Unsupervised Learning by Generative Adversarial Networks

Yanbei Chen, Shaogang Gong

Queen Mary University of London

yanbei.chen@qmul.ac.uk, s.gong@qmul.ac.uk

Overview

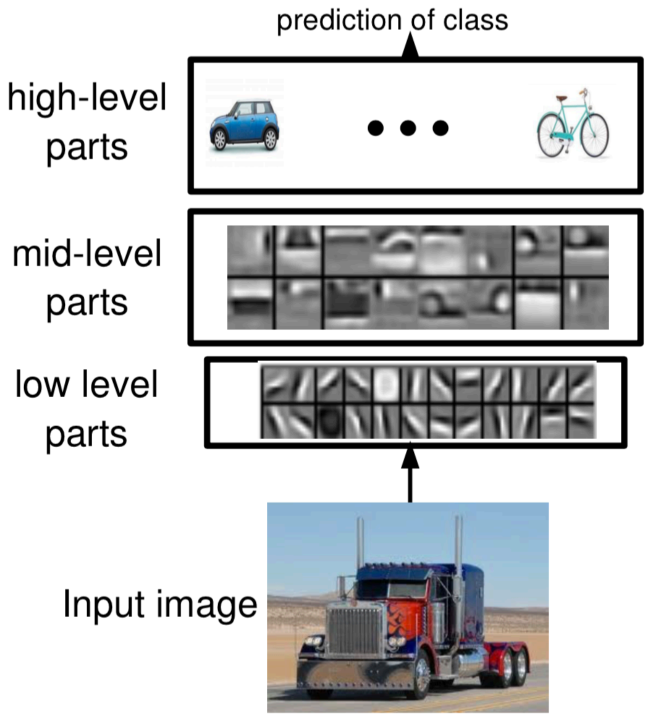
- **Supervised Learning**
- **Unsupervised Learning & Why Unsupervised Learning**
- **Background on Unsupervised Learning**
  - Auto-Encoder (2000s - 2013) – One of the simplest form in unsupervised learning
- **Generative Adversarial Networks (2014-present)**
- **Research Frontiers in GANs**
  - Different variations of GANs
  - Applications based on GANs

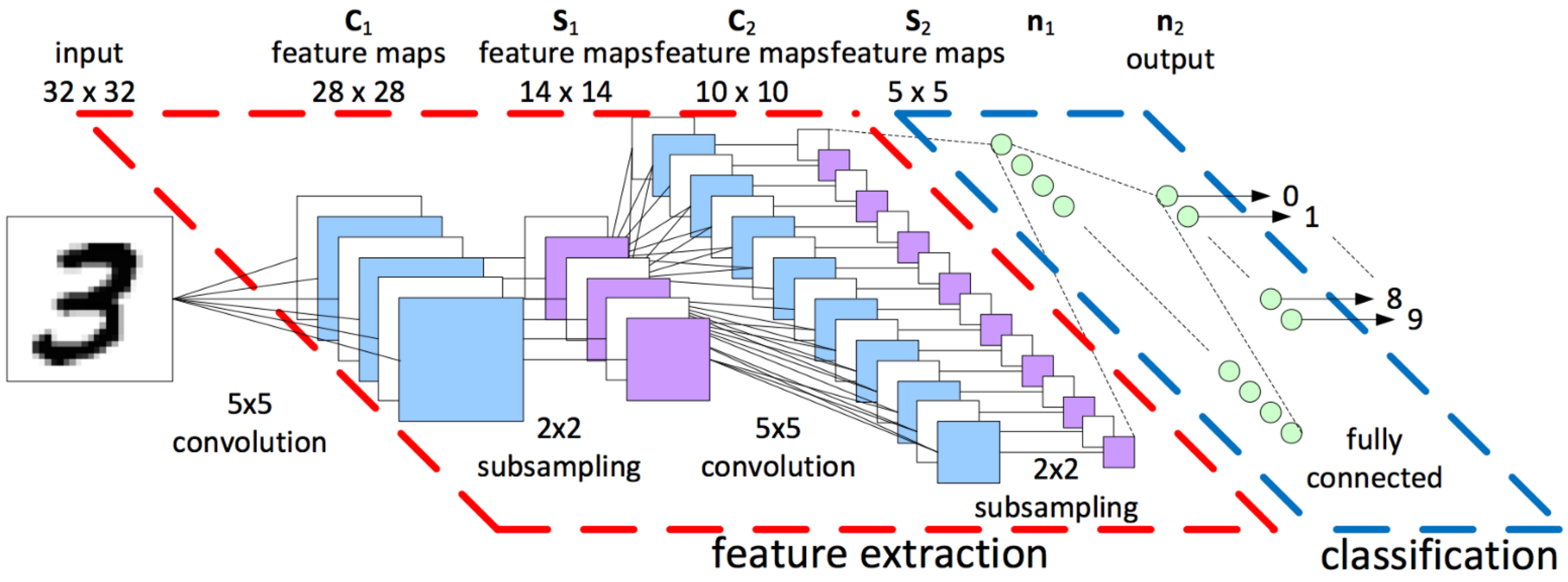
Overview

- **Supervised Learning**

Supervised Learning

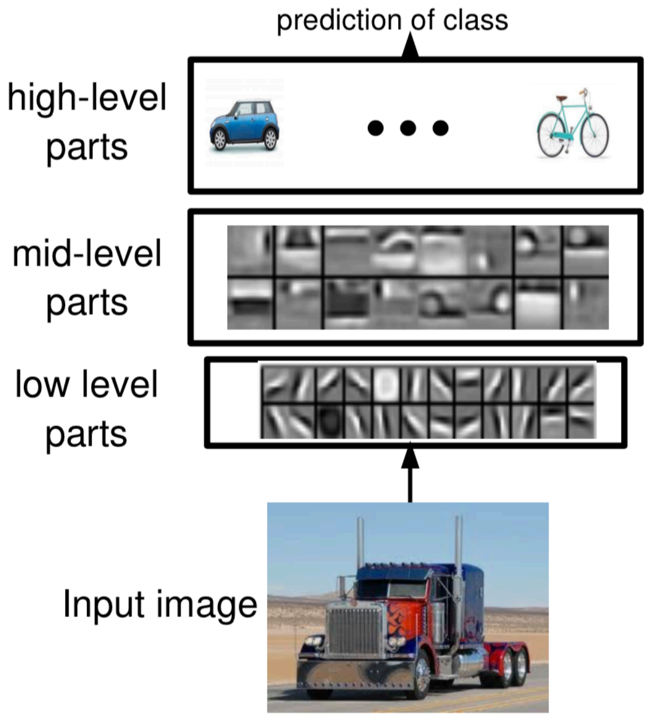
- **Data:** $\{(x_i,\ y_i)\}_{i=1,2,\ldots,n}$
- **Goal:** Learn a mapping function: $f(): x \rightarrow y$
- **Examples:** Image classification

