CS 2770: Computer Vision

Generative Adversarial Networks

Prof. Adriana Kovashka
University of Pittsburgh
April 12, 17, 2018

Plan for today

- Generative models: What are they?
- Techniques: Generative Adversarial Networks
  - Conditional GANs
  - Cycle-consistency loss
  - Dealing with sparse data
  - *Applications interleaved with techniques*
- Generation before GANs
- Generation for synthetic training data

Supervised vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x -> y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.

Supervised vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x -> y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.

Cat

Classification

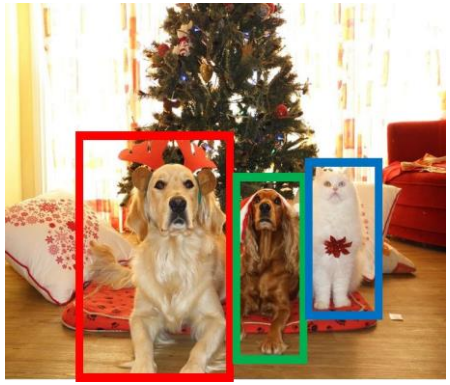
Supervised vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x -> y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.



DOG, **DOG**, **CAT**

Object Detection

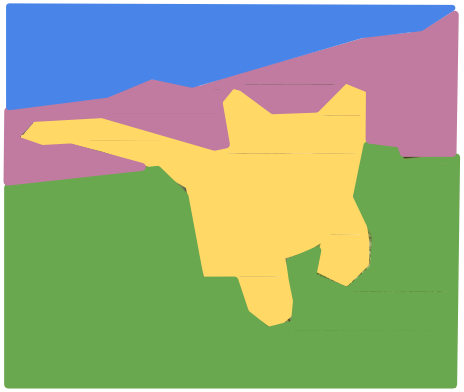
Supervised vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x -> y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.



GRASS, **CAT**, **TREE**, **SKY**

Semantic Segmentation

Supervised vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x -> y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.

A cat sitting on a suitcase on the floor

Image captioning

Supervised vs Unsupervised Learning

Unsupervised Learning

Data: x
Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.

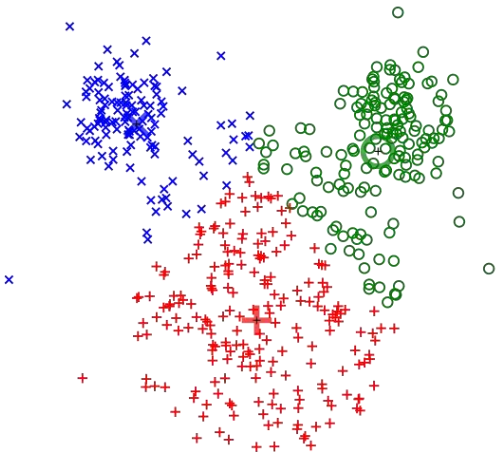
Supervised vs Unsupervised Learning

Unsupervised Learning

Data: x
Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.



K-means clustering

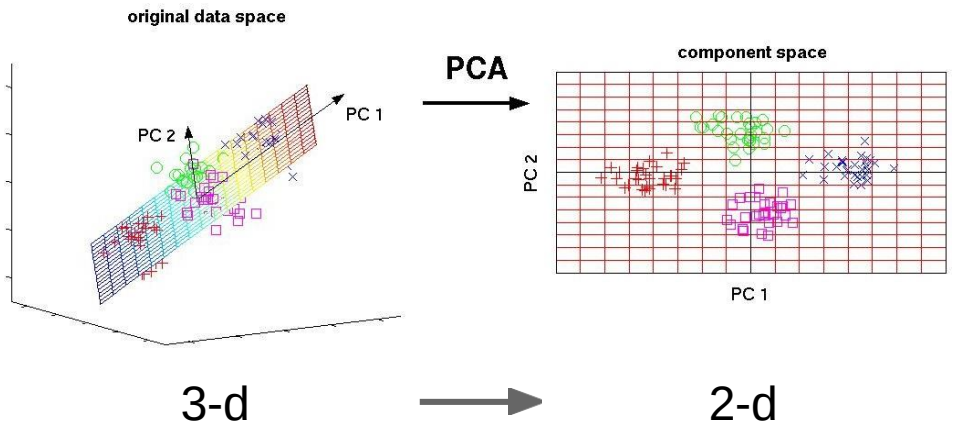
Supervised vs Unsupervised Learning

Unsupervised Learning

Data: x
Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.



Principal Component Analysis
(Dimensionality reduction)

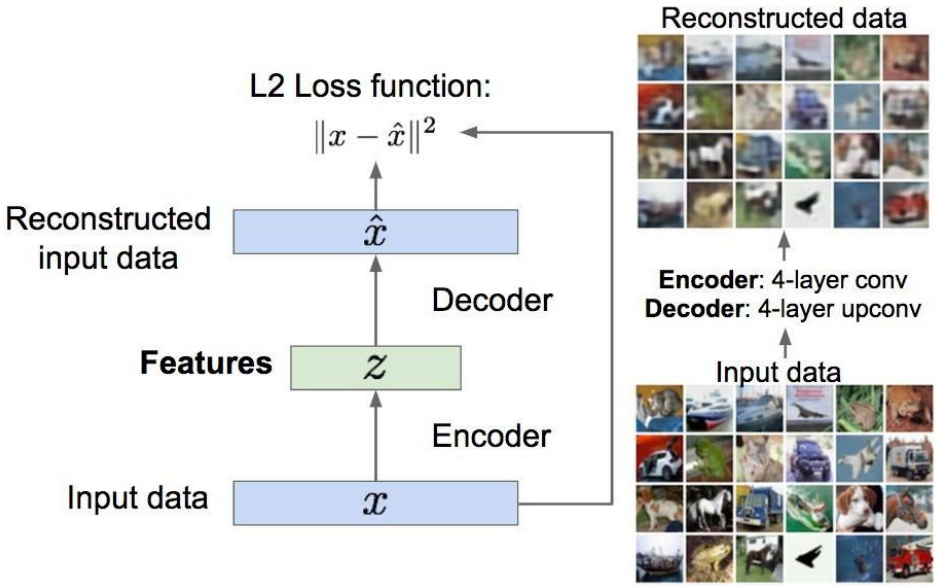
Supervised vs Unsupervised Learning

Unsupervised Learning

Data: x
Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.



Autoencoders
(Feature learning)

Supervised vs Unsupervised Learning

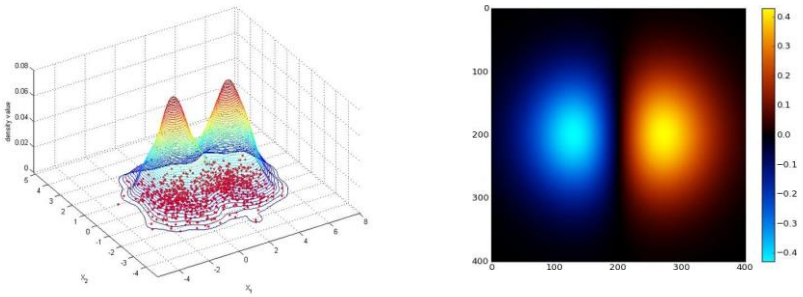
Unsupervised Learning

Data: x
Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.

1-d density estimation



2-d density estimation

Supervised vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x -> y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.

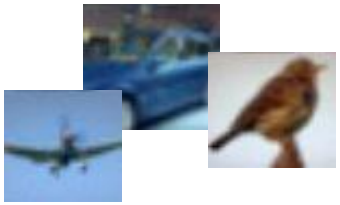
Unsupervised Learning

Data: x
Just data, **no labels!**

Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.

Generative Models

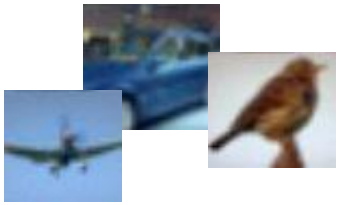


Training data ~ $p_{data}(x)$

Generated samples ~ $p_{model}(x)$

Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

Generative Models



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

Generated samples ~ $p_{model}(x)$

Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

Addresses density estimation, a core problem in unsupervised learning

Several flavors:

- Explicit density estimation: explicitly define and solve for $p_{model}(x)$
- Implicit density estimation: learn model that can sample from $p_{model}(x)$ w/o explicitly defining it

Why Generative Models?

- Realistic samples for artwork, super-resolution, colorization, etc.



- Generative models can be used to enhance training datasets with diverse synthetic data
- Generative models of time-series data can be used for simulation

Adapted from Serena Young

Taxonomy of Generative Models

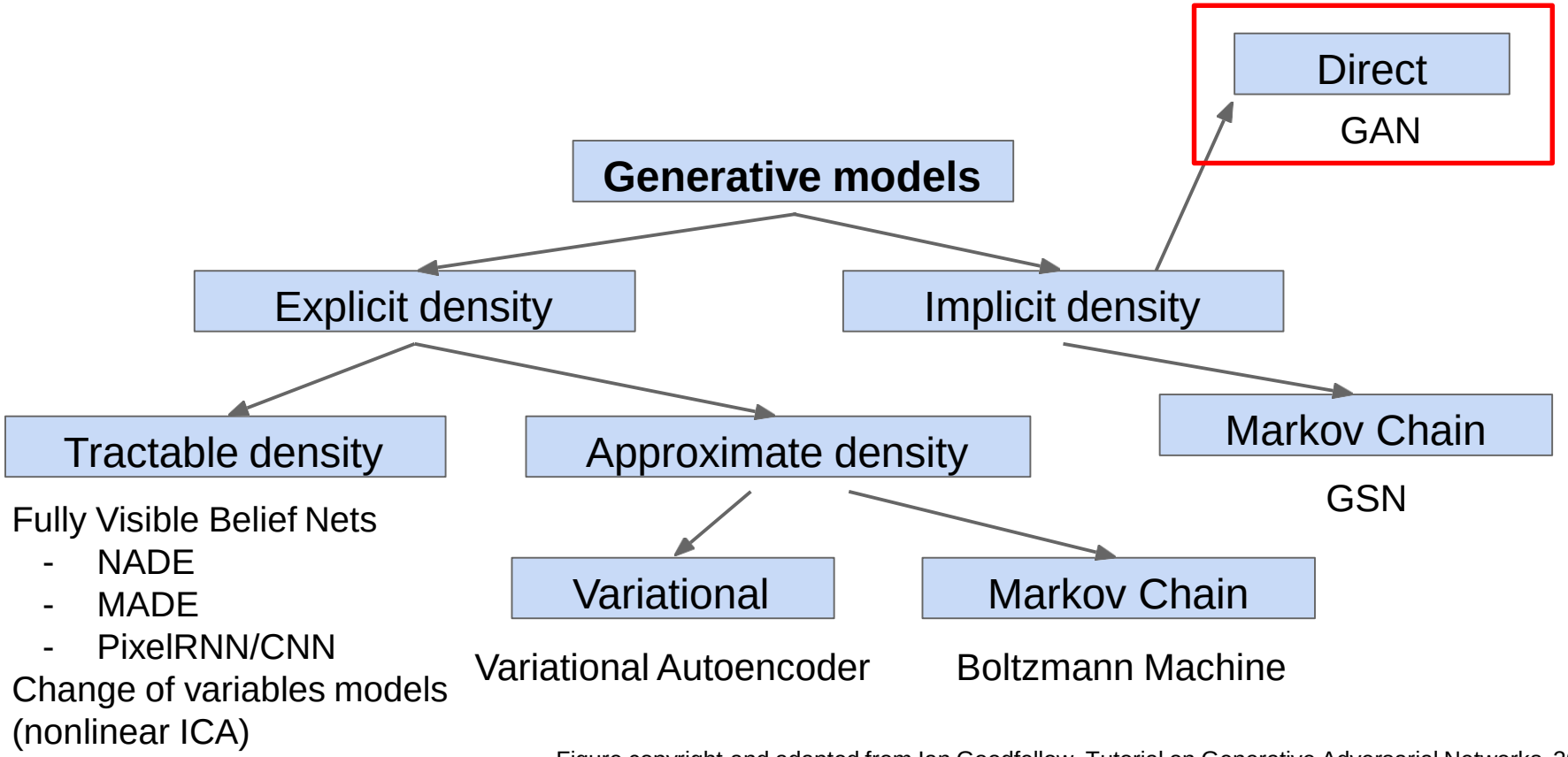


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

Generative Adversarial Networks

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Problem: Want to sample from complex, high-dimensional training distribution. No direct way to do this!

Solution: Sample from a simple distribution, e.g. random noise. Learn transformation to training distribution.

Q: What can we use to represent this complex transformation?

Generative Adversarial Networks

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

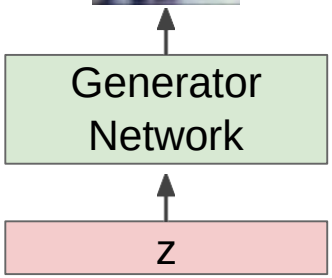
Problem: Want to sample from complex, high-dimensional training distribution. No direct way to do this!

Solution: Sample from a simple distribution, e.g. random noise. Learn transformation to training distribution.

Q: What can we use to represent this complex transformation?

A: A neural network!

