eriklindernoren / Keras-GAN

Keras implementations of Generative Adversarial Networks.

#deep-learning #gan #keras #generative-adversarial-networks #neural-networks

180 commits | 1 branch | 0 releases | 13 contributors | MIT

Branch: master | New pull request | Create new file | Upload files | Find File | Clone or download

eriklindernoren Merge pull request #85 from andrearama/master ... Latest commit 44d3320 28 days ago

aae	Clean up in training loop	a year ago
acgan	changed the discriminator objective function for ACGAN	7 months ago
assets	WGAN (GP): Resolves #37. + clean up of handling input shapes of laten...	a year ago
bgan	Update bgan.py	9 months ago
bigan	Clean up in training loop	a year ago
ccgan	updated instance normalization import	2 months ago
cgan	Update cgan.py	9 months ago
cogan	Update cogan.py	9 months ago
context_encoder	Update context_encoder.py	5 months ago
cyclegan	updated instance normalization import	2 months ago
dcgan	Fix input shape of generator	9 months ago
discogan	updated instance normalization import	2 months ago
dualgan	Clean up in training loop	a year ago
gan	removed hard-coded instances of self.latent_dim = 100	11 months ago
infogan	Clean up in training loop	a year ago
lsgan	Update lsgan.py	9 months ago
pix2pix	updated instance normalization import	2 months ago
pixelda	updated instance normalization import	2 months ago
sgan	same fix for sgan	2 months ago
srgan	updated instance normalization import	2 months ago
wgan	Fix image rescaling to [0,1]	2 months ago
wgan_gp	Fix image rescaling to [0,1]	2 months ago
.gitignore	Logo	a year ago
LICENSE	Initial commit	2 years ago
README.md	Update README.md	a year ago
requirements.txt	cleaned up requirements.txt	11 months ago

README.md

Keras-GAN

Collection of Keras implementations of Generative Adversarial Networks (GANs) suggested in research papers. These models are in some cases simplified versions of the ones ultimately described in the papers, but I have chosen to focus on getting the core ideas covered instead of getting every layer configuration right. Contributions and suggestions of GAN varieties to implement are very welcomed.

See also: PyTorch-GAN

Table of Contents

Installation

```
$ git clone https://github.com/eriklindernoren/Keras-GAN
$ cd Keras-GAN/
$ sudo pip3 install -r requirements.txt
```

Implementations

AC-GAN

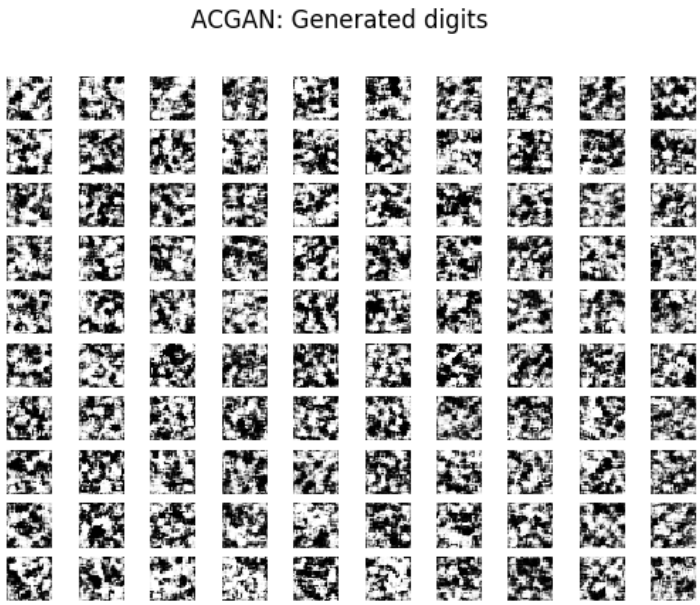
Implementation of *Auxiliary Classifier Generative Adversarial Network*.

