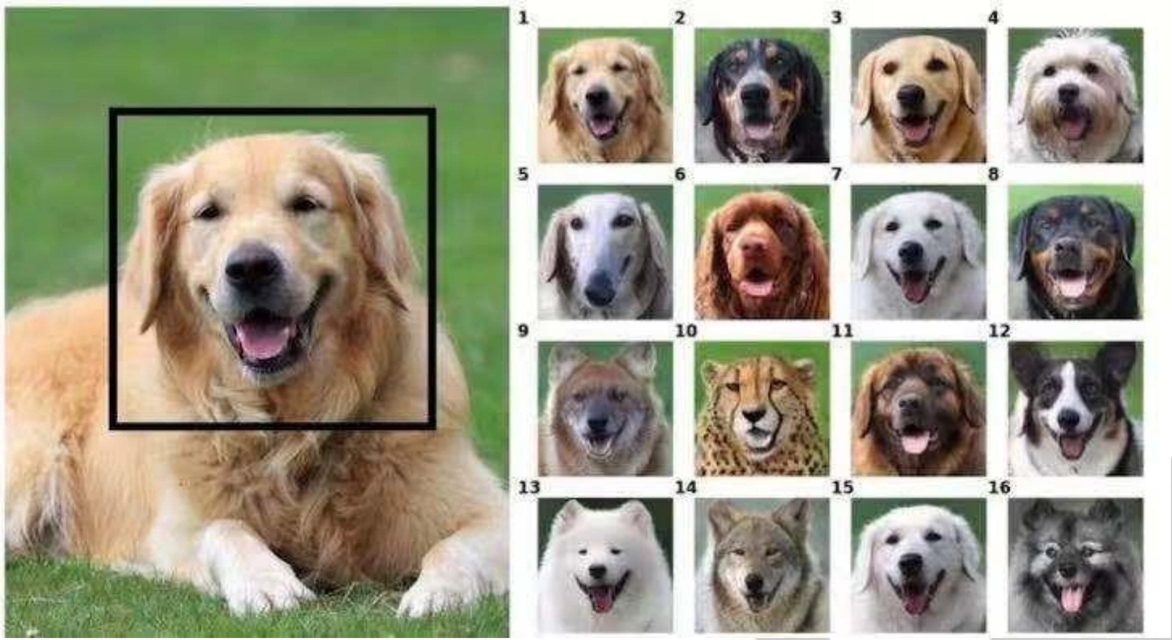
Generate Dog Images with Generative Adversarial Networks (GAN)



