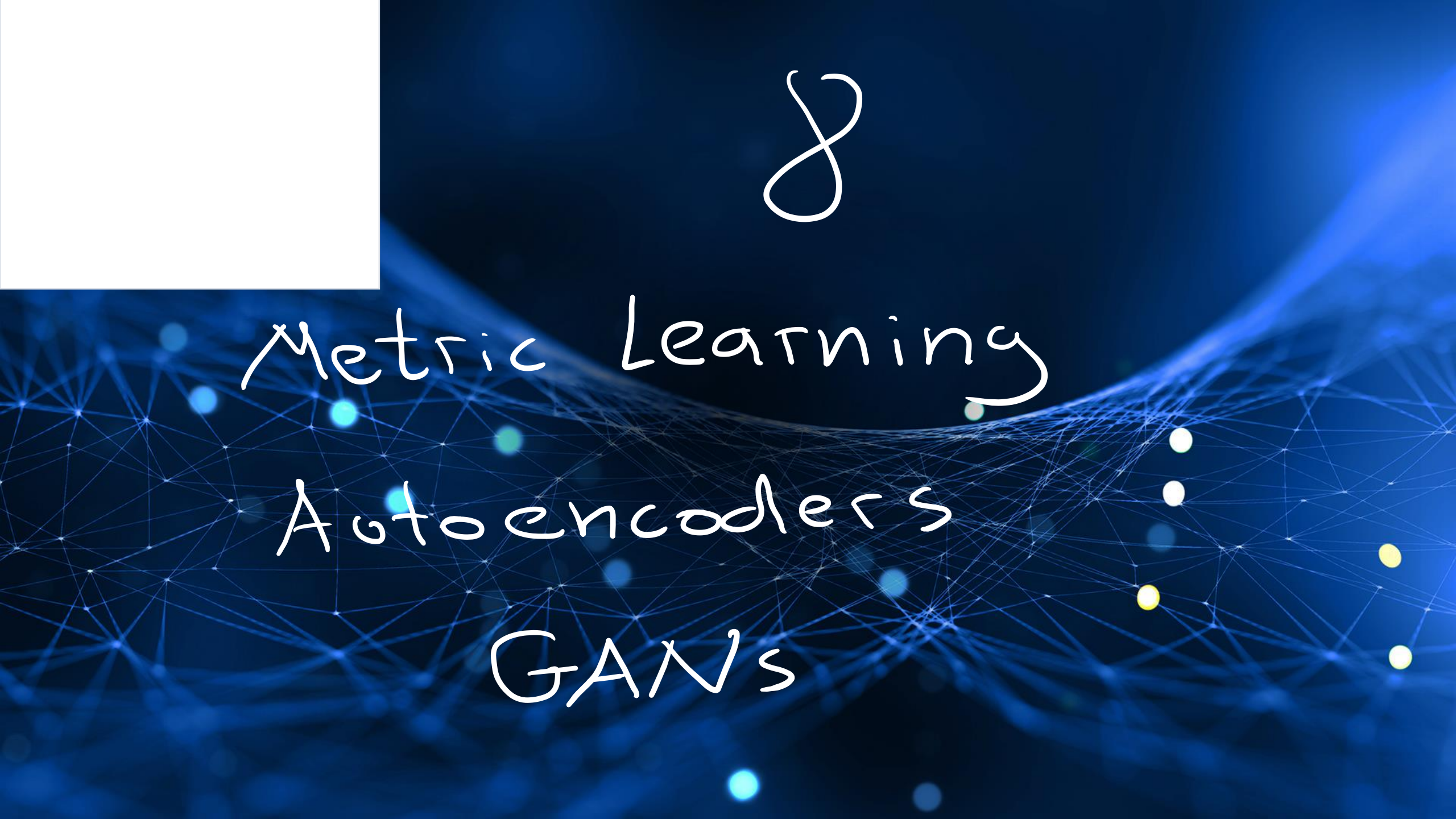
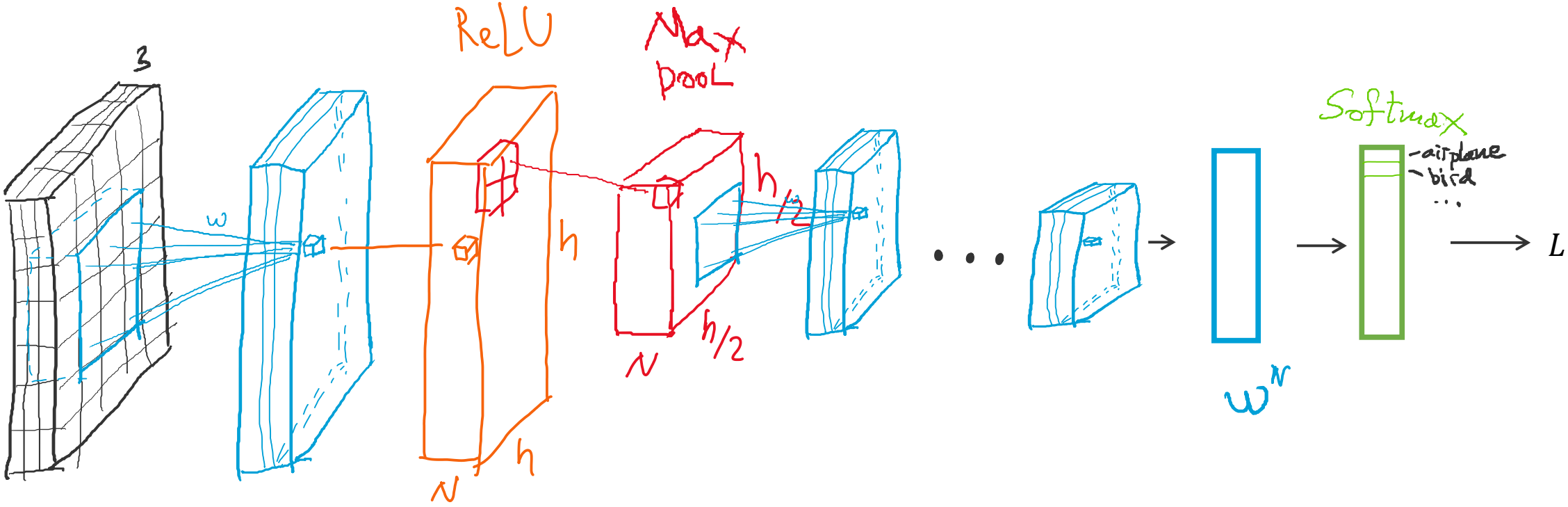
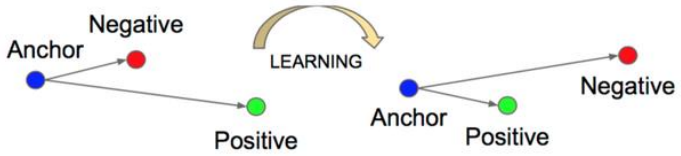

8
Metric Learning
Autoencoders
GANs

Beyond Classification



Metric Learning

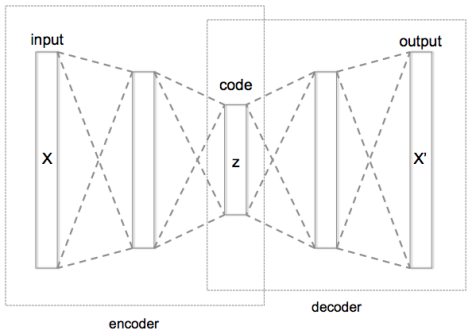
Triplet loss



embedding

Unsupervised learning

Autoencoders



GANs

Latent space

Face recognition

Distractors

1 Million Photos

690,572 Unique Users

Training Set

4.7 Million Photos

672,057 Unique Identities

7 Mean photos / person (3 min, 2469 max)

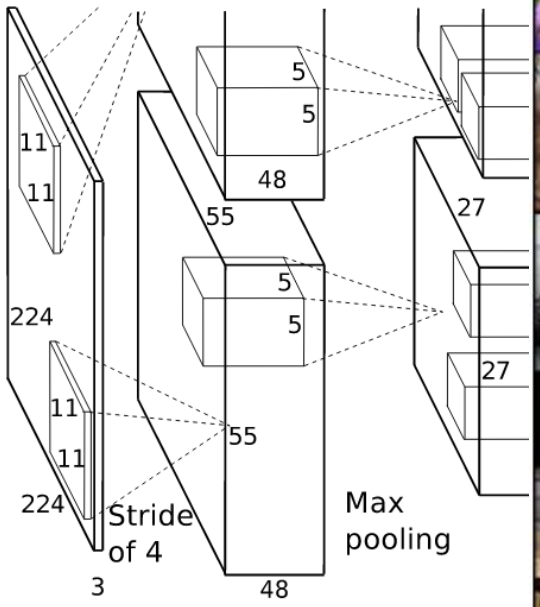
Megaface

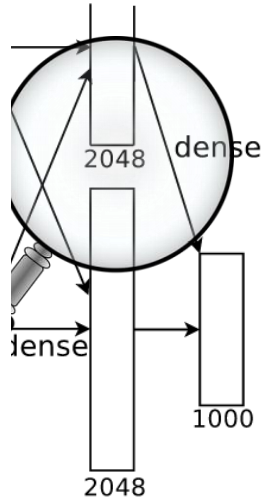
Кто из людей в тренировочных данных изображен на новом фото?

На двух новых фото один и тот же человек или нет?

Даны 3 фото нового человека.
Изображен ли он на еще одном фото?

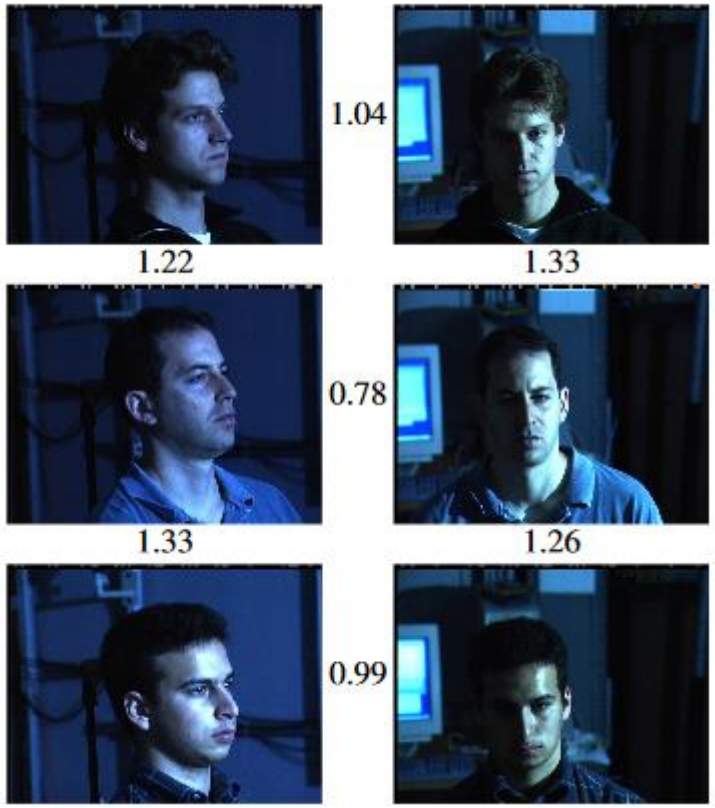
Признаки предпоследнего слоя Embedding

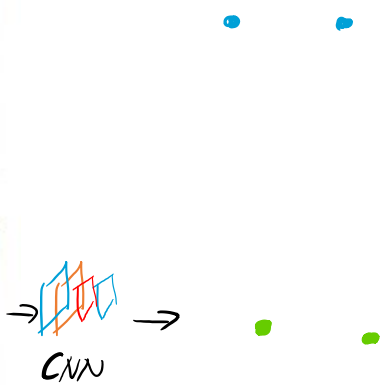




AlexNet’12

Metric Learning





Кто из людей в тренировочных данных изображен на новом фото?

