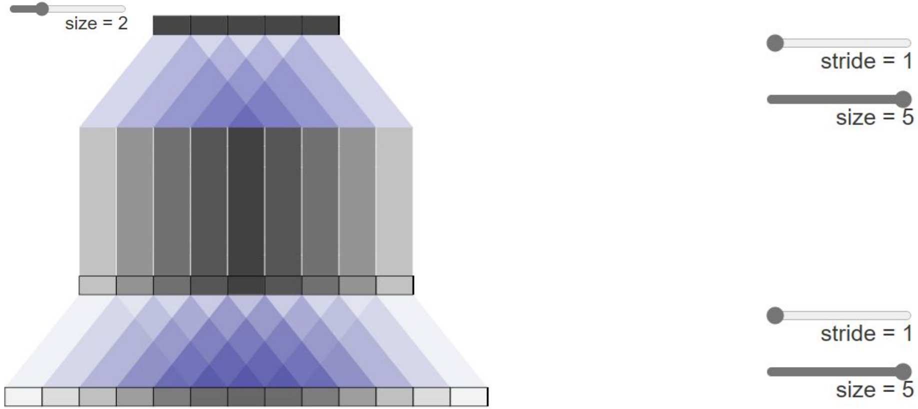


Results and Problems: Checkerboard Artifacts



Deconvolution and Checkerboard Artifacts

AUGUSTUS ODENA — Google Brain | VINCENT DUMOULIN — Université de Montréal | CHRIS OLAH — Google Brain | Oct. 17 2016 | Citation: Odena, et al., 2016