Code

Paper: https://arxiv.org/abs/1610.09585

Example

```
$ cd acgan/
$ python3 acgan.py
```



Adversarial Autoencoder

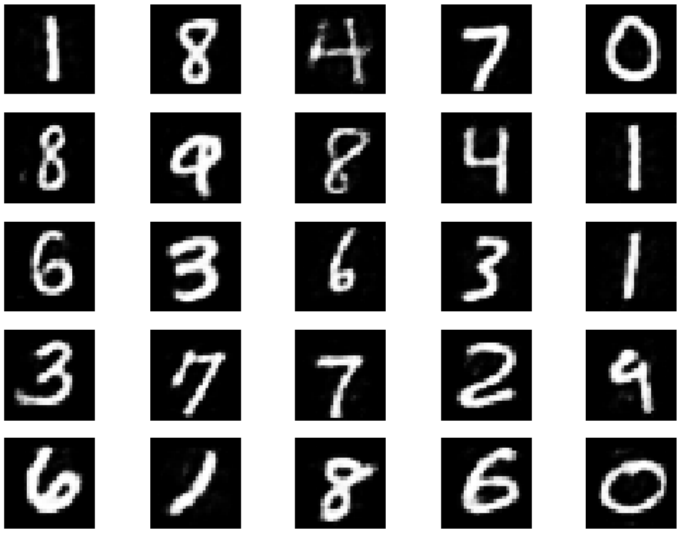
Implementation of *Adversarial Autoencoder*.

Code

Paper: https://arxiv.org/abs/1511.05644

Example

```
$ cd aae/
$ python3 aae.py
```



BiGAN

Implementation of *Bidirectional Generative Adversarial Network*.

Code

Paper: https://arxiv.org/abs/1605.09782

Example

```
$ cd bigan/
$ python3 bigan.py
```

BGAN

Implementation of *Boundary-Seeking Generative Adversarial Networks*.

Code

Paper: https://arxiv.org/abs/1702.08431

Example

```
$ cd bgan/
$ python3 bgan.py
```

CC-GAN

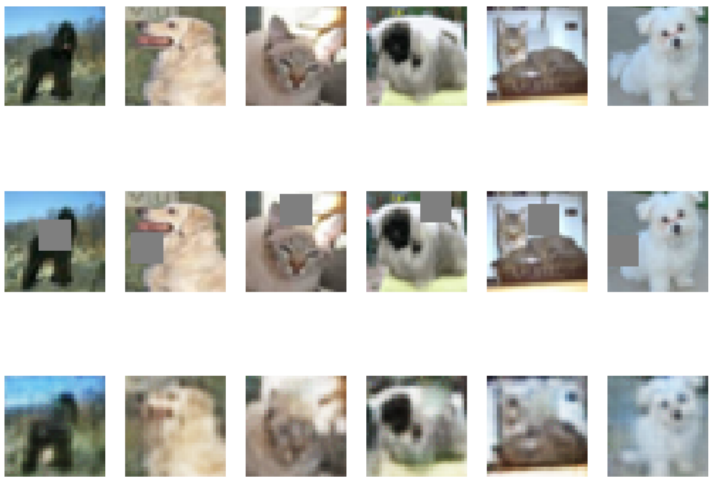
Implementation of *Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks*.

Code

Paper: https://arxiv.org/abs/1611.06430

Example

```
$ cd ccgan/
$ python3 ccgan.py
```



CGAN

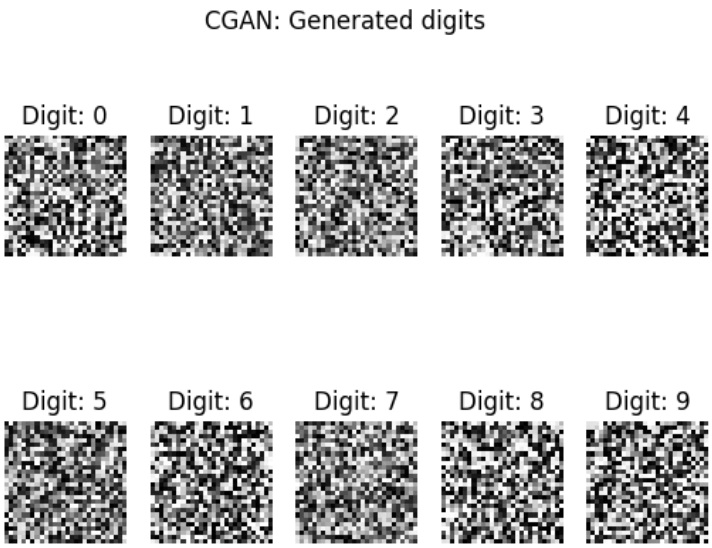
Implementation of *Conditional Generative Adversarial Nets*.