Outline

Introduction of GAN

Data

Description Model

Description

Results

Potential Improvement

Introduction of GAN

Application of GAN in the Real World

Snap Chat Babyface Filter

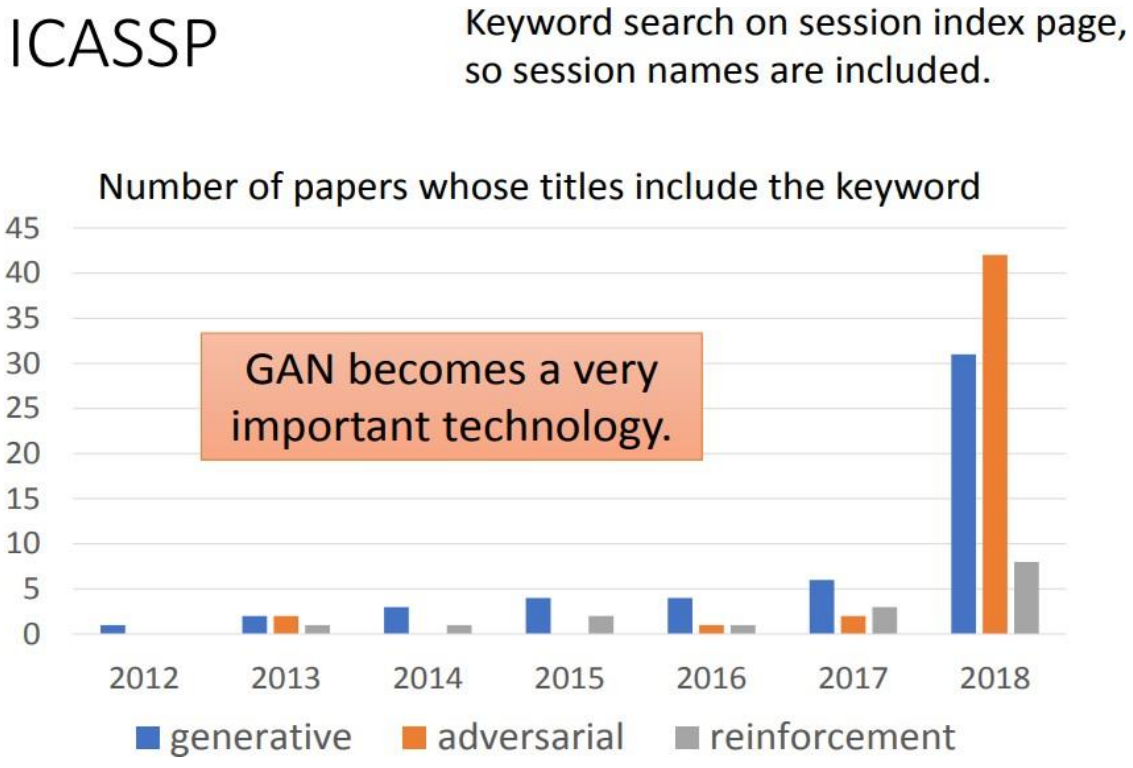
Shakespearean poetry generator

Random Music generator

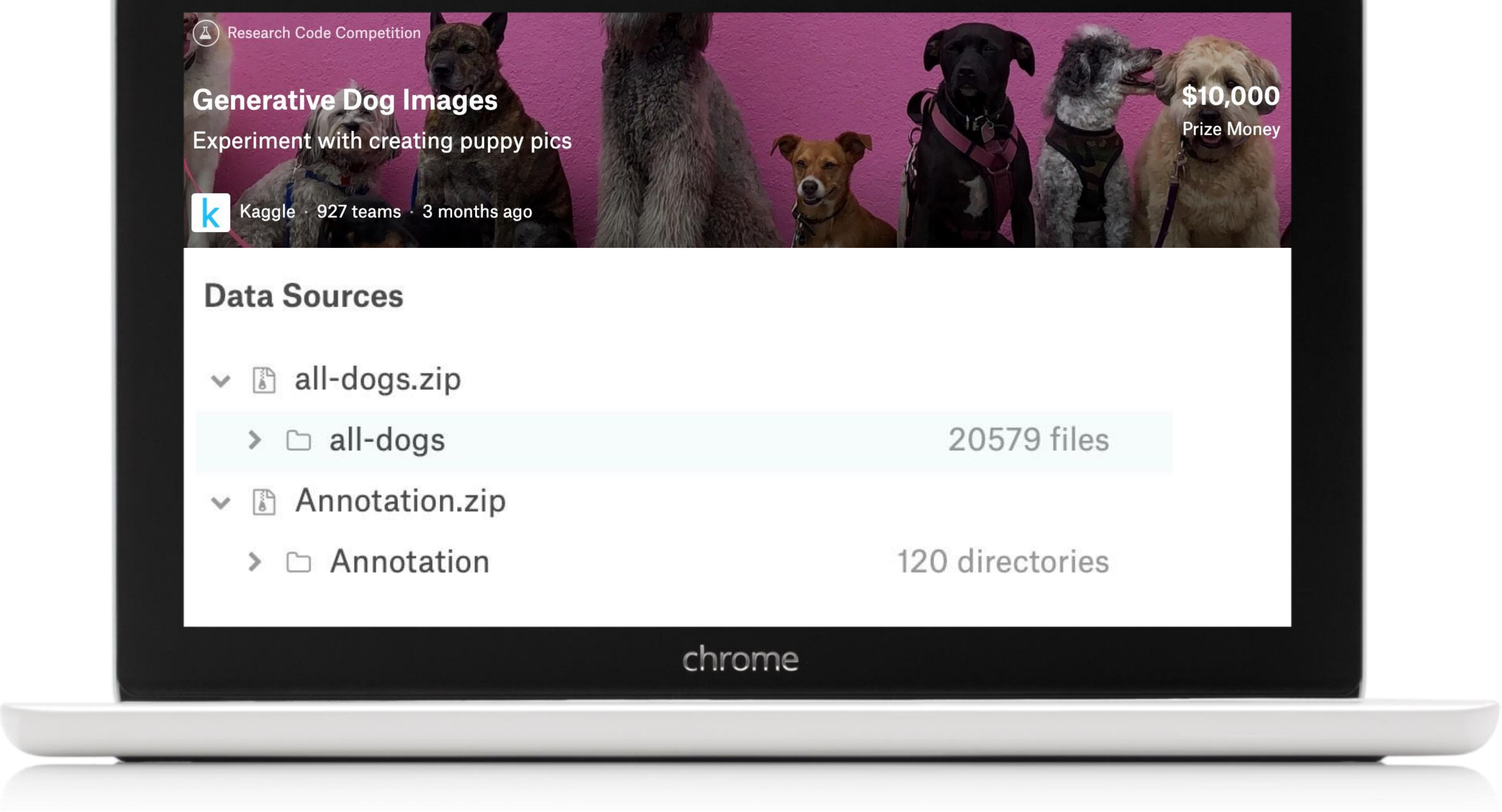
Introduction of GAN

Background

GAN is a very new stuff and has a promising future. Especially in last few years, GAN was developed with an exponential increment. In other words, it is almost an



