Image Repair Using Deep Convolutional General Adversarial Networks

Martin Trpceski and Sean Gleason

ECE 208, Spring 2025

Introduction

Old images are often subject to damage due to weathering leaving scratches, stains, reduction of contrast as well as limitations of old imaging techniques such as over exposure.

Manual Image restoration services do exist, but this requires extensive care and time to be taken by skilled professionals, and thus can end up being expensive.

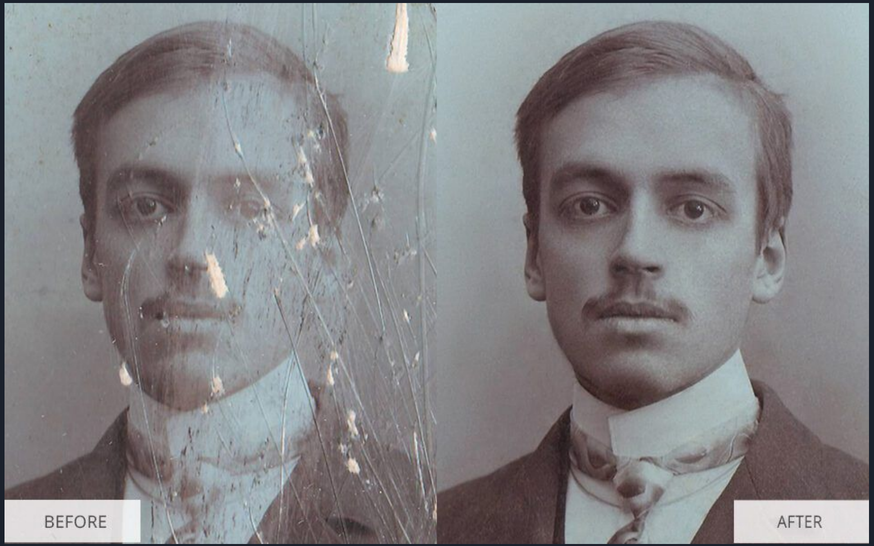


Image source: https://blog.icons8.com/articles/best-photo-restoration-services/

Generative Adversarial Networks (GANs) are a fairly new machine learning method that have shown to be robust in a variety of generative tasks through diffusion [1].

The subset of GANs used for image applications are called Deep Convolutional GANs (DCGANs)

The purpose of this project is to use DCGANs for image repair.

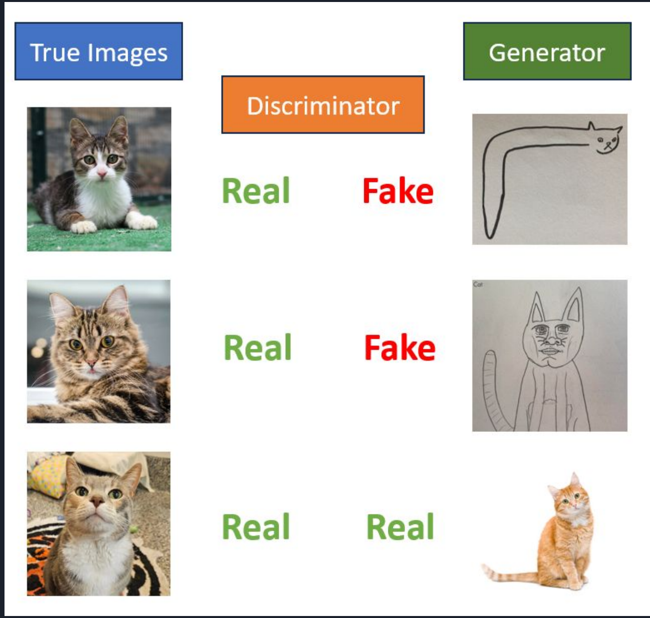


Image sources:
https://www.fourpawsusa.org/our-stories/publications-guides/a-cats-personality
https://www.boredpanda.com/funny-poorly-drawn-cats/
https://www.greensboro.carolinavet.com/site/greensboro-specialty-veterinary-blog/2023/03/15/how-to-choose-cat-breed
https://www.reddit.com/r/drawing/comments/wcd0lt/my_cat/?rdt=58196
https://www.facebook.com/photo.php?fbid=1030686362439913&id=100064956751160&set=a.325146289660594
https://www.istockphoto.com/photos/domestic-cat

[1] Zhendong Wang, undefined., et al, "Diffusion-GAN: Training GANs with Diffusion," 2023.