Code

Paper:https://arxiv.org/abs/1411.1784

Example

```
$ cd cgan/
$ python3 cgan.py
```



Context Encoder

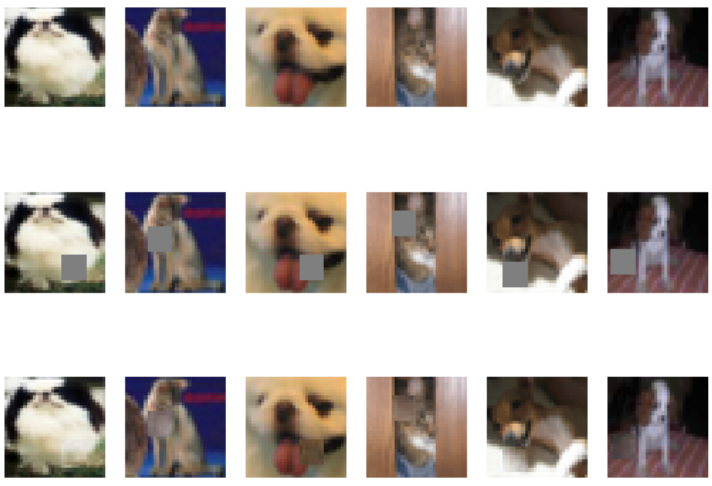
Implementation of *Context Encoders: Feature Learning by Inpainting*.

Code

Paper: https://arxiv.org/abs/1604.07379

Example

```
$ cd context_encoder/
$ python3 context_encoder.py
```



CoGAN

Implementation of *Coupled generative adversarial networks*.

Code

Paper: https://arxiv.org/abs/1606.07536

Example

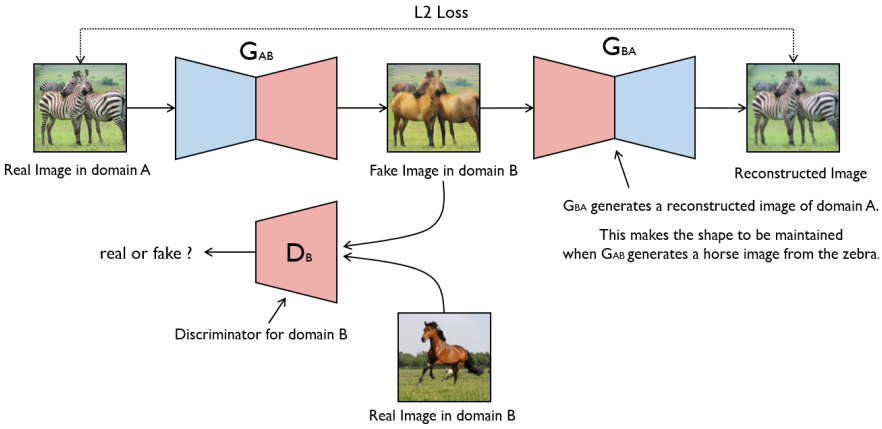
```
$ cd cogan/
$ python3 cogan.py
```

CycleGAN

Implementation of *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*.