Output: Sample from training distribution



Input: Random noise

Training GANs: Two-player game

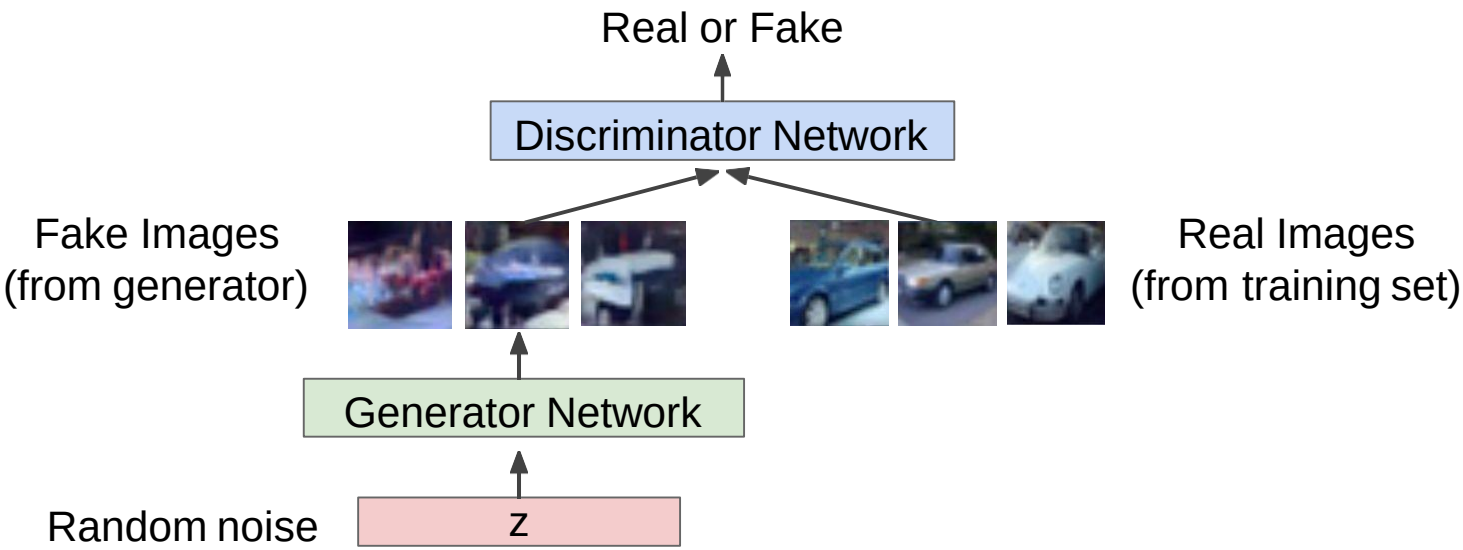
Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images

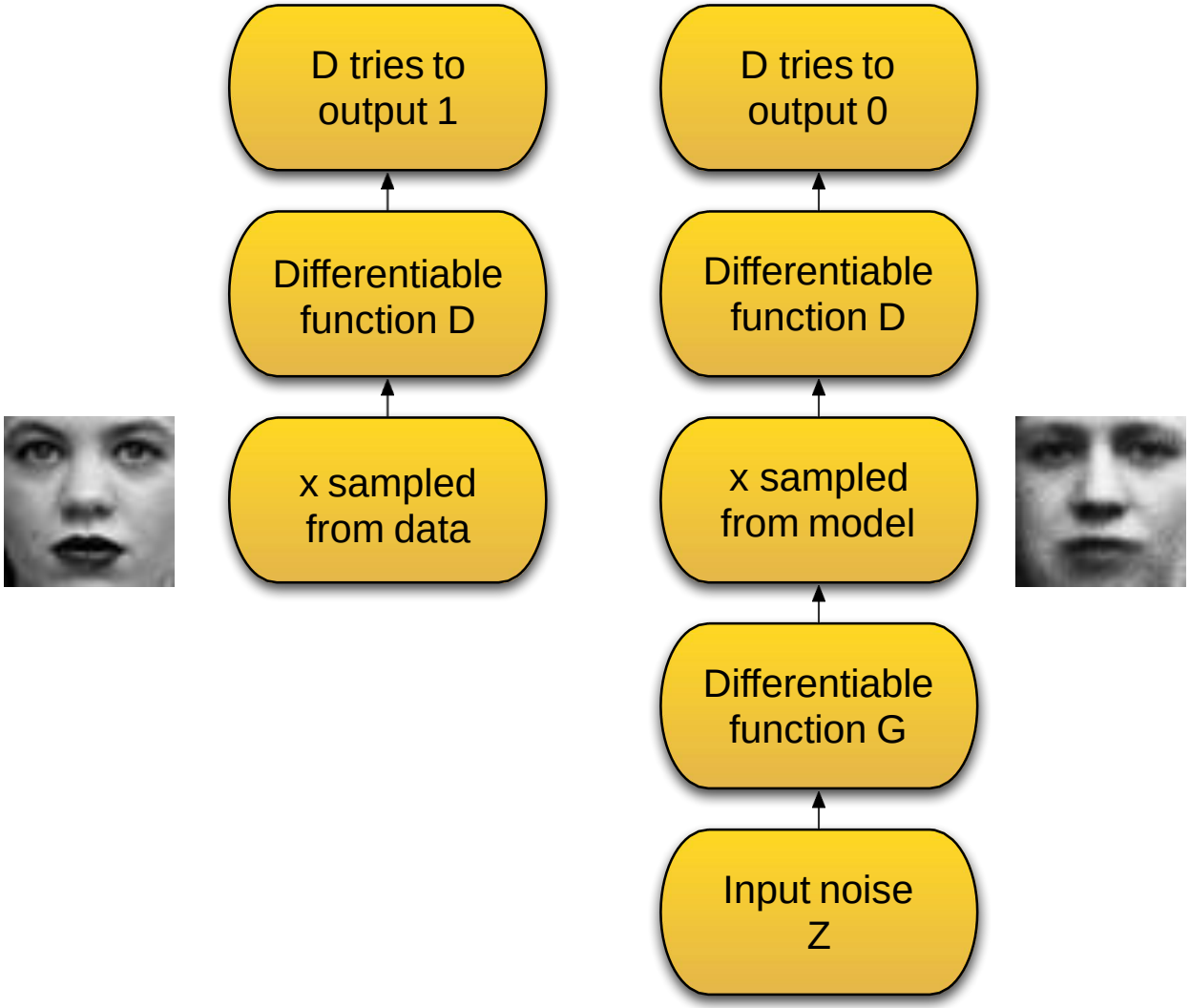
Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images

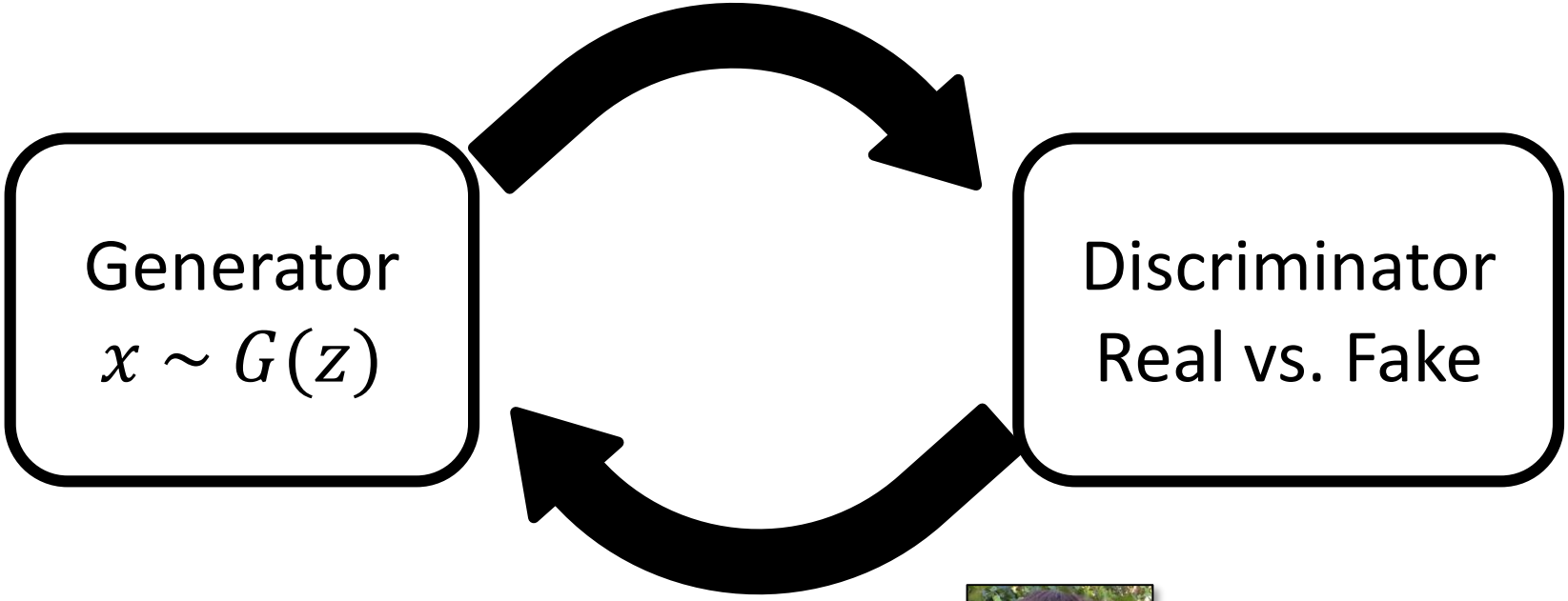


Adversarial Networks Framework



Ian Goodfellow

Adversarial Networks Framework



[Goodfellow et al. 2014]

Training GANs: Two-player game

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images

Train jointly in **minimax game**

Minimax objective function:

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

Training GANs: Two-player game

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images

Train jointly in **minimax game**

Discriminator outputs likelihood in (0,1) of real image

Minimax objective function:

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

Training GANs: Two-player game

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images

Train jointly in **minimax game**

Discriminator outputs likelihood in (0,1) of real image

Minimax objective function:

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

- Discriminator (θ_d) wants to **maximize objective** such that D(x) is close to 1 (real) and D(G(z)) is close to 0 (fake)
- Generator (θ_g) wants to **minimize objective** such that D(G(z)) is close to 1 (discriminator is fooled into thinking generated G(z) is real)

Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Minimax objective function:

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

Alternate between:

1. **Gradient ascent** on 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. **Gradient descent** on generator

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

Training GANs: Two-player game

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Minimax objective function:

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

Alternate between:

1. **Gradient ascent** on discriminator

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

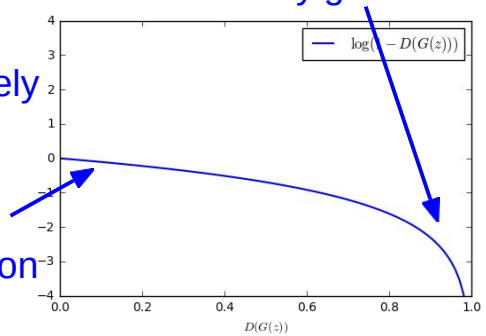
2. **Gradient descent** on generator

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

In practice, optimizing this generator objective does not work well!

Gradient signal dominated by region where sample is already good

When sample is likely fake, want to learn from it to improve generator. But gradient in this region is relatively flat!



Training GANs: Two-player game

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Minimax objective function:

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

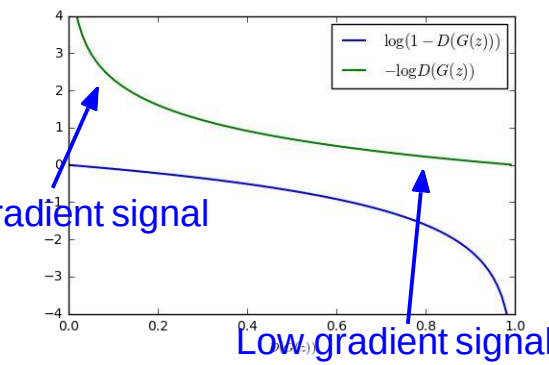
Alternate between:

1. **Gradient ascent** on 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. **Instead: Gradient ascent** on generator, different objective

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



Instead of minimizing likelihood of discriminator being correct, now maximize likelihood of discriminator being wrong.
Same objective of fooling discriminator, but now higher gradient signal for bad samples => works much better! Standard in practice.