https://distill.pub/2016/deconv-checkerboard/

Design and Implementation

UNet-VAE Hybrid Architecture

Encoder 1: 1x150x150 -> 64x74x74

Encoder 2: 64x74x74 -> 64x36x36

Encoder 3: 64x36x36 -> 64x17x17

VAE: 64x17x17 -> 64x17x17

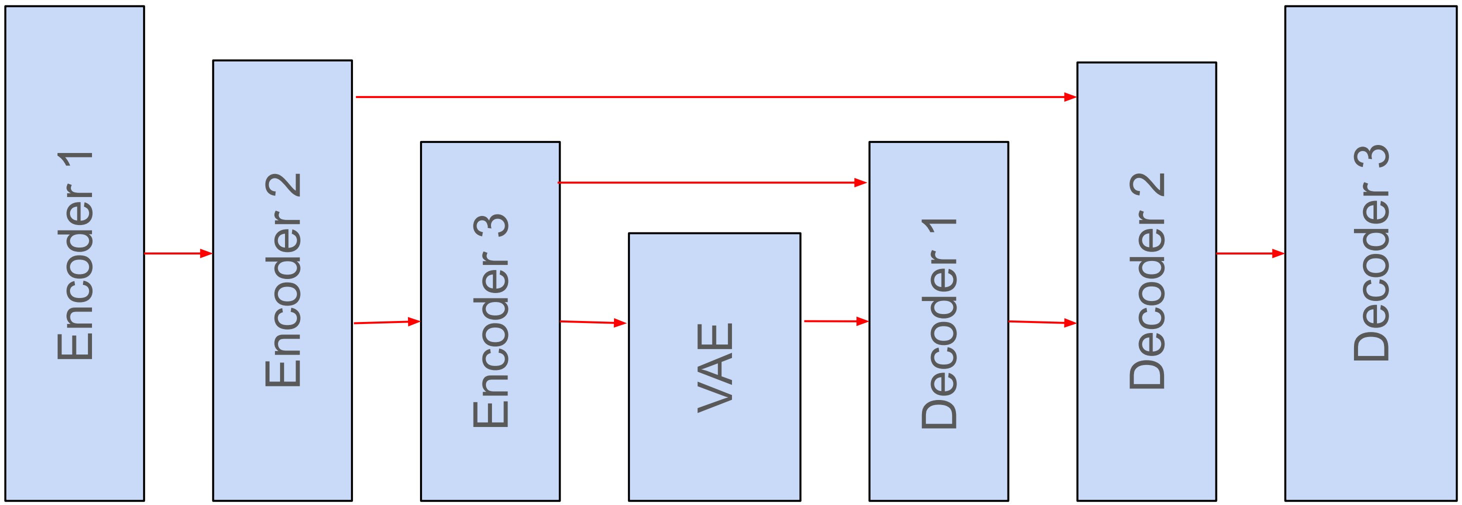
Decoder 1: 128x17x17 -> 64x38x38

Decoder 2: 128x38x38 -> 64x80x80

Decoder 3: 64x80x80 -> 64x164x164

Final: 64x164x164 -> 3x150x150

Datasets: [5,6,7] combination of 36k images


Encoder 1
Encoder 2
Encoder 3
VAE
Decoder 1
Decoder 2
Decoder 3

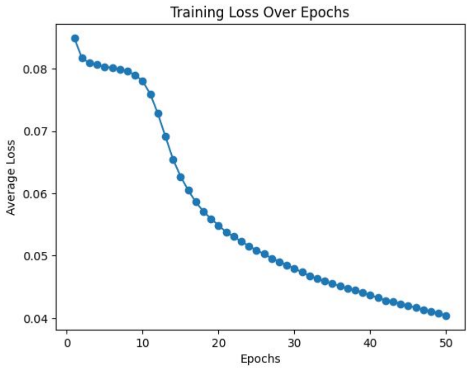
Results: A Good Model

- 3 layer K=4 S=2 Encoder
- VAE
- 3 layer K=6 S=1 Decoder

- Channel size
- Interpolation in skip connections

- 4 layer K=3 S=2 Encoder
- VAE
- 4 layer K=3 S=2 Decoder

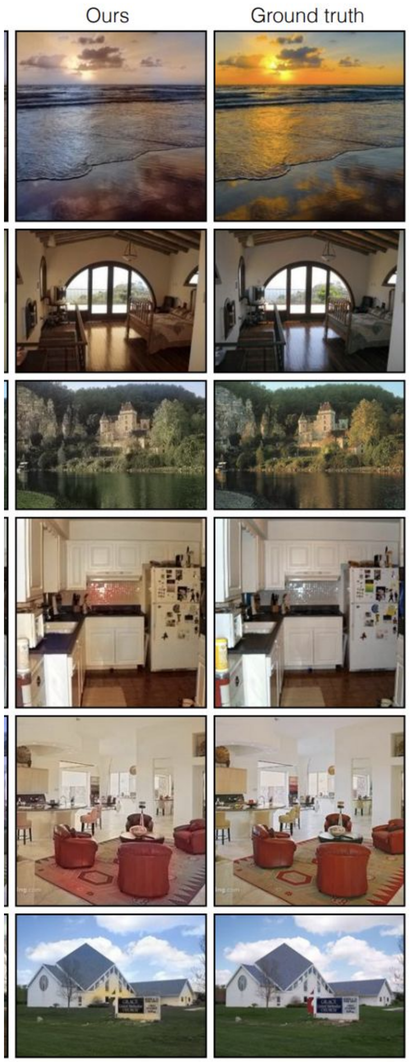
- Insufficient resources (Colab, time, dataset)