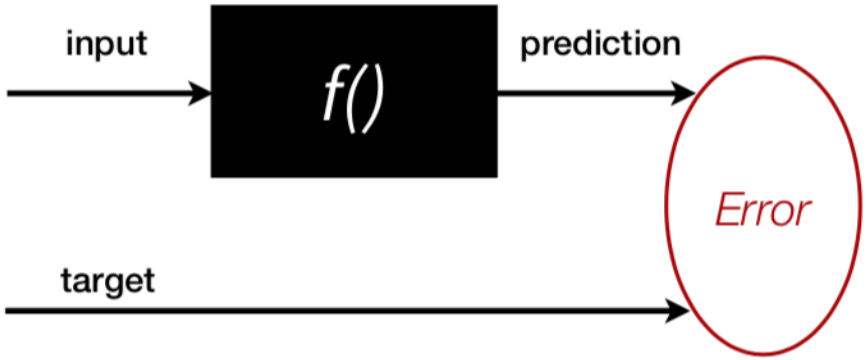




Supervised Learning

- **Data:** $\{(x_i,\ y_i)\}_{i=1,2,\ldots,n}$, label y_i is given during training
- **Goal:** Learn a mapping function: $f\,()$: $x \rightarrow y$
- **Examples:** Image classification





Objective: Learn $f\,()$ to predict image classes by minimizing a cross entropy loss:

$$L(x, y; \theta) = -\sum y_i \log(p(c_i|x_i))$$

Overview

- Supervised Learning
- **Unsupervised Learning & Why Unsupervised Learning**

Unsupervised Learning

- **Data:** $\{(x_i)\}_{i=1,2,\ldots,n}$, no label y_i is given during training
- **Goal:** Learn some underlying structure of data x
- **Examples:** clustering

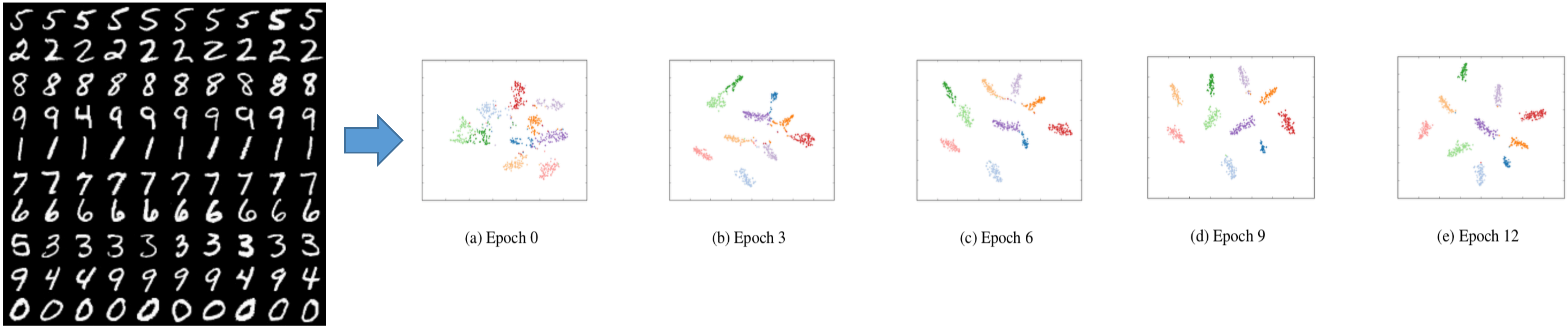


Figure copyright and adapted from Xie et al. [ICML2016]Unsupervised Deep Embedding for Clustering Analysis

Why Unsupervised Learning?

➢ Exploitation of redundant unlabelled data: often easy to acquire from different search engines.

Why Unsupervised Learning?

- Generate realistic images for various real-world image processing applications such as: image style transfer; semantic manipulation; image colorization, etc.

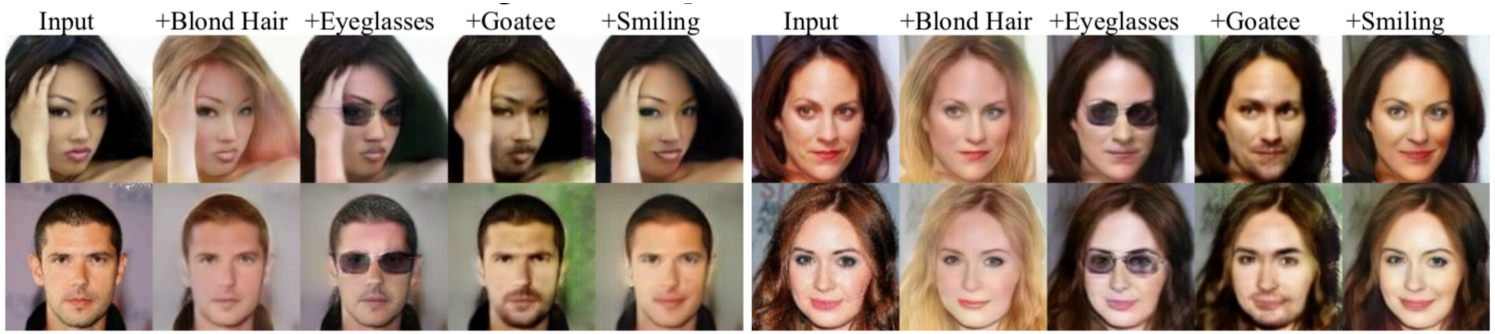


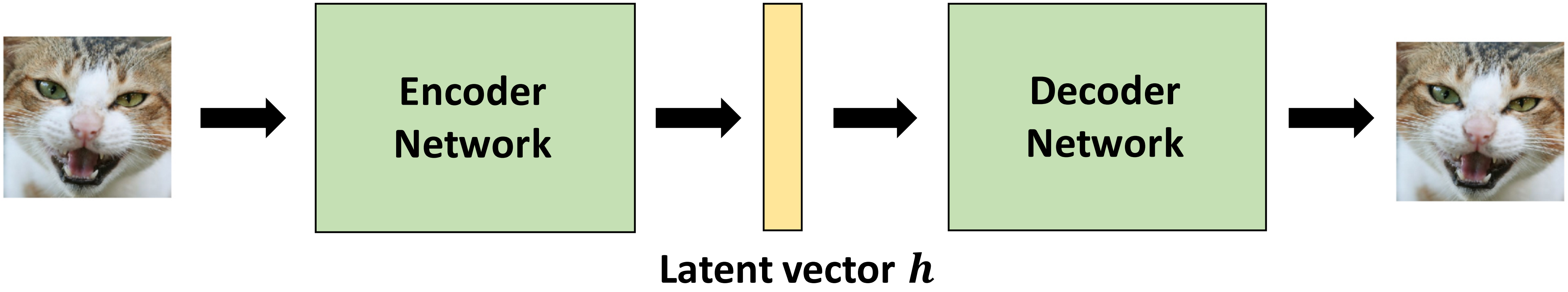
Figure copyright and adapted from Liu et al. [NIPS2017]Unsupervised Image-to-Image Translation Networks
Isola et al. [CVPR2017]Image-to-Image Translation with Conditional Adversarial Networks

Overview

- Supervised Learning
- Unsupervised Learning & Why Unsupervised Learning
- **Background on Unsupervised Learning**
  - Auto-Encoder (2000s - 2013) – One of the simplest form in unsupervised learning

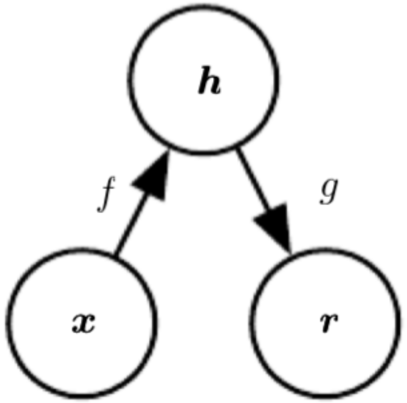
Background on Unsupervised Learning: Auto-Encoder

- **Auto-Encoder:** A type of neural network that learns to generate the output data r as close as to the input x as possible.
- **Architecture:** Auto-Encoder is consist of an encoder network (map input x to latent code h) and a decoder network (map h to reconstructed input r).