Code

Paper: https://arxiv.org/abs/1703.10593



Example

```
$ cd cyclegan/
$ bash download_dataset.sh apple2orange
$ python3 cyclegan.py
```

DCGAN

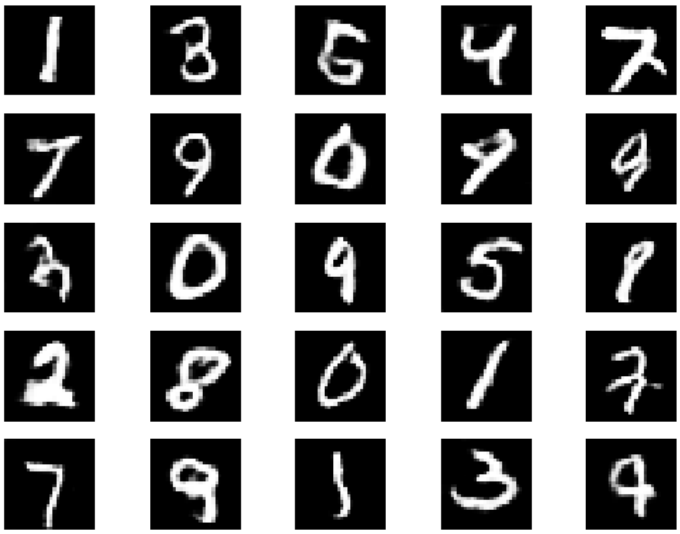
Implementation of *Deep Convolutional Generative Adversarial Network*.

Code

Paper: https://arxiv.org/abs/1511.06434

Example

```
$ cd dcgan/
$ python3 dcgan.py
```

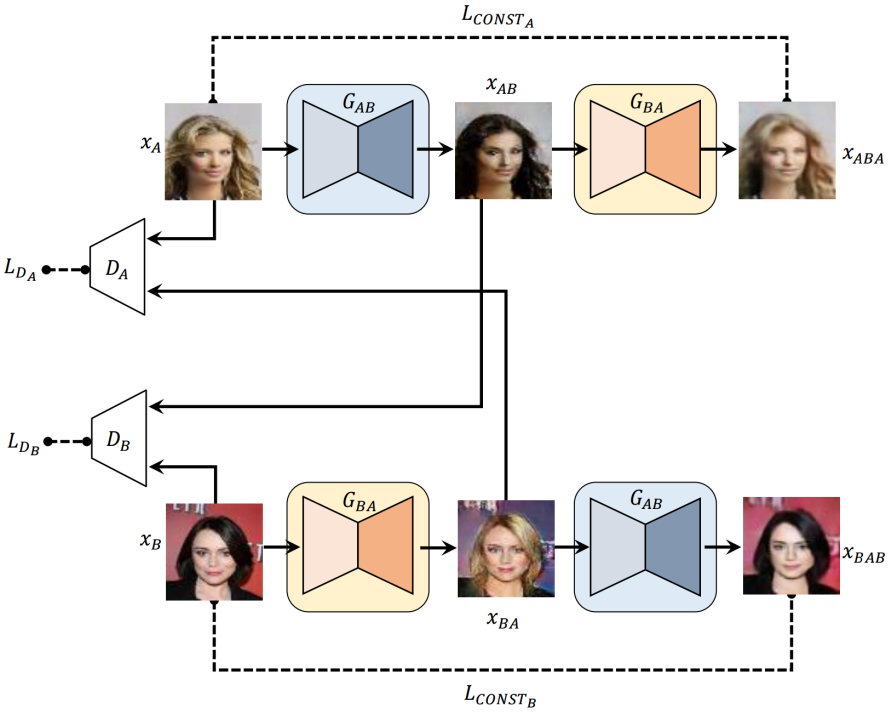


DiscoGAN

Implementation of *Learning to Discover Cross-Domain Relations with Generative Adversarial Networks*.

Code

Paper: https://arxiv.org/abs/1703.05192



Example

```
$ cd discogan/
$ bash download_dataset.sh edges2shoes
$ python3 discogan.py
```

DualGAN

Implementation of *DualGAN: Unsupervised Dual Learning for Image-to-Image Translation*.

