SE125 Machine Learning

Generative Adversarial Networks

Yue Ding

School of Software, Shanghai Jiao Tong University
dingyue@sjtu.edu.cn

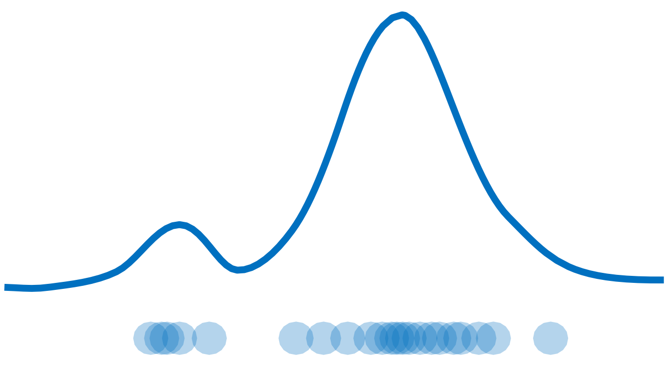
Which face is fake?

Generative modeling

Goal: Take as input training samples from some distribution and learn a model that represents that distribution

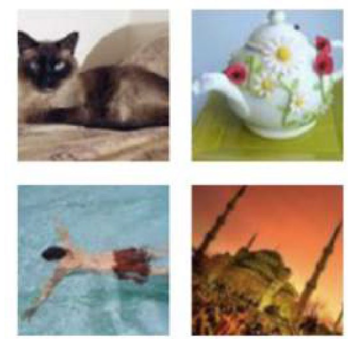
Density Estimation

samples



Sample Generation

Input samples



Generated samples

Training data ~ $P_{data}(x)$

Generated ~ $P_{model}(x)$

How can we learn $P_{model}(x)$ similar to $P_{data}(x)$?

Why generative models? Debiasing

Capable of uncovering **underlying latent variables** in a dataset

vs

Homogeneous skin color, pose

Diverse skin color, pose, illumination

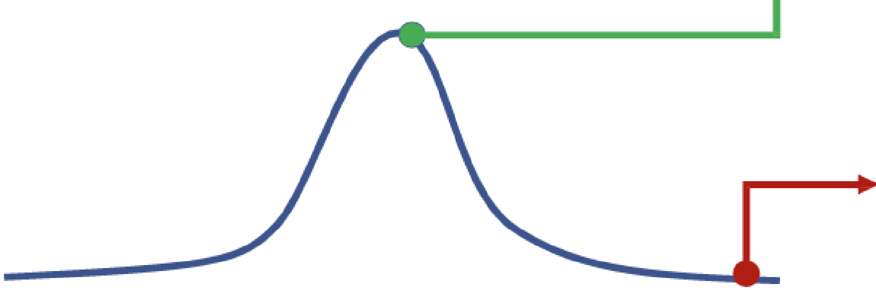
How can we use latent distributions to create fair and representative datasets?

Why generative models? Outlier detection

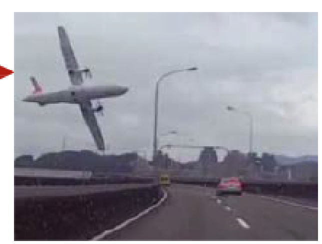
- **Problem:** How can we detect when we encounter something new or rare?
- **Strategy:** Leverage generative models, detect outliers in the distribution
- Use outliers during training to improve even more!

95% of Driving Data:

(1) sunny, (2) highway, (3) straight road



Detect outliers to avoid unpredictable behavior when training



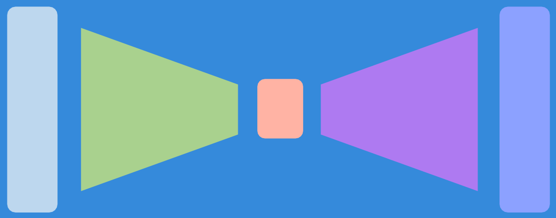
Edge Cases

Harsh Weather

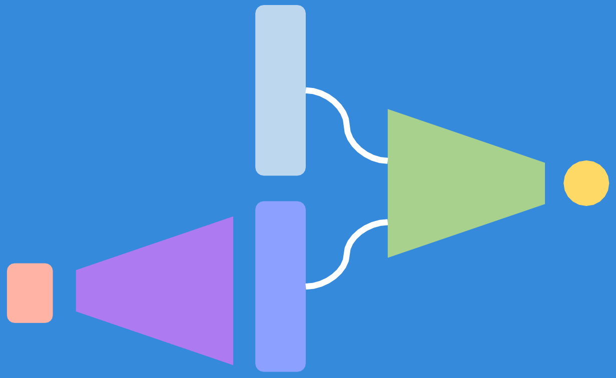
Pedestrians

Latent variable models

Autoencoders and Variational Autoencoders (VAEs)



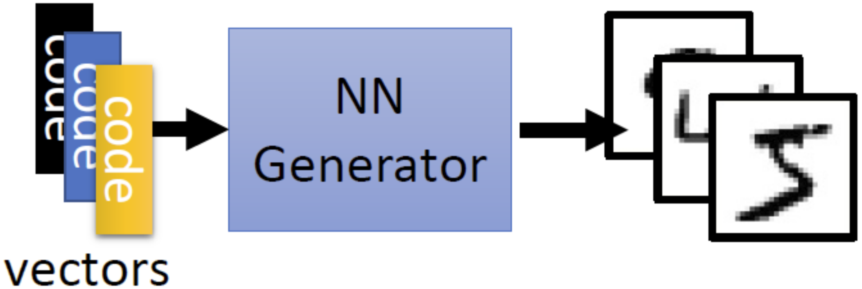
Generative Adversarial Networks (GANs)