Training GANs: Two-player game

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Putting it together: GAN training algorithm

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:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D_{\theta_d}(x^{(i)}) + \log(1 - D_{\theta_d}(G_{\theta_g}(z^{(i)}))) \right]$$

  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 ascending its stochastic gradient (improved objective):

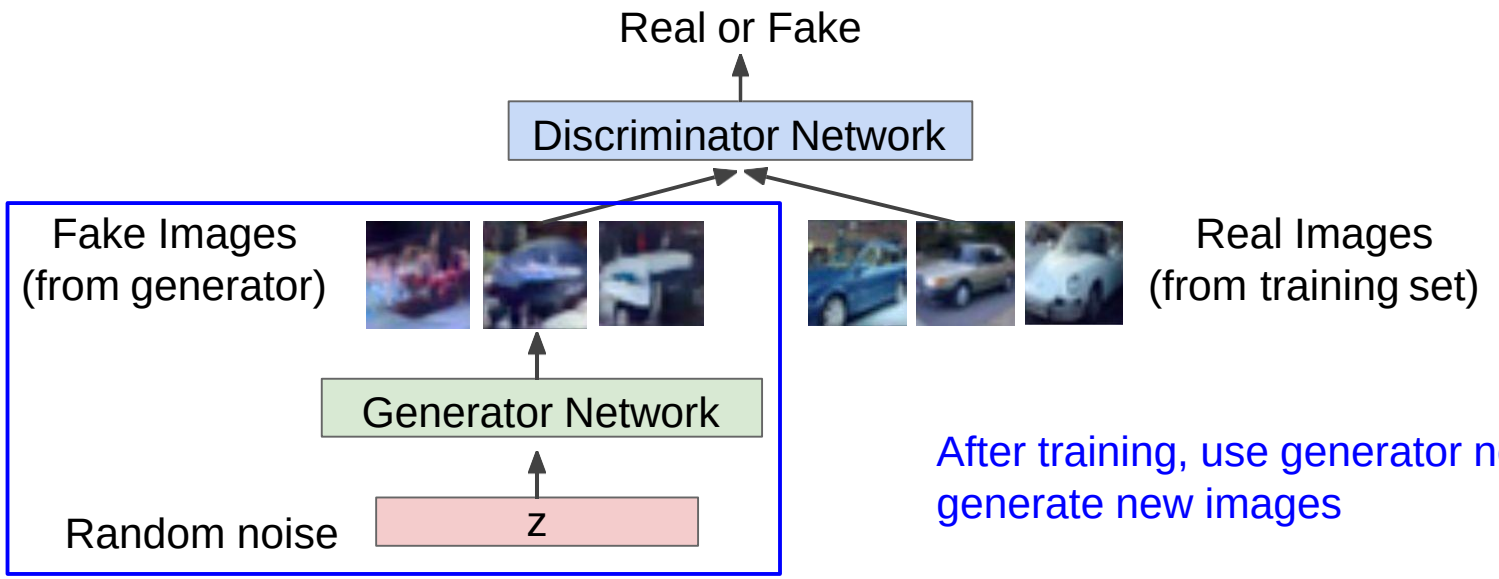
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(D_{\theta_d}(G_{\theta_g}(z^{(i)})))$$

end for

Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images



After training, use generator network to generate new images

Generative Adversarial Nets

Samples from the model look amazing!

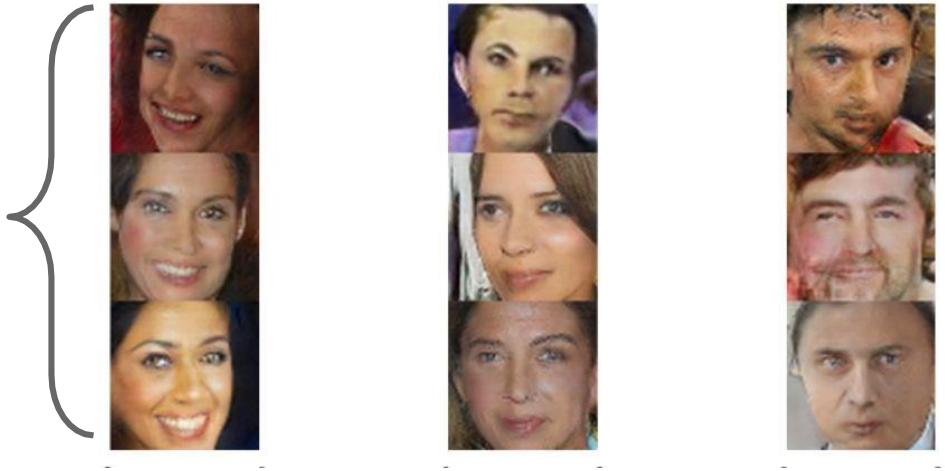
Radford et al, ICLR 2016

Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016

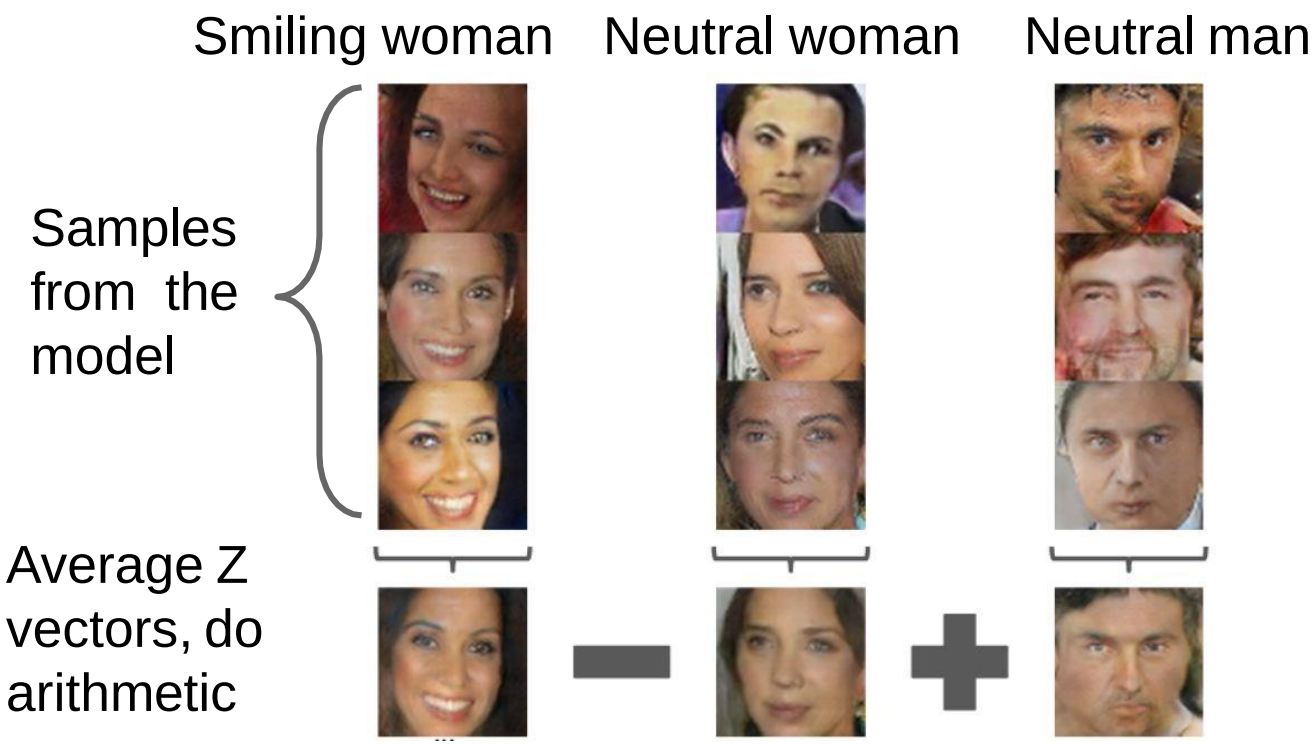
Smiling woman Neutral woman Neutral man

Samples from the model



Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016



Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016

Smiling woman Neutral woman Neutral man

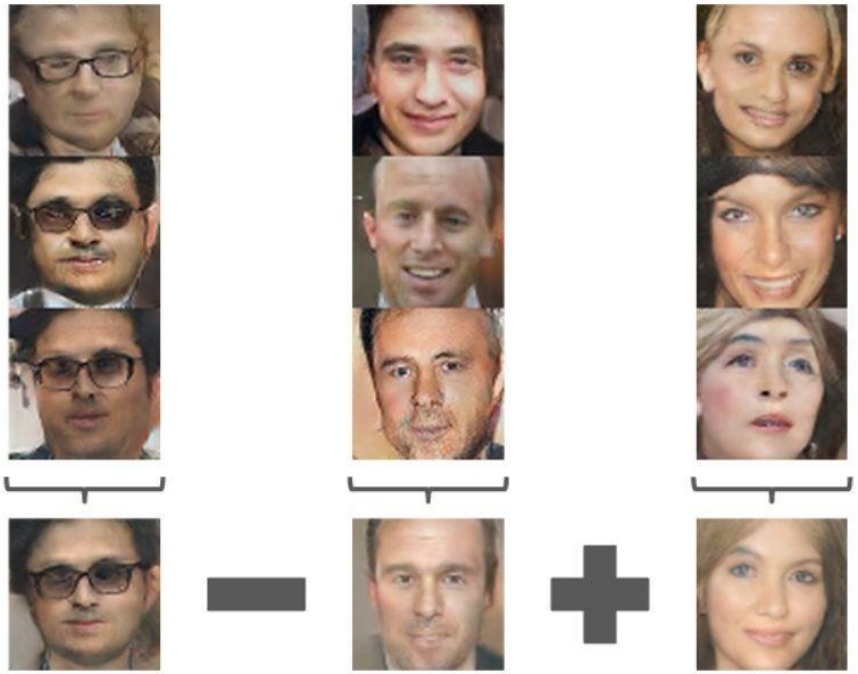
Samples from the model

Average Z vectors, do arithmetic

Smiling Man

Generative Adversarial Nets: Interpretable Vector Math

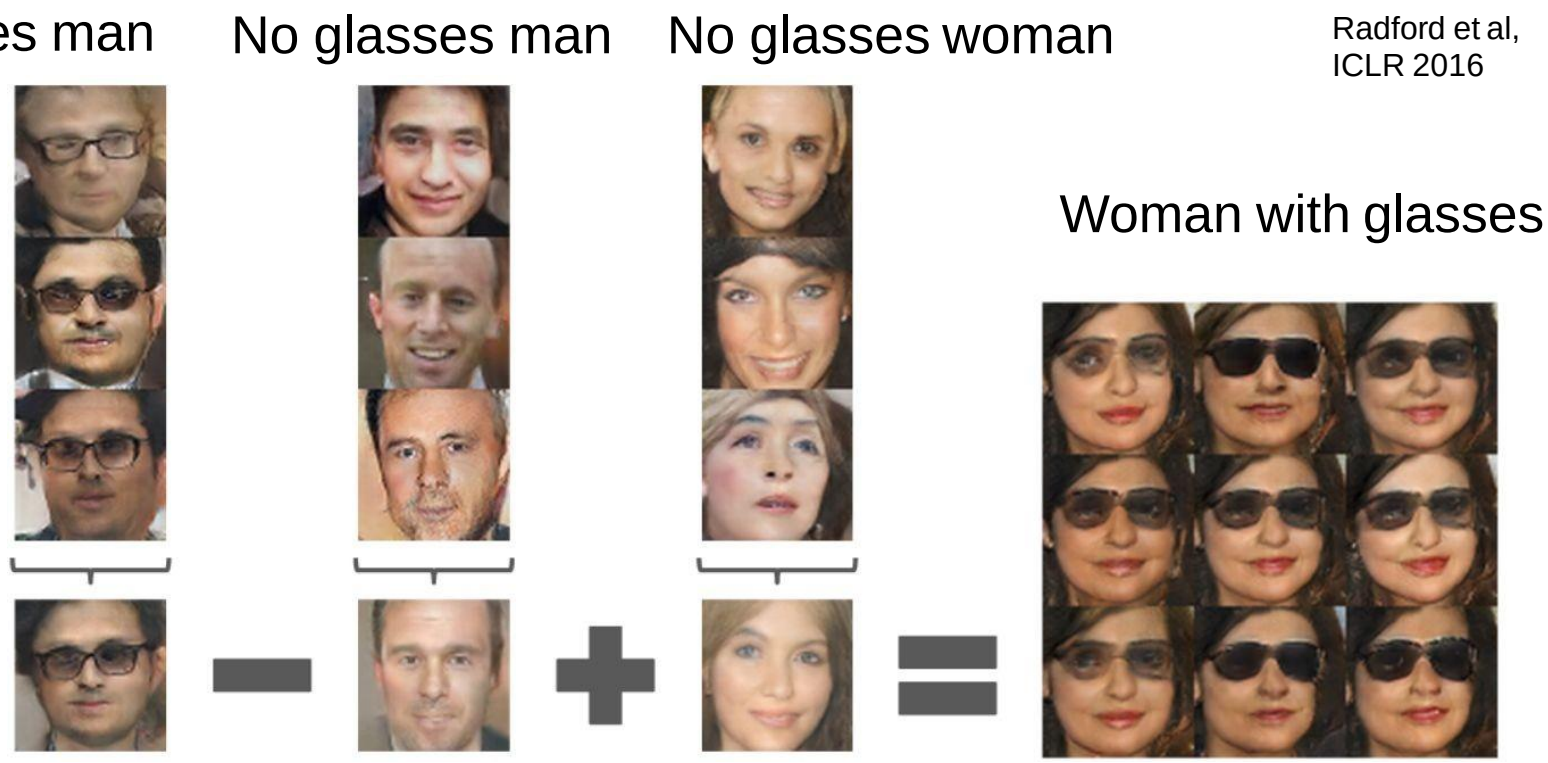
Glasses man No glasses man No glasses woman



Radford et al,
ICLR 2016

Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016



What is in this image?