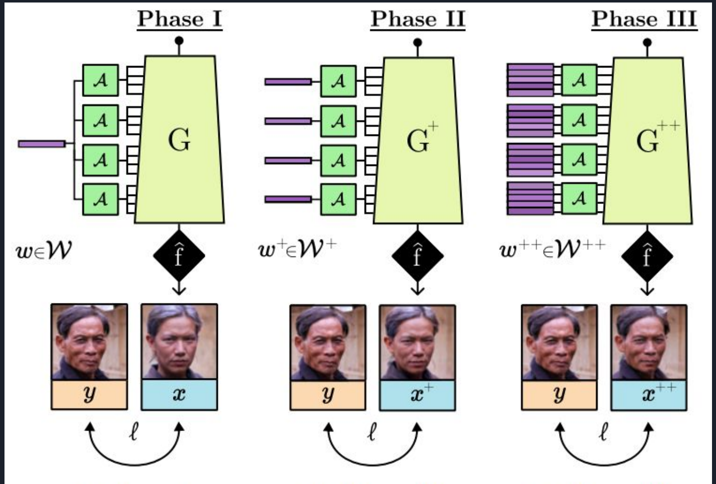
Methods

For objective evaluation, a dataset of 10 classes from the Imagenette dataset, with pre-processing included to make the images mimic old photos was created, consisting of 9469 training images.

For the subjective evaluation, a smaller dataset of old images from the Library of Congress was used [2].

[2] "Daguerreotypes" Library of Congress, Prints & Photographs Division, [reproduction number, e.g., LC-USZ62-12345]

A benchmark model was also used: a modified, inverted version of the StyleGAN network to leverage multi-stage latent space expansion [3].



(a) Phase I (b) Phase II (c) Phase III

Algorithm 1: Robust StyleGAN inversion.

Output: restored image $\boldsymbol{x}^{++}$

`# Phase I`

$\boldsymbol{w} = \mathbb{E}_{\tilde{\boldsymbol{w}} \in \mathcal{W}}[\tilde{\boldsymbol{w}}]$

for `1 to 150` **do**

  $\boldsymbol{x} \leftarrow G(\boldsymbol{w})$

  $\boldsymbol{w} \leftarrow 0.08 \bar{\nabla}_{\boldsymbol{w}} \ell(\hat{f}(\boldsymbol{x}), \boldsymbol{y})$

`# Phase II`

$\boldsymbol{w}^{+} = \mathrm{repeat}(\boldsymbol{w}, N_L)$

for `1 to 150` **do**

  $\boldsymbol{x}^{+} \leftarrow G^{+}(\boldsymbol{w}^{+})$

  $\boldsymbol{w}^{+} \leftarrow 0.02 \bar{\nabla}_{\boldsymbol{w}} \ell(\hat{f}(\boldsymbol{x}^{+}), \boldsymbol{y})$

`# Phase III`

$\boldsymbol{w}^{++} = \mathrm{repeat}(\boldsymbol{w}^{+}, N_F)$

for `1 to 150` **do**

  $\boldsymbol{x}^{++} \leftarrow G^{++}(\boldsymbol{w}^{++})$

  $\boldsymbol{w}^{++} \leftarrow 0.005 \bar{\nabla}_{\boldsymbol{w}^{++}} \ell(\hat{f}(\boldsymbol{x}^{++}), \boldsymbol{y})$

return $\boldsymbol{x}^{++}$

[3] Y. Poirier-Ginter, J. Lalonde, "Robust Unsupervised StyleGAN Image Restoration," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 22292-22301.

The custom DCGAN consisted of a straightforward implementation of a binary classifier for the discriminator and a simple convolutional network for the generator.

Layer (type)	Output Shape	Param #
ConvTranspose2d-1	[-1, 256, 259, 259]	4,096
BatchNorm2d-2	[-1, 256, 259, 259]	512
ReLU-3	[-1, 256, 259, 259]	0
ConvTranspose2d-4	[-1, 128, 518, 518]	524,288
BatchNorm2d-5	[-1, 128, 518, 518]	256
ReLU-6	[-1, 128, 518, 518]	0
ConvTranspose2d-7	[-1, 64, 1036, 1036]	131,072
BatchNorm2d-8	[-1, 64, 1036, 1036]	128
ReLU-9	[-1, 64, 1036, 1036]	0
MaxPool2d-10	[-1, 64, 259, 259]	0
ConvTranspose2d-11	[-1, 32, 260, 260]	32,768
BatchNorm2d-12	[-1, 32, 260, 260]	64
ReLU-13	[-1, 32, 260, 260]	0
MaxPool2d-14	[-1, 32, 257, 257]	0
ConvTranspose2d-15	[-1, 1, 256, 256]	512
Tanh-16	[-1, 1, 256, 256]	0