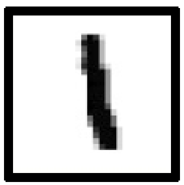
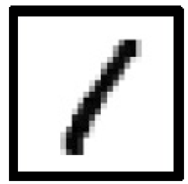
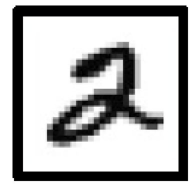
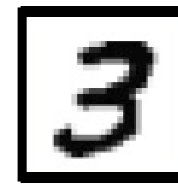


What is a latent variable?

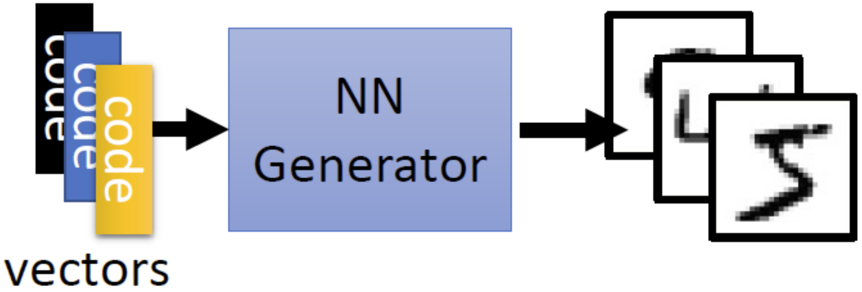
Can we learn the true explanatory factors, e.g. latent variables, from only observed data?

Generator

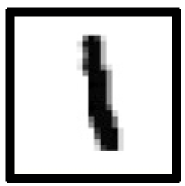
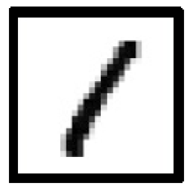
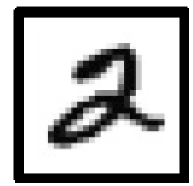
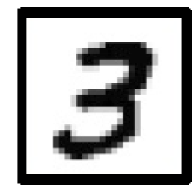


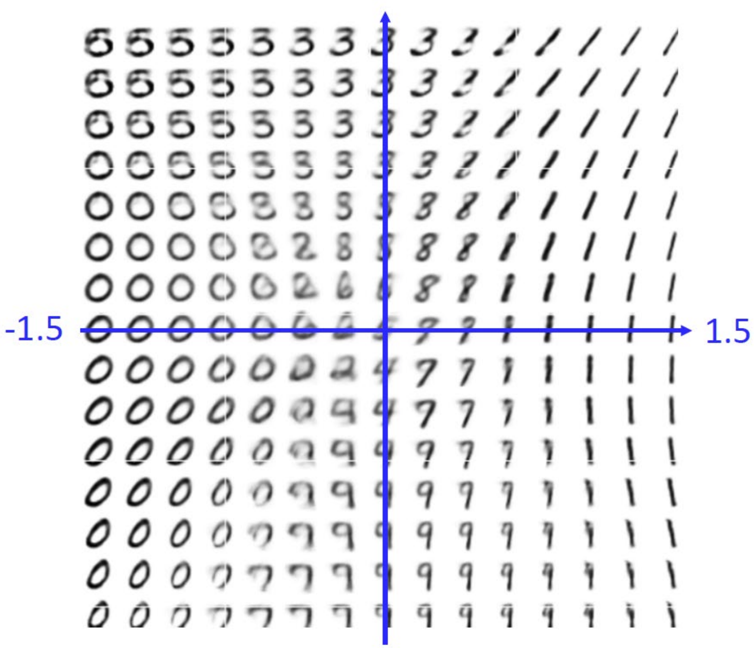
code:	$\begin{bmatrix} 0.1 \\ -0.5 \end{bmatrix}$	$\begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}$	$\begin{bmatrix} 0.2 \\ -0.1 \end{bmatrix}$	$\begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}$
Image:				

Generator

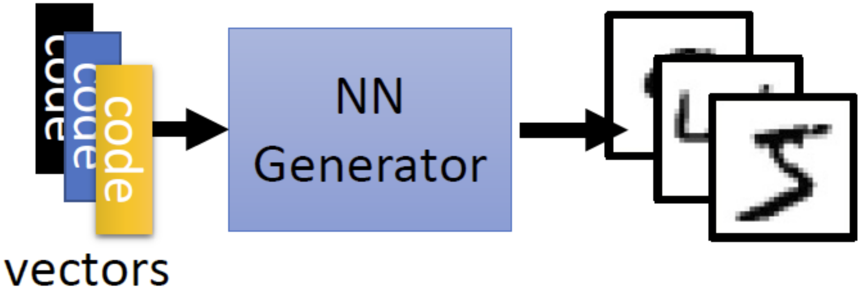


code: $\begin{bmatrix} 0.1 \\ -0.5 \end{bmatrix}$ $\begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}$ $\begin{bmatrix} 0.2 \\ -0.1 \end{bmatrix}$ $\begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}$

Image:    