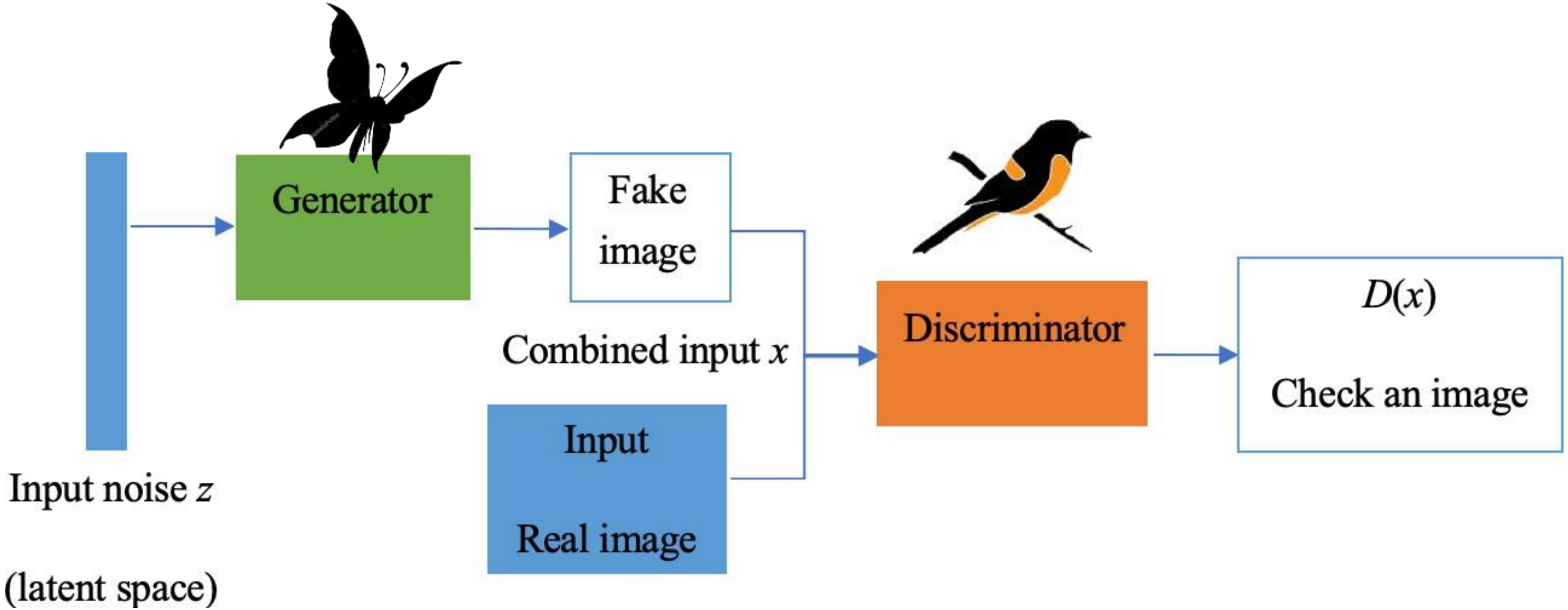
Data Source


Research Code Competition
Generative Dog Images
Experiment with creating puppy pics
$10,000
Prize Money
Kaggle · 927 teams · 3 months ago
Data Sources
all-dogs.zip
all-dogs
20579 files
Annotation.zip
Annotation
120 directories
chrome

Data Description

	Dog Breed	Number of Observations
0	n02085620-Chihuahua	152
1	n02085782-Japanese_spaniel	185
2	n02085936-Maltese_dog	252
3	n02086079-Pekinese	149
4	n02086240-Shih-Tzu	214
.	.	.
.	.	.
.	.	.
116	n02113978-Mexican_hairless	155
117	n02115641-dingo	156
118	n02115913-dhole	150
119	n02116738-African_hunting_dog	169

Description of Models - The Basic Concept

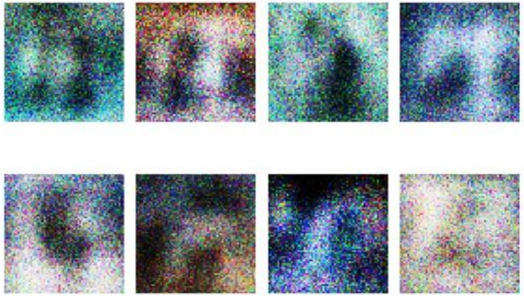


Structure of GAN

Input random vectors into generator to generate fake image. Discriminator is responsible to classify the fake and real image. As the discriminator becomes stricter, the generator must generate more realistic image to cheat the discriminator.