Generator

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 128, 128]	512
LeakyReLU-2	[-1, 32, 128, 128]	0
Conv2d-3	[-1, 64, 64, 64]	32,768
BatchNorm2d-4	[-1, 64, 64, 64]	128
LeakyReLU-5	[-1, 64, 64, 64]	0
Conv2d-6	[-1, 128, 32, 32]	131,072
BatchNorm2d-7	[-1, 128, 32, 32]	256
LeakyReLU-8	[-1, 128, 32, 32]	0
Conv2d-9	[-1, 256, 16, 16]	524,288
BatchNorm2d-10	[-1, 256, 16, 16]	512
LeakyReLU-11	[-1, 256, 16, 16]	0
MaxPool2d-12	[-1, 256, 4, 4]	0
Conv2d-13	[-1, 256, 2, 2]	1,048,576
BatchNorm2d-14	[-1, 256, 2, 2]	512
LeakyReLU-15	[-1, 256, 2, 2]	0
Flatten-16	[-1, 1024]	0
Linear-17	[-1, 100]	102,500
ReLU-18	[-1, 100]	0
Linear-19	[-1, 100]	10,100
ReLU-20	[-1, 100]	0
Linear-21	[-1, 2]	202
Softmax-22	[-1, 2]	0

Discriminator

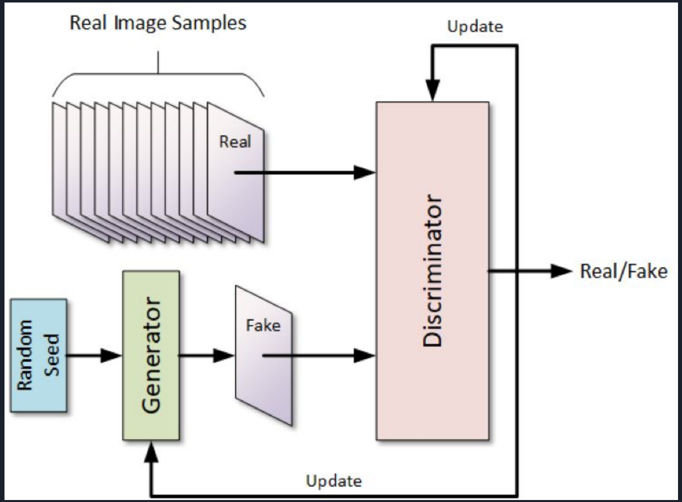


Image source:
https://semiengineering.com/knowledge_centers/artificial-intelligence/neural-networks/generative-adversarial-network-gan/

Coding was conducted using Kaggle to allow for collaboration. Pytorch was used to develop and train the custom models.

Image source: https://datascientest.com/en/pytorch-all-about-this-framework

Image source: https://opendatascience.com/10-tips-to-get-started-with-kaggle/

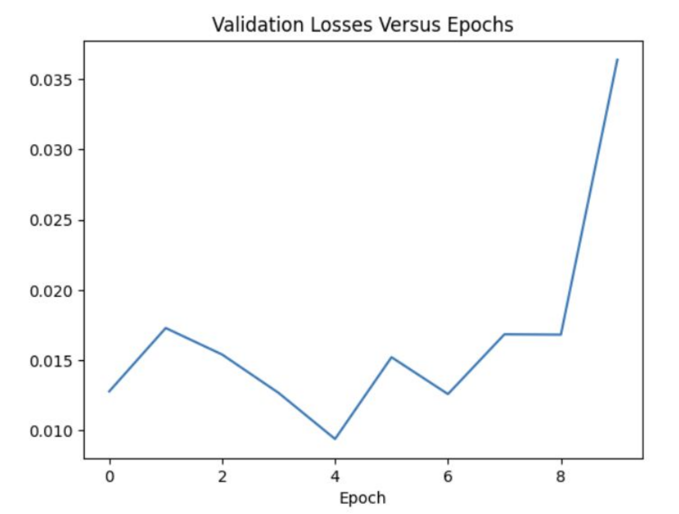
To evaluate the models both a subjective evaluation and objective evaluation was used.