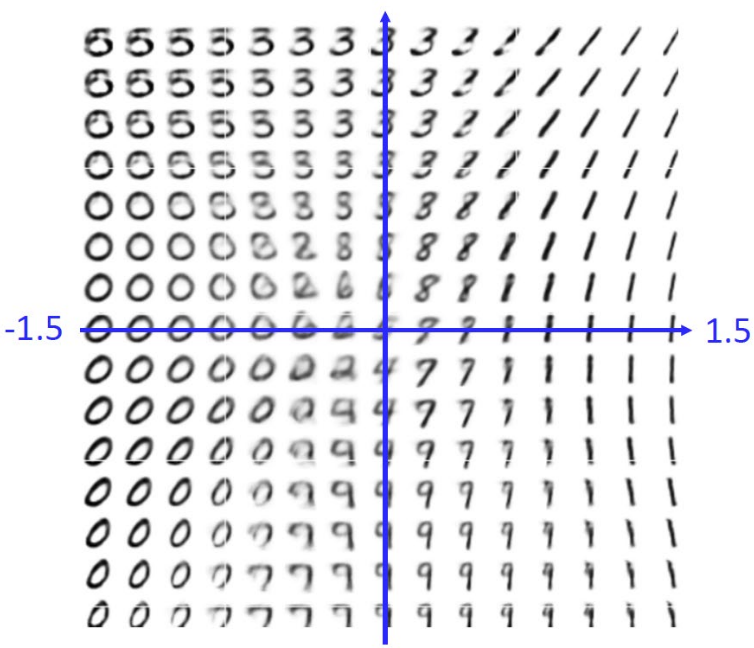


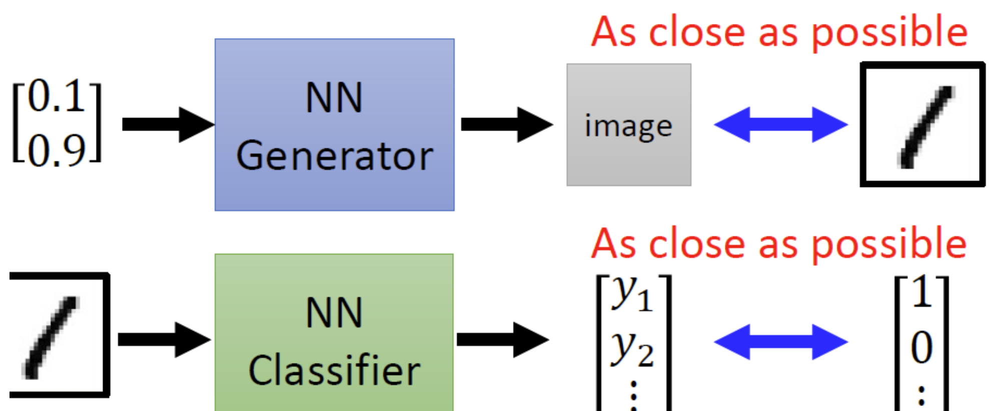
Generator



code: $\begin{bmatrix}0.1\\-0.5\end{bmatrix}$ $\begin{bmatrix}0.1\\0.9\end{bmatrix}$ $\begin{bmatrix}0.2\\-0.1\end{bmatrix}$ $\begin{bmatrix}0.3\\0.2\end{bmatrix}$

Image:





Generative Adversarial Nets

Ian J. Goodfellow,[*] Jean Pouget-Abadie,[†] Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,[‡] Aaron Courville, Yoshua Bengio[§]
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Yann LeCun’s comment

What are some recent and potentially upcoming breakthroughs in deep learning?

Yann LeCun, Director of AI Research at Facebook and Professor at NYU

Written Jul 29 · Upvoted by Joaquin Quiñonero Candela, Director Applied Machine Learning at Facebook and Nikhil Garg, I lead a team of Quora engineers working on ML/NLP problems

……

The most important one, in my opinion, is adversarial training (also called GAN for Generative Adversarial Networks). This is an idea that was originally proposed by Ian Goodfellow when he was a student with Yoshua Bengio at the University of Montreal (he since moved to Google Brain and recently to OpenAI).

This, and the variations that are now being proposed is the most interesting idea in the last 10 years in ML, in my opinion.

https://www.quora.com/What-are-some-recent-and-potentially-upcoming-breakthroughs-in-deep-learning