Simple GAN with MLP

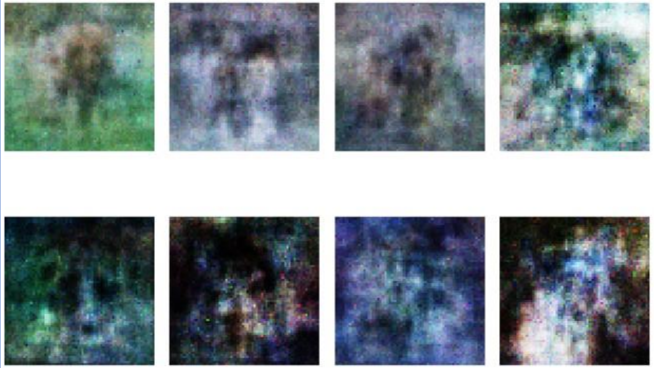
- After 40 epochs



Simple GAN with MLP: Problems

Imbalance between Generator and Discriminator.

- Even with larger epochs, the generated images are still blurred. → Stop learning anymore.
- Generator always cannot compete with Discriminator.
  - Intuitively, creativity is more difficult than criticism. In fact, it tends to be easy to distinguish an artwork is real or fake. However, without seeing the real artwork, it is really hard to create a fake artwork which looks just like the real one.
- Mathematically, the gradient descent of Generator will be vanishing.

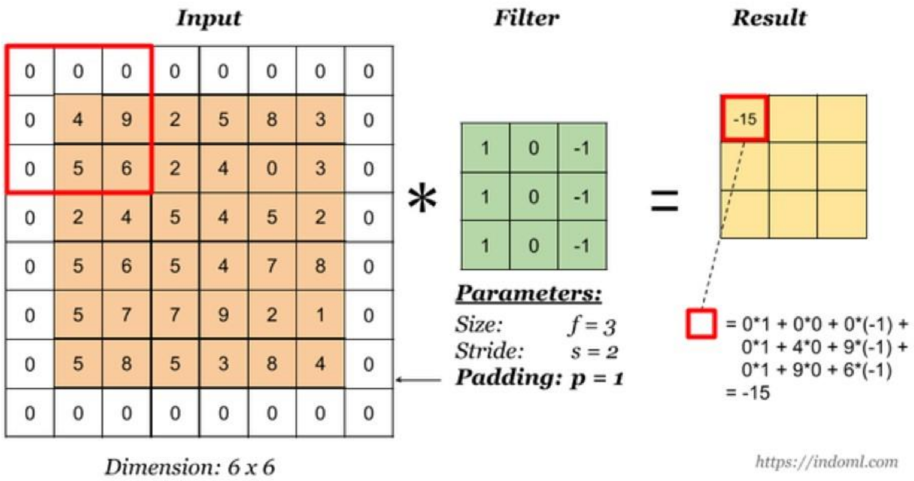


Potential solutions

- A trivial approach is adjusting training times of Generator and Discriminator separately.
  - In practice, it helps a little bit, but it also makes the training process more unstable.
- MLP based Generator cannot focus on the detail features of an image. → Create a deeper Generator structure with convolution layers.
  - Deep Convolution GAN (DCGAN)
- Some other approaches
  - spectral normalization.
  - finding a loss function and an activation function with stable and non-vanishing gradients.

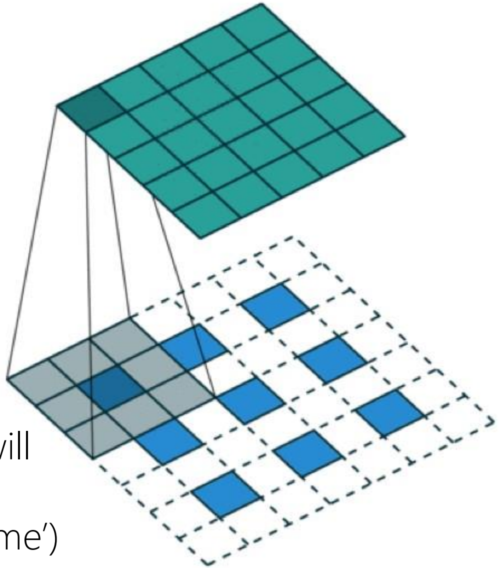
DCGAN - Upsampling & Downsampling

- To construct a DCGAN
  - Generator → Upsampling
  - Discriminator → Downsampling
- Downsampling
  - MLP: Diminishing number of neurons
  - Convolution (with stride)