Просто K-nearest neighbor

На двух новых фото один и тот же человек или нет?

Просто отсечка по расстоянию

Даны 3 фото нового человека.
Изображен ли он на еще одном фото?

Добавить новые точки, снова KNN

FaceNet'15

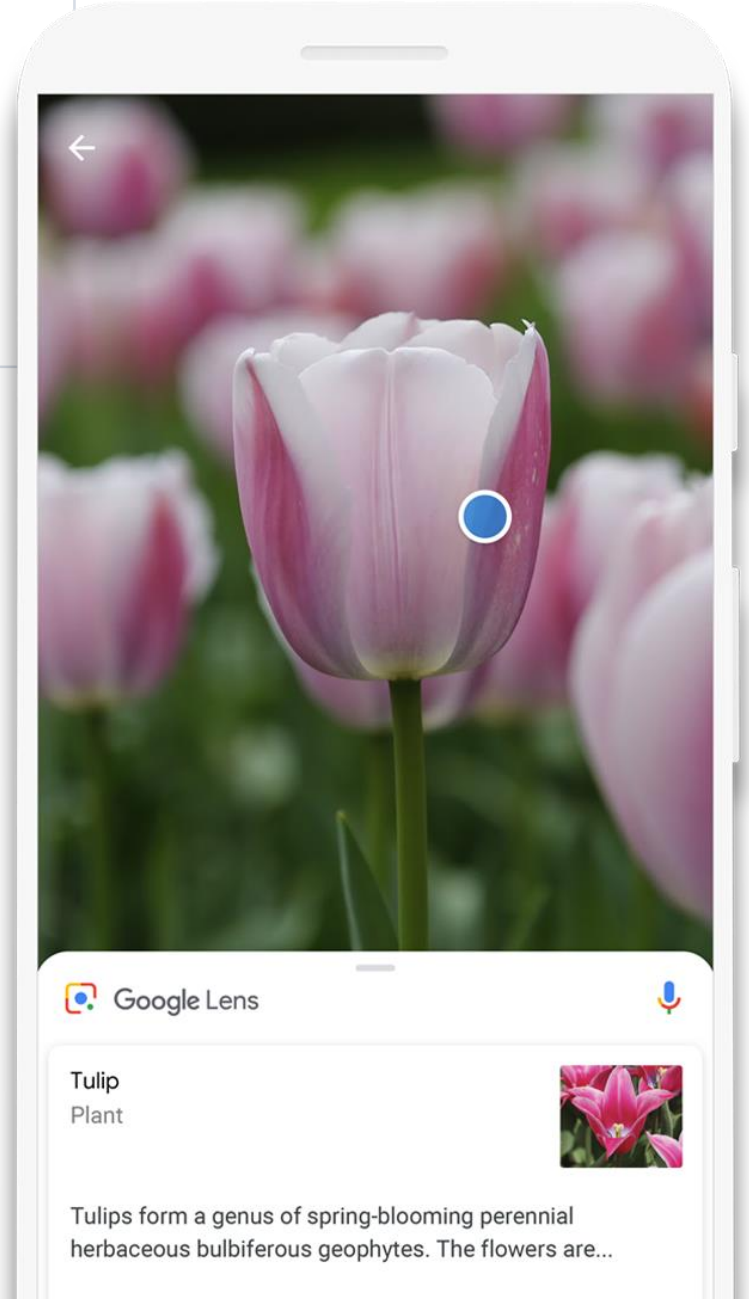
Triplet Loss

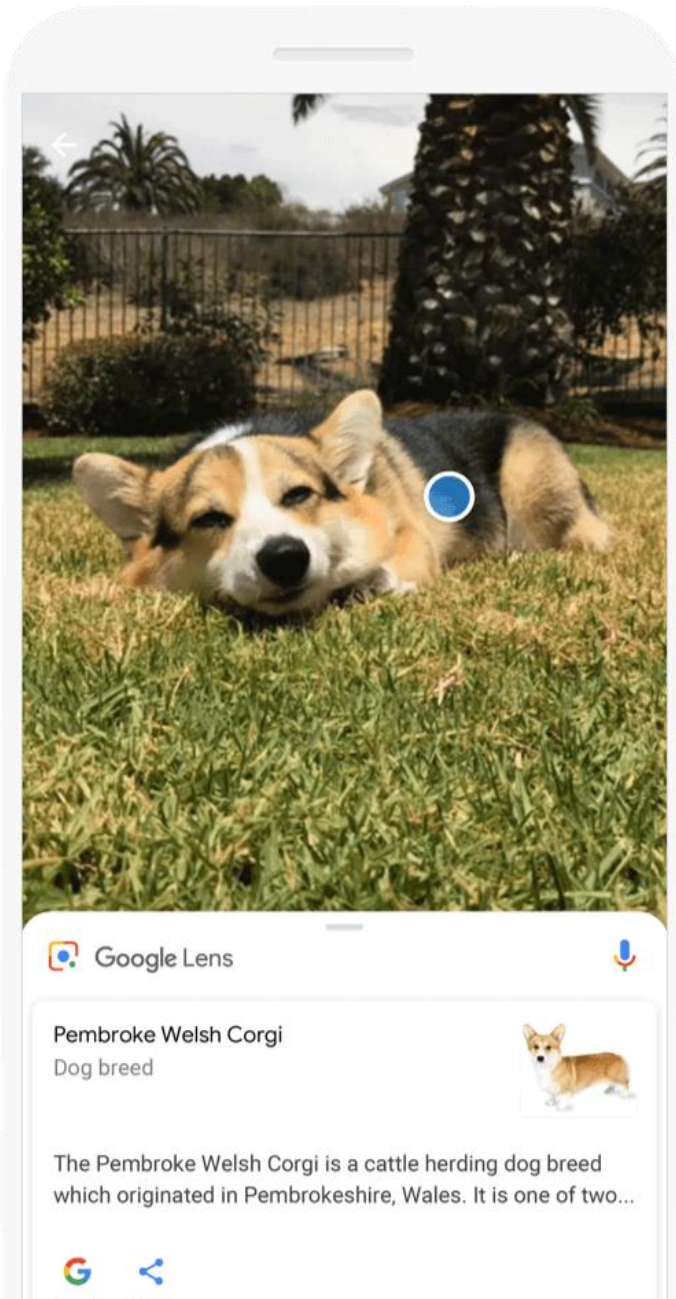
Figure 2. **Model structure.** Our network consists of a batch input layer and a deep CNN followed by L_2 normalization, which results in the face embedding. This is followed by the triplet loss during training.

Figure 3. The **Triplet Loss** minimizes the distance between an *anchor* and a *positive*, both of which have the same identity, and maximizes the distance between the *anchor* and a *negative* of a different identity.

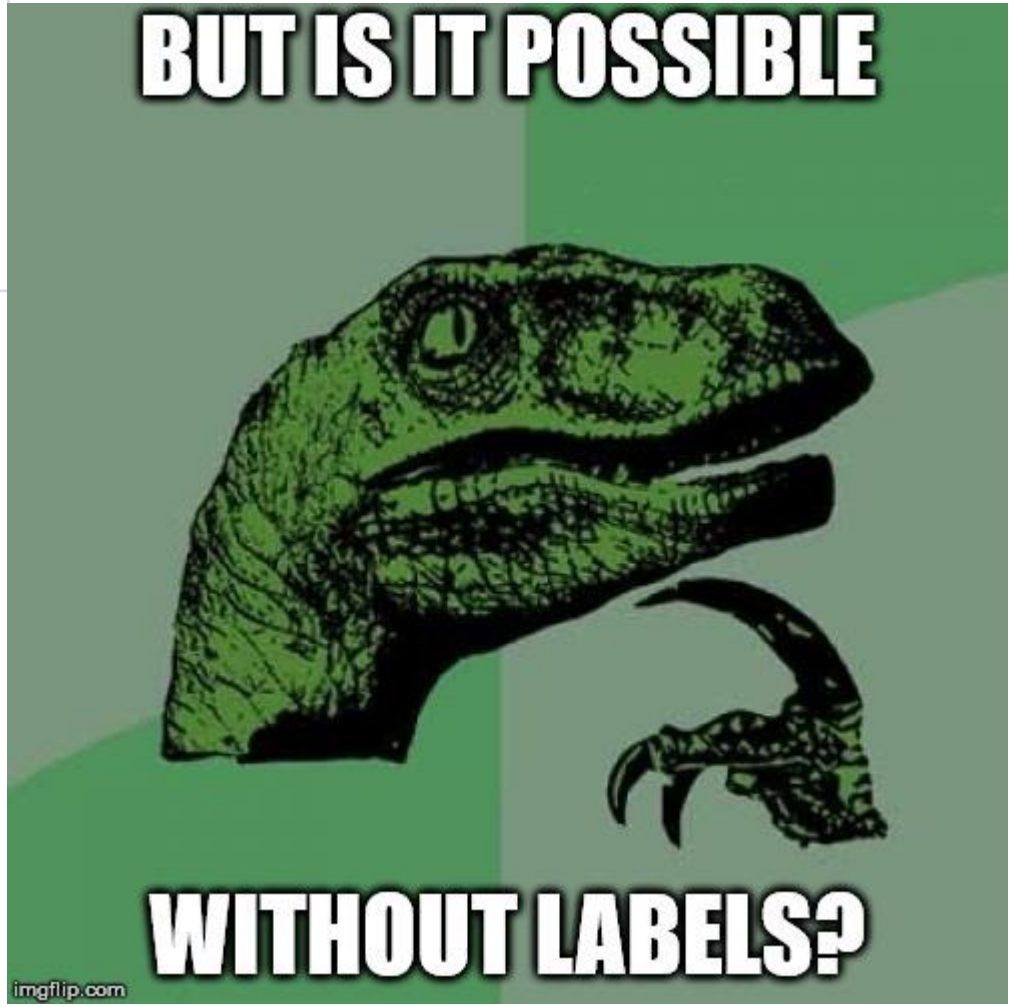
$$L = \sum_{i}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

[FaceNet'15]


Google Lens
Tulip
Plant
Tulips form a genus of spring-blooming perennial herbaceous bulbiferous geophytes. The flowers are...

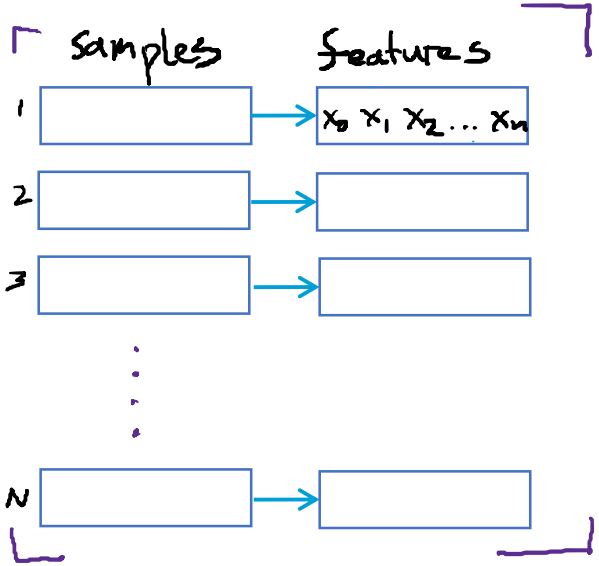

Google Lens
Pembroke Welsh Corgi
Dog breed
The Pembroke Welsh Corgi is a cattle herding dog breed which originated in Pembrokeshire, Wales. It is one of two...