(Yeh et al., 2016)

Generative modeling reveals a face

(Yeh et al., 2016)

Creative Adversarial Networks

CAN: Top ranked by human subjects

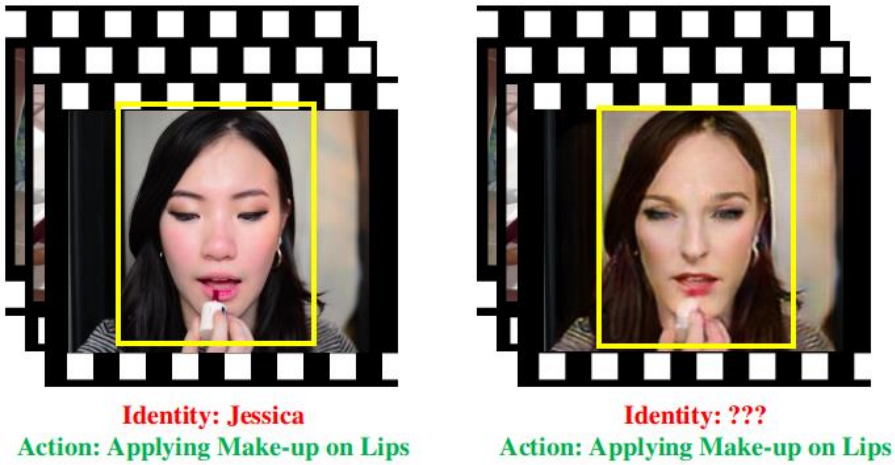
(Elgammal et al., 2017)

Artificial Fashion: vue.ai

Celebrities Who Never Existed

Karras et al., "Progressive Growing of GANs for Improved Quality, Stability, and Variation", ICLR ***2018***

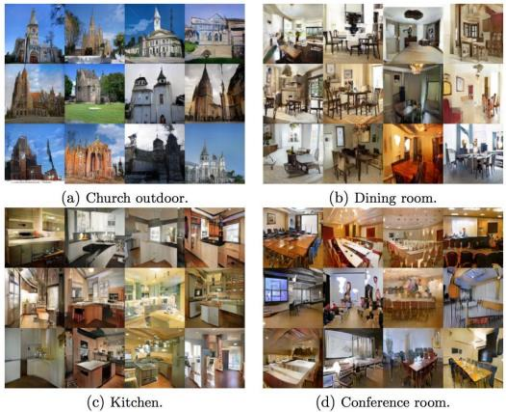
GANs for Privacy (Action Detection)



Ren et al., "Learning to Anonymize Faces for Privacy Preserving Action Detection", arxiv 2018

2017: Year of the GAN

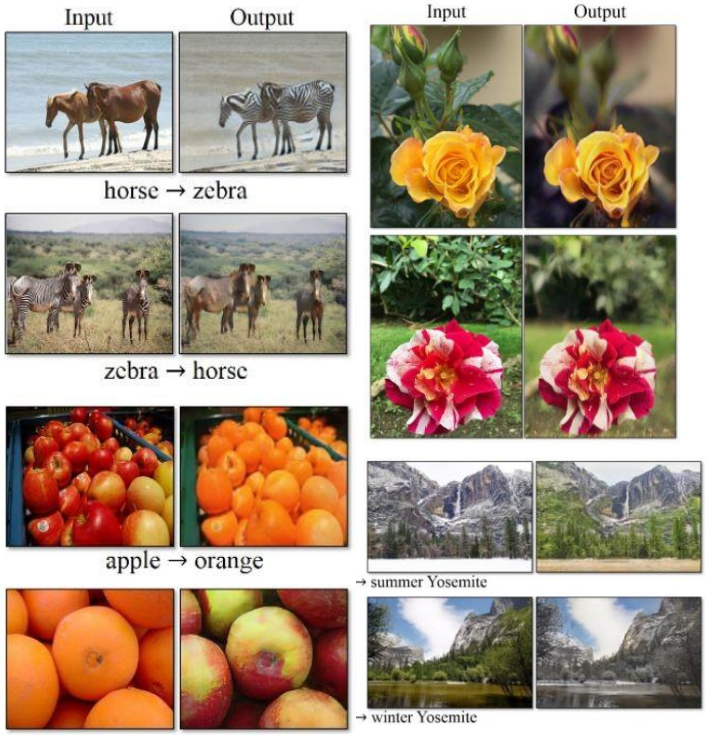
Better training and generation



LSGAN. Mao et al. 2017.

BEGAN. Bertholet et al. 2017.

Source->Target domain transfer



CycleGAN. Zhu et al. 2017.

Text -> Image Synthesis

Reed et al. 2017.

Many GAN applications

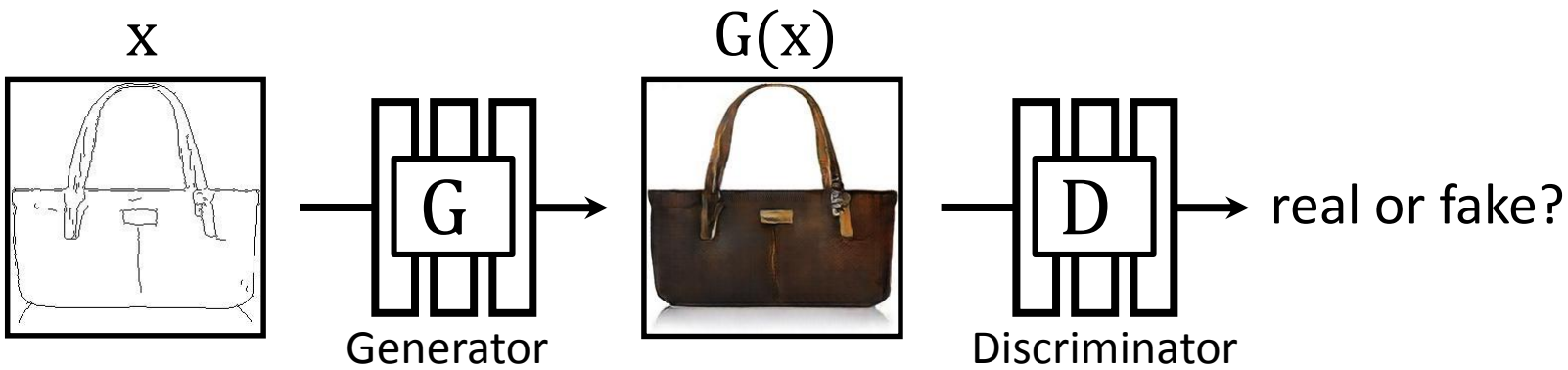
Pix2pix. Isola 2017. Many examples at https://phillipi.github.io/pix2pix/

Serena Young

Plan for today

- Generative models: What are they?
- Techniques: Generative Adversarial Networks
  - Conditional GANs
  - Cycle-consistency loss
  - Dealing with sparse data
  - *Applications interleaved with techniques*
- Generation before GANs
- Generation for synthetic training data

GANs

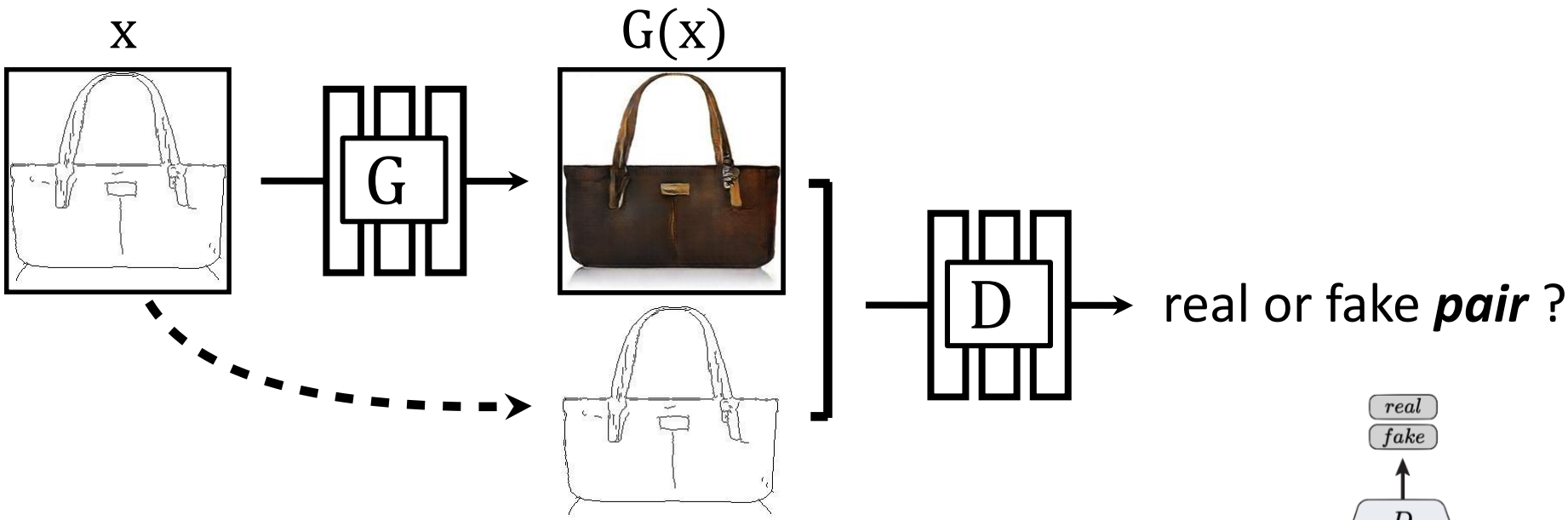


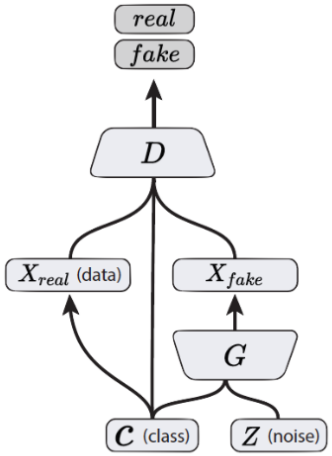
G: generate fake samples that can fool D

D: classify fake samples vs. real images

[Goodfellow et al. 2014]

Conditional GANs





Conditional GAN
(Mirza & Osindero, 2014)

Adapted from Jun-Yan Zhu

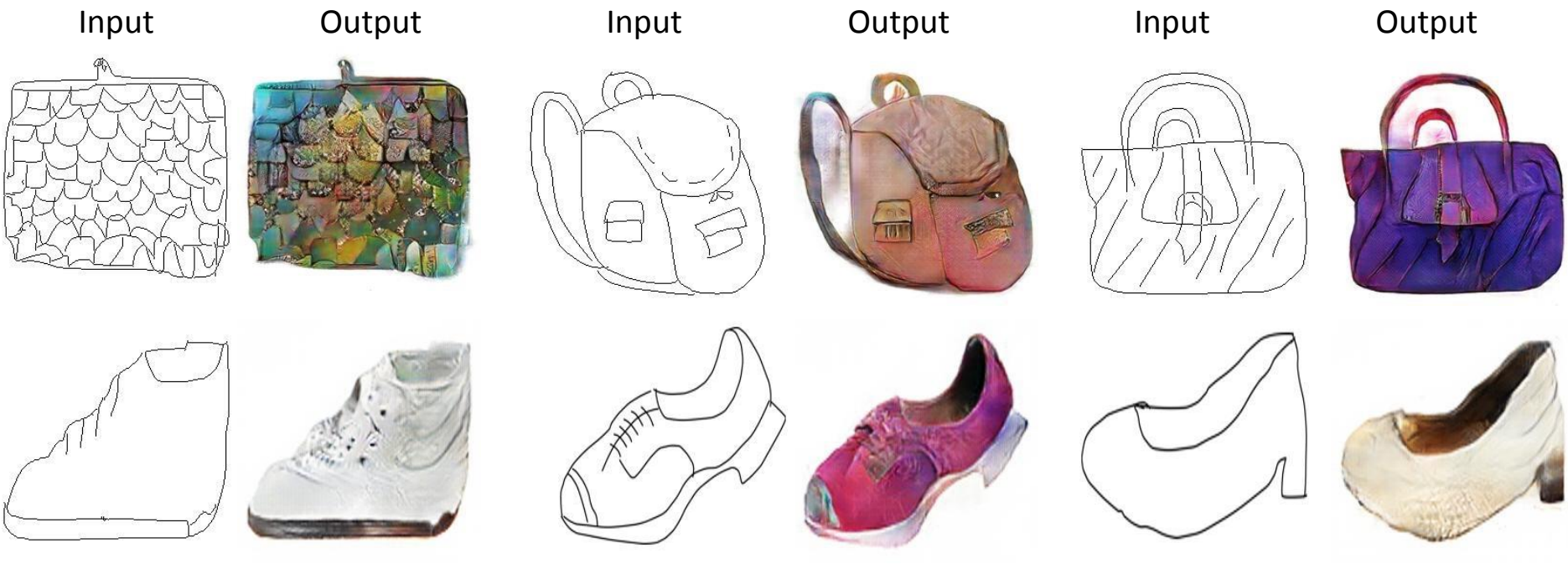
Image Credit: Figure 2 in Odena, A., Olah, C. and Shlens, J., 2016. Conditional image synthesis with auxiliary classifier GANs. *arXiv preprint arXiv:1610.09585*.