All Kinds of GAN …

https://github.com/hindupuravinash/the-gan-zoo

GAN

ACGAN

BGAN

CGAN

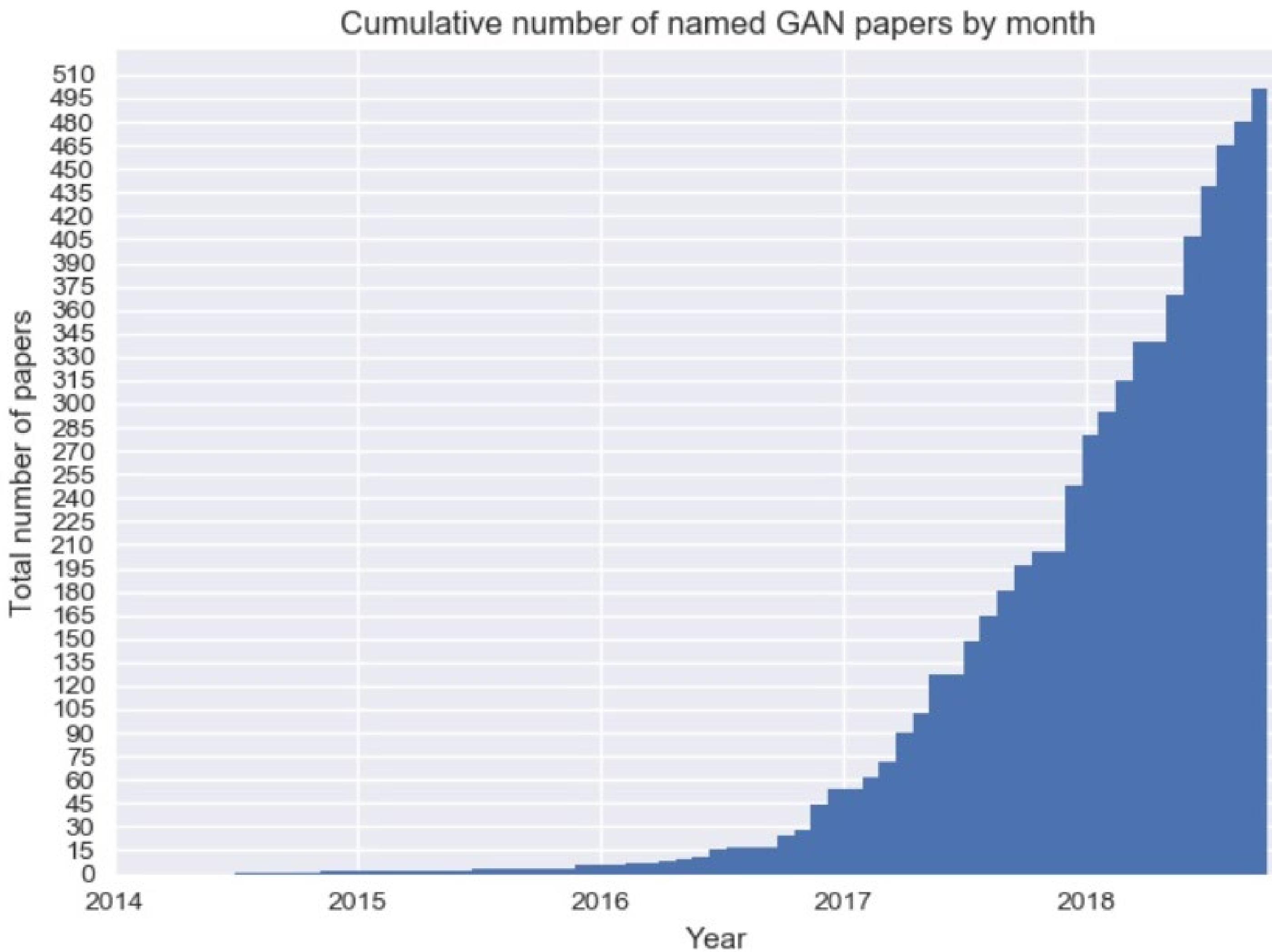
DCGAN

EBGAN

fGAN

GoGAN

⋮



Basic Idea of GAN

- The **prisoner‘s dilemma**. Two members of a criminal gang are arrested and imprisoned, the prosecutors offer each prisoner a bargain. Each prisoner is given the opportunity either to betray the other by testifying that the other committed the crime, or to cooperate with the other by remaining silent. The possible outcomes are:

- If A and B each betray the other, each of them serves ***two*** years in prison
- If A betrays B but B remains silent, A will ***be set free*** and B will serve ***three years*** in prison
- If A remains silent but B betrays A, A will serve ***three years*** in prison and B will ***be set free***
- If A and B both remain silent, both of them will serve ***only one year*** in prison (on the lesser charge).

Prisoner's dilemma payoff matrix

B / A	**B stays silent**	**B betrays**
A stays silent	-1 / -1	0 / -3
A betrays	-3 / 0	-2 / -2

囚徒困境、纳什均衡和帕雷托最优

- **纳什均衡**（Nash equilibrium），无一参与者可以通过独自行动而增加收益的策略组合。
  - 囚徒困境是一个非零和博弈，**反映个人最佳选择并非团体最佳选择**。
- 在**重复的囚徒困境**中，博弈被反复地进行。作为反复接近无限的数量，纳什均衡趋向于**帕雷托最优**。**帕累托最优可以是合作博弈，而纳什均衡只能是非合作博弈。**
- **帕雷托最优**（Pareto optimality），也称为**帕雷托效率**（Pareto efficiency），是指资源分配的一种理想状态。
  - 给定固有的一群人和可分配的资源，如果从一种分配状态到另一种状态的变化中，在没有使任何人境况变坏的前提下，使得至少一个人变得更好，这就是**帕雷托改善**。**帕雷托最优的状态**就是不可能再有更多的帕雷托改善的状态；换句话说，**不可能再改善某些人的境况，而不使任何其他人受损**。

Basic idea of GAN

Image Generation

$$\begin{bmatrix} 0.3 \\ -0.1 \\ \vdots \\ -0.7 \end{bmatrix} \begin{bmatrix} 0.1 \\ -0.1 \\ \vdots \\ 0.7 \end{bmatrix} \begin{bmatrix} -0.3 \\ 0.1 \\ \vdots \\ 0.9 \end{bmatrix} \rightarrow \boxed{\text{NN Generator}} \rightarrow$$

In a specific range

Sentence Generation

$$\begin{bmatrix} 0.3 \\ -0.1 \\ \vdots \\ -0.7 \end{bmatrix} \begin{bmatrix} 0.1 \\ -0.1 \\ \vdots \\ 0.2 \end{bmatrix} \begin{bmatrix} -0.3 \\ 0.1 \\ \vdots \\ 0.5 \end{bmatrix} \rightarrow \boxed{\text{NN Generator}} \rightarrow$$

How are you?