Code

Paper: https://arxiv.org/abs/1704.02510

Example

```
$ cd dualgan/
$ python3 dualgan.py
```

GAN

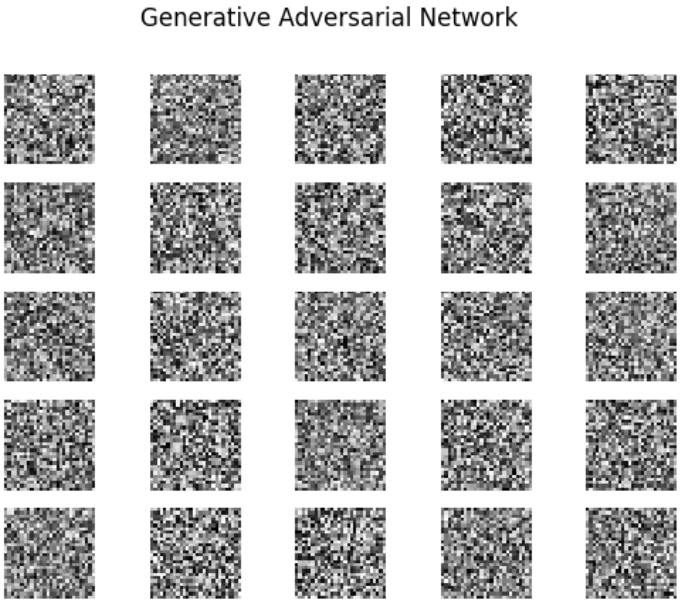
Implementation of *Generative Adversarial Network* with a MLP generator and discriminator.

Code

Paper: https://arxiv.org/abs/1406.2661

Example

```
$ cd gan/
$ python3 gan.py
```



InfoGAN

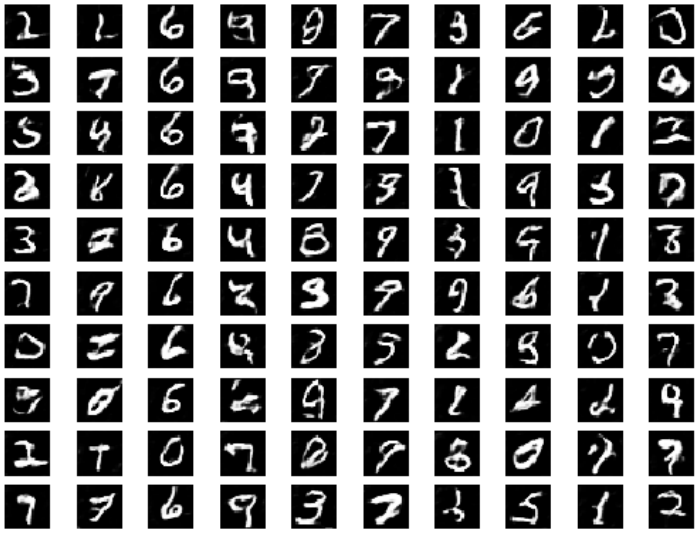
Implementation of *InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets*.

Code

Paper: https://arxiv.org/abs/1606.03657

Example

```
$ cd infogan/
$ python3 infogan.py
```



LSGAN

Implementation of *Least Squares Generative Adversarial Networks*.

Code

Paper: https://arxiv.org/abs/1611.04076

Example

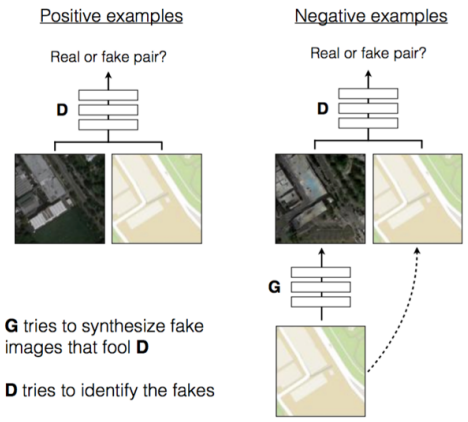
```
$ cd lsgan/
$ python3 lsgan.py
```

Pix2Pix

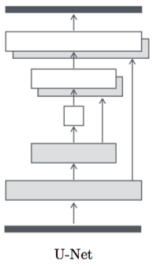
Implementation of *Image-to-Image Translation with Conditional Adversarial Networks*.