[Google Lens]


BUT IS IT POSSIBLE
WITHOUT LABELS?
imgflip.com

Unsupervised learning

data



НЕТ МЕТОК

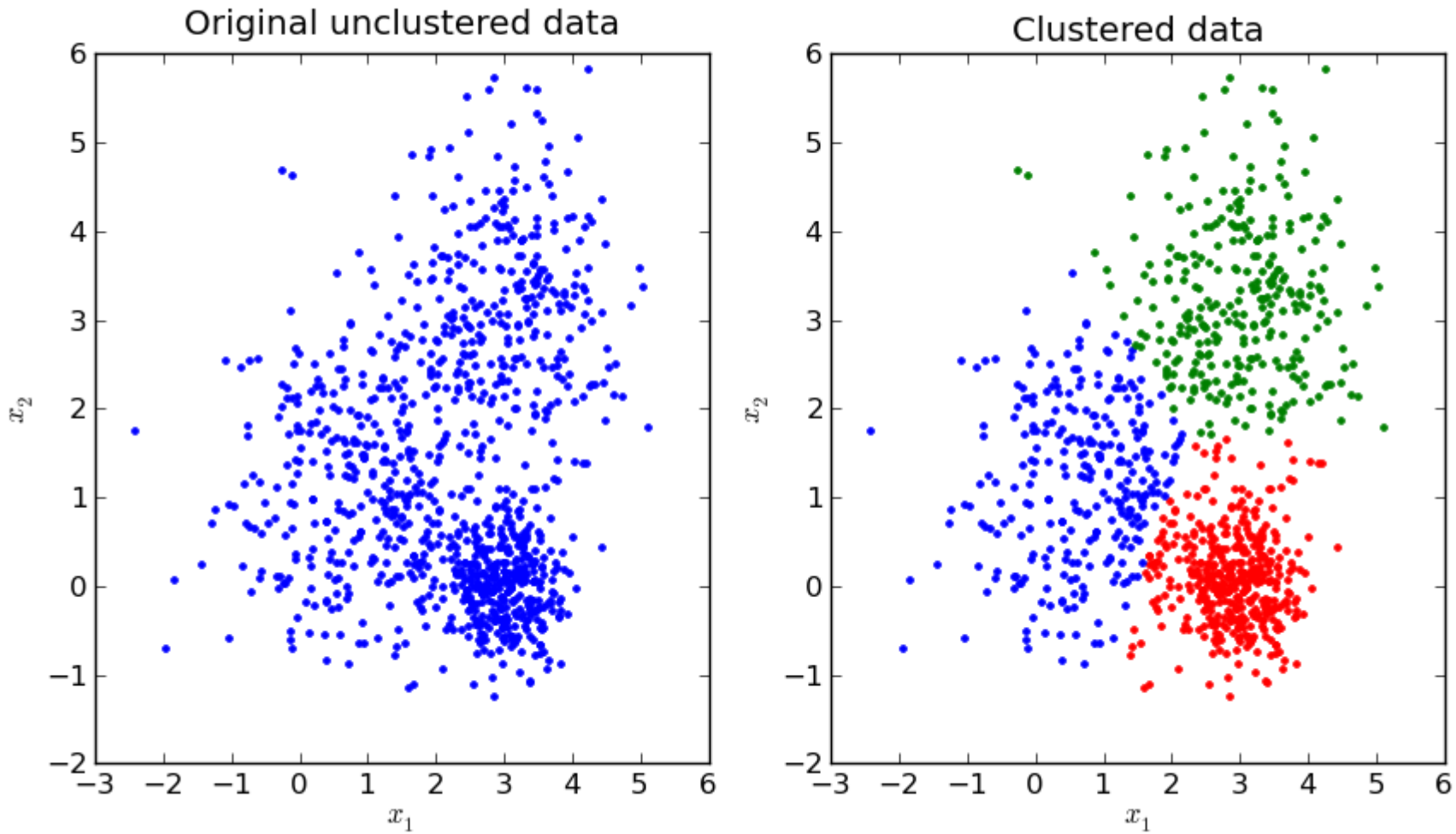
ML algorithm

Model

Structure of the data

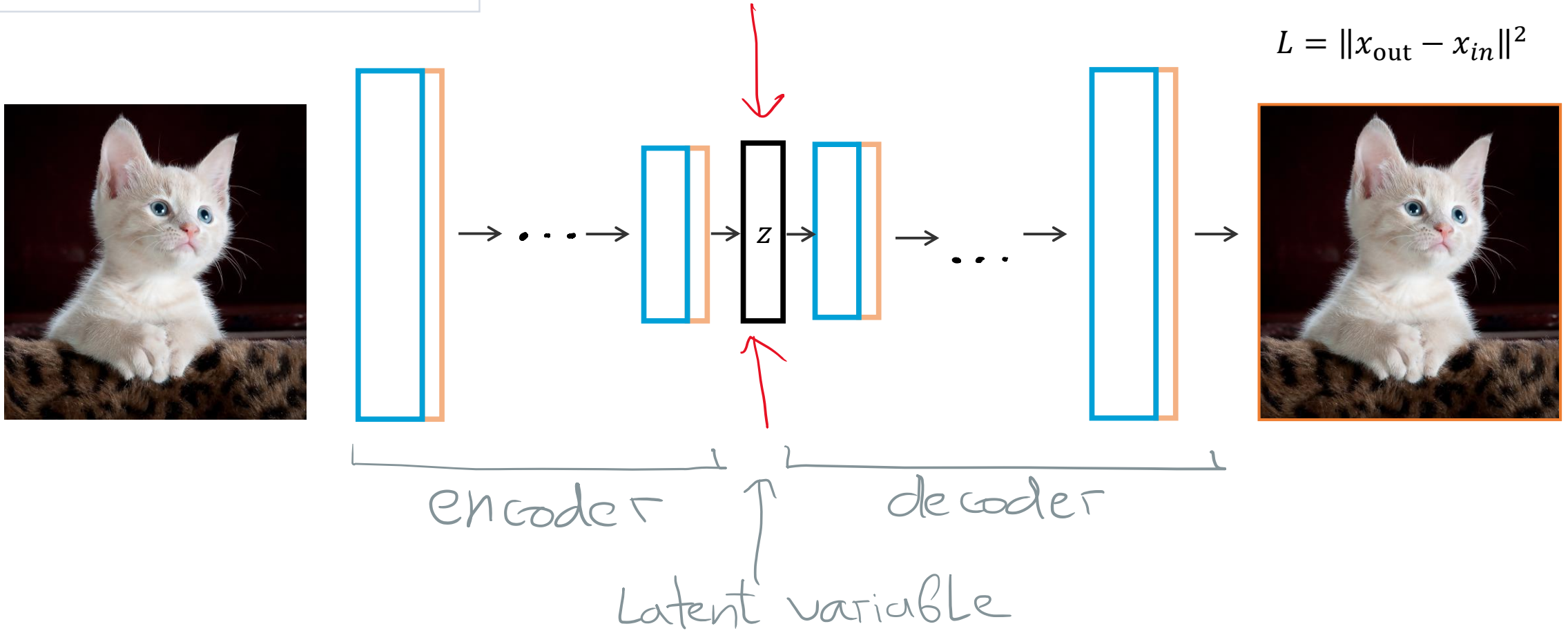
- Clustering
- Outlier detection
- Learning features
- Generation
- ....

Пример: K-means clustering

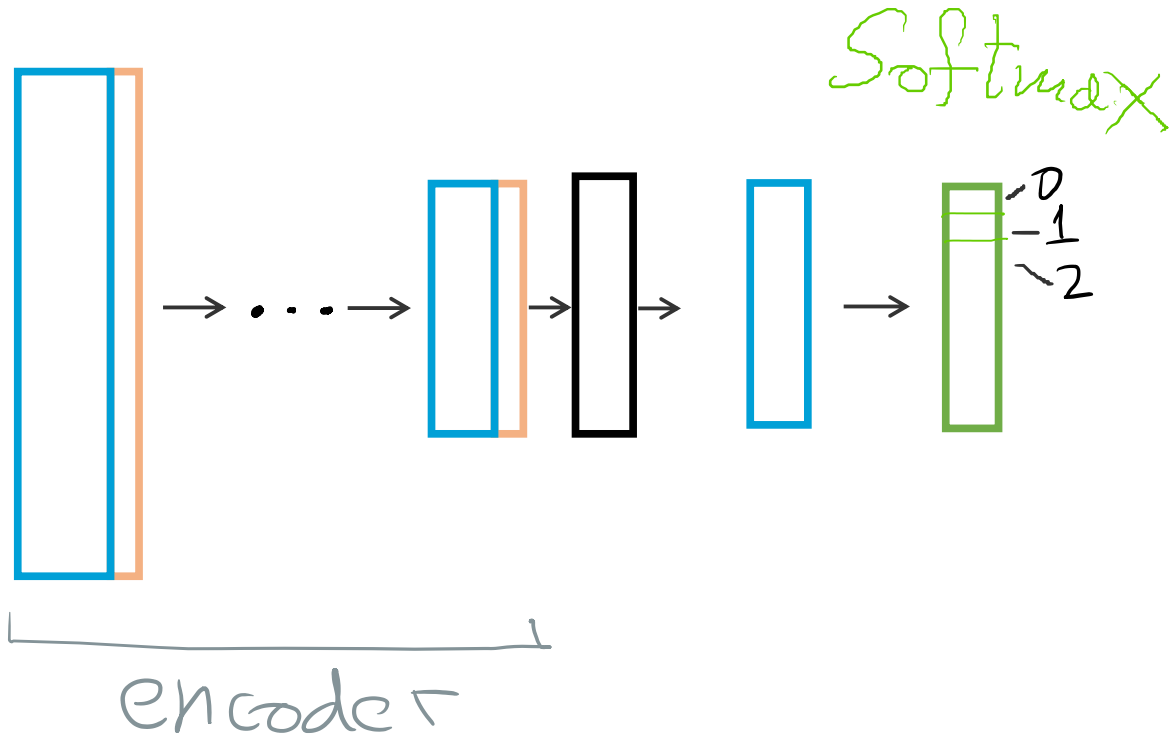


Source

Autoencoder

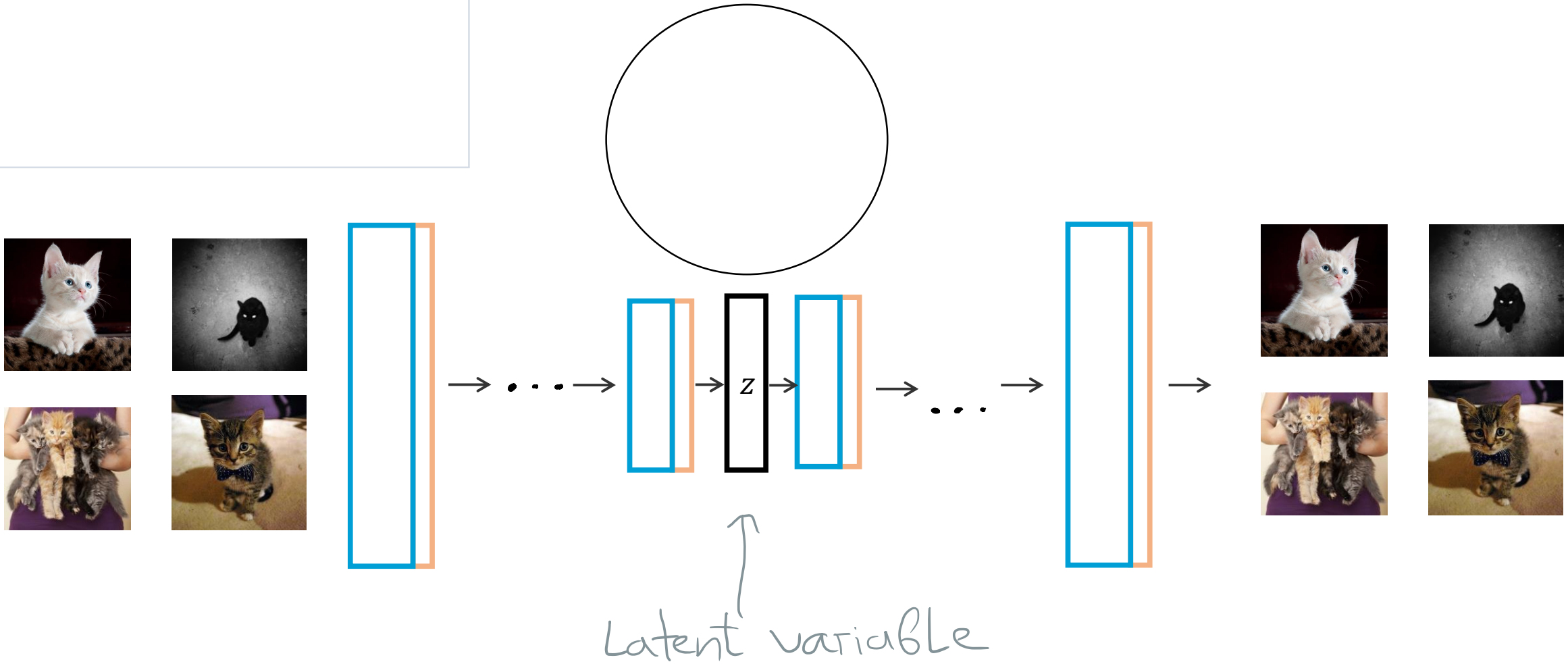


Что с этим делать дальше?