Good morning.

Good afternoon.

Powered by: http://mattya.github.io/chainer-DCGAN/

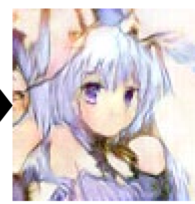
Basic Idea of GAN

It is a neural network (NN), or a function.

vector → Generator → image

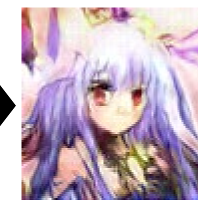
image: high dimensional vector

$\begin{bmatrix} 0.1 \\ -3 \\ \vdots \\ 2.4 \\ 0.9 \end{bmatrix}$ → Generator →



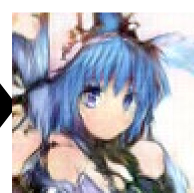
Each dimension of input vector represents some characteristics.

$\begin{bmatrix} \boxed{3} \\ -3 \\ \vdots \\ 2.4 \\ 0.9 \end{bmatrix}$ → Generator →



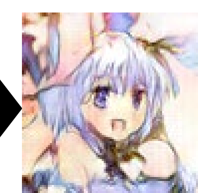
Longer hair

$\begin{bmatrix} 0.1 \\ 2.1 \\ \vdots \\ \boxed{5.4} \\ 0.9 \end{bmatrix}$ → Generator →



blue hair

$\begin{bmatrix} 0.1 \\ -3 \\ \vdots \\ 2.4 \\ \boxed{3.5} \end{bmatrix}$ → Generator →



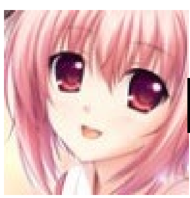
Open mouth

Basic Idea of GAN

It is a neural network (NN), or a function.

image → Discri-minator → scalar

Larger value means real, smaller value means fake.

 → Discri-minator → 1.0

 → Discri-minator → 1.0

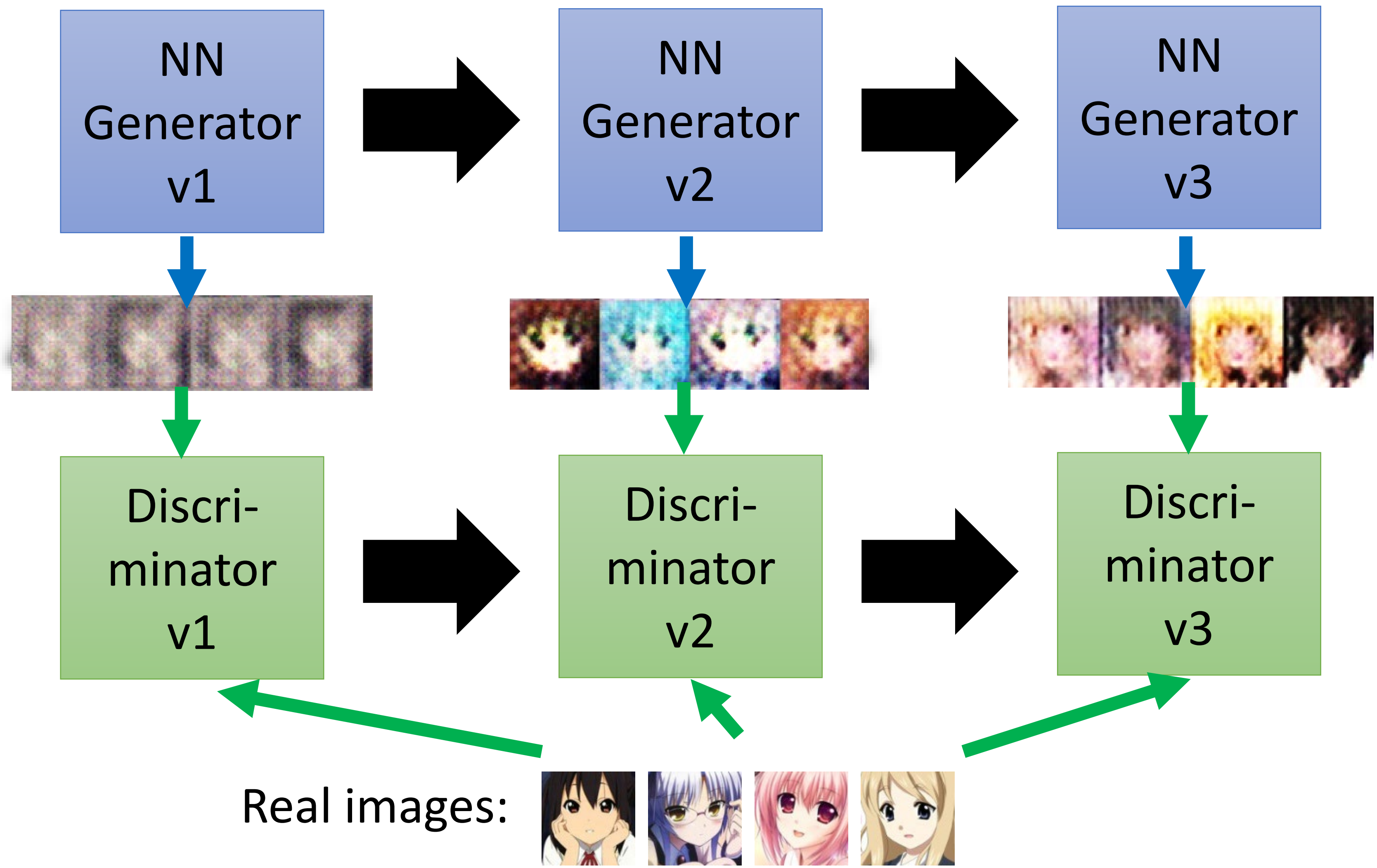
 → Discri-minator → 0.1

 → Discri-minator → 0.1

Basic Idea of GAN

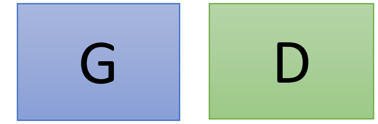
This is where the term "***adversarial***" comes from.

You can explain the process in different ways…….



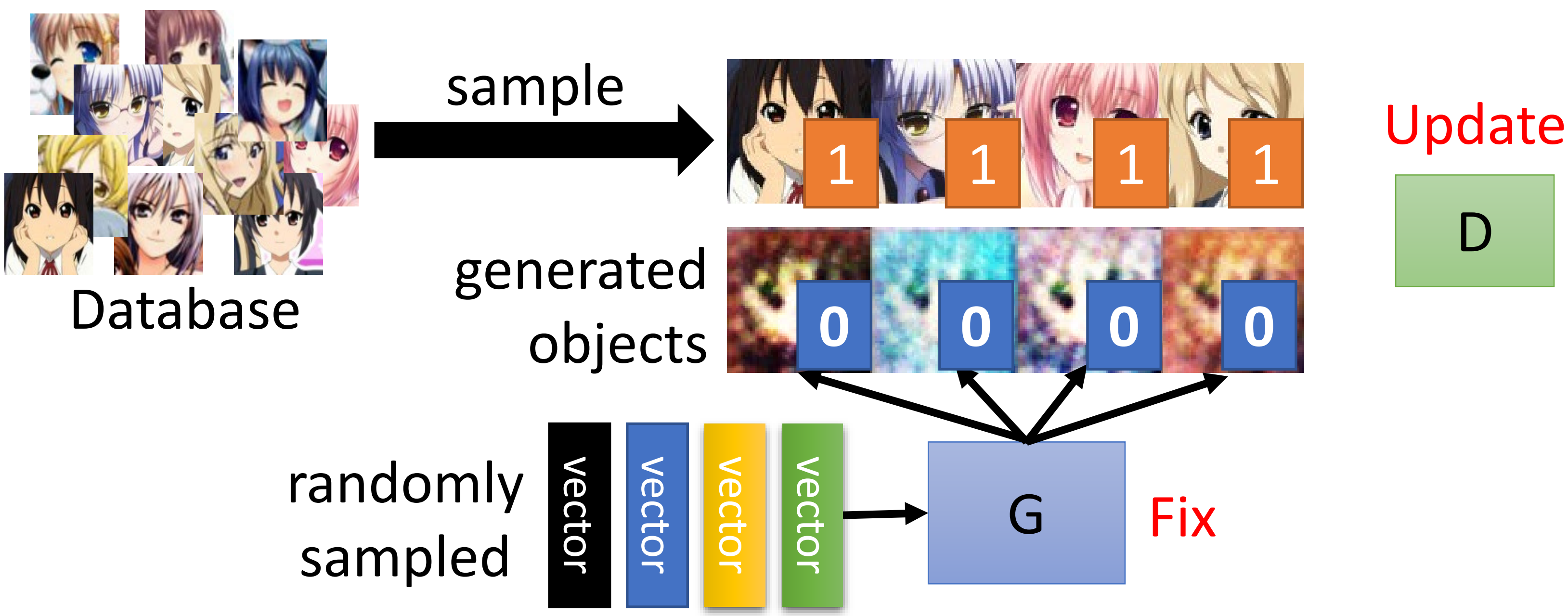
Algorithm

- Initialize generator and discriminator



- In each training iteration:

Step 1: Fix generator G, and update discriminator D



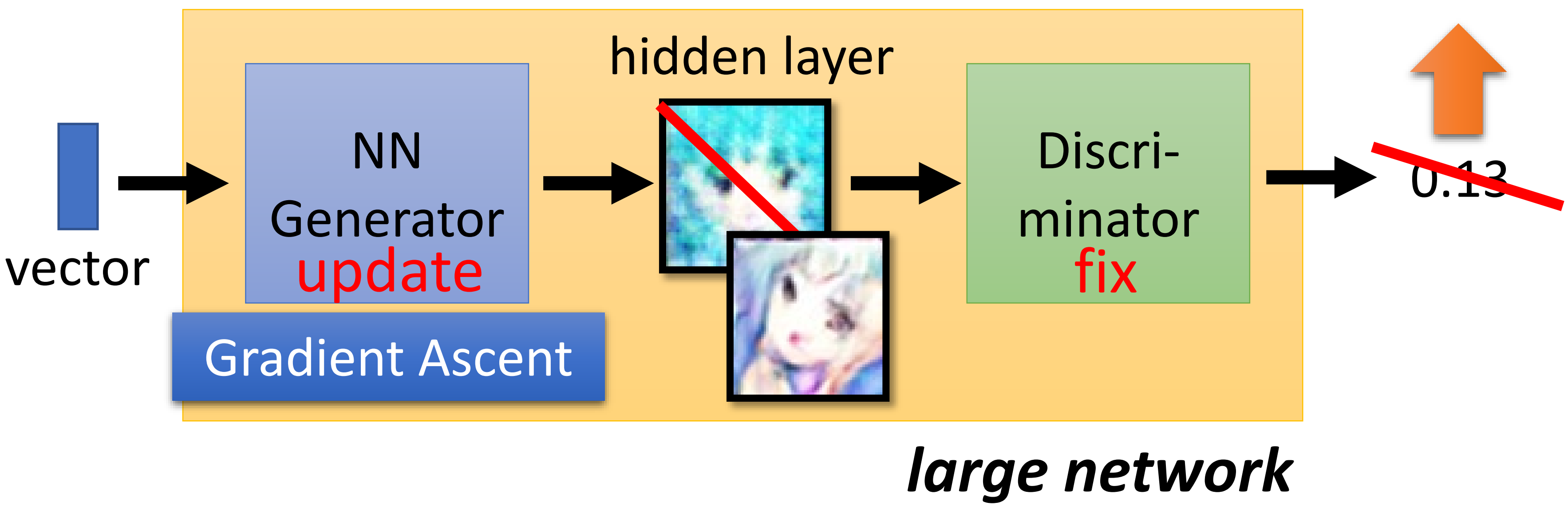
Discriminator learns to assign high scores to real objects and low scores to generated objects.

Algorithm

- Initialize generator and discriminator
- In each training iteration:

Step 2: Fix discriminator D, and update generator G

Generator learns to "fool" the discriminator



Algorithm Initialize θ_d for D and θ_g for G

- In each training iteration:

Learning D

- Sample m examples $\{x^1, x^2, \dots, x^m\}$ from database
- Sample m noise samples $\{z^1, z^2, \dots, z^m\}$ from a distribution
- Obtaining generated data $\{\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^m\}$, $\tilde{x}^i = G(z^i)$
- Update discriminator parameters θ_d to maximize
  - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} logD(x^i) + \frac{1}{m}\sum_{i=1}^{m} log\left(1 - D(\tilde{x}^i)\right)$
  - $\theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d)$

Learning G

- Sample m noise samples $\{z^1, z^2, \dots, z^m\}$ from a distribution
- Update generator parameters θ_g to maximize
  - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} log\left(D\left(G(z^i)\right)\right)$
  - $\theta_g \leftarrow \theta_g + \eta \nabla \tilde{V}(\theta_g)$

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

---

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ 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 $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ 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

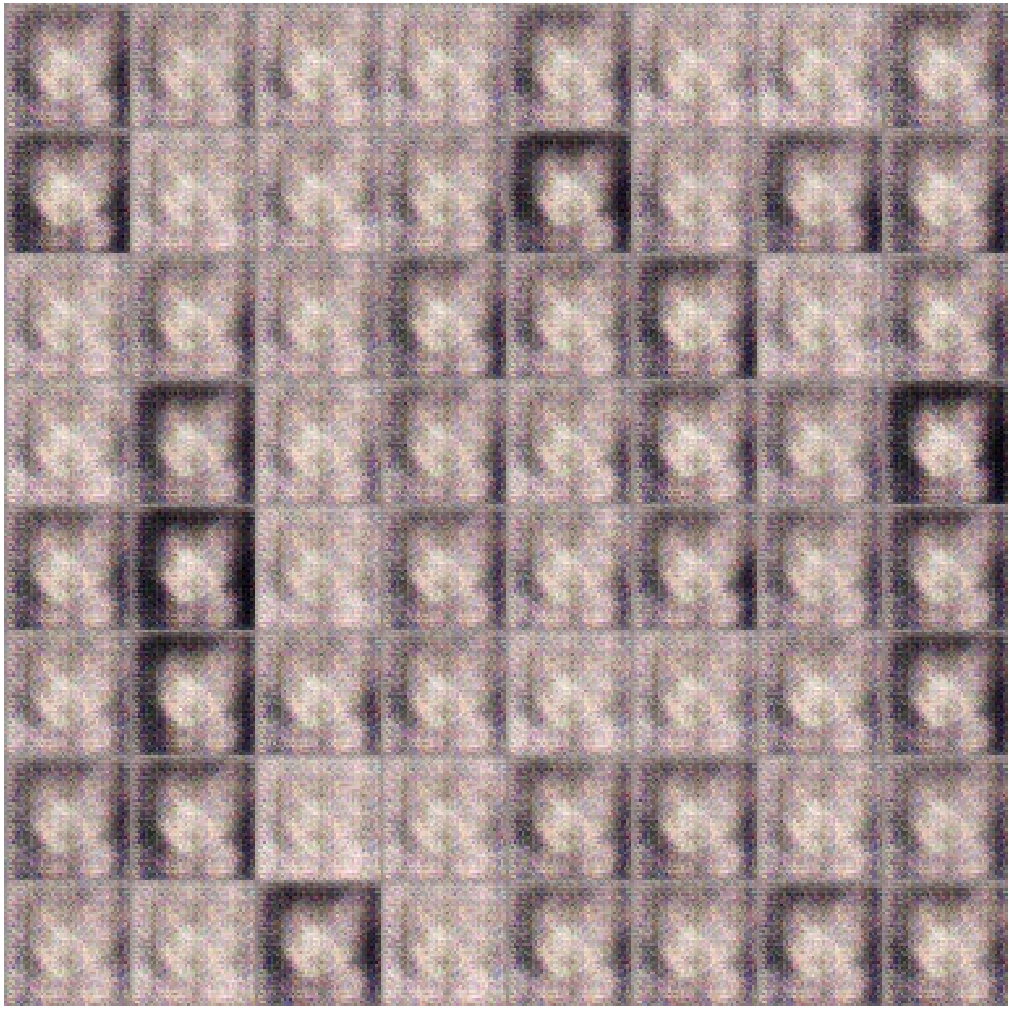
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

D and *G* play the following two-player **minimax game** with value function *V (G;D)*

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

Ian J. Goodfellow et.al, Generative Adversarial Nets. NIPS 2014: 2672-2680

Anime Face Generation



100 updates

Source of training data: https://zhuanlan.zhihu.com/p/24767059

Anime Face Generation

1000 updates

Anime Face Generation

2000 updates

Anime Face Generation

5000 updates

Anime Face Generation

10,000 updates

Anime Face Generation

20,000 updates

Anime Face Generation

50,000 updates

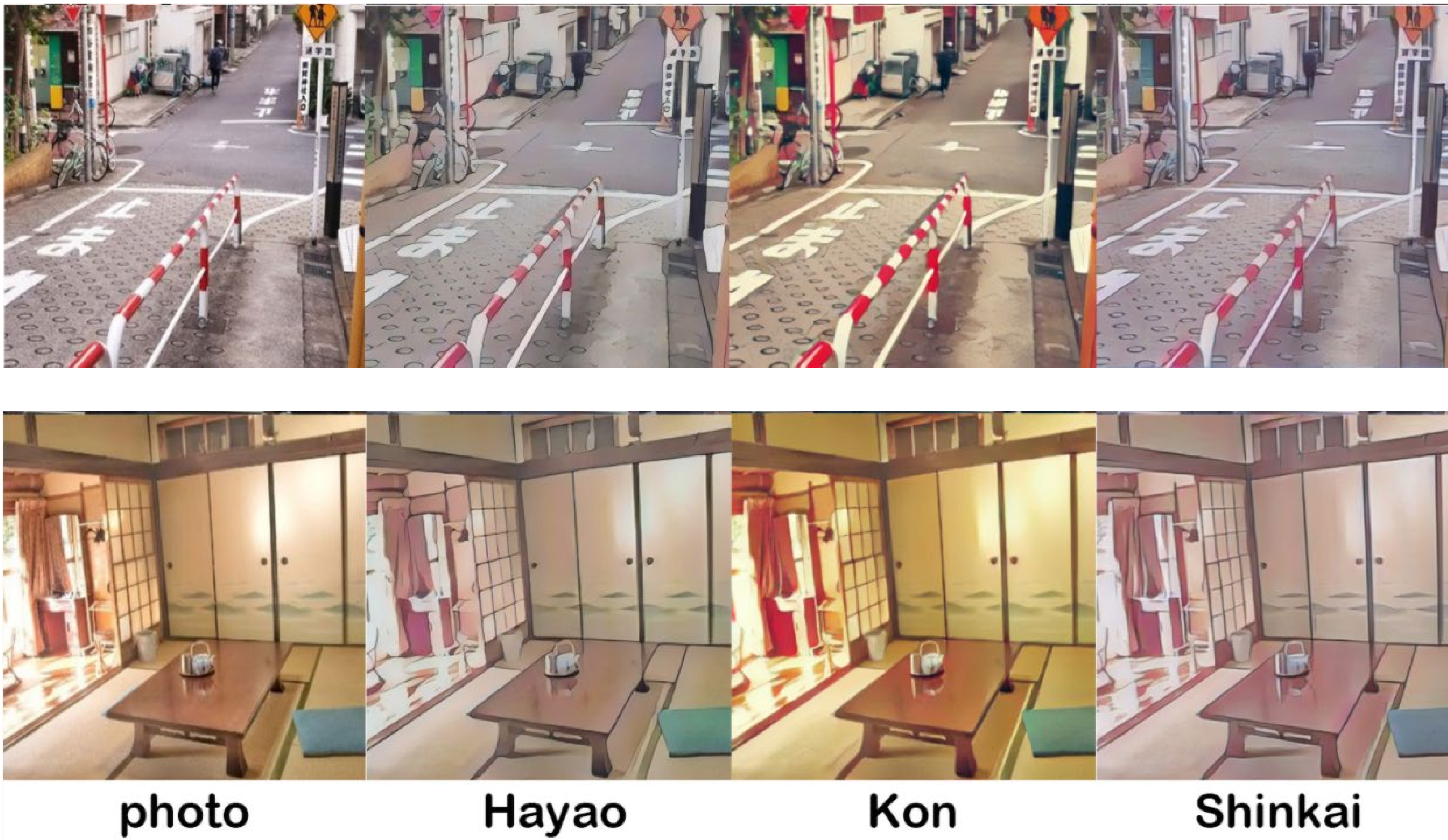
The faces generated by machine.

AnimeGAN

AnimeGAN

Photo CartoonGAN ComixGAN AnimeGAN

AnimeGAN v2



References and Acknowledgement

- Deep Generative Models, MIT 6.S191, Alexander Amini, 2019
- Introduction of Generative Adversarial Network (GAN), Hung-yi Lee