Code

Paper: https://arxiv.org/abs/1611.07004



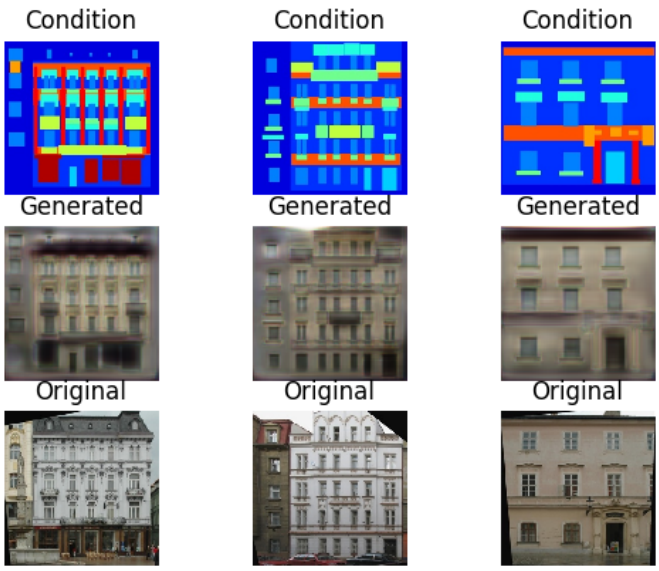
Discriminator



Generator

Example

```
$ cd pix2pix/
$ bash download_dataset.sh facades
$ python3 pix2pix.py
```



PixelDA

Implementation of *Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks*.

Code

Paper: https://arxiv.org/abs/1612.05424

MNIST to MNIST-M Classification

Trains a classifier on MNIST images that are translated to resemble MNIST-M (by performing unsupervised image-to-image domain adaptation). This model is compared to the naive solution of training a classifier on MNIST and evaluating it on MNIST-M. The naive model manages a 55% classification accuracy on MNIST-M while the one trained during domain adaptation gets a 95% classification accuracy.

```
$ cd pixelda/
$ python3 pixelda.py
```

Method	Accuracy
Naive	55%
PixelDA	95%

SGAN

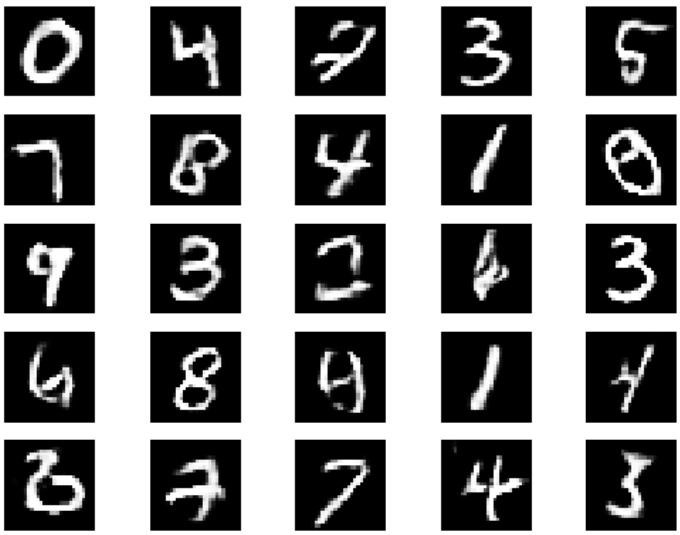
Implementation of *Semi-Supervised Generative Adversarial Network*.