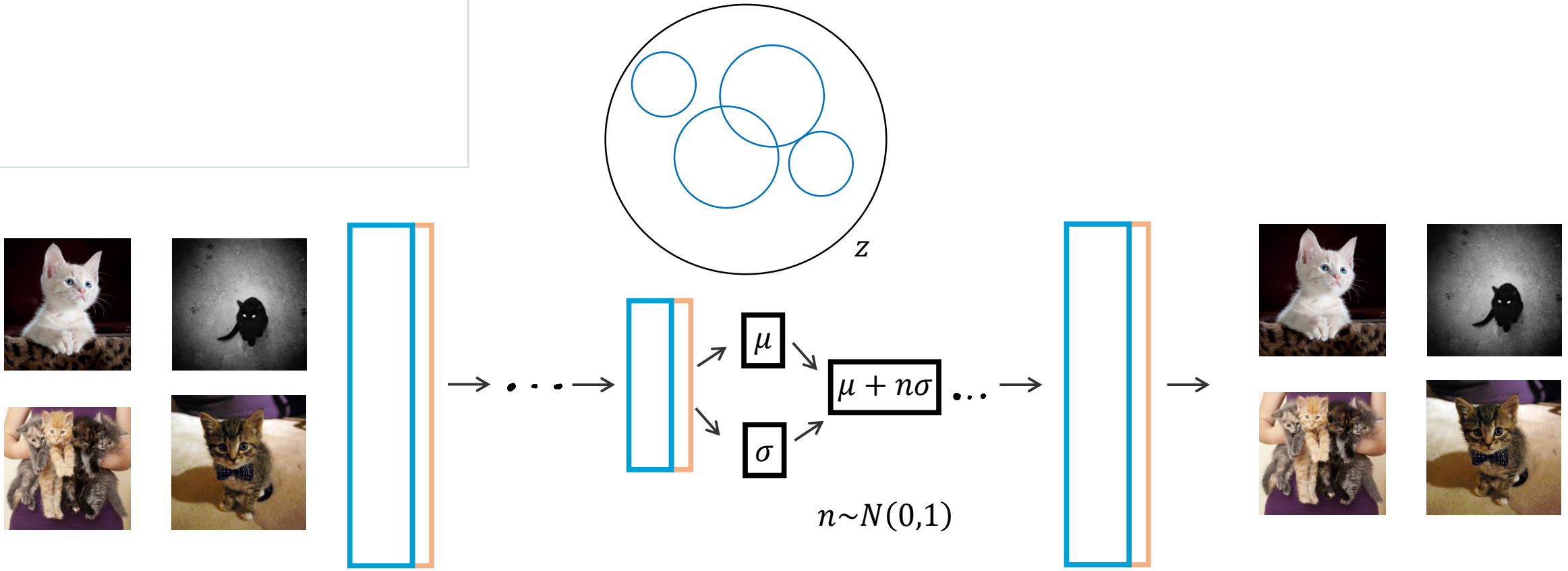



ТАК СЕБЕ
ПРАЗДНИК,
ЕСЛИ
ЧЕСТНО

overfitting



Variational Autoencoder



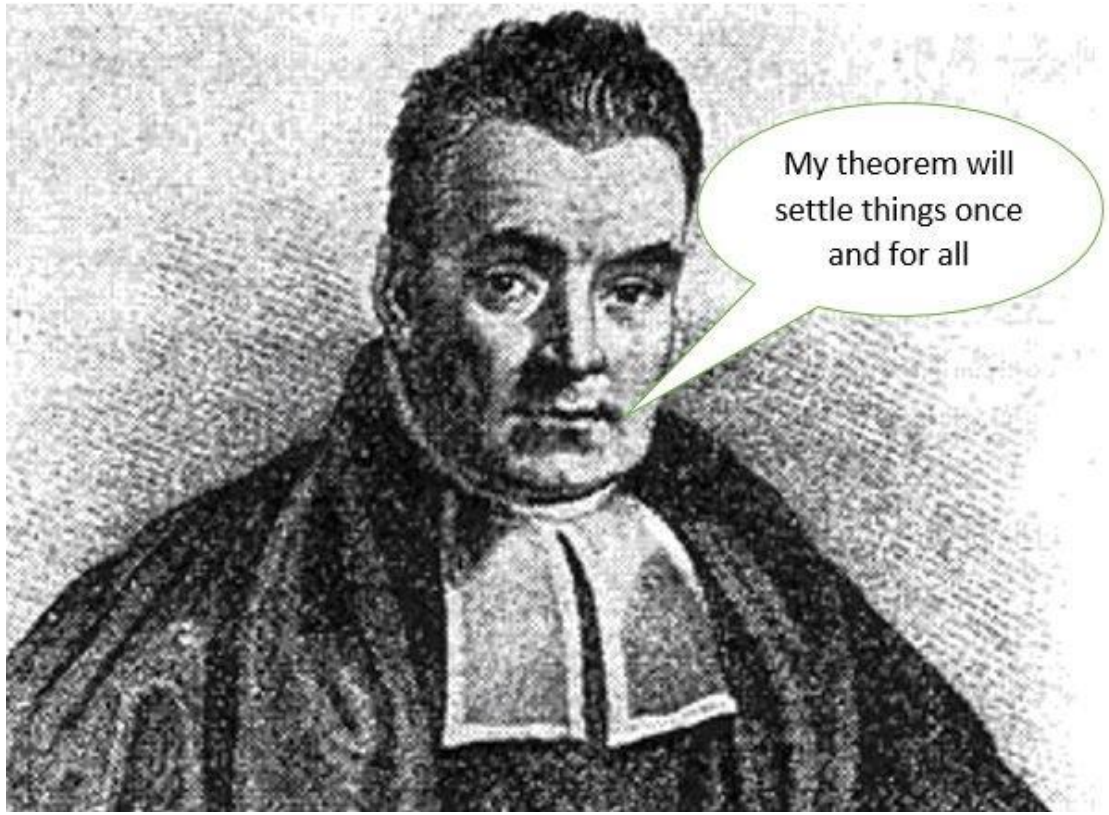
$$L_{\mathrm{KL}} = D_{KL}\left(\mu, \sigma \,||N(0,1)\right)$$

$$= \frac{1}{2}\sum_{j=0}^{N}\left(1 + \log\sigma_j^2 - \mu_j^2 - \sigma_j^2\right)$$

$$L_{\mathrm{r}} = \|x_{\mathrm{out}} - x_{in}\|^2$$

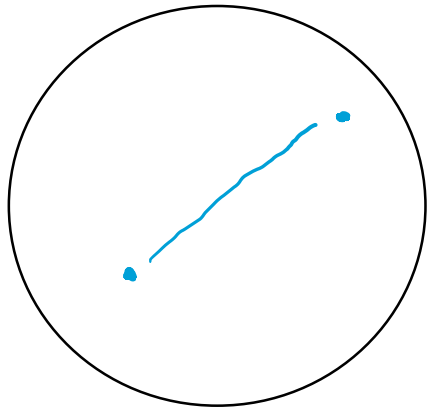
Bayesian perspective

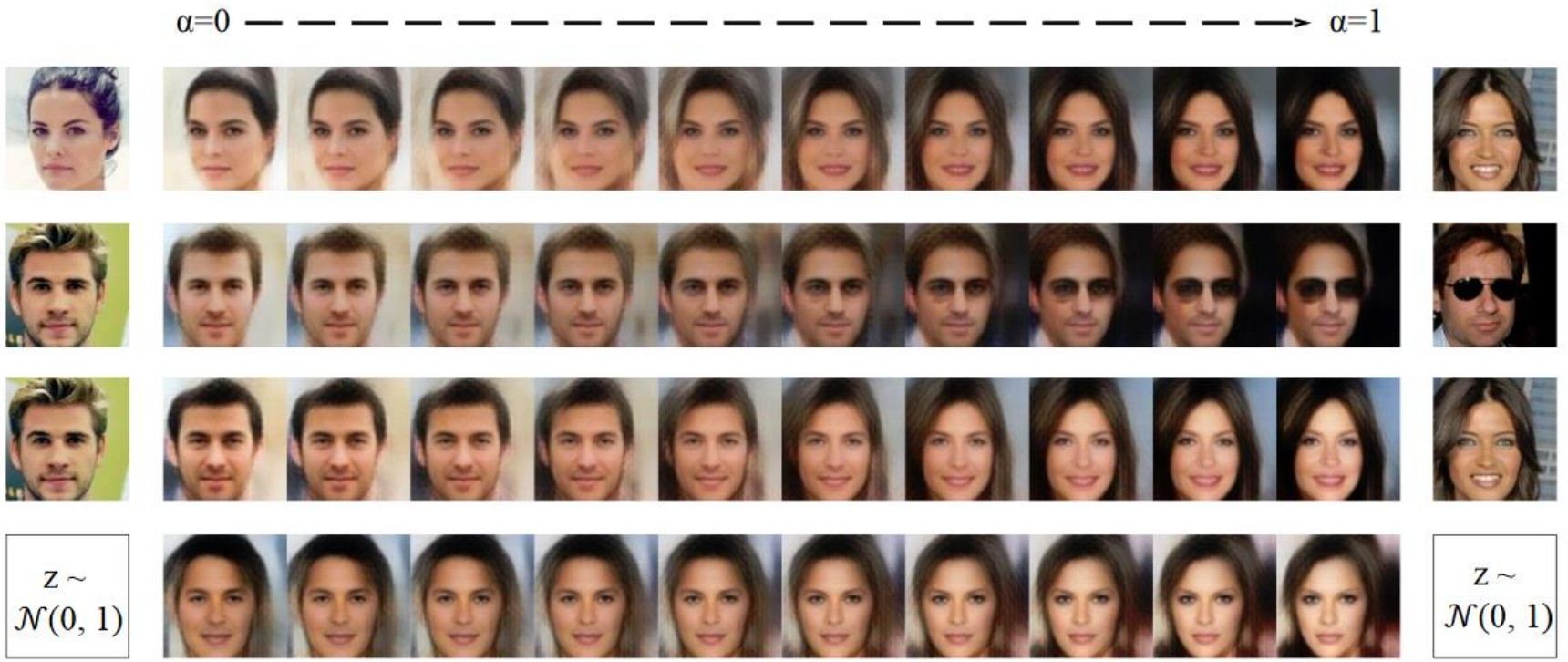
$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$