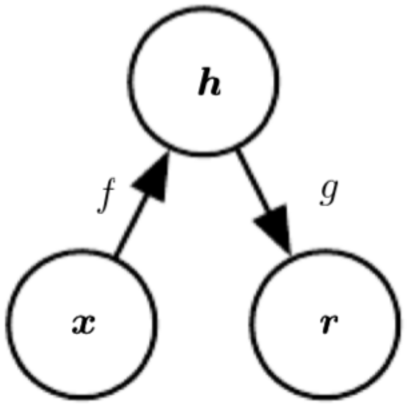
Background on Unsupervised Learning: Auto-Encoder

- **Goal:** Learn encoder f and decoder (f, g): $x \rightarrow h \rightarrow r$
  - **Encoder:** Project the higher-dimensional input data x into a lower-dimensional latent space
  - **Decoder:** Reconstruct the lower-dimensional latent code h into an output r that resemble input x



Background on Unsupervised Learning: Auto-Encoder

- **Goal:** Learn encoder f and decoder g: f, g: $x \rightarrow h \rightarrow r$



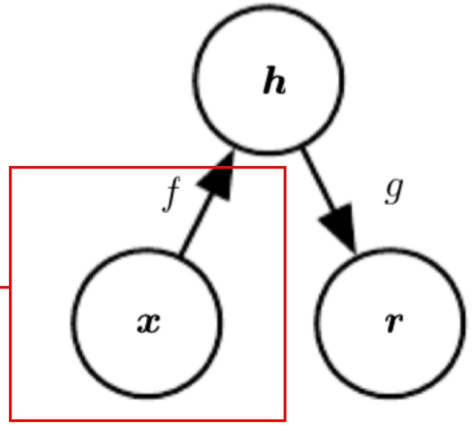
- **Objective:** Minimizing reconstruction L2 loss:

$$L(x, y; \theta) = -\frac{1}{M}\sum_{i=1}^{M} ||x_i - r_i||^2$$

Background on Unsupervised Learning: Auto-Encoder

- **Goal:** Learn encoder f and decoder g: f, g: $x \to h \to r$

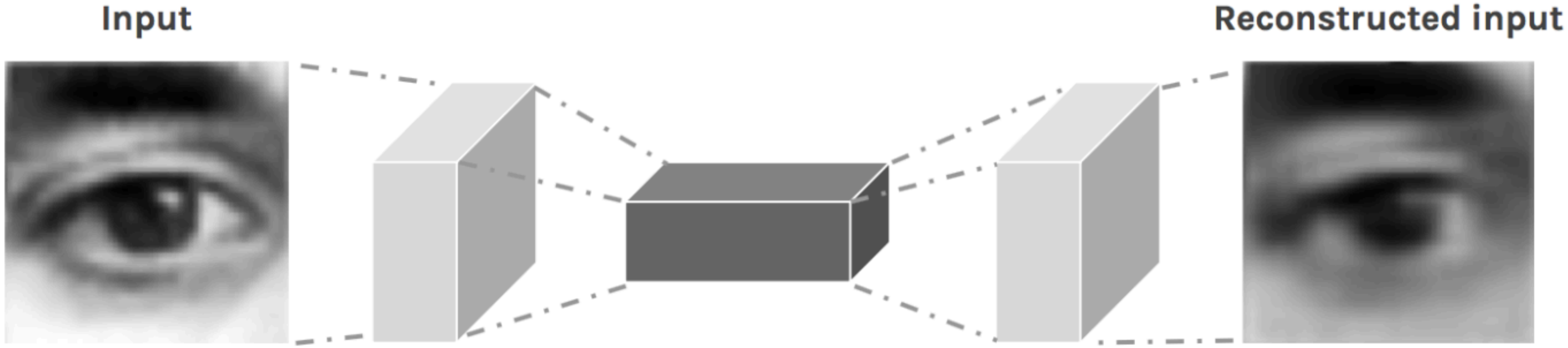
Use the encoder as
a feature extractor
to extract features



- **Outcome:** The encoder can be used as a feature extractor since the lower-dimensional latent code h can capture the input data characteristics for reconstructing the input data.

Background on Unsupervised Learning: Auto-encoder

- **Pros:**
  - 1) Compress data; 2) Extract features;
  - 3) Image Restoration; 4) Image Colorization.
- **Cons:**
  - Likely to generate blurry images of low quality that do not stimulate real images.



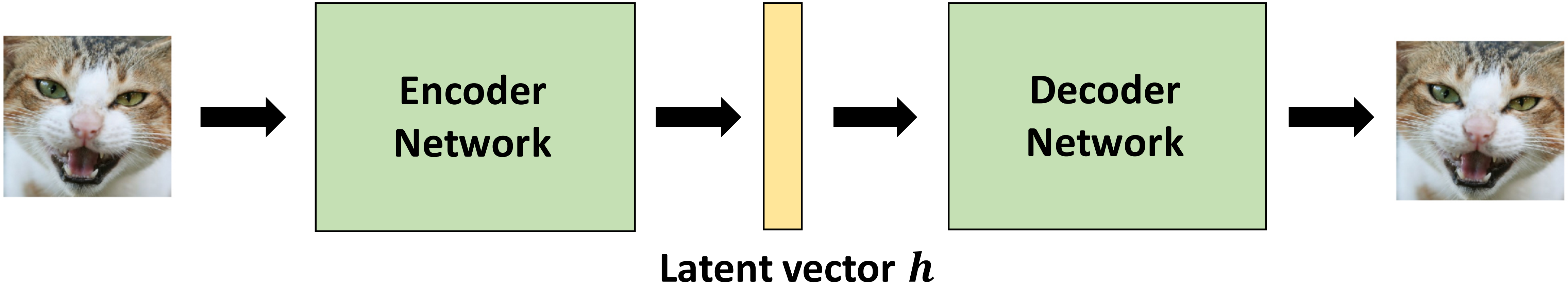
Interesting applications based on auto-encoder: 1) Zhang et al. [ECCV2016]Colorful Image Colorization; 2) Mao et al. [NIPS2016]Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections

Overview

- Supervised Learning
- Unsupervised Learning & Why Unsupervised Learning
- Background on Unsupervised Learning
    - Auto-Encoder (2000s - 2013) – One of the simplest form in unsupervised learning
- **Generative Adversarial Networks (2014-present)**

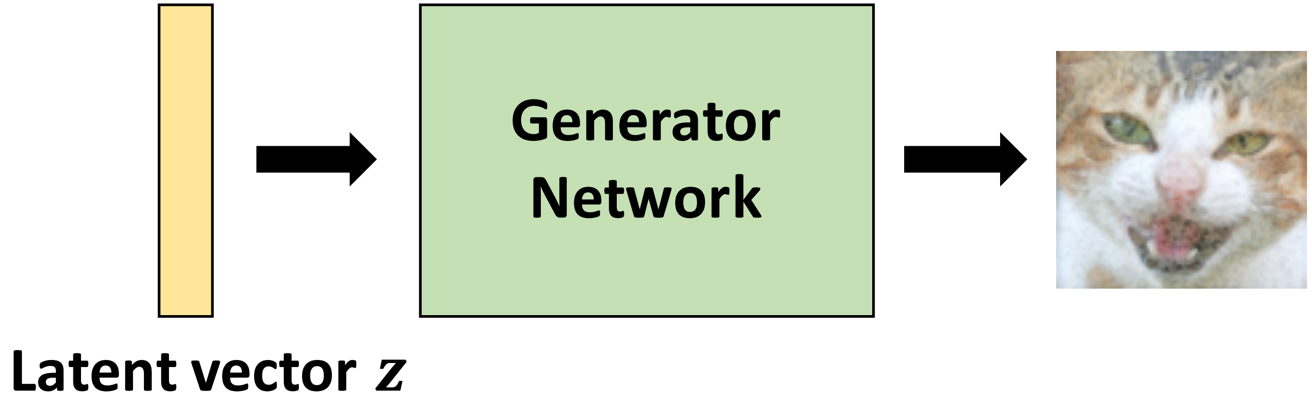
Generative Adversarial Networks (GANs)

- **GANs:** A type of generative models that learn to generate samples through a two-player minimax game.
- **Contrary to Auto-Encoder** that use an image as an input to generate the corresponding output that resembles the input.