Edges → Images

Edges from [Xie & Tu, 2015]

Pix2pix / CycleGAN

Sketches → Images

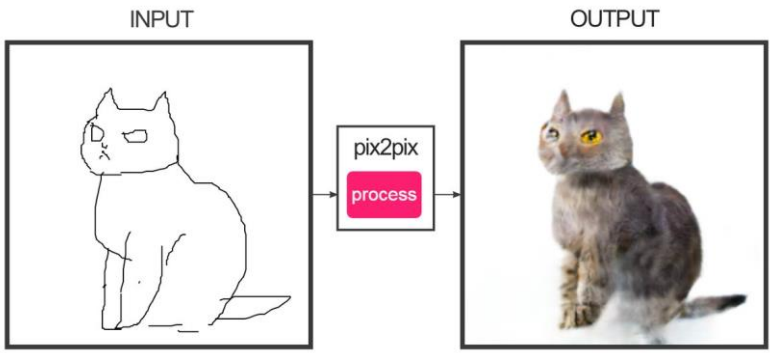


Trained on Edges → Images

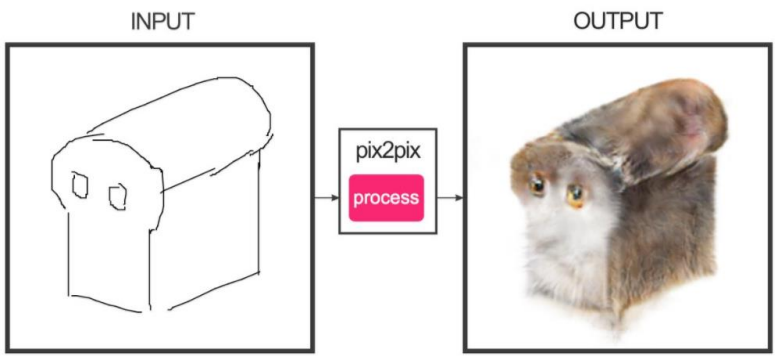
Data from [Eitz, Hays, Alexa, 2012]

Pix2pix / CycleGAN

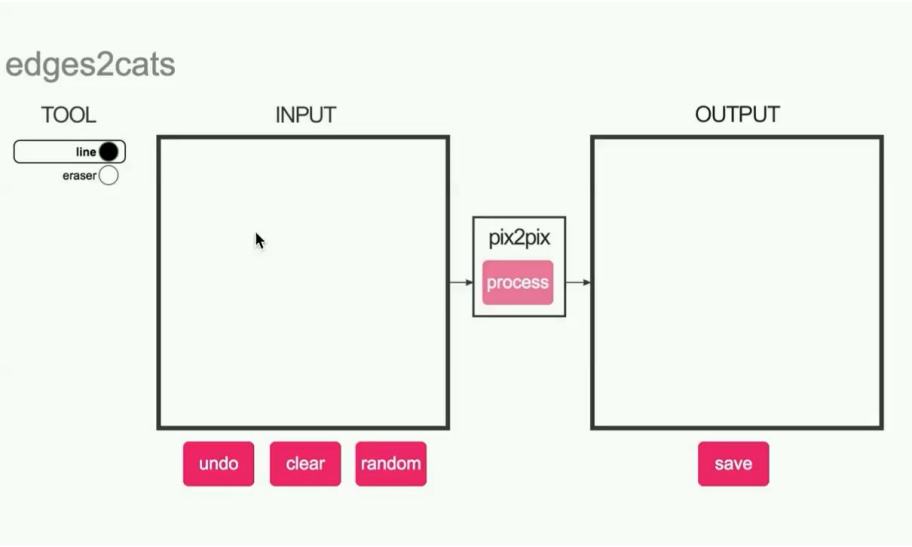
#edges2cats [Christopher Hesse]



@gods_tail



Ivy Tasi @ivymyt

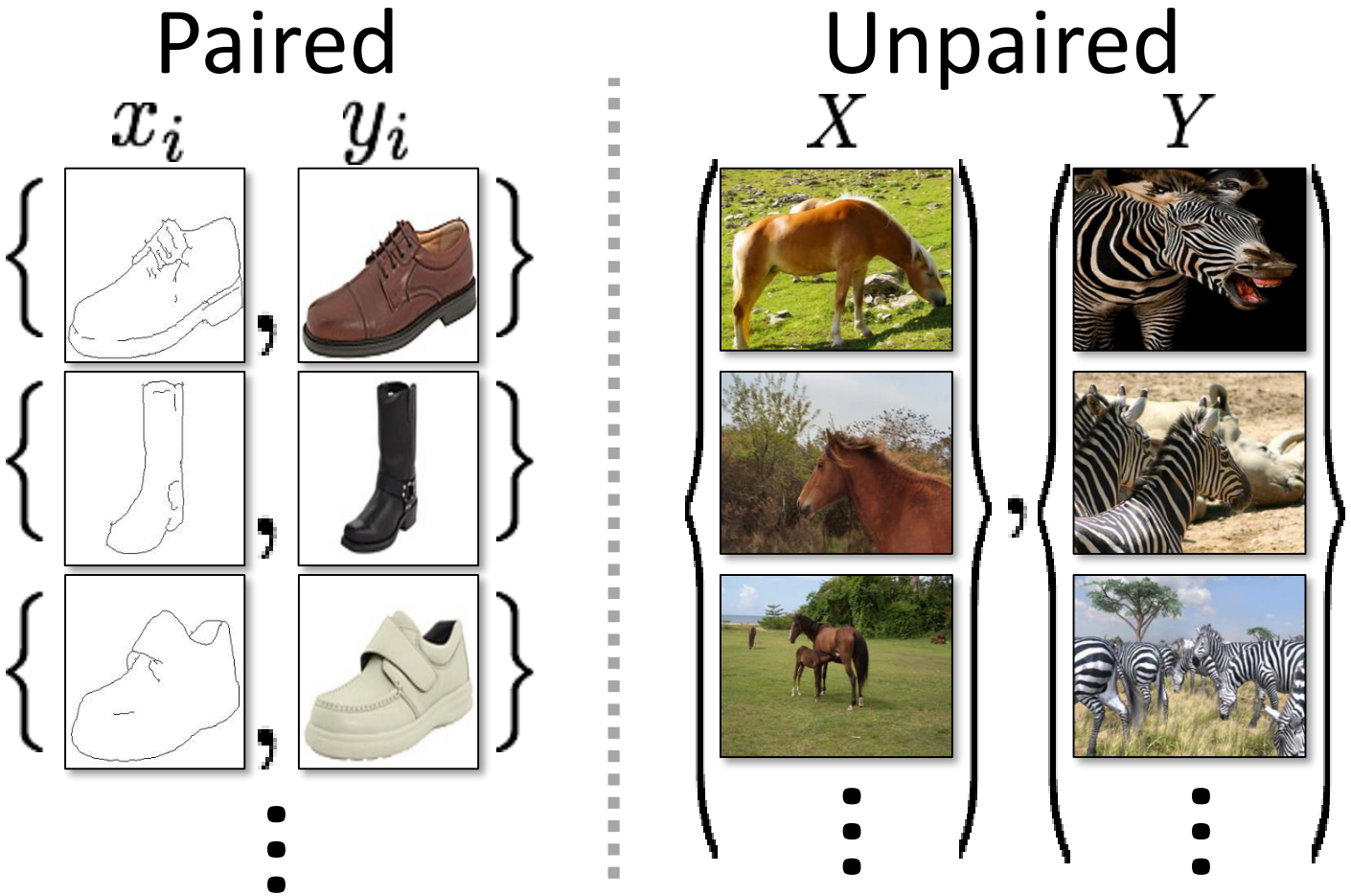


@matthematician

Vitaly Vidmirov @vvid

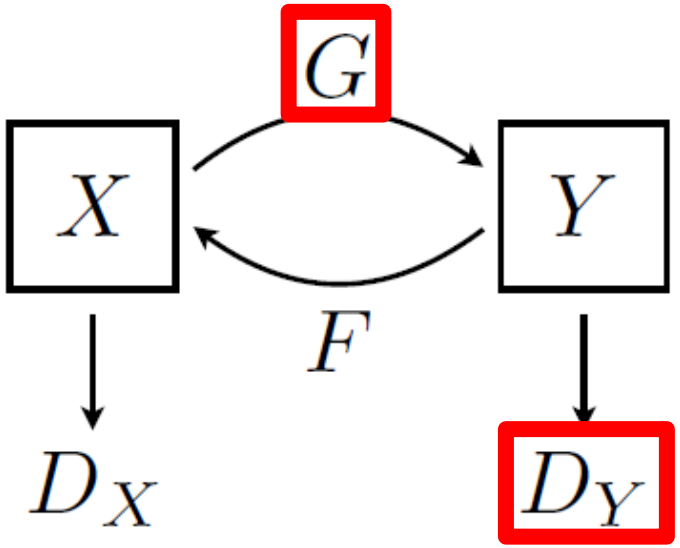
https://affinelayer.com/pixsrv/

Pix2pix / CycleGAN



Jun-Yan Zhu

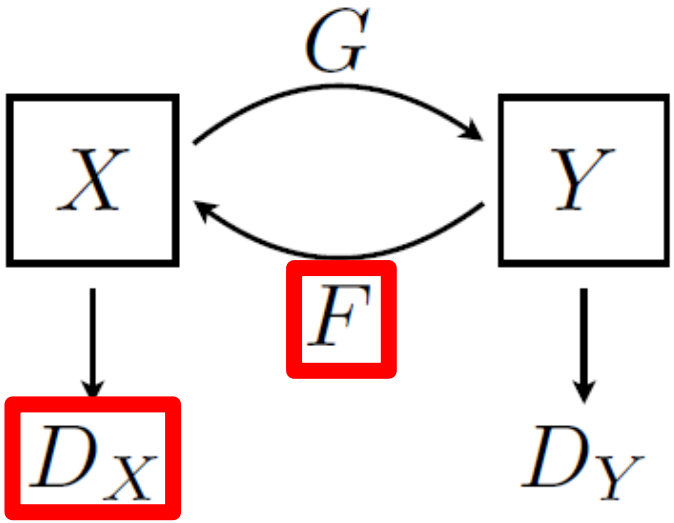
Cycle Consistency



Discriminator $\mathrm{D_Y}$: $L_{GAN}(G(x), y)$
Real zebras vs. generated zebras

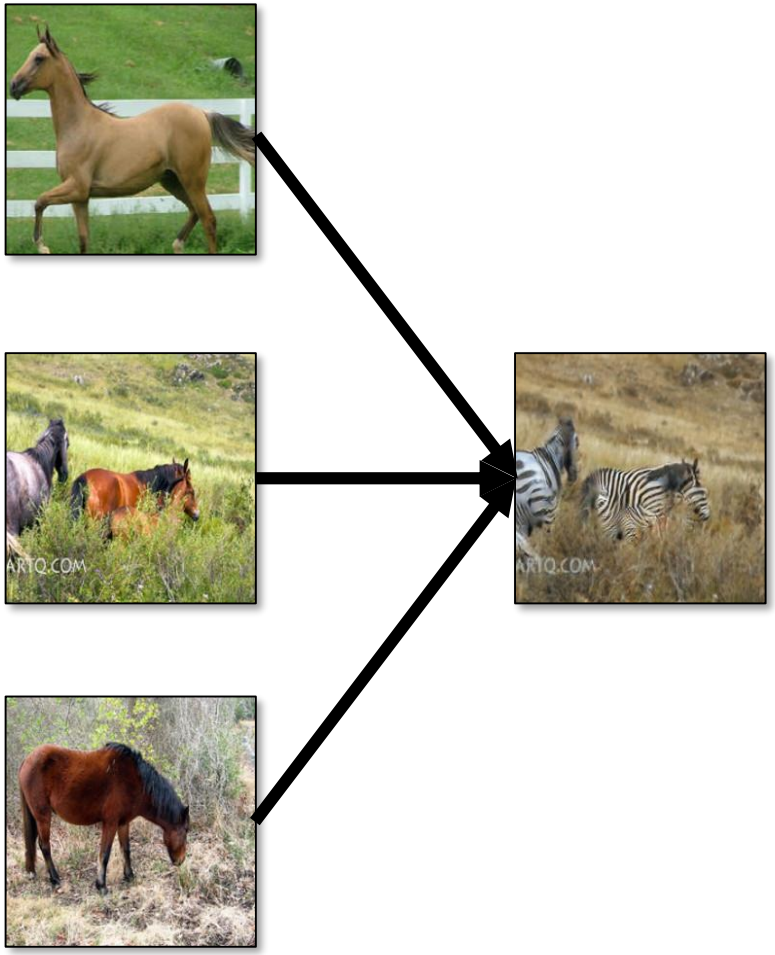
Zhu et al., "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks", ICCV 2017