Если интересно:
[Tutorial on Variational Autoencoders'16]
[Auto-Encoding Variational Bayes'13]

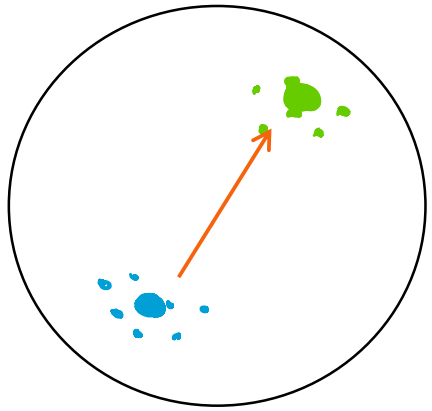
Интерполяции



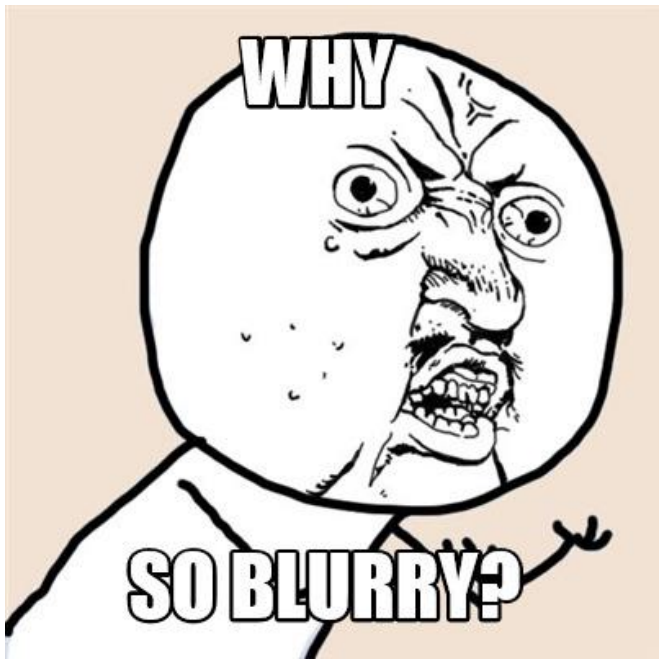


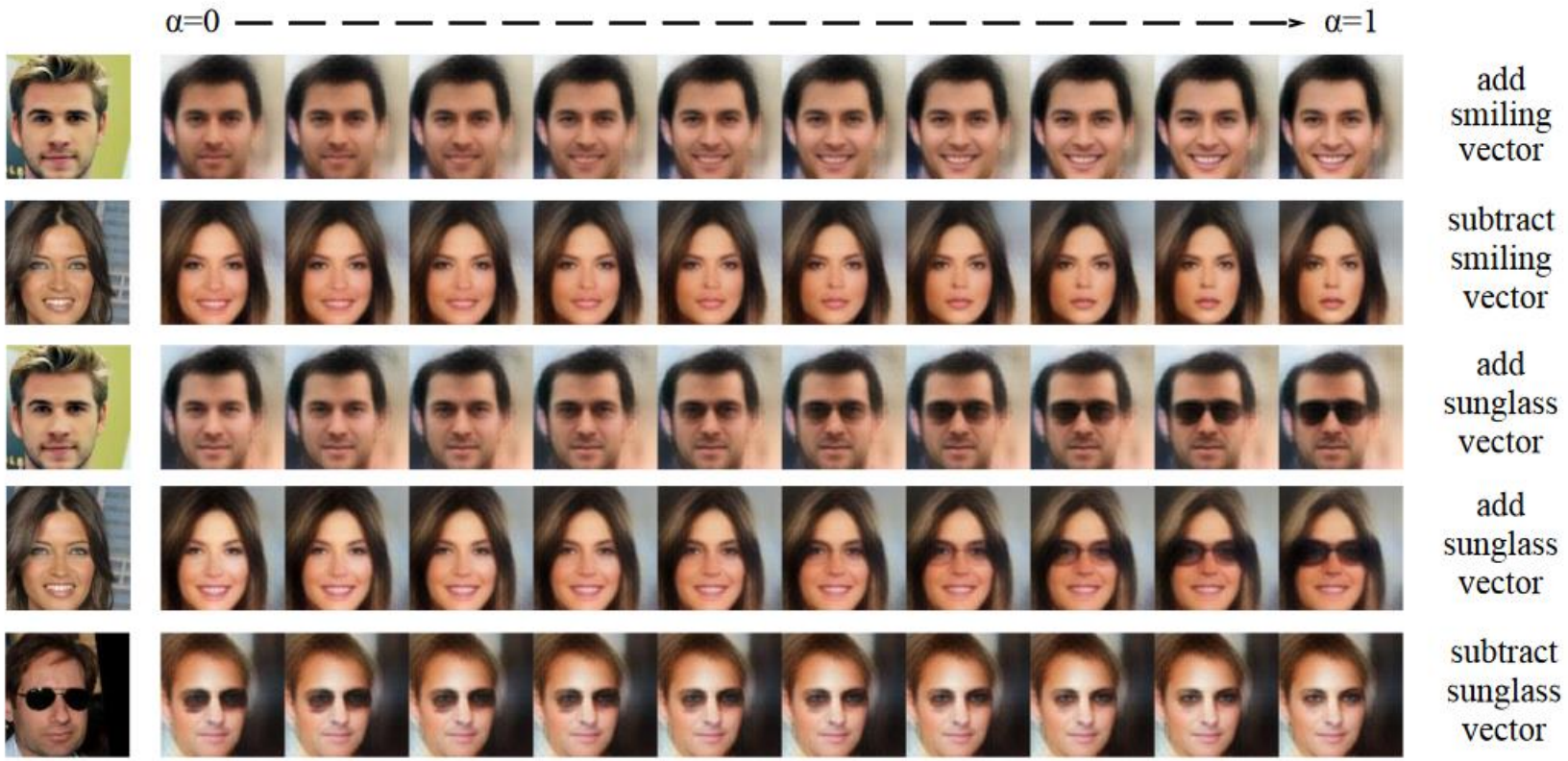
[Deep Feature Consistent Variational Autoencoder'16]

Операции с latent space



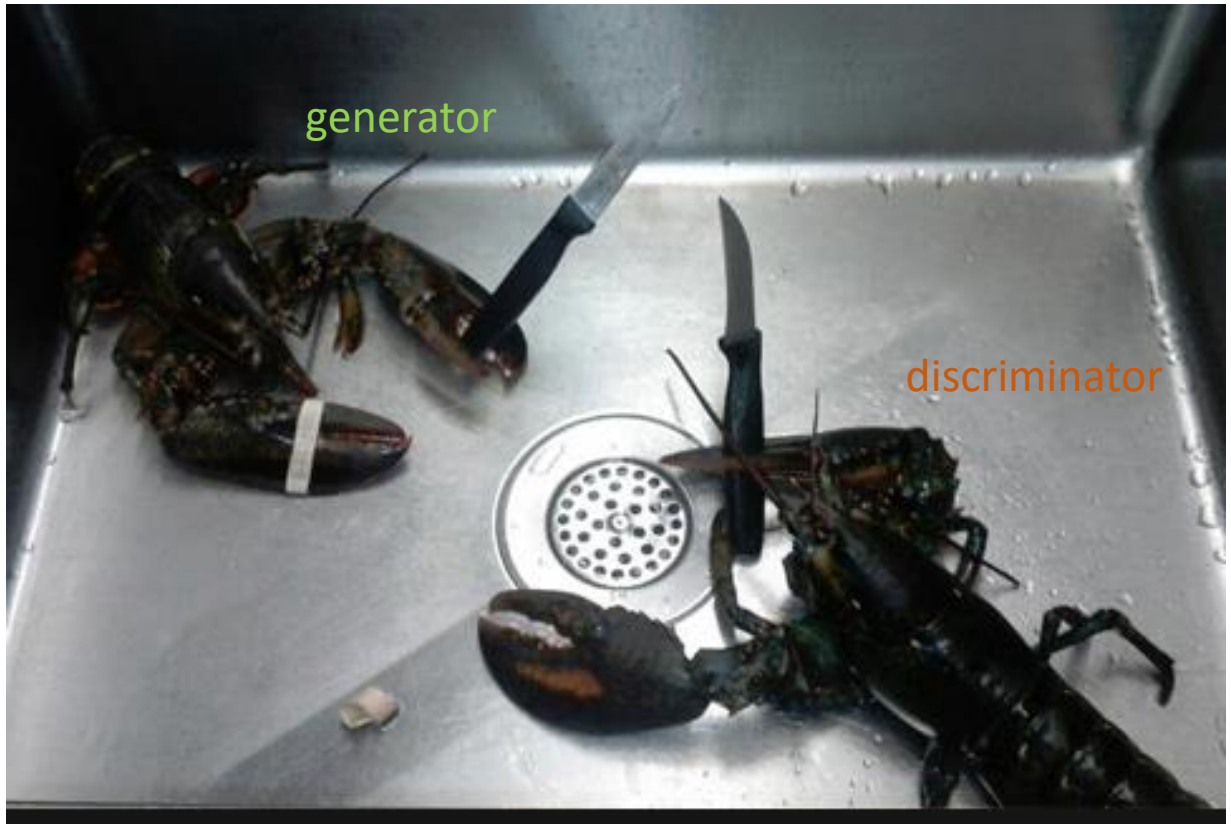
[Deep Feature Consistent Variational Autoencoder'16]



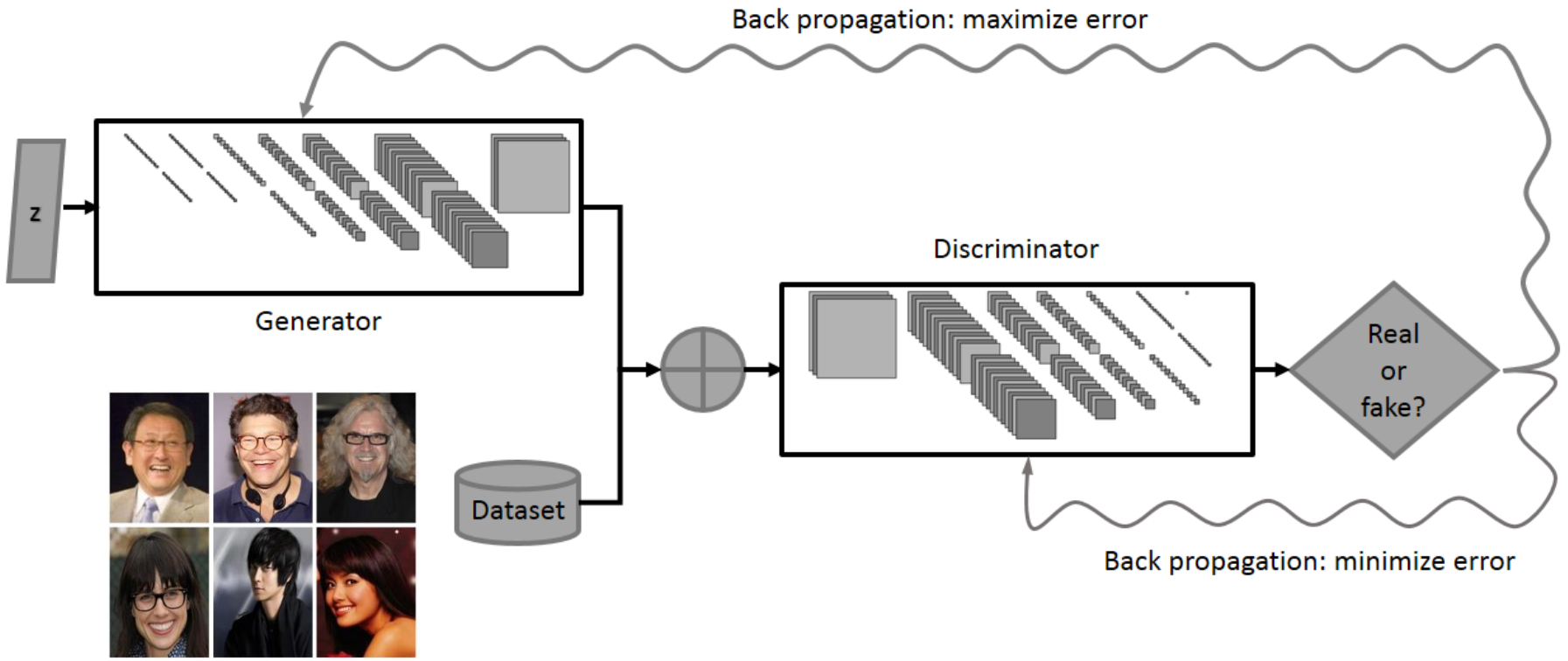


[Deep Feature Consistent Variational Autoencoder'16]