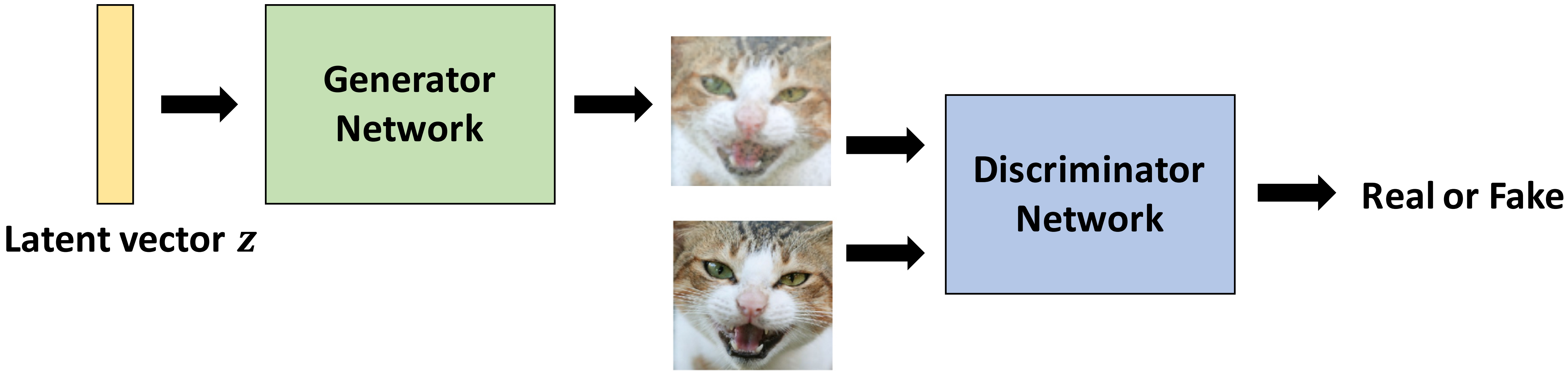
Generative Adversarial Networks (GANs)

- **GANs:** A type of generative models that learn to generate samples through a two-player minimax game.
- **Contrary to Auto-Encoder**, GANs aim to generate samples from a simple distribution, i.e. use Gaussian random noise as input z.



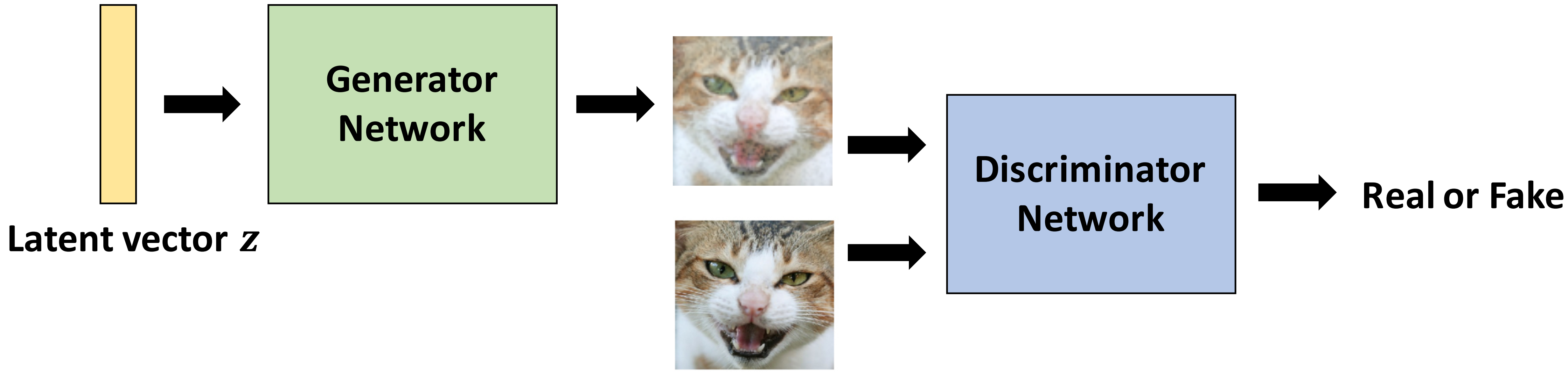
Generative Adversarial Networks (GANs)

- **Architecture:** generator network + discriminator network



Generative Adversarial Networks (GANs)

- **Architecture:** generator network + discriminator network
  - **Generator Network:** learn to fool the discriminator by generating real-looking images (as real as possible to the groundtruth)
  - **Discriminator Network:** learn to differentiate between the generated images (fake) and the groundtruth images (real)



Generative Adversarial Networks (GANs)

- **Architecture:** generator network + discriminator network
  - **Generator Network:** learn to fool the discriminator by generating real-looking images (as real as possible to the groundtruth)
  - **Discriminator Network:** learn to differentiate between the generated images (fake) and the groundtruth images (real)
- **Objective (two-player minimax game):**

$$\min_{\theta_g}\max_{\theta_d}\left[\mathbb{E}_{x\sim p_{data}}\log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data x}} + \mathbb{E}_{z\sim p(z)}\log(1-\underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data G(z)}})\right]$$

- Remark: Discriminator outputs a value between 0 and 1 to denote the likelihood of being real

Generative Adversarial Networks (GANs)

- **Objective (two-player minimax game):**

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data x}} + \mathbb{E}_{z \sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data G(z)}}) \right]$$

- Remark: Discriminator outputs a value between 0 and 1 to denote the likelihood of being real
- **Discriminator Network:**
  - maximise the objective such that D(x) is close to 1, i.e. when given real images x; while D(G(z)) is close to 0, i.e. when given fake images G(z).
- **Generator Network:**
  - minimise the objective such that D(G(z)) is close to 1, i.e. try to generate image G(z) look as genuine as the real image x

Generative Adversarial Networks (GANs)

- **Objective (two-player minimax game): :**

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data x}} + \mathbb{E}_{z \sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data G(z)}}) \right]$$

- **Training GANs:** alternative updates on discriminator and generator
  1. Take the gradient descent on the discriminator

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

  2. Take the gradient descent on the generator

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

Generative Adversarial Networks (GANs)

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 $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ 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
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