Cycle Consistency



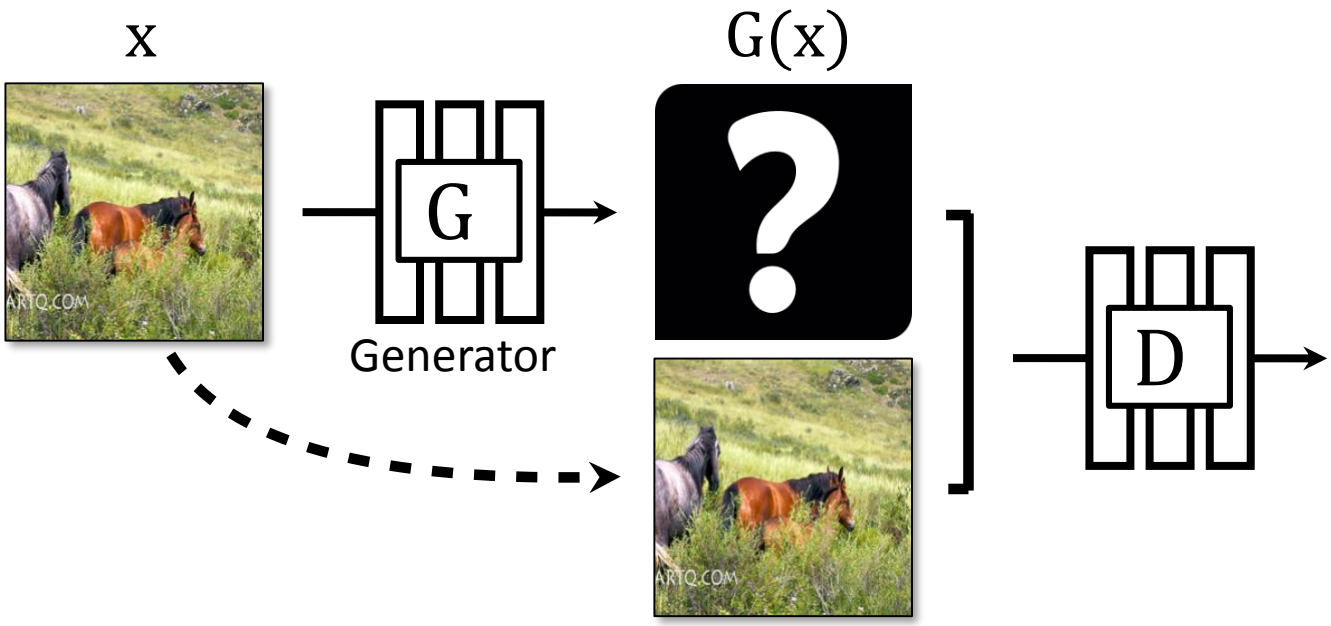
Discriminator $\mathrm{D_Y}$: $L_{GAN}(G(x), y)$
Real zebras vs. generated zebras
Discriminator $\mathrm{D_X}$: $L_{GAN}(F(y), x)$
Real horses vs. generated horses

Zhu et al., "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks", ICCV 2017



mode collapse!

Zhu et al., “Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks”, ICCV 2017

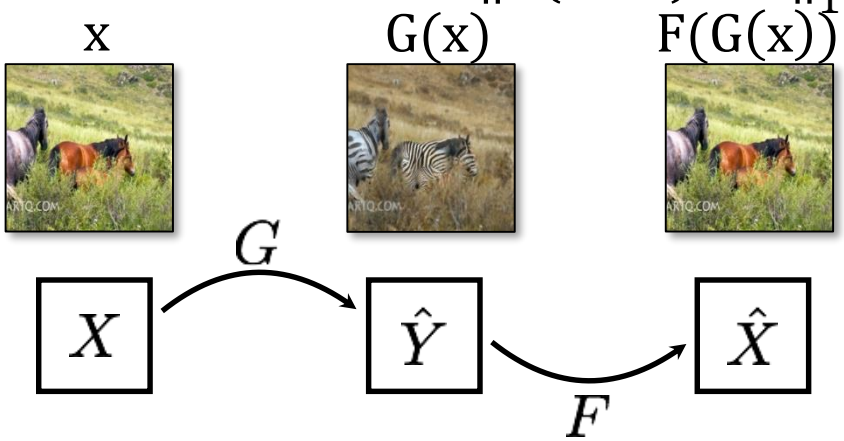


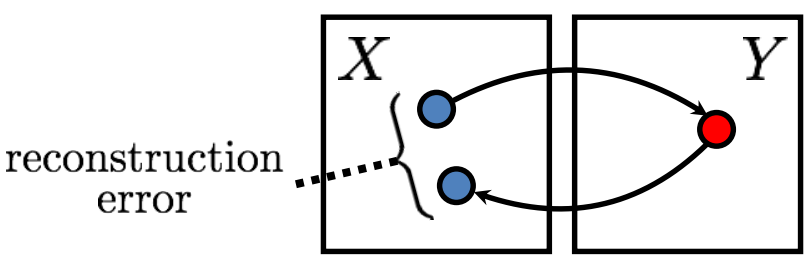
No input-output pairs!

Zhu et al., "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks", ICCV 2017

Cycle Consistency

Forward cycle loss: $\|F(G(x)) - x\|_1$

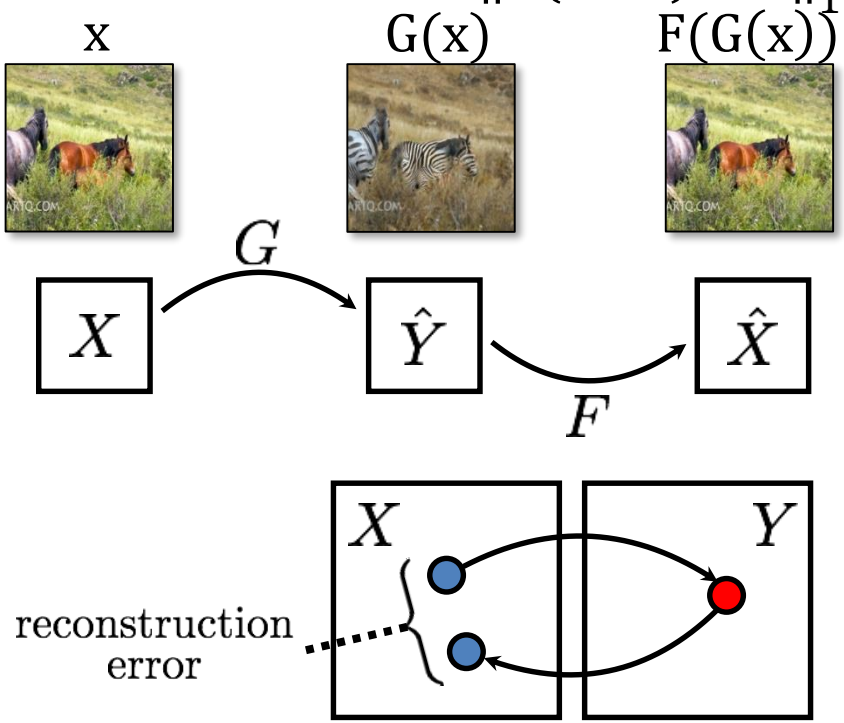




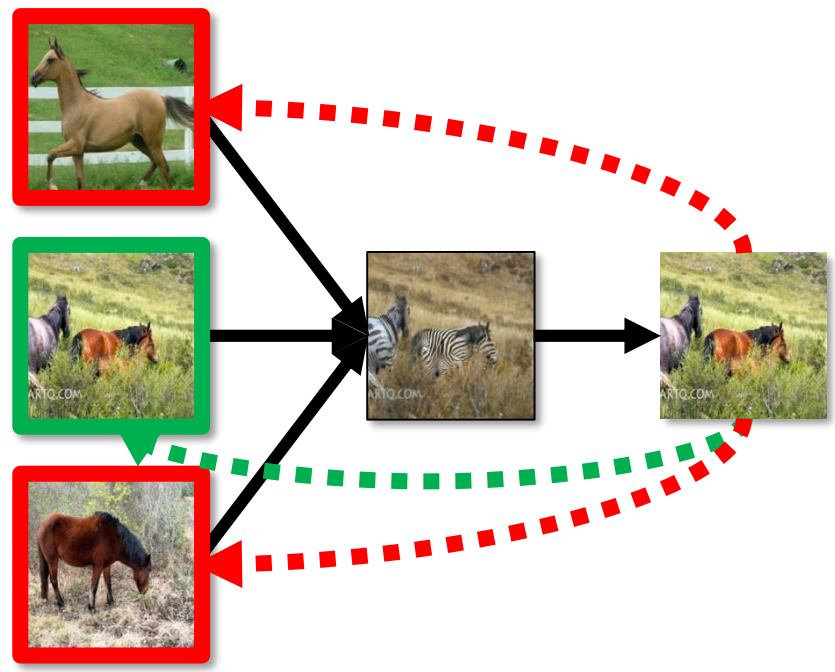
Zhu et al., "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks", ICCV 2017

Cycle Consistency

Forward cycle loss: $\|F(G(x)) - x\|_1$



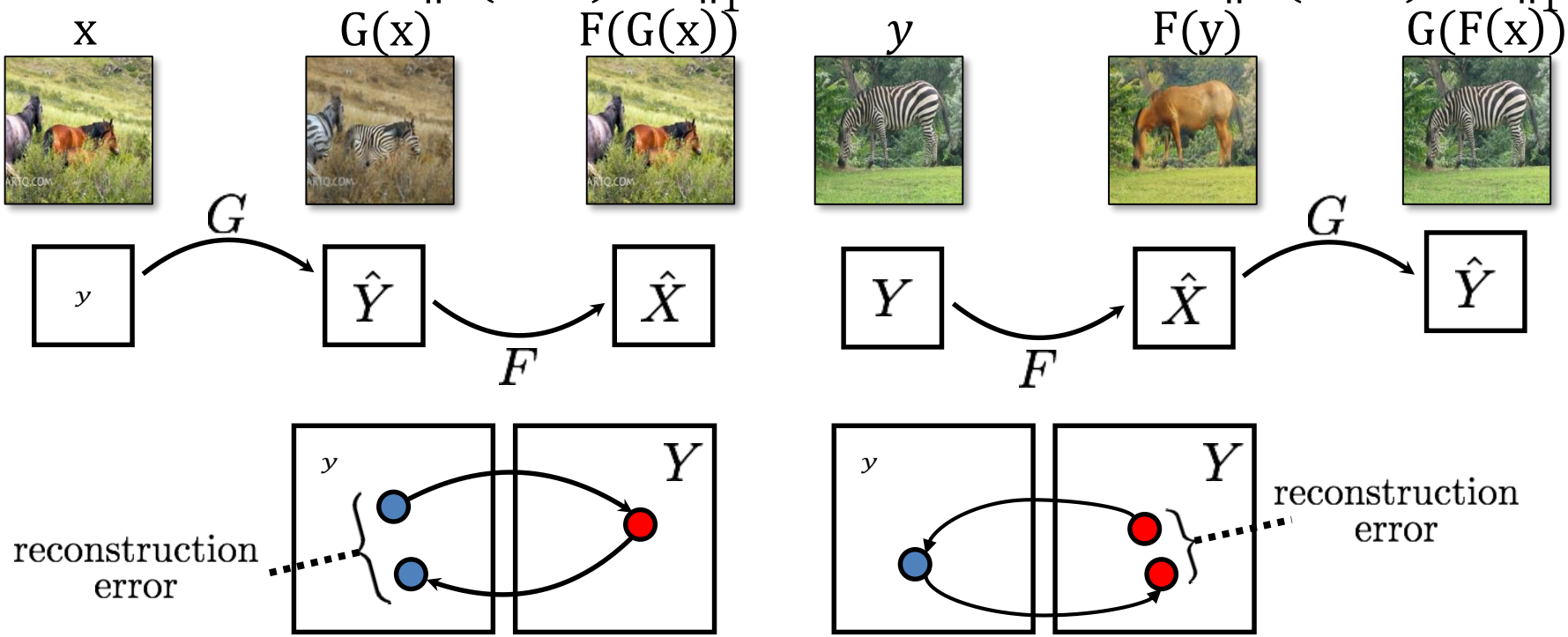
Small cycle loss



Cycle Consistency

Forward cycle loss: $\|F(G(x)) - x\|_1$

Backward cycle loss: $\|G(F(y)) - y\|_1$



Zhu et al., "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks", ICCV 2017

Training Details: Objective

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))],$$

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1].$$

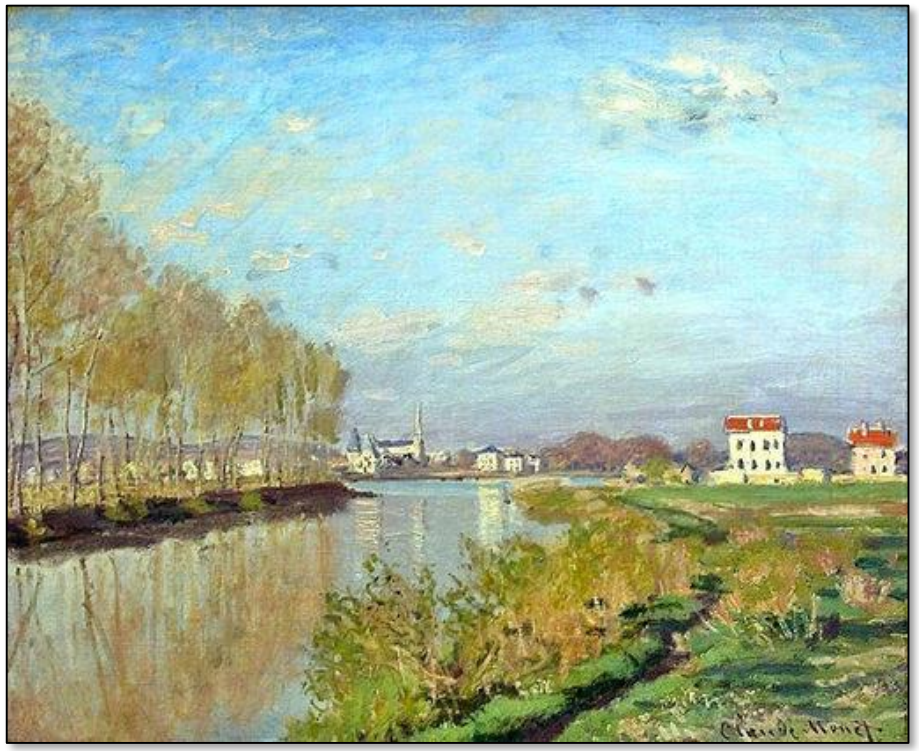
$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F),$$