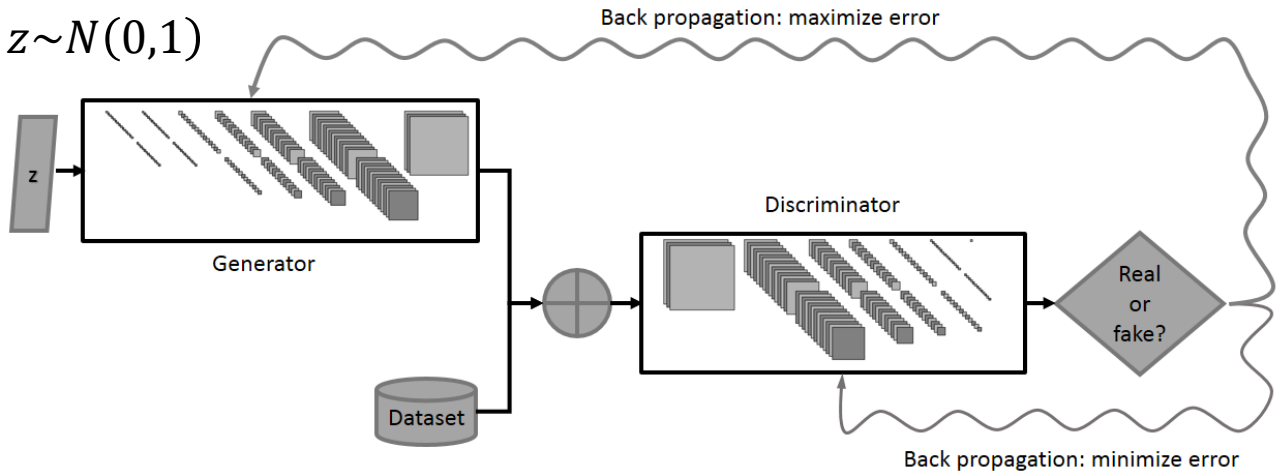
Generative Adversarial Networks (GANs)



Generative Adversarial Networks (GANs)



Generative Adversarial Networks (GANs)



for number of training iterations **do**
  for k steps **do**
    • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    • Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
    • Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

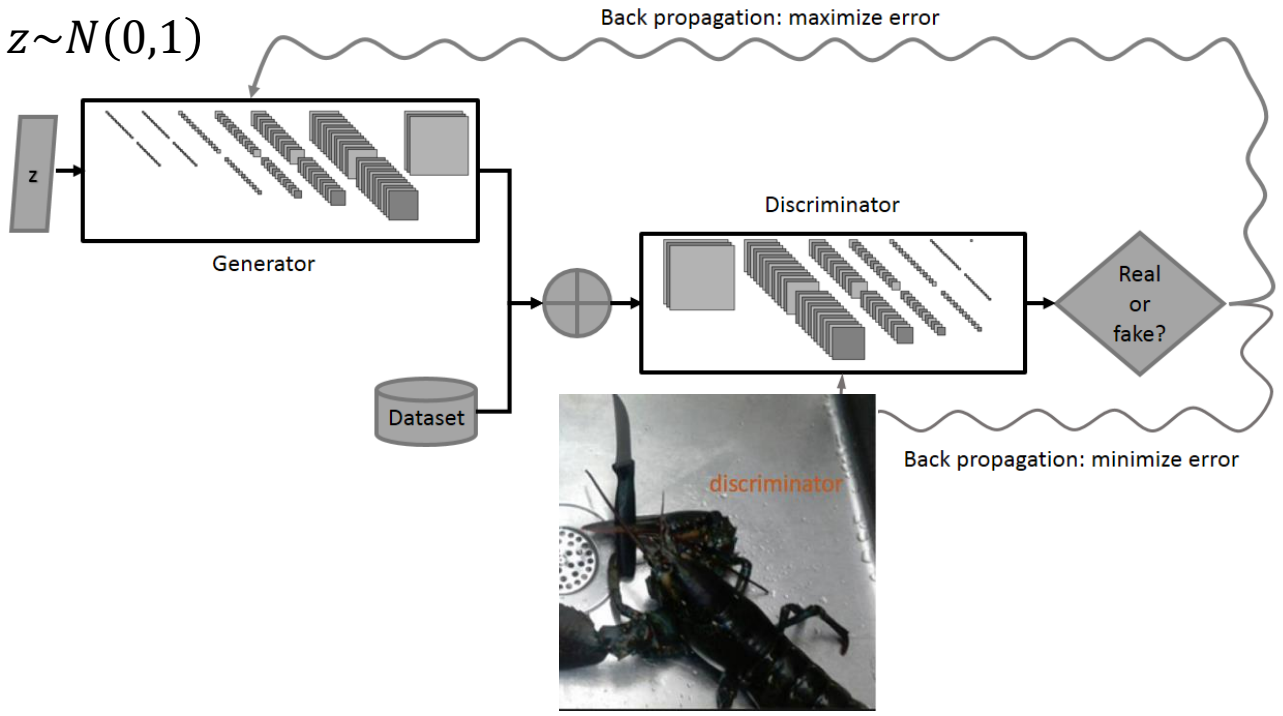
  end for
  • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  • Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

Generative Adversarial Networks'14

Generative Adversarial Networks (GANs)



for number of training iterations **do**
  for k steps **do**
    - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
    - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

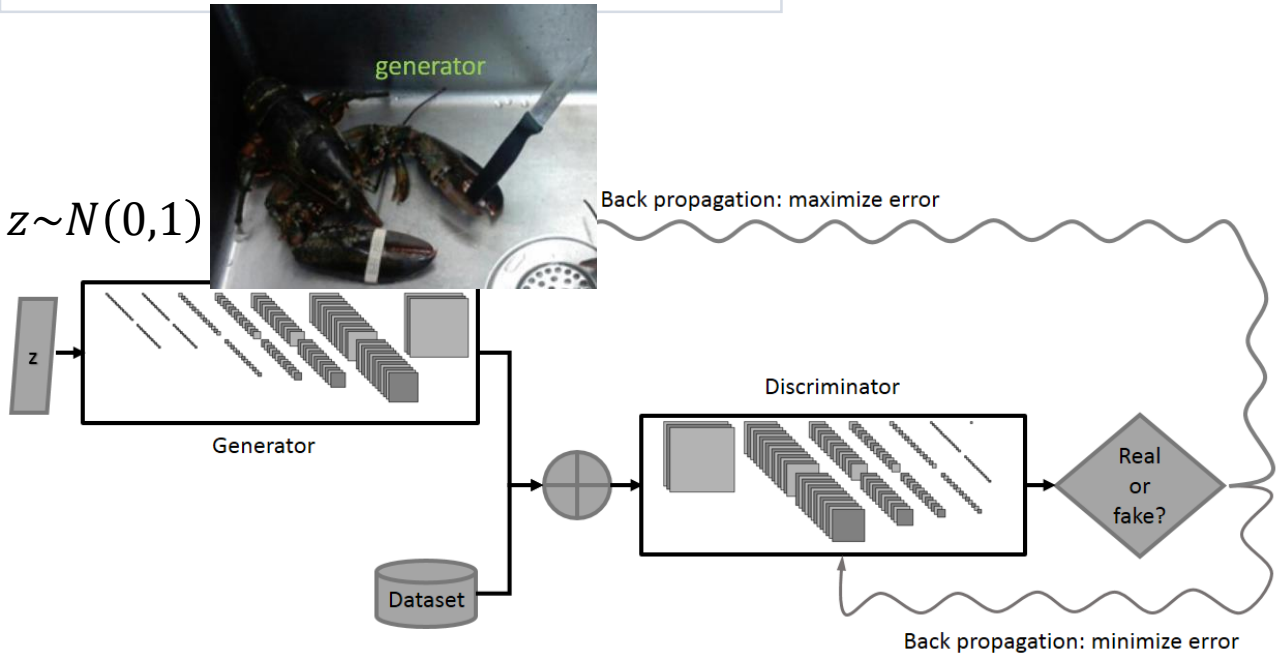
  end for
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

Generative Adversarial Networks'14

Generative Adversarial Networks (GANs)



for number of training iterations **do**
  for k steps **do**
    - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
    - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

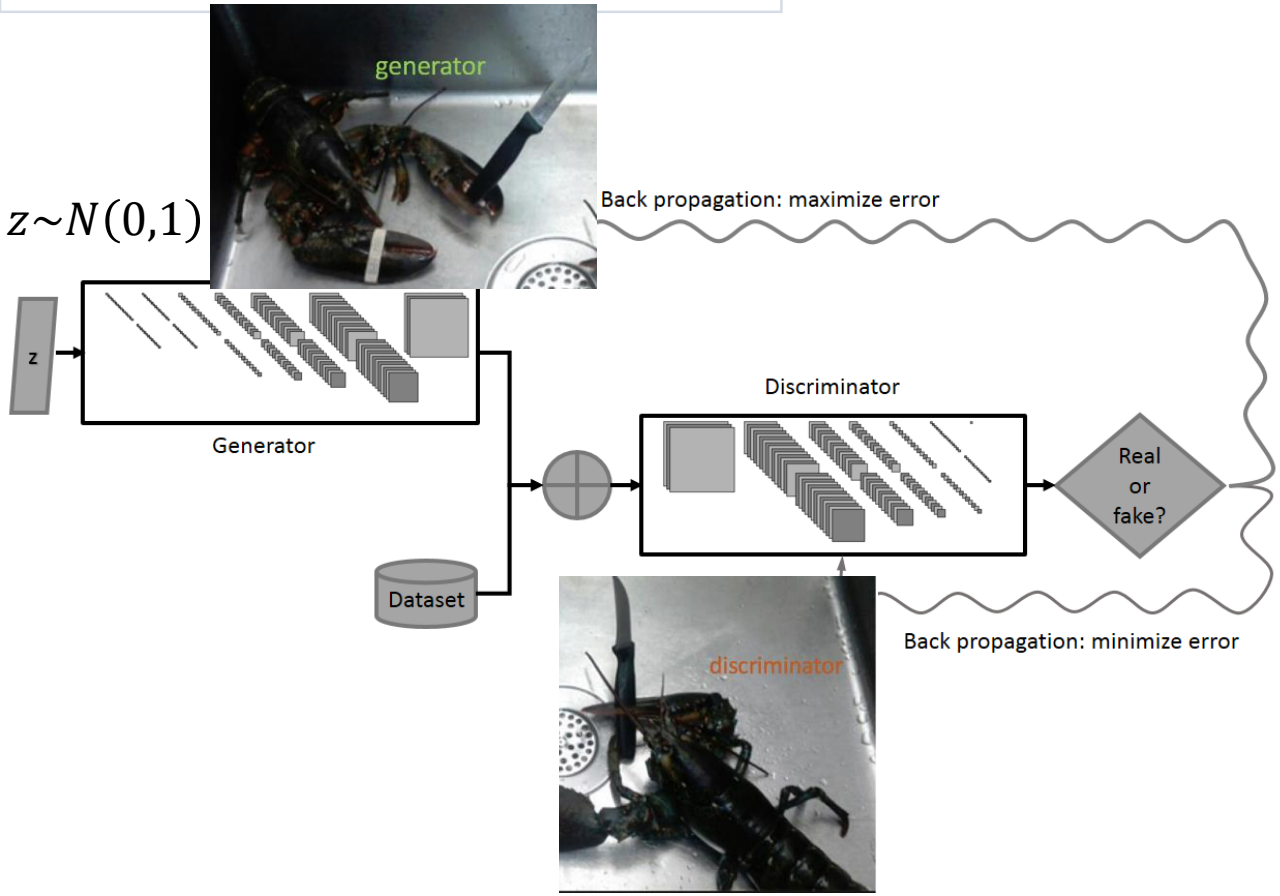
  end for
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