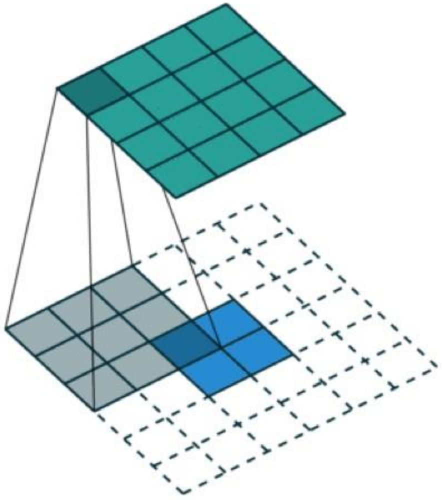


DCGAN - Upsampling & Downsampling

- Upsampling
  - MLP: Increasing number of neurons
  - Transposed convolution (with stride)
    - Pretty same as convolution
    - Stride concept is transposed
    - Padding concept is transposed
    - E.g. If Conv2D(stride=(2, 2), padding='same') will reduce a size from 6x6 to 3x3. TransposedConv2D(stride=(2, 2), padding='same') will increase a size from 3x3 to 6x6.



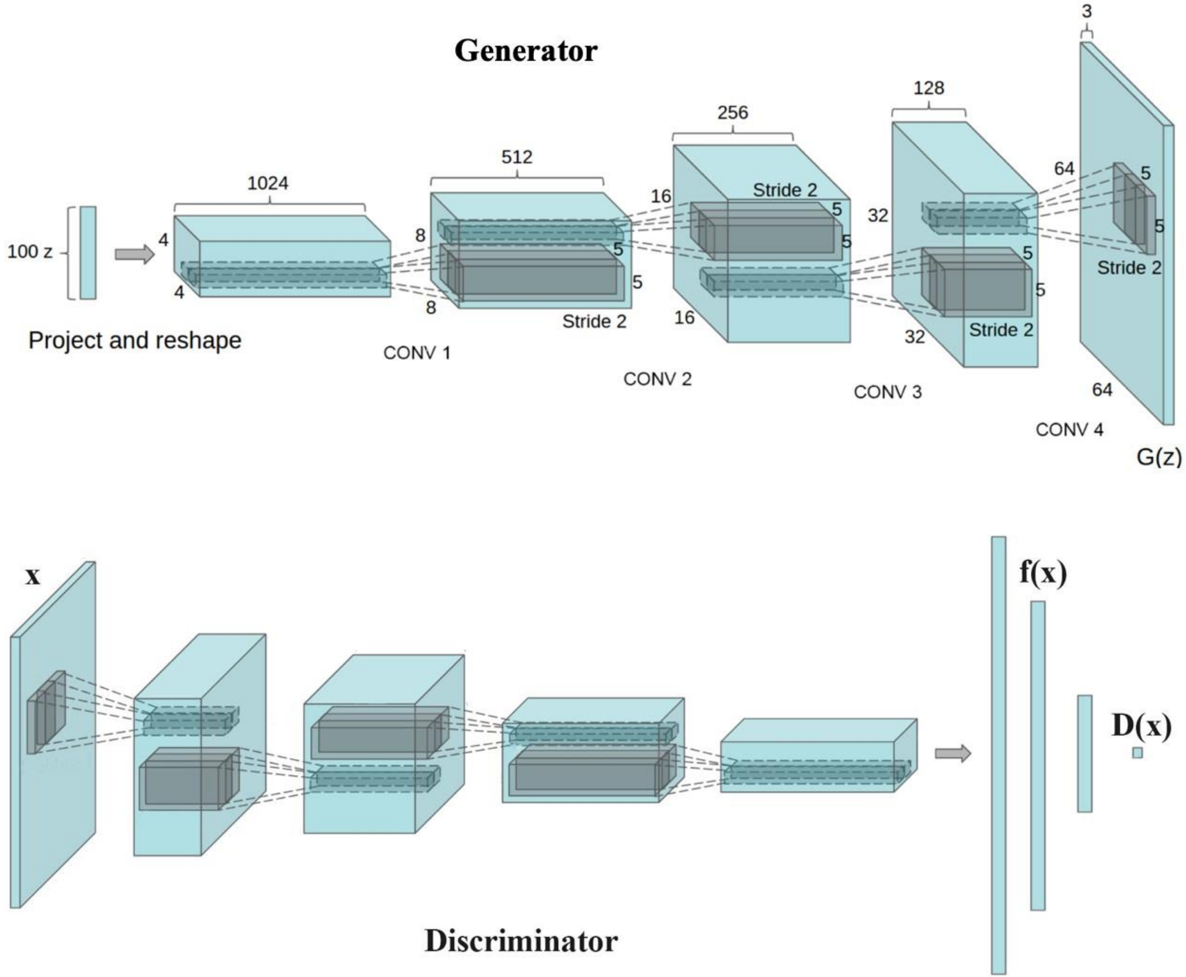
Stride(2, 2), Padding=1



Stride(1, 1), Padding = 0

DCGAN Construction

- Upsampling
  - Transposed convolution
- Downsampling
  - Replace all max pooling with convolutional stride.
- Activation
  - LeakyReLU