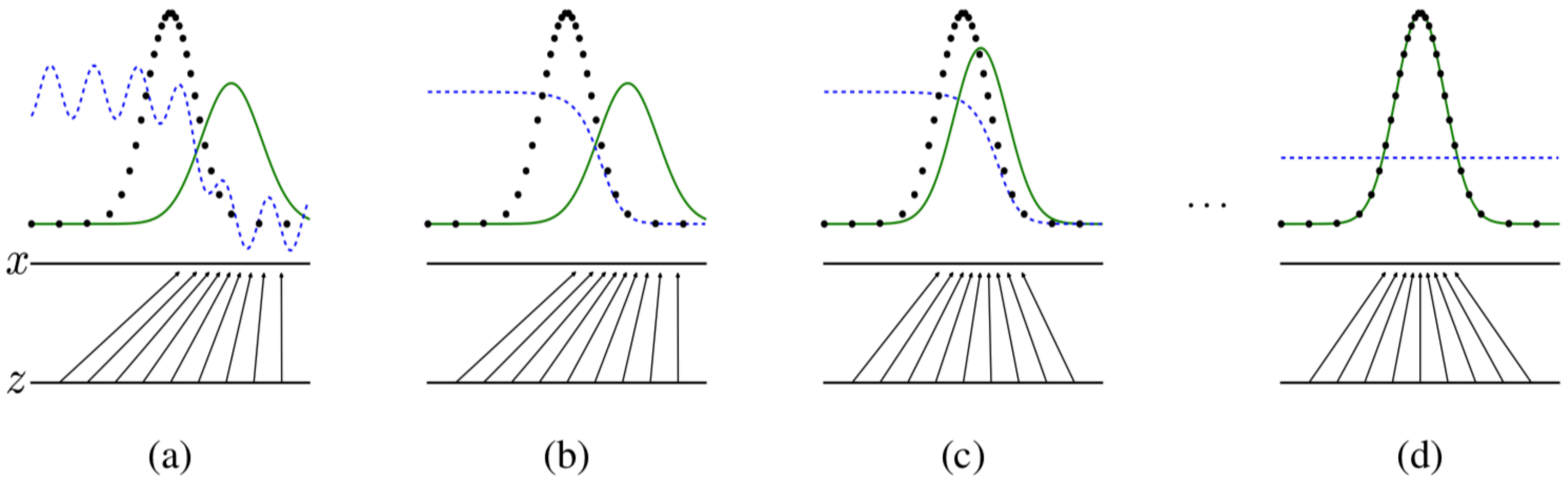
An improved objective:

- **Update the generator by ascending its stochastic gradient**

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(D_{\theta_d}(G_{\theta_g}(z^{(i)})))$$

Generative Adversarial Networks (GANs)

- **Evolution of the training process:**

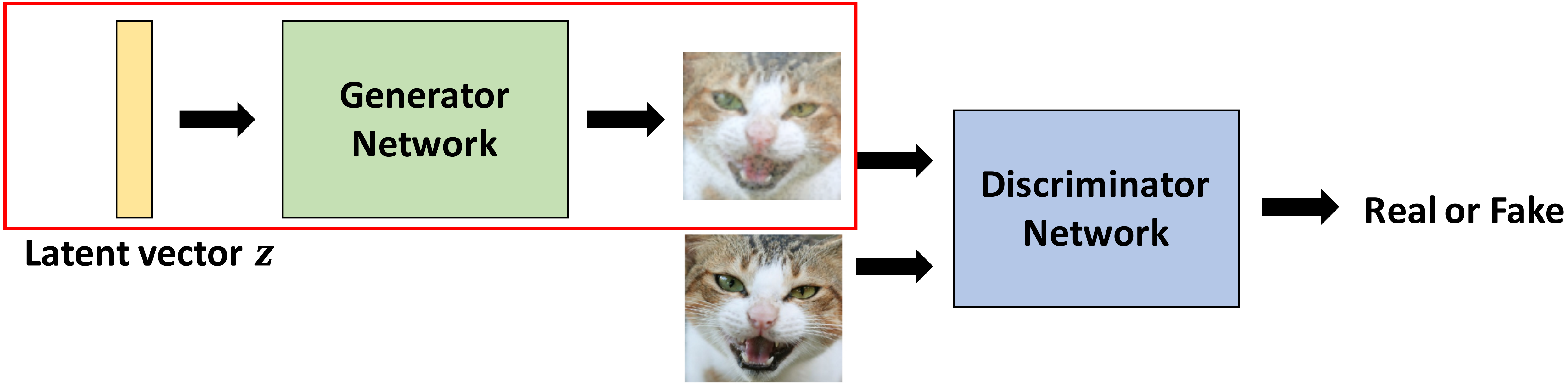


- Blue dashed line: distribution of discriminator $D(x)$.
- Green solid line: distribution of generate $p_g(x)$.
- Black dotted line: true data distribution $p_{data}(x)$.
- **Global optimum:** $\boldsymbol{D(x) = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)} = \frac{1}{2}}$.

Figure copyright and adapted from Goodfellow et al. [NIPS2014] Generative Adversarial Nets

Generative Adversarial Networks (GANs)

- **Outcome:** Use the generator to generate new images given input random noise z sampled from a simple distribution



Generative Adversarial Networks (GANs)

- **Visualization:**
  - Generated samples are not memorized samples from the training set

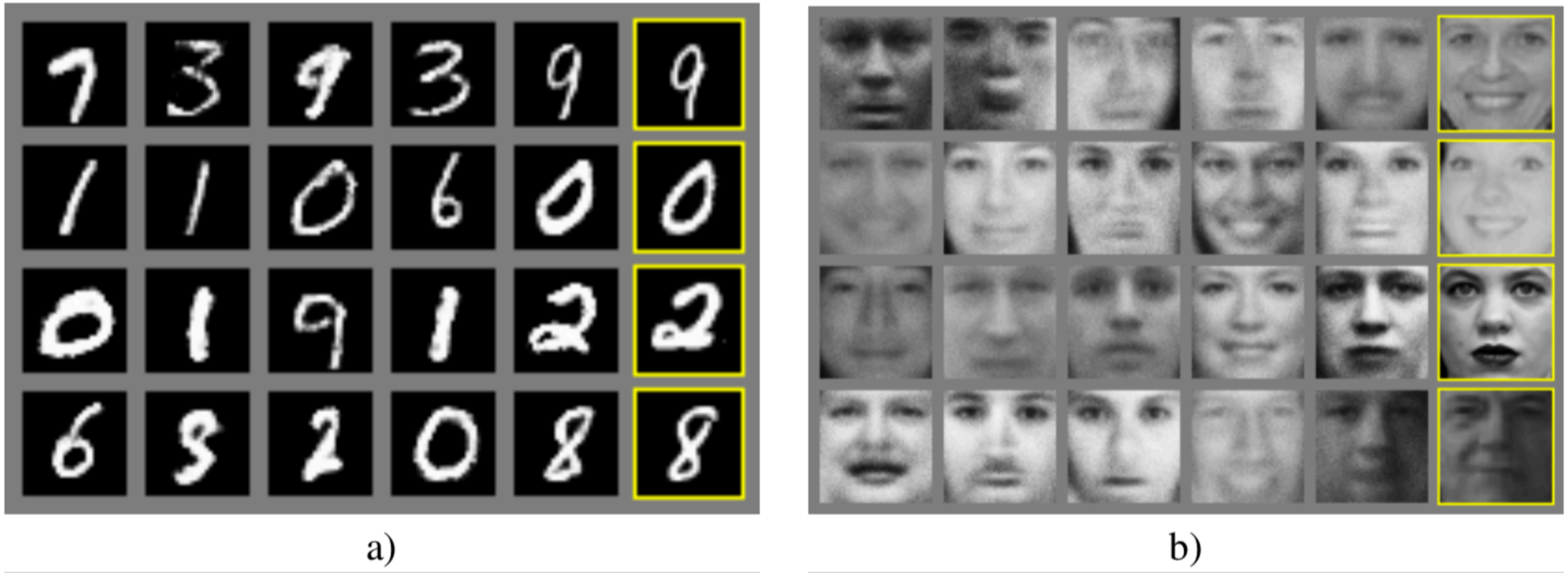


Figure copyright and adapted from Goodfellow et al. [NIPS2014] Generative Adversarial Nets