[Generative Adversarial Networks'14]

Generative Adversarial Networks (GANs)



for number of training iterations **do**

  for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

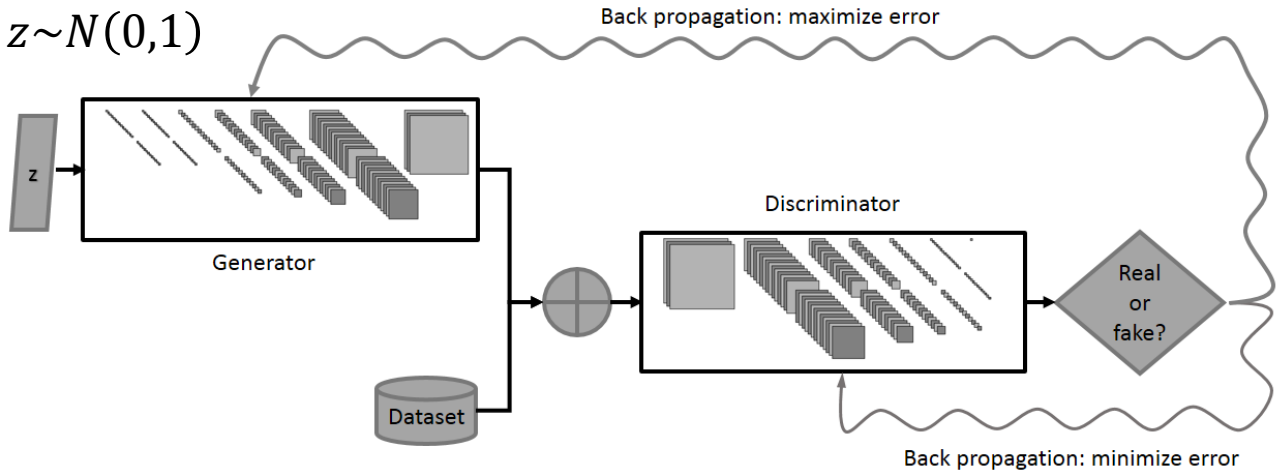
- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

Generative Adversarial Networks'14

Generative Adversarial Networks (GANs)

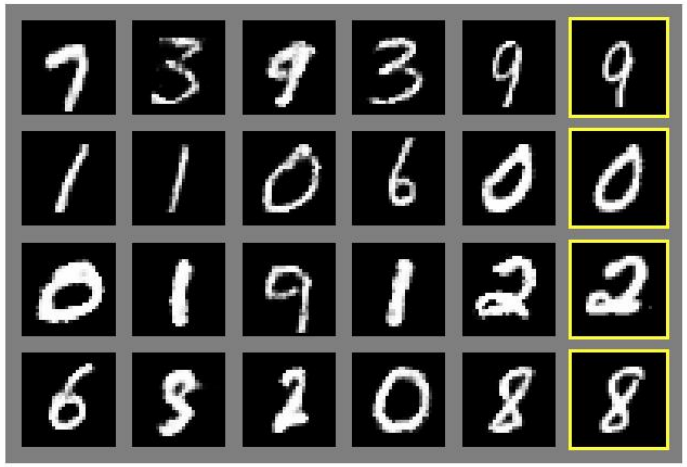


In other words, D and G play the following two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \quad (1)$$

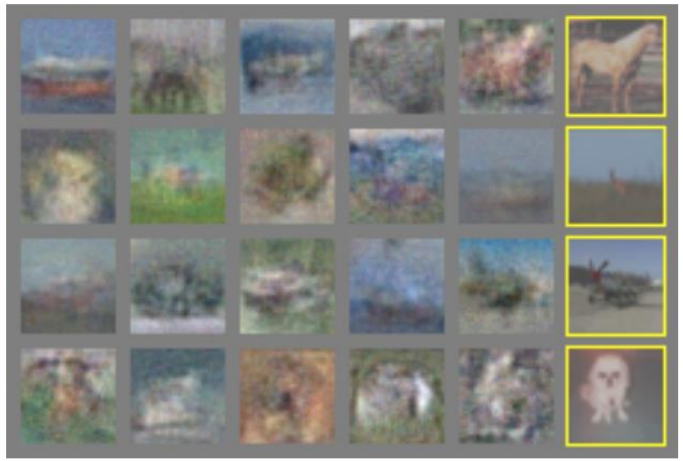
Generative Adversarial Networks'14

Если все это сделать аккуратно…

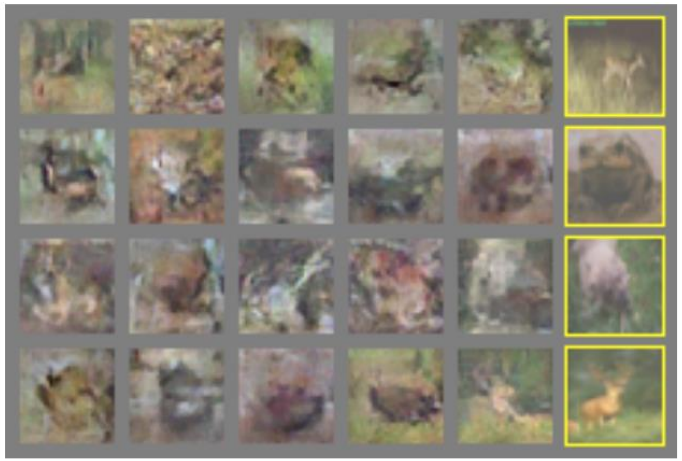


a)

b)



c)



d)

[Generative Adversarial Networks’14]

Soumith Chintala
@soumithchintala

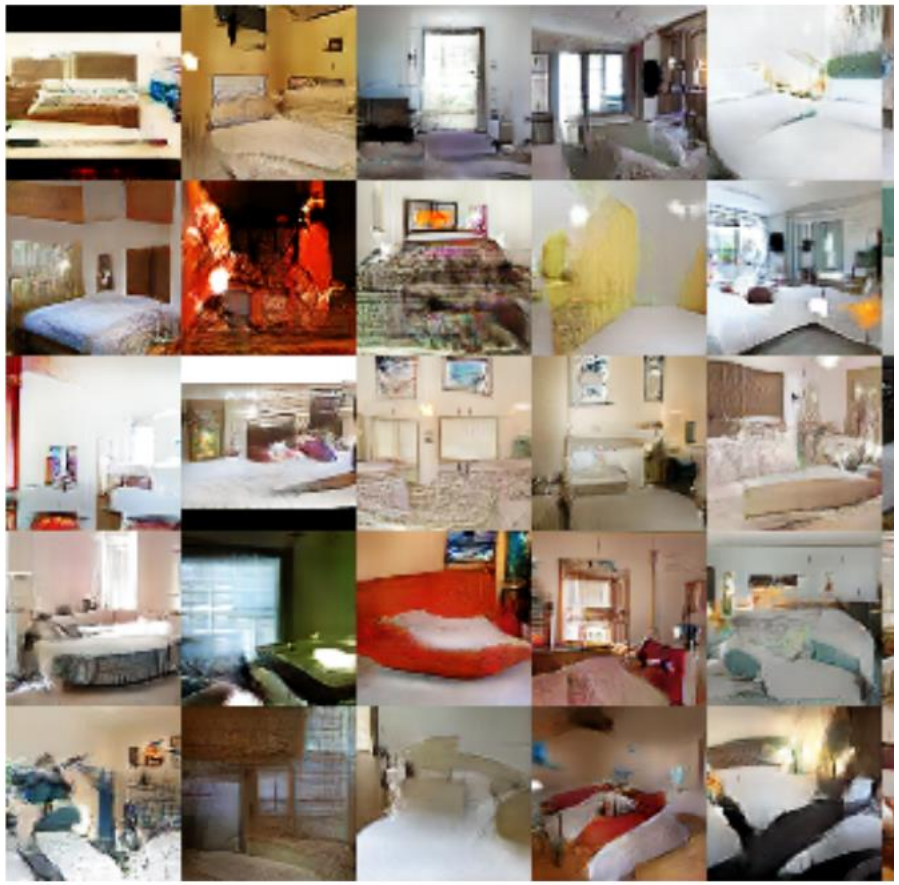
Following

How to Train a GAN? I'm as clueless as you. Best I'll do is summarize the tricks, hacks and untold dark rituals that we've all used so far!