Hard to tune

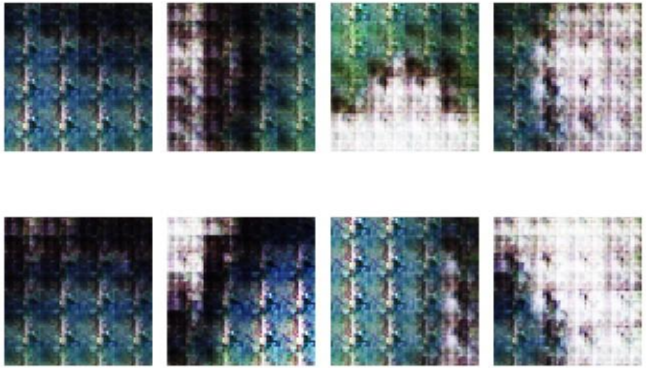
- High sensitivity to hyperparameter. Even the performance of GANs varies with different random seeds. Tuning a GAN should be very patient.

- Solutions. Looking at the gradients and loss changes is the most efficient way to help with tuning. After that, the only thing we should keep is our patience.

DCGAN - Not well-tuned

- Comparing with MLP structure, a well-tuned deep convolution structure helps the networks recognize the more details inside an image. However, if the convolution structure is not tuned well, the result will be worse than simple GAN with MLP. With the operation of convolution, the generated images may be cut in blocks.

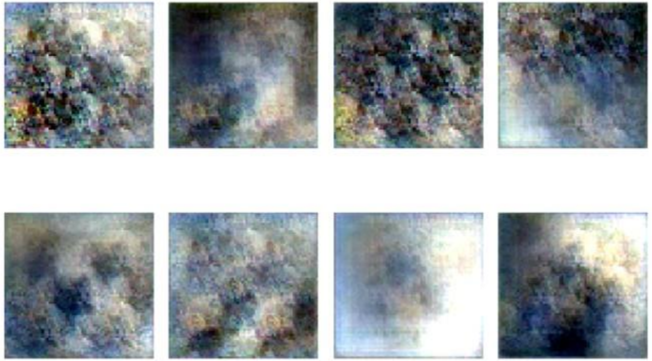
- CNN creates blocks in images



DCGAN - Not well-tuned

- Even if the blocks problem is mitigated, DCGAN may fail to generate target-like images. It means both of Discriminator and Generator are learning in the wrong directions.
- Most generated images look similar. → Mode collapse
  - Although the image quality is improved in some cases, the mode collapse problem is still existed in DCGAN.
  - (mode collapse means when generator produces limited varieties of outputs and often converging to specific modes or patterns,ignoring the diversity present in the training data.