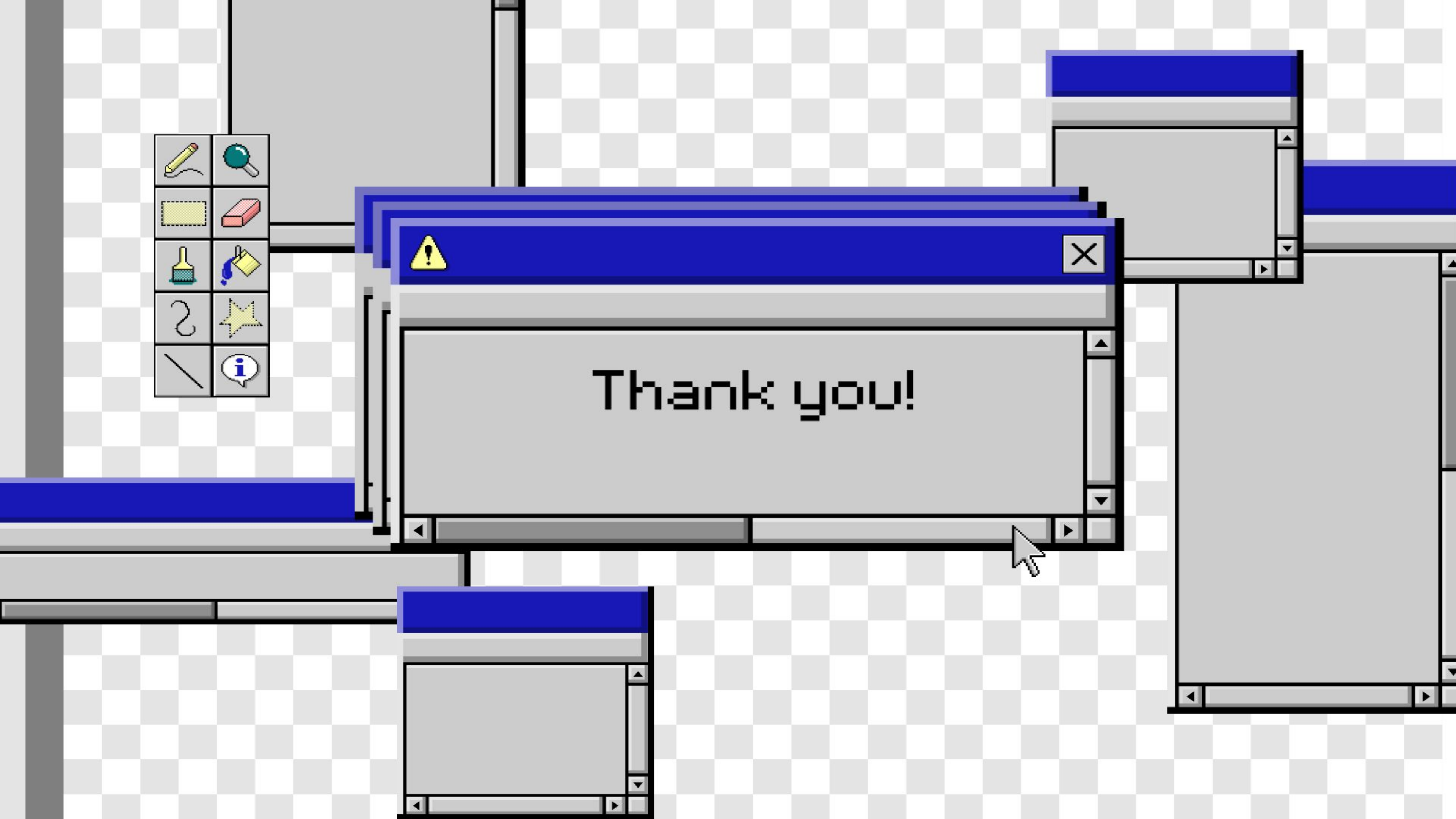


Recap and further work

- Deeper architectures
- Transfer learning
- Leveraging color spaces
- Transposed convolution (U-Net)
- Skip connection

- Conditional GAN
- Variety of loss functions
    - Reweighting the loss for emphasis of rare colors
    - Penalizing dominant output colors




Thank you!

References

[1] Jason Antic. jantic/deoldify: A deep learning based project for colorizing and restoring old images (and video!). https://github.com/jantic/DeOldify, 2019.

[2] A. Salmona, L. Bouza, and J. Delon, “Deoldify: A review and implementation of an automatic colorization method,” Image Processing On Line, vol. 12, pp. 347–368, 2022. doi:10.5201/ipol.2022.403

[3] R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” Computer Vision – ECCV 2016, pp. 649–666, 2016. doi:10.1007/978-3-319-46487-9_40

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234-241. Available: https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

[5] TheBlackMamba31. (2022, August). Landscape Image Colorization. Retrieved January 24, 2024 from https://www.kaggle.com/datasets/theblackmamba31/landscape-image-colorization.

[6] Arnaud58. (2022, August). Landscape Pictures. Retrieved January 24, 2024 from https://www.kaggle.com/datasets/arnaud58/landscape-pictures.

[7] Mirflickr. (n.d.). MIRFLICKR - 25k Dataset. Retrieved January 24, 2024 from https://press.liacs.nl/mirflickr/mirdownload.html.

[8] Dumoulin, V., & Olah, C. (2016, October). Deconvolution and Checkerboard Artifacts. Distill. Retrieved January 24, 2024 from https://distill.pub/2016/deconv-checkerboard/.