$$G^*, F^* = \arg\min_{G,F} \max_{D_x, D_Y} \mathcal{L}(G, F, D_X, D_Y).$$

Zhu et al., "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks", ICCV 2017

Pix2pix / CycleGAN

Monet’s paintings → photos

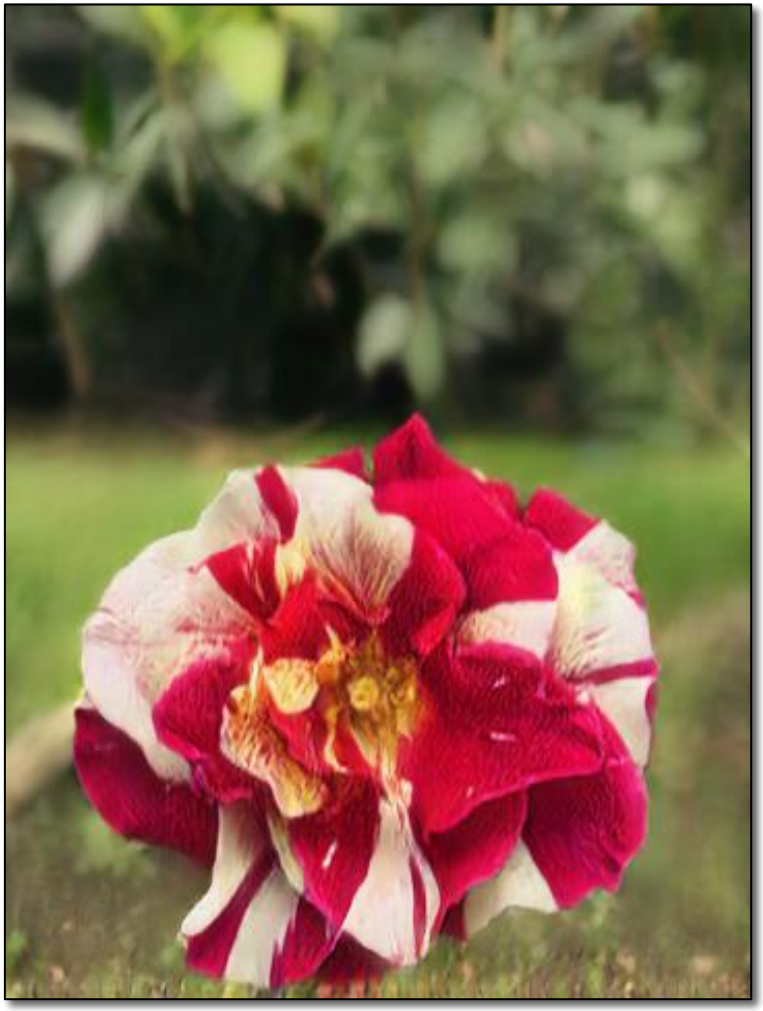


Pix2pix / CycleGAN

Monet’s paintings → photos

Pix2pix / CycleGAN

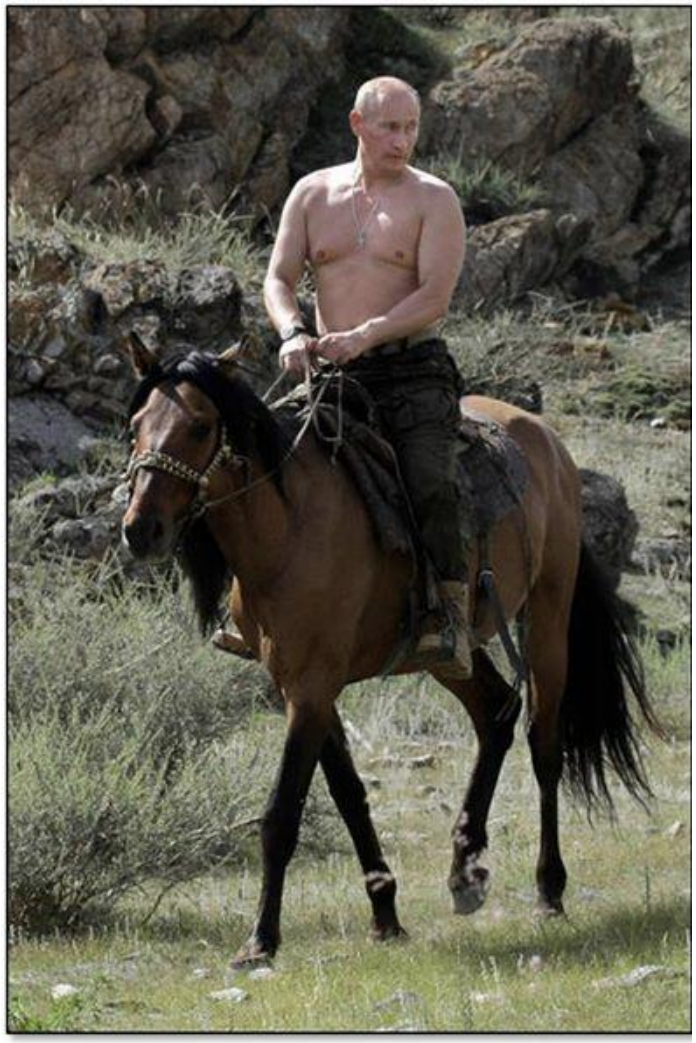
Pix2pix / CycleGAN



Pix2pix / CycleGAN

Pix2pix / CycleGAN

Pix2pix / CycleGAN



Pix2pix / CycleGAN

Pix2pix / CycleGAN

#CycleGAN

Monet → Thomas Kinkade @David Fouhey

Resurrecting Ancient Cities @ Jack Clark

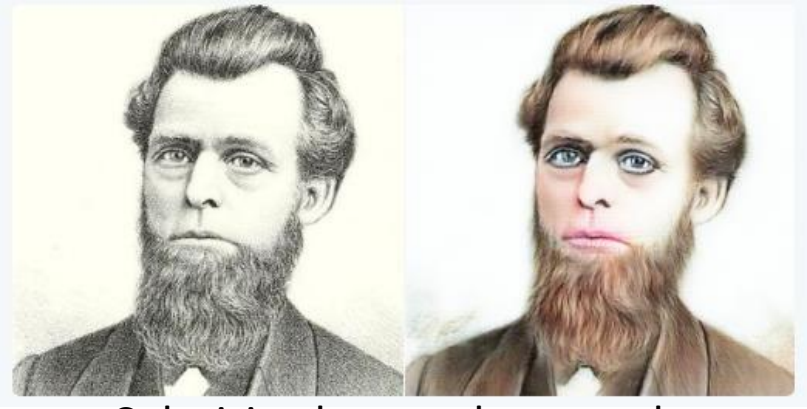
Birds @Matt Powell

Bear → Panda @Matt Powell

Pix2pix / CycleGAN

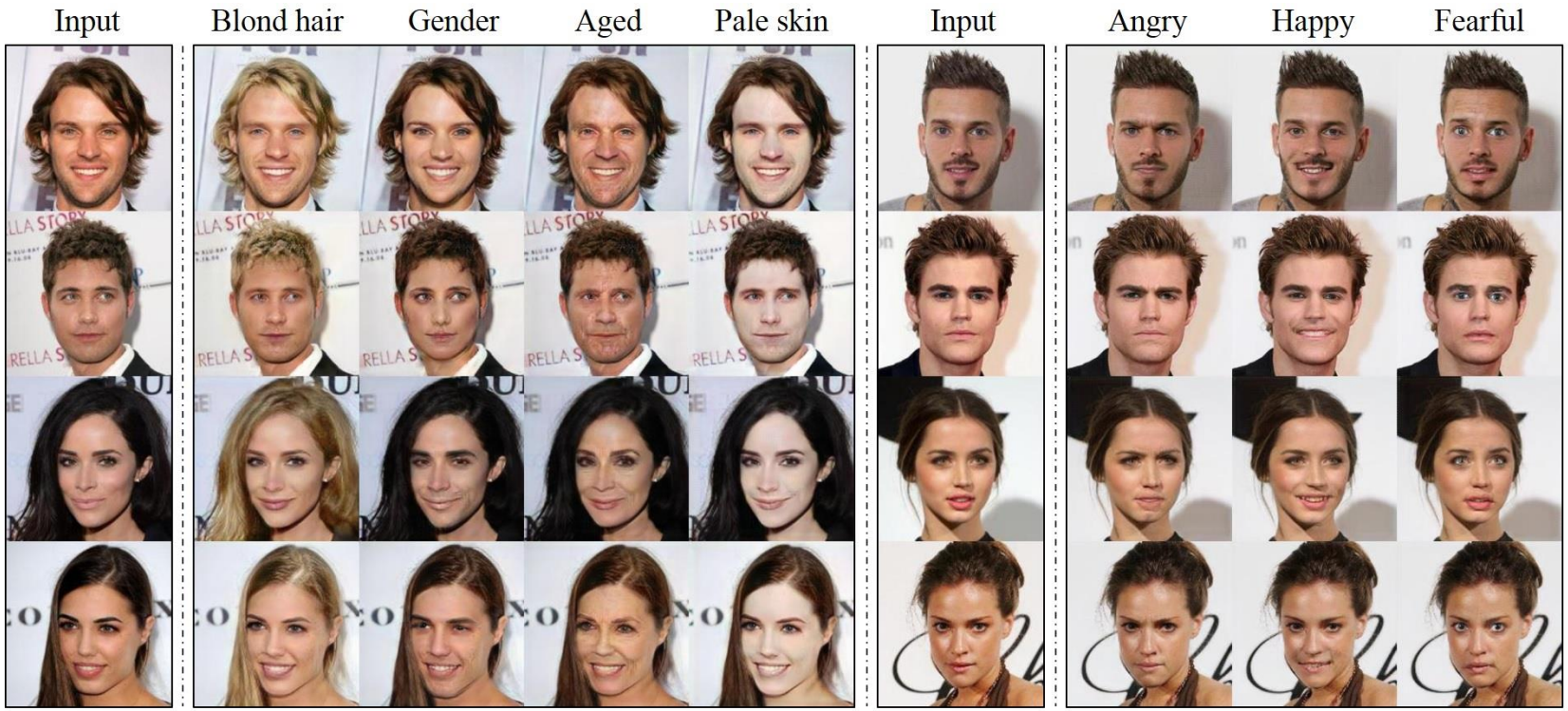
#CycleGAN

Portrait to Dollface
@Mario Klingemann



Colorizing legacy photographs
@Mario Klingemann

StarGAN

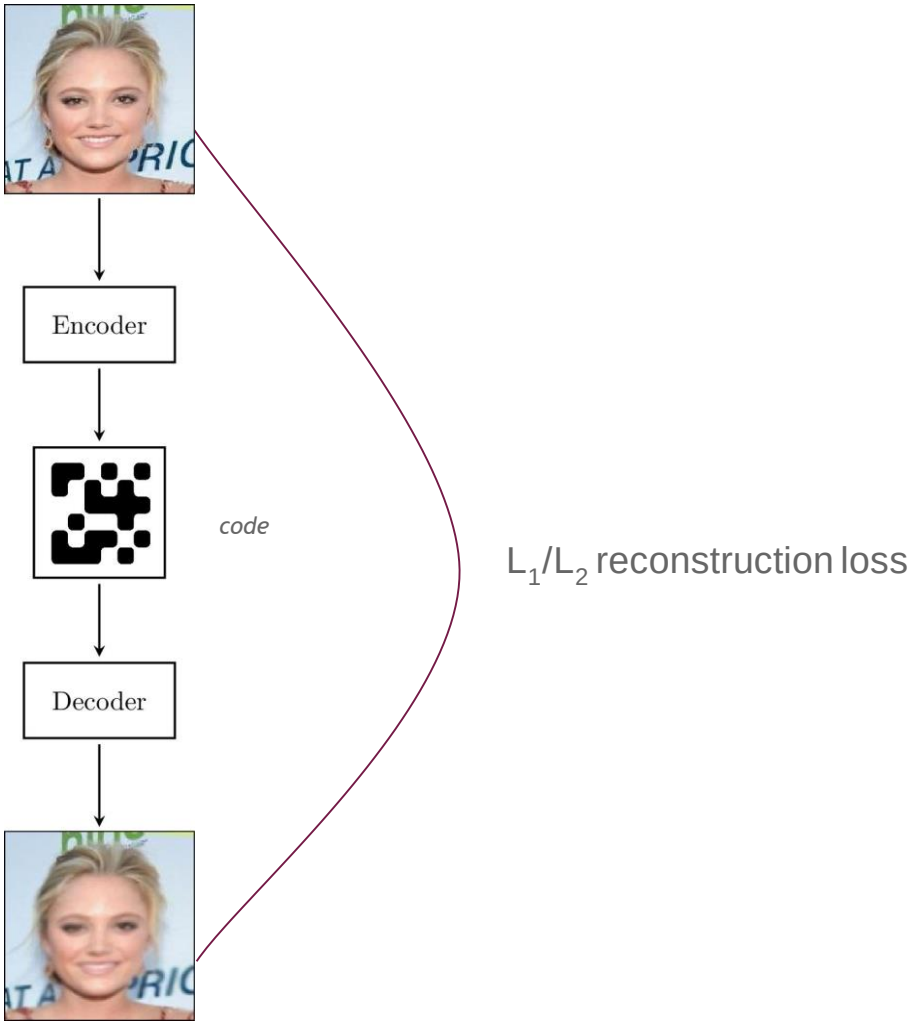


Choi et al., "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation", **CVPR 2018**