- Fail to learn to generate dog-like images



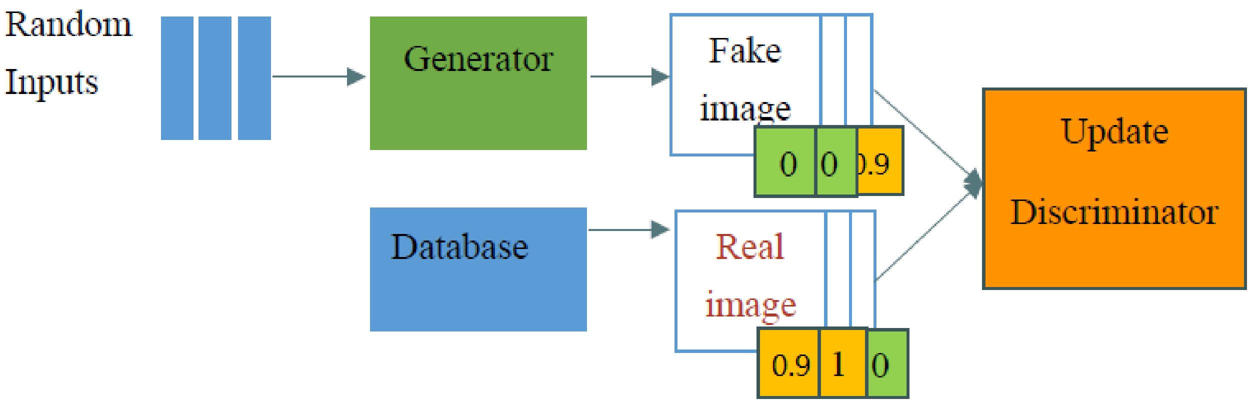
- Model collapse

One trick makes a great improvement

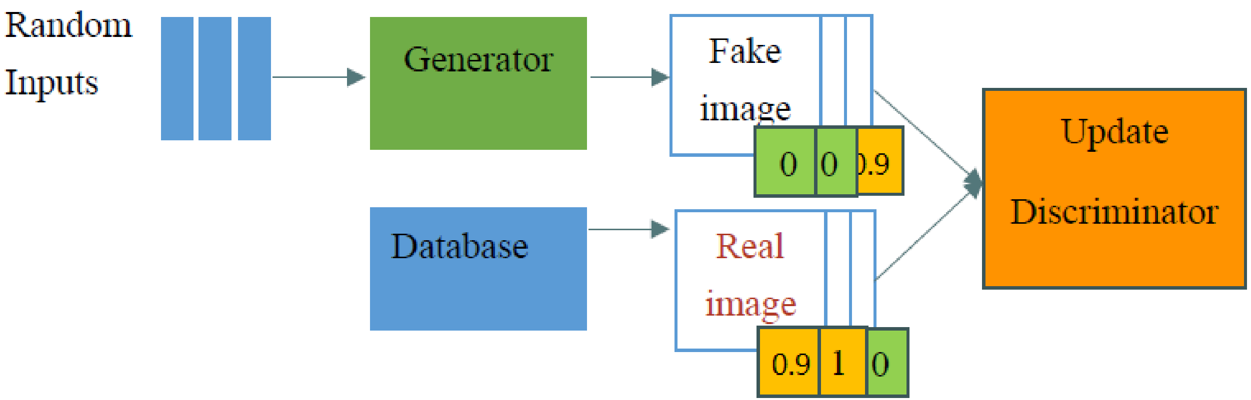
- Using soft and noisy labels can help GANs to be stable a lot. Without such changes, our model cannot create a clear image. Smoothing the positive labels (like 0.9-1.0).

- Soft and noisy labe
  Let Discriminator r

- To be noticed, we
  (Goodfellow, 2016)



One trick ı



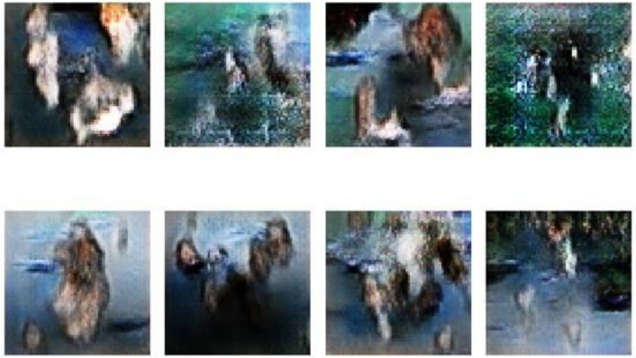
- Using soft and noi
  cannot create a cl

- Soft and noisy labels make a balance between Discriminator and Generator in this competition. Let Discriminator not be over-confident and let Generator not be under-confident.

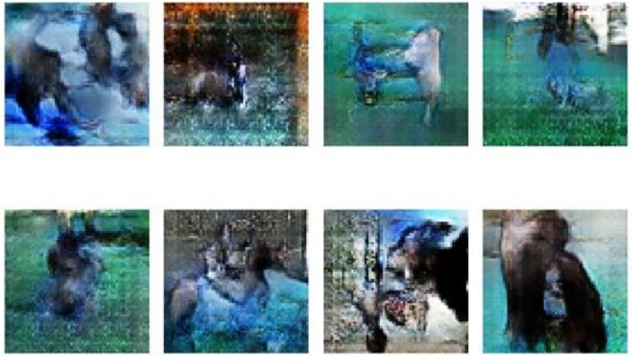
- To be noticed, we should only smooth the one-sided label, particularly the positive labels. (Goodfellow, 2016)

DCGAN - Performance

- In the early training step, it can generate some clear image but not dog-like.
- After 5000 epochs



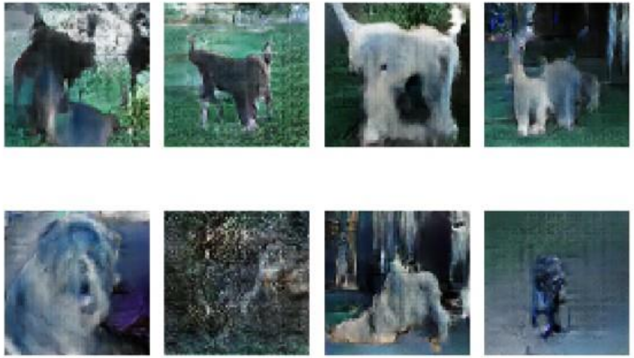
- Some images generated look like a dog, but most of them are still not dog-like.
- After 10000 epochs