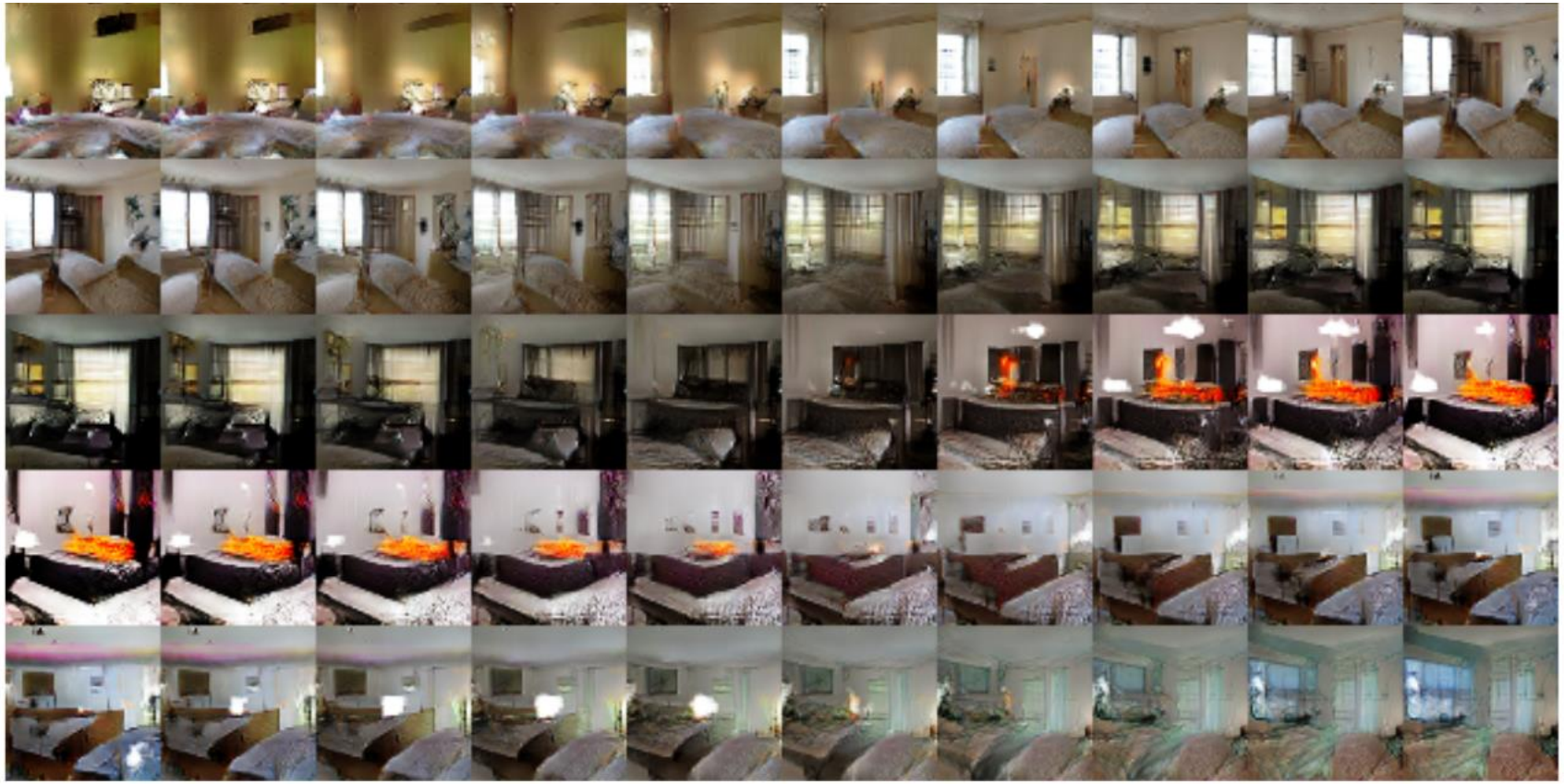
10:08 AM - 3 Dec 2016

Twitter

DCGAN'15

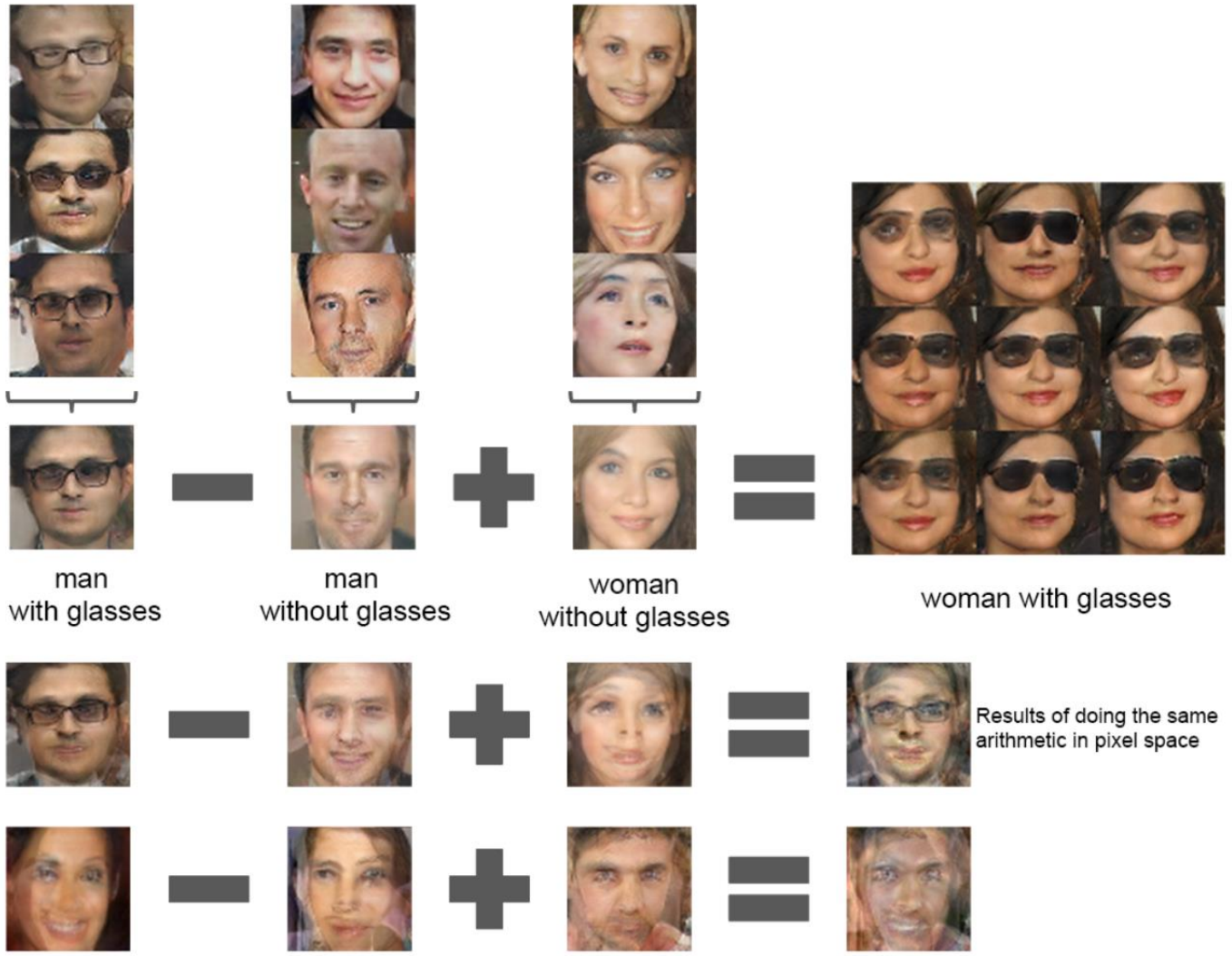


Интерполяции



DCGAN’15

Арифметика в latent space



DCGAN'15


BACK
TO THE PRESENT

- Sdf-GAN - Sdf-GAN: Semi-supervised Depth Fusion with Multi-scale Adversarial Networks
- SEGAN - SEGAN: Speech Enhancement Generative Adversarial Network
- SeGAN - SeGAN: Segmenting and Generating the Invisible
- SegAN - SegAN: Adversarial Network with Multi-scale L1 Loss for Medical Image Segmentation
- Sem-GAN - Sem-GAN: Semantically-Consistent Image-to-Image Translation
- SeqGAN - SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient (github)
- SeUDA - Semantic-Aware Generative Adversarial Nets for Unsupervised Domain Adaptation in Chest X-ray Segmentation
- SG-GAN - Semantic-aware Grad-GAN for Virtual-to-Real Urban Scene Adaption (github)
- SG-GAN - Sparsely Grouped Multi-task Generative Adversarial Networks for Facial At
- SGAN - Texture Synthesis with Spatial Generative Adversarial Networks
- SGAN - Stacked Generative Adversarial Networks (github)
- SGAN - Steganographic Generative Adversarial Networks

GAN zoo

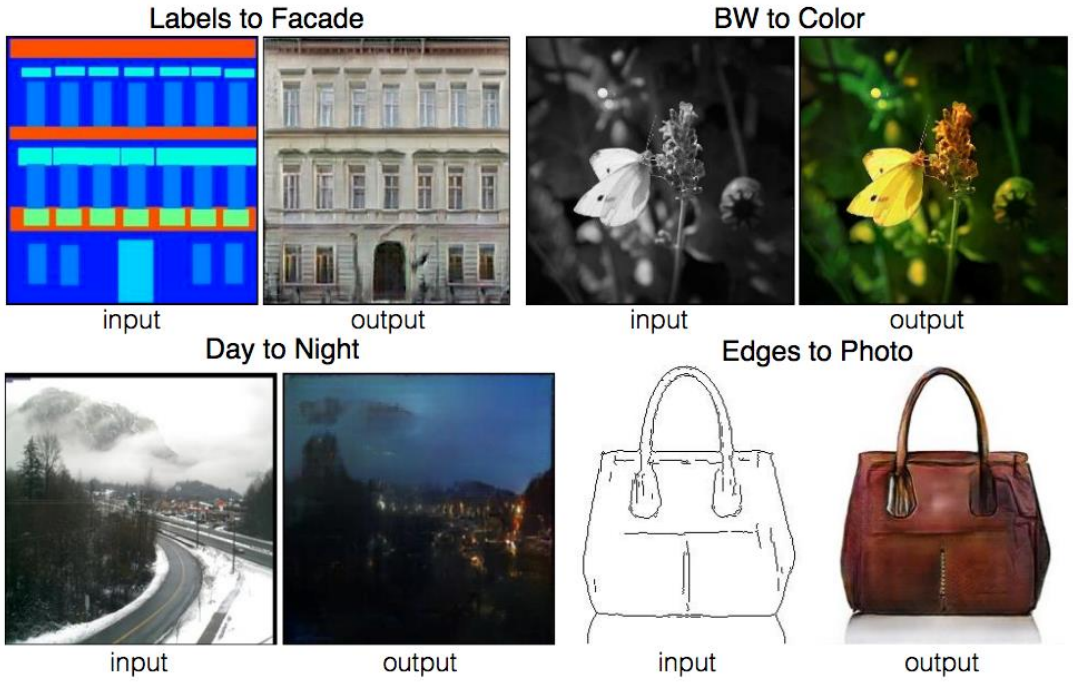
StyleGAN'18

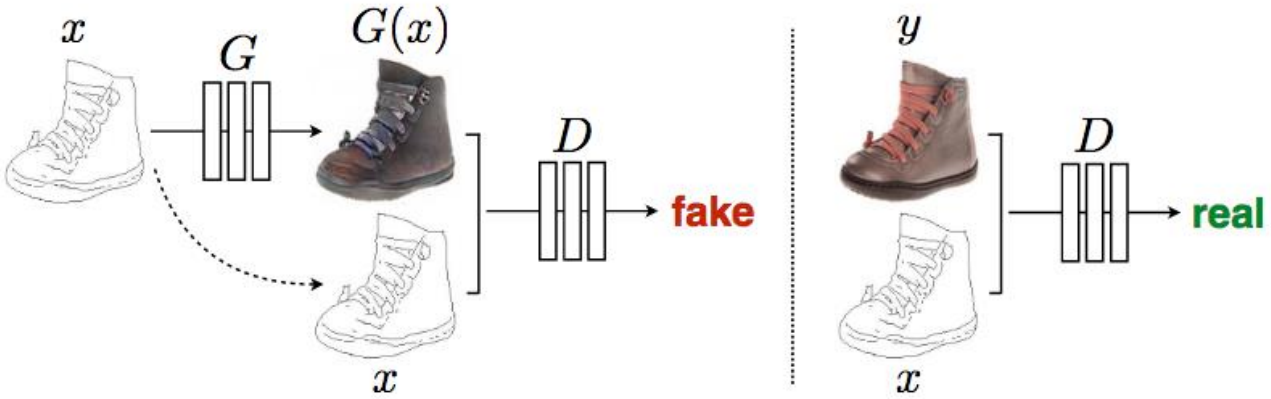
Produced by a GAN (generative adversarial netw
StyleGAN (Dec 2018) - Karras et al. and Nvidia
Original GAN (2014) - Goodfellow et al.
Don't panic. Learn about how it works.
Help me figure out what was learned by this AI he
Click for another person Link to image

thispersondoesnotexist.com


УЖЕ УХОДИТЕ?
А РАЗВЕ ЕЩЁ ЧТО-НИБУДЬ ОСТАЛОСЬ?

Pix2Pix'17


Labels to Facade
input
output
BW to Color
input
output
Day to Night
input
output
Edges to Photo
input
output


x
G
$G(x)$
D
fake
x
y
D
real
x

Rendering with GANs

[Video-to-Video Synthesis'18]