Plan for today

- Generative models: What are they?
- Techniques: Generative Adversarial Networks
  - Conditional GANs
  - Cycle-consistency loss
  - Dealing with sparse data
  - *Applications interleaved with techniques*
- Generation before GANs
- Generation for synthetic training data

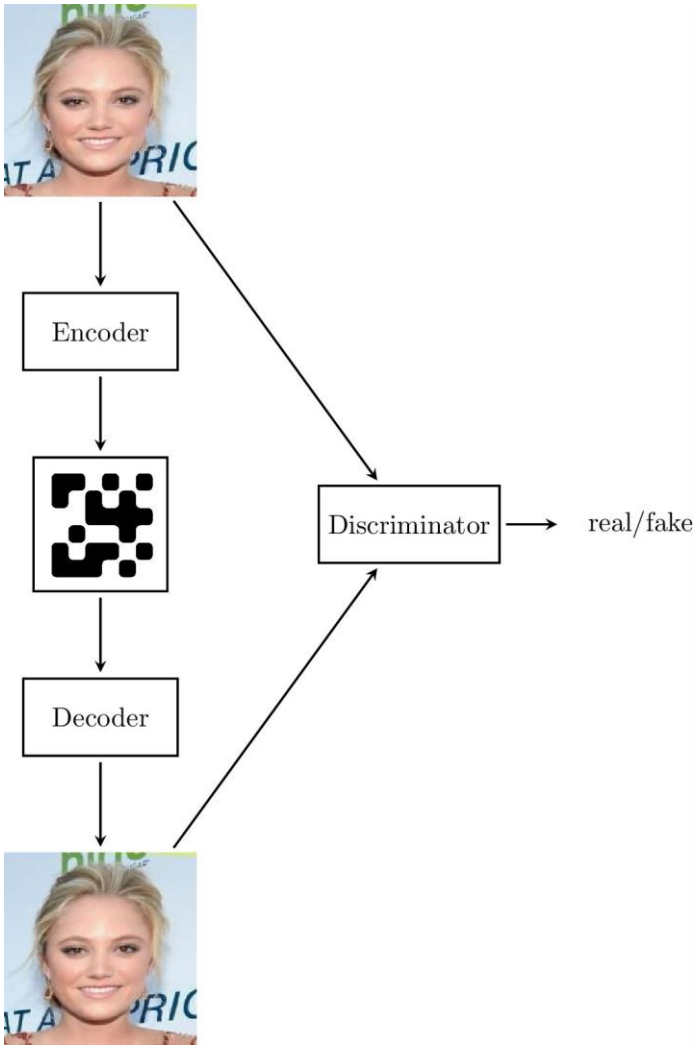
Autoencoders



Mihaela Rosca

Adversarial Autoencoders

Improve reconstruction quality by adding a GAN loss.



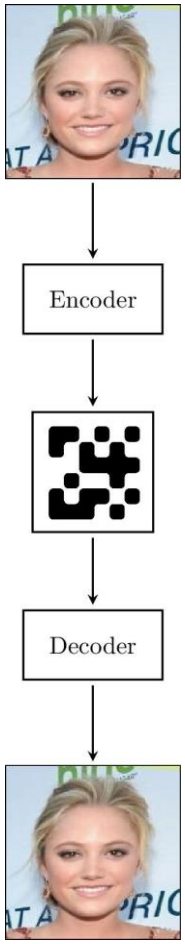
Adversarial Autoencoders
A. Makhzani, J. Shlens, N.Jaitly, I. Goodfellow, B.Frey

Autoencoder GANs

- Combine the reconstruction power of autoencoders with the sampling power of GANs.
- By construction, autoencoders learn to cover the entire training data.
- Work on the codespace - not data space.

Adapted from Mihaela Rosca

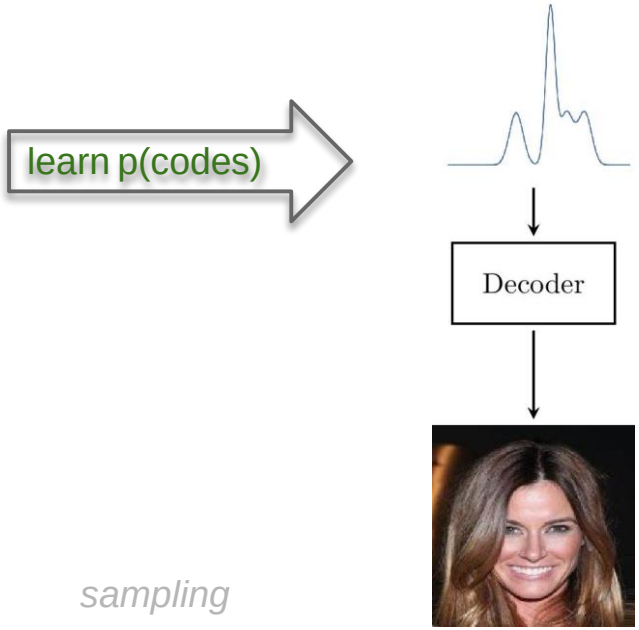
Learning the code distribution



reconstructing

Assumption

learning the code distribution is simpler than learning data distribution



sampling

OUR WORK

Mihaela Rosca

Plan for today

- Generative models: What are they?
- Techniques: Generative Adversarial Networks
  - Conditional GANs
  - Cycle-consistency loss
  - Dealing with sparse data
  - *Applications interleaved with techniques*
- Generation before GANs
- Generation for synthetic training data

Neural Style Transfer

Gatys et al., "Image Style Transfer Using Convolutional Neural Networks", CVPR 2016

SIGGRAPH 2001 EXPLORE INTERACTION AND DIGITAL IMAGES

+

=

Image Quilting for Texture Synthesis & Transfer

Alexei Efros *(UC Berkeley)*

Bill Freeman *(MERL)*

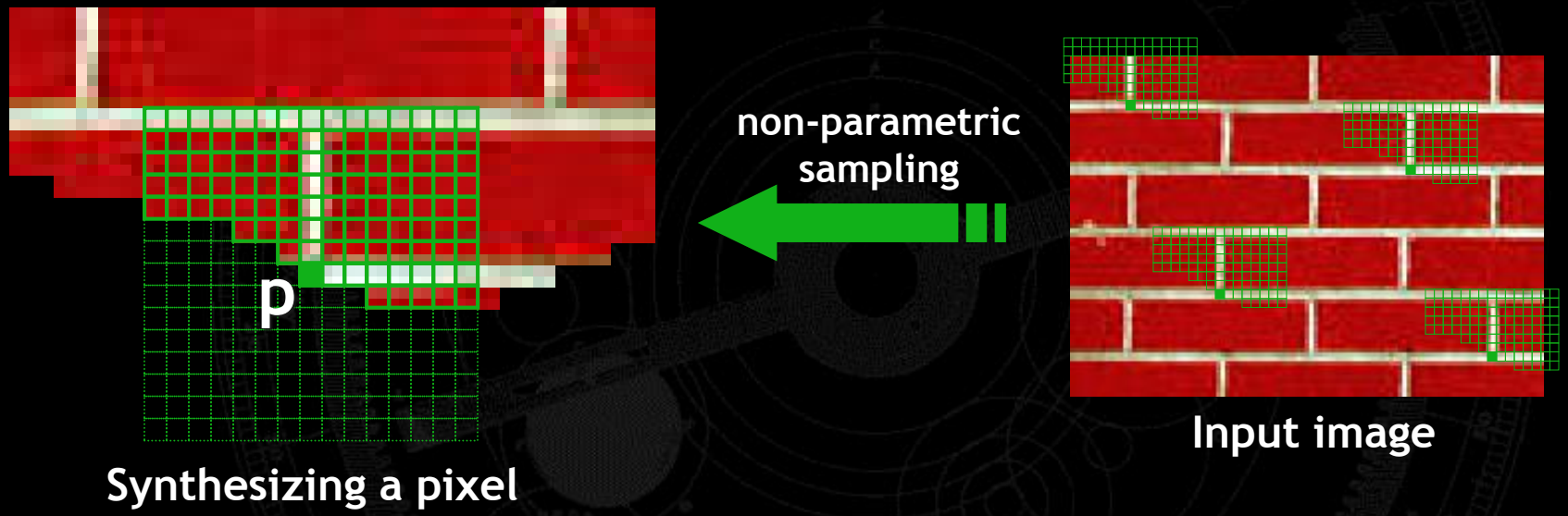
Efros & Leung ’99

[Shannon,’48] proposed a way to generate English-looking text using N-grams:

- Assume a generalized Markov model
- Use a large text to compute prob. distributions of each letter given N-1 previous letters
- Starting from a seed repeatedly sample this Markov chain to generate new letters
- Also works for whole words

WE NEED TO EAT CAKE

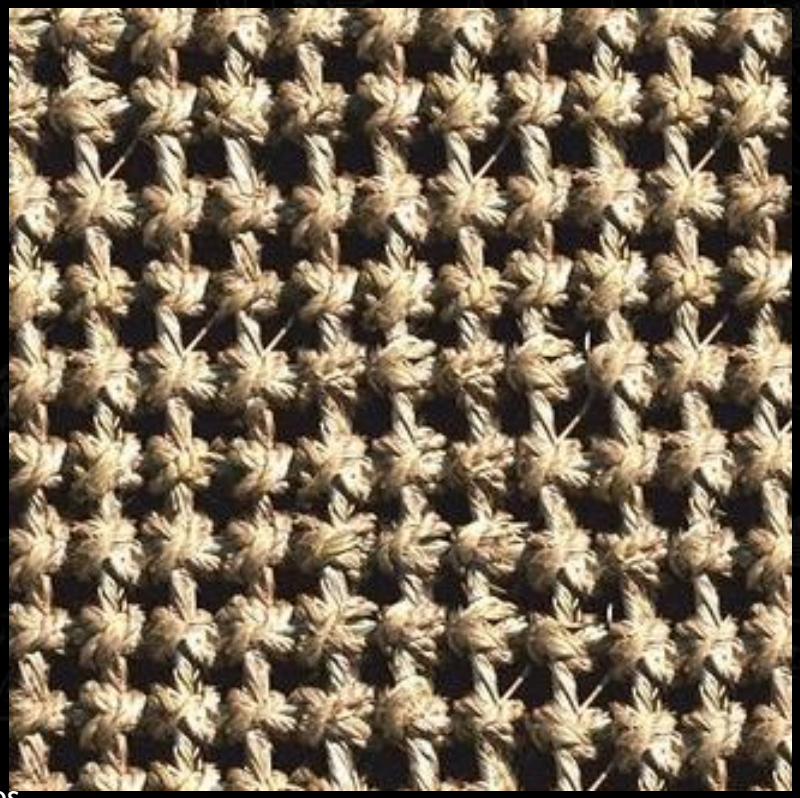
Efros & Leung ’99



Assuming Markov property, compute P(p|N(p))

- Building explicit probability tables infeasible
- Instead, let’s *search the input image* for all similar neighborhoods — that’s our histogram for **p**

To synthesize p, just pick one match at random



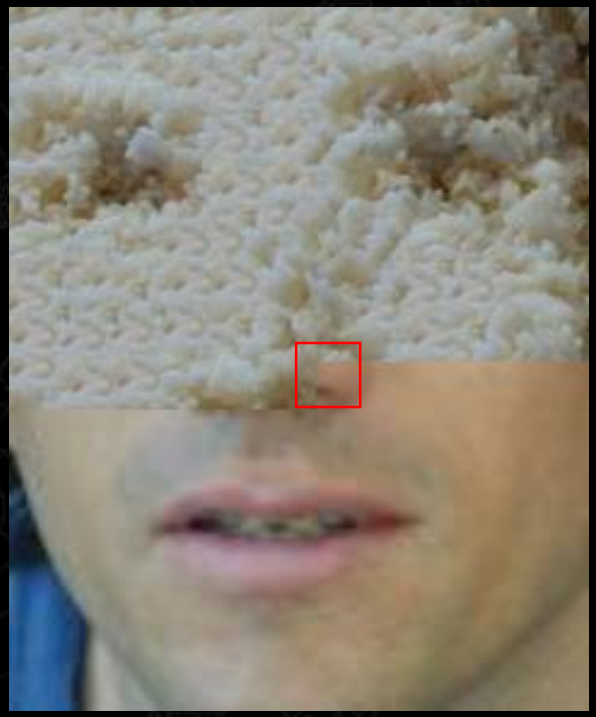
Efros

Efros

Texture Transfer

Take the texture