Generative Adversarial Networks (GANs)

- **Visualization:**
  - Changing the shape or type of the digits by interpolating between coordinates in $\boldsymbol{z}$ space

Figure copyright and adapted from Goodfellow et al. [NIPS2014] Generative Adversarial Nets

Overview

- Supervised Learning
- Unsupervised Learning & Why Unsupervised Learning
- Background on Unsupervised Learning
  - Auto-Encoder (2000s - 2013) – One of the simplest form in unsupervised learning
- Generative Adversarial Networks (2014-present)
- **Research Frontiers in GANs**
  - Different variations of GANs
  - Applications based on GANs

Research Frontiers in GANs: Different variations of GANs - DCGAN

- DCGAN: Deep Convolutional GAN

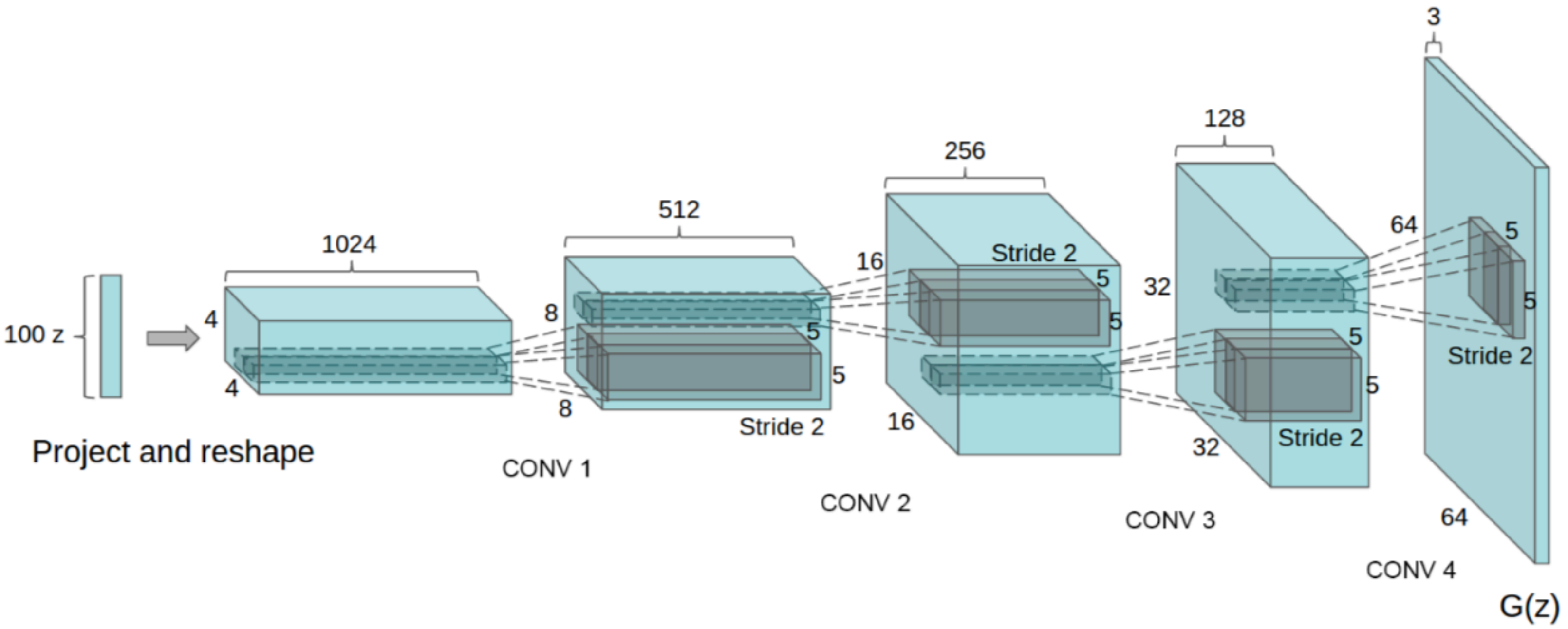


Figure copyright and adapted from Radford et al.
[ICLR2016]Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

Research Frontiers in GANs: Different variations of GANs - DCGAN

- DCGAN: Deep Convolutional GAN

Figure copyright and adapted from Radford et al.
[ICLR2016]Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

Research Frontiers in GANs: Different variations of GANs - DCGAN

- DCGAN: Deep Convolutional GAN – Interpolation on Z

Figure copyright and adapted from Radford et al.
[ICLR2016]Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