DCGAN - Performance

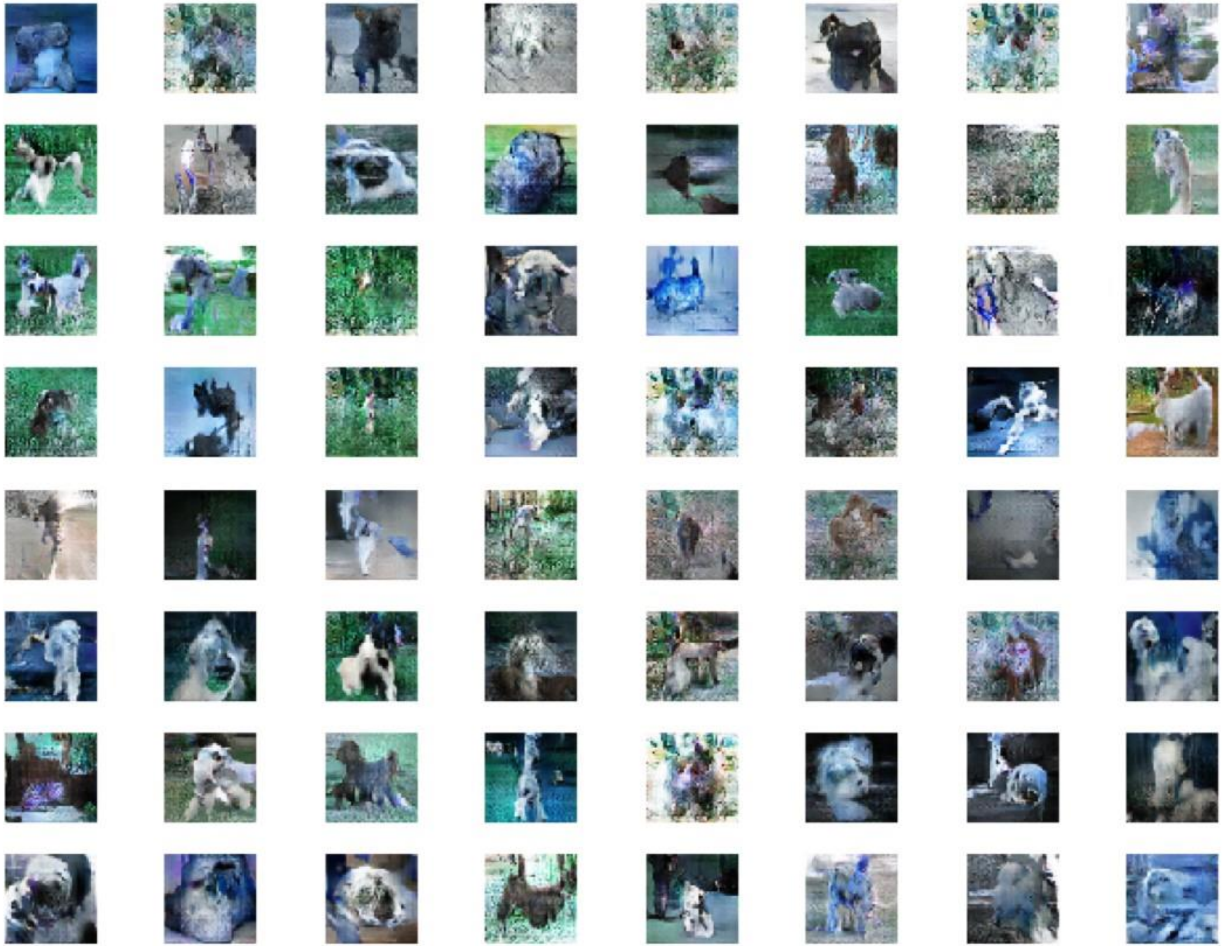
- It is not easy to generate clear dog-like images without mode collapse.
- After 40 epochs

- the improvement is not significant. This DCGAN model almost achieves its maximum performance.
- After 60 epochs



DCGAN

- A glance of the random generated dogs.



Mode collapse

- Issue: Generator only produces low-diversity outputs. A complete mode collapse, which is not common, means the Generator just makes a trick to create only one type of image to fool the Discriminator. A partial mode collapse, which always happens, is a hard problem in GAN to solve.
- Solutions: Mode collapse is still a difficult problem in most GANs. Nevertheless, there are some tricky ways to disperse kind of collapse like Conditional GAN (CGAN). CGAN inputs the label of one or more real data into the model as a condition, so that the model is affected by the label. Here, our dog dataset contains 120 kinds of dogs. Therefore, we try to use CGAN to solve the problem of mode collapse.

Q & A