Code

Paper: https://arxiv.org/abs/1606.01583

Example

```
$ cd sgan/
$ python3 sgan.py
```

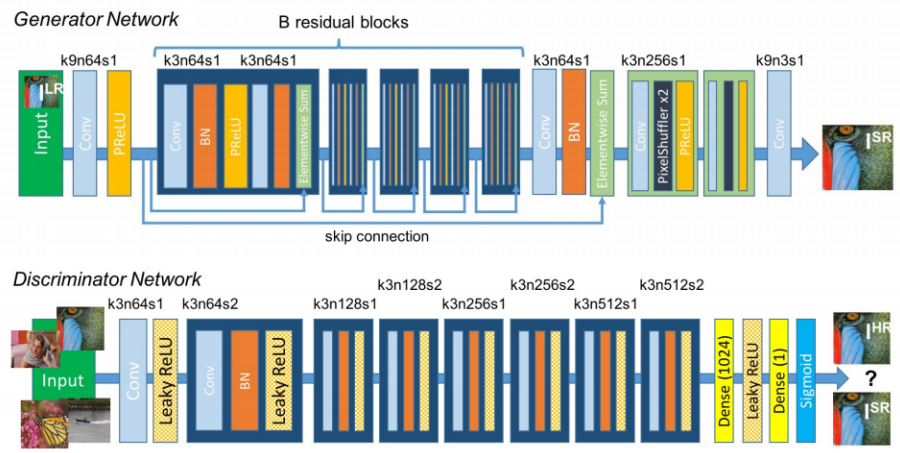


SRGAN

Implementation of *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*.

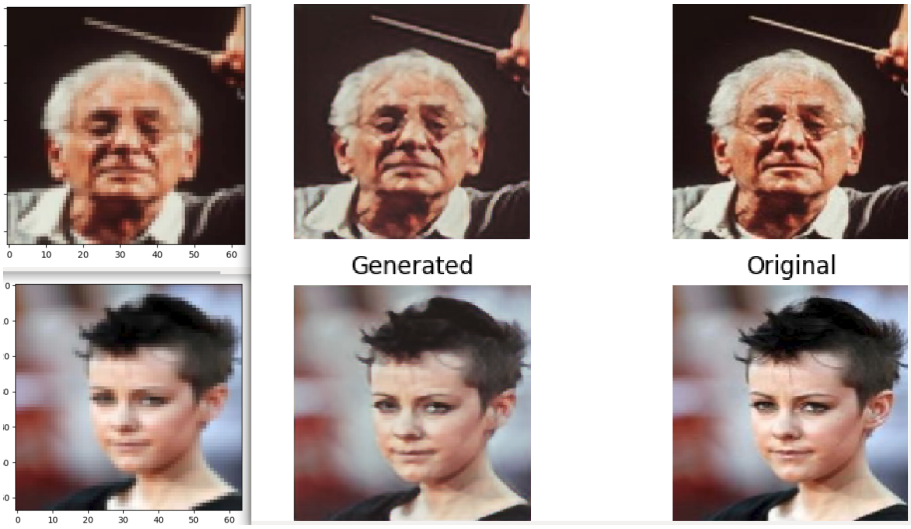
Code

Paper: https://arxiv.org/abs/1609.04802



Example

```
$ cd srgan/
<follow steps at the top of srgan.py>
$ python3 srgan.py
```



WGAN

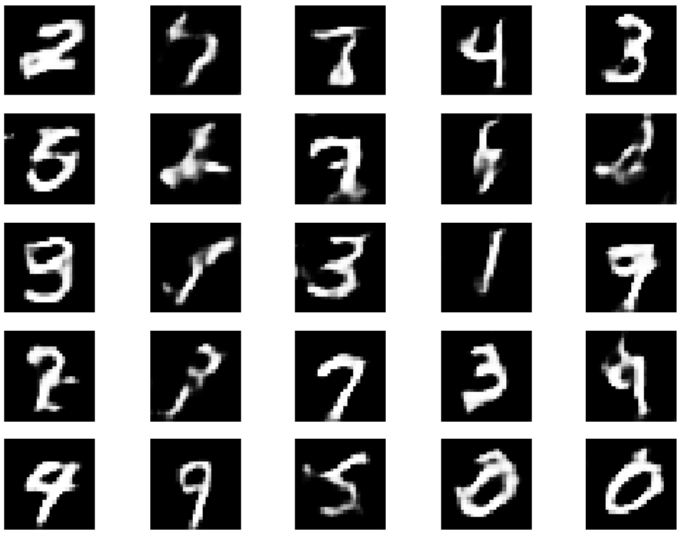
Implementation of *Wasserstein GAN* (with DCGAN generator and discriminator).

Code

Paper: https://arxiv.org/abs/1701.07875

Example

```
$ cd wgan/
$ python3 wgan.py
```



WGAN GP

Implementation of *Improved Training of Wasserstein GANs*.

Code

Paper: https://arxiv.org/abs/1704.00028

Example

```
$ cd wgan_gp/
$ python3 wgan_gp.py
```