Research Frontiers in GANs: Different variations of GANs - DCGAN

- DCGAN: Deep Convolutional GAN - Vector arithmetic for visual concepts

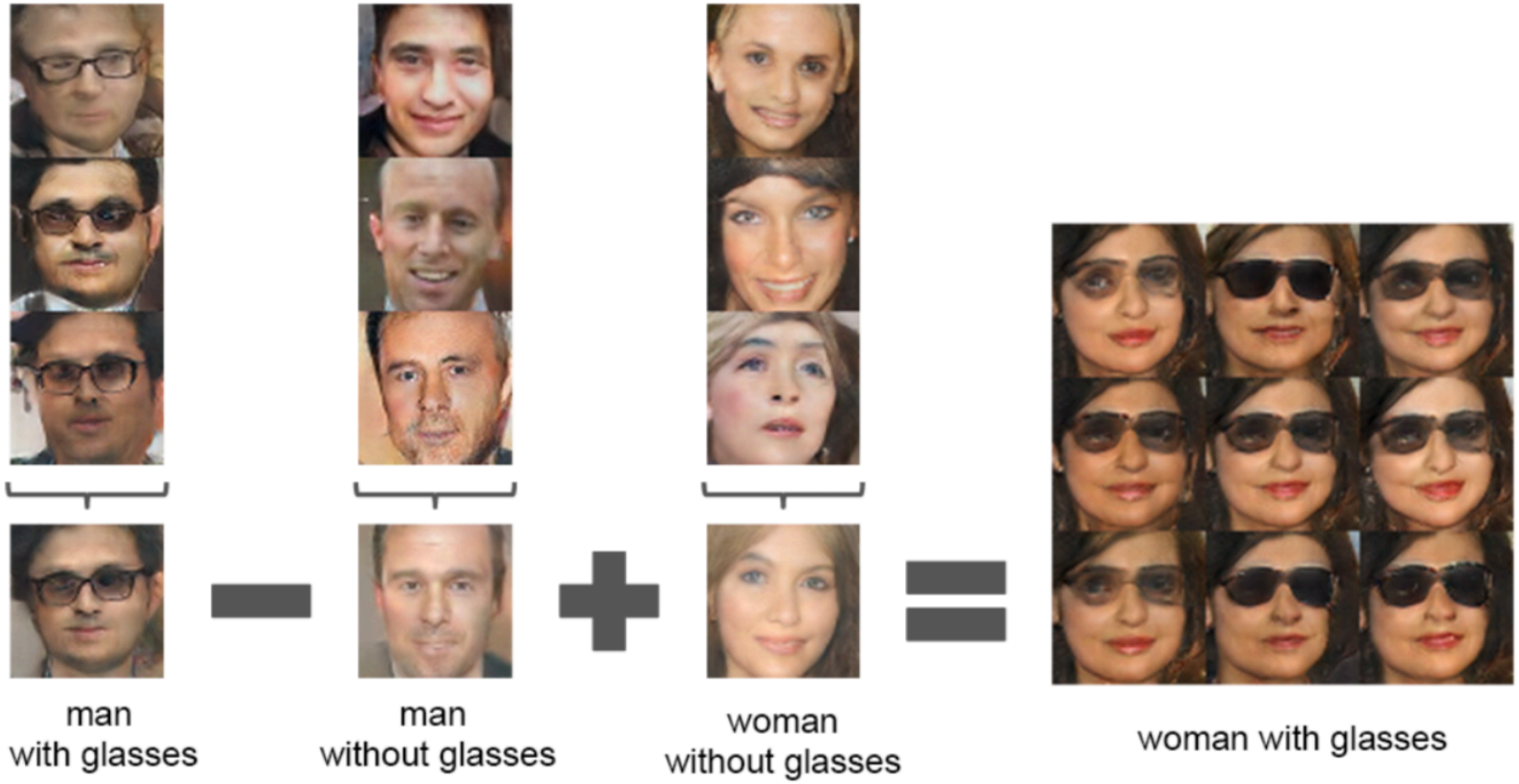
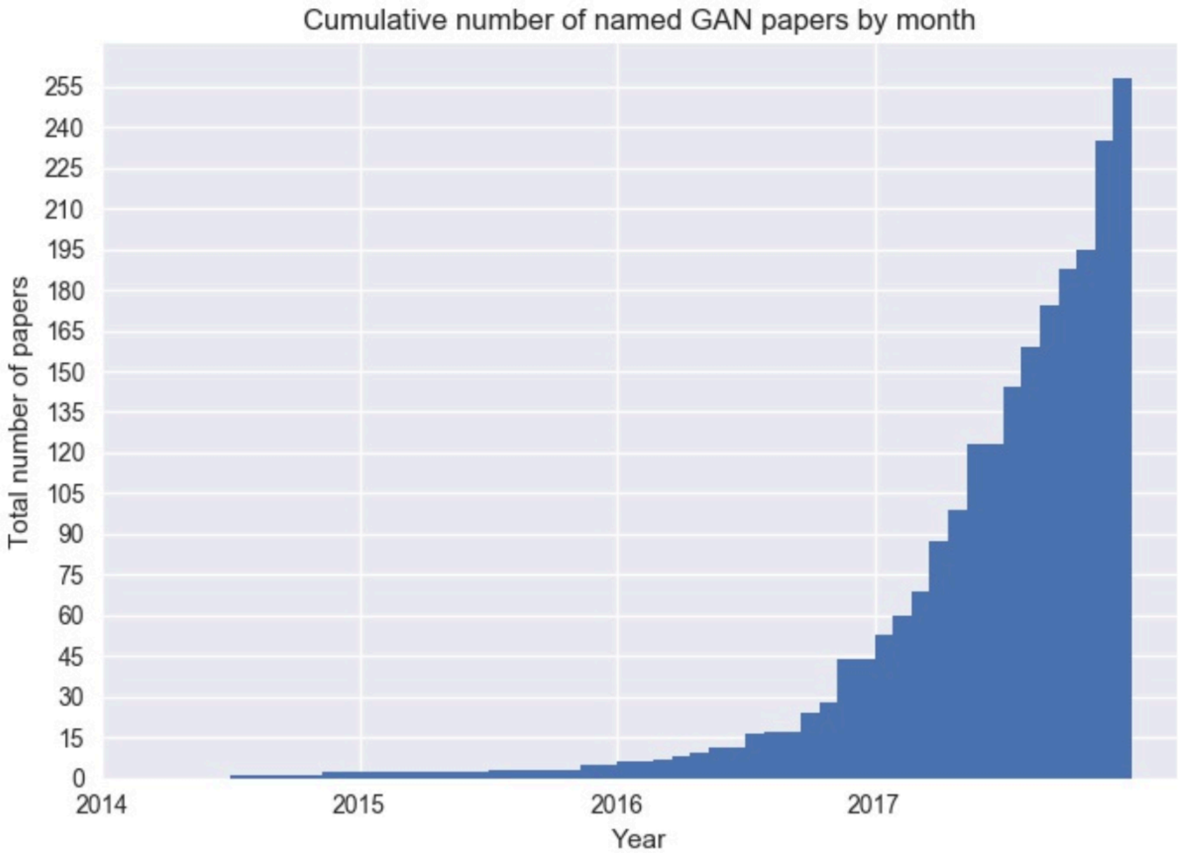


Figure copyright and adapted from Radford et al.
[ICLR2016]Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

Research Frontiers in GANs: Different variations of GANs - DCGAN

- Different variations of GANs are created since 2014.



More details refer to https://github.com/hindupuravinash/the-gan-zoo.