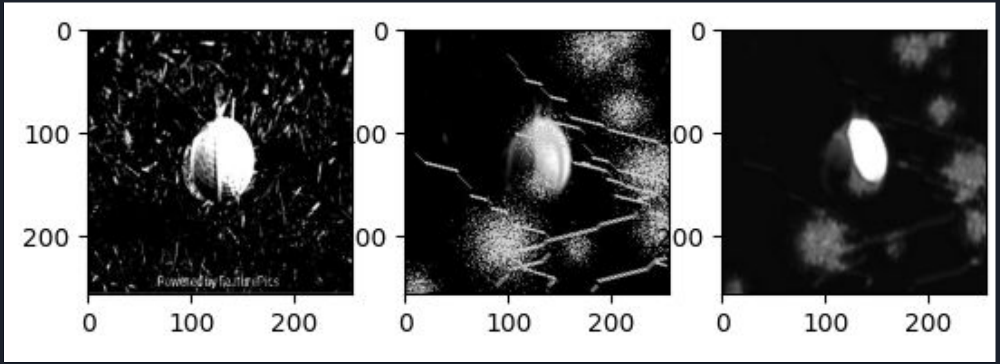
Objective evaluation -> loss values

Subjective evaluation -> direct observation of the model output

Results

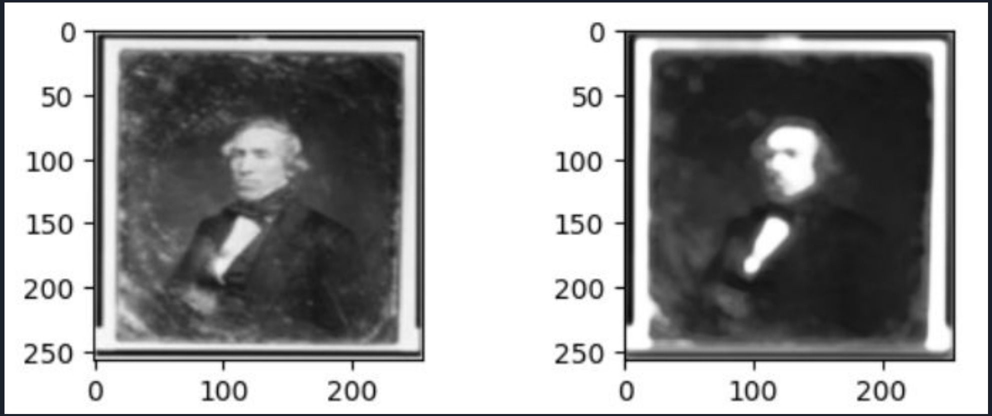
Below shows the validation loss versus epchs and examples of the reconstructed images from the best custom DCGAN.





Example output from the imagenette dataset showing the clean image(left), damaged image (center) and restored image(right)

When using the model on the subjective evaluation dataset, it can be seen that the GAN seems to make the images blurrier.



Example output from the daguerreotypes dataset showing the original image(left) and restored image(right)

The code provided by the researchers for the benchmark, StyleGAN, model was somewhat unclear and seriously time consuming to run, and thus we were unable to collect data on this model.

Discussion

The custom build GAN seemed to just make the images blurrier.

It is able to improve upon certain issues, so it is possible that multiple different models would be needed to repair different types of problems.

Limitations

DCGANs are very resource intensive, so training was very slow, making it difficult to fine tune the model hyperparameters and architecture and prevented the use of a larger dataset.

GANs are also generally difficult to train due to the fact that there are 2 models that must be taken into account and that loss values are often highly abstracted from actual successful performance.

There were also several technical difficulties using Kaggle and limitations for GPU resources.

Contributions

Martin Trpceski - Researched benchmark models, designed the custom model, developed the training code for the custom models, worked on presentation and paper.

Sean Gleason - Developed image processing code, created the dataset, ran the training code with Kaggle, conducted subjective evaluation, worked on the presentation and paper.

References

[1] Zhendong Wang, undefined., et al, "Diffusion-GAN: Training GANs with Diffusion," 2023.
[2] "Daguerreotypes" Library of Congress, Prints & Photographs Division, [reproduction number, e.g., LC-USZ62-12345]
[3] Y. Poirier-Ginter, J. Lalonde, "Robust Unsupervised StyleGAN Image Restoration," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 22292-22301.

Image Repair Using Deep Convolutional General Adversarial Networks

Martin Trpceski and Sean Gleason

ECE 208, Spring 2025