Research Frontiers in GANs: Applications based on GANs

- **image translation, image style transfer (CycleGAN)**

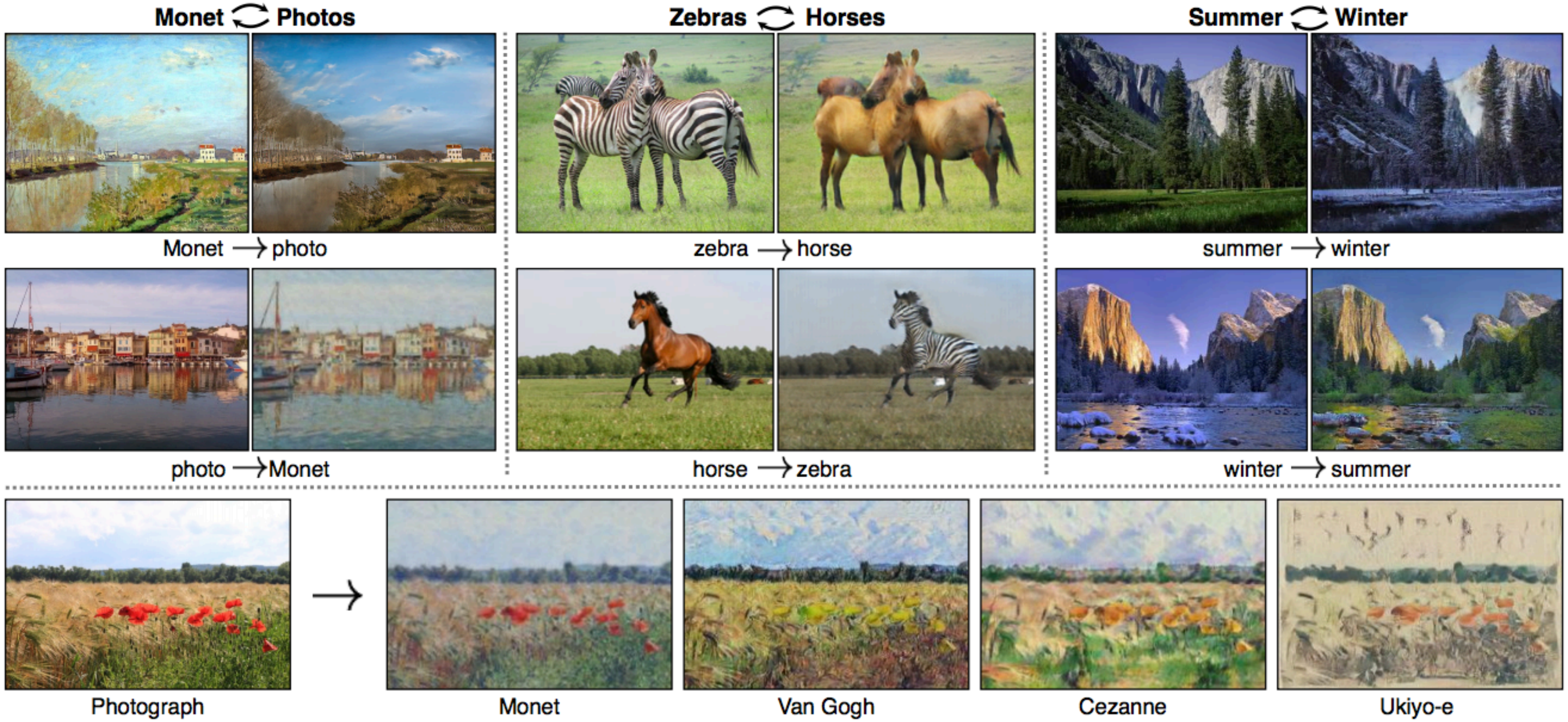


Figure copyright and adapted from Zhu et al.
[CVPR2017]Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Research Frontiers in GANs: Applications based on GANs

- **image super-resolution**

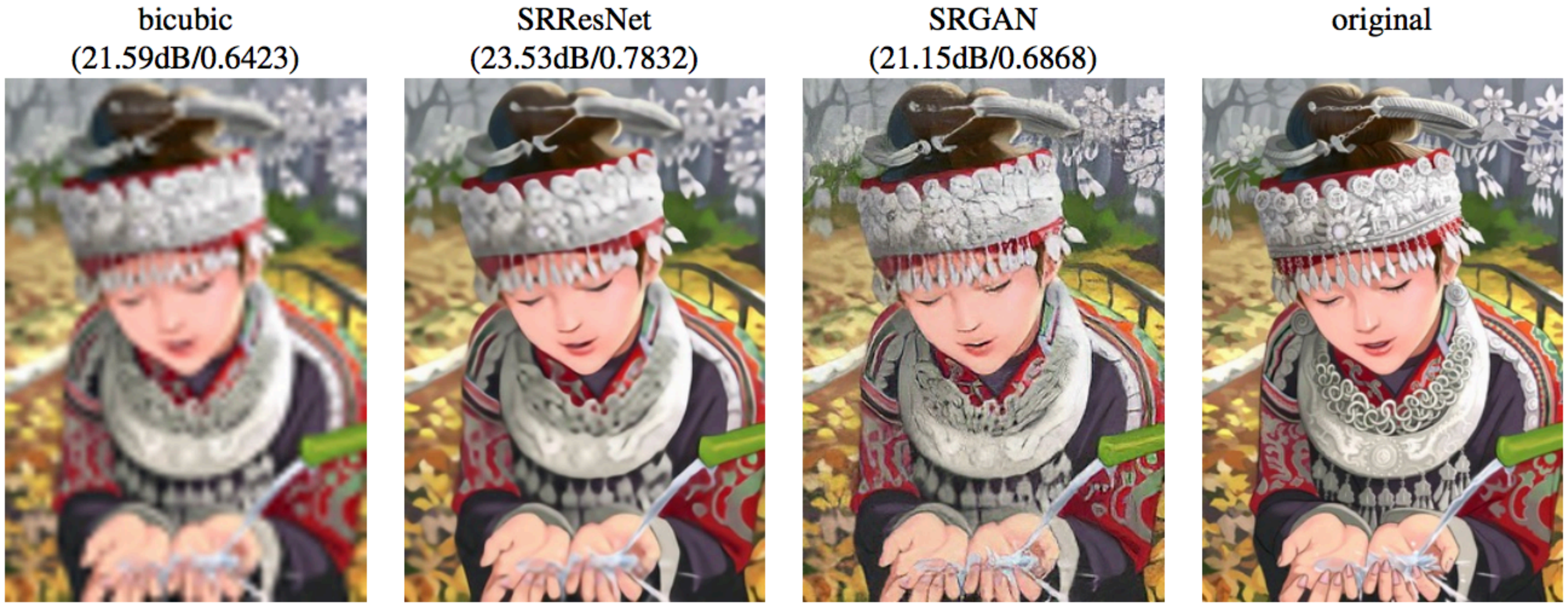


Figure copyright and adapted from Ledig et al.
[CVPR2017]Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

Research Frontiers in GANs: Applications based on GANs

- **data augmentation:** train CNN using refined simulated data by GAN to avoid the expensive human labeling

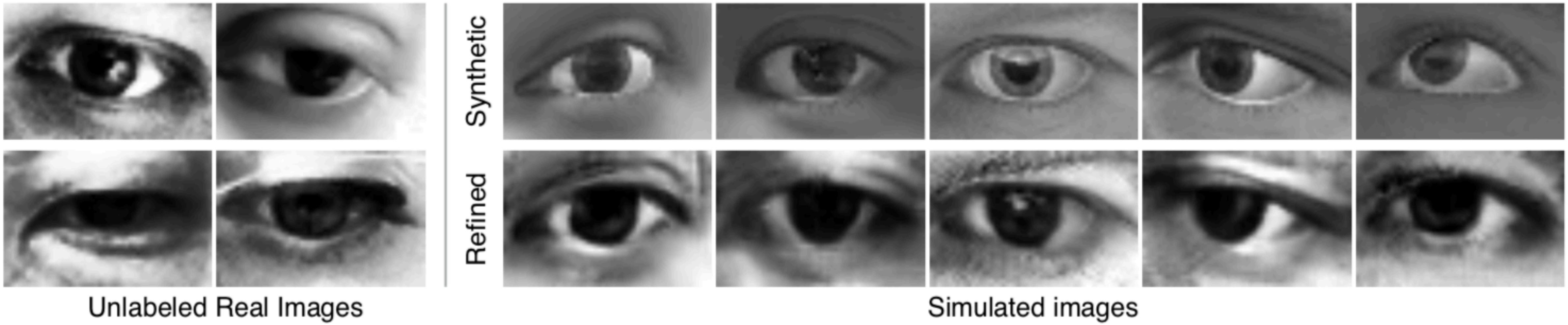


Figure copyright and adapted from Shrivastava et al.
[CVPR2017]Learning from Simulated and Unsupervised Images through Adversarial Training

References

1. Isola et al. [CVPR2017]Image-to-Image Translation with Conditional Adversarial Networks
2. Zhu et al. [CVPR2017]Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
3. Shrivastava et al. [CVPR2017]Learning from Simulated and Unsupervised Images through Adversarial Training
4. Ledig et al. [CVPR2017]Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network
5. Liu et al. [NIPS2017]Unsupervised Image-to-Image Translation Networks

Questions to think about on GANs

- Understand the network architecture:
  1. What does the network architecture look like?
  2. What is the objective to be optimised for the generator & discriminator respectively?

- Tell whether the model is learn in a *supervised* or *unsupervised* way:
  1. Is there any training labels used during training?

- Identify the potential application scenarios for the model:
  1. What can GANs do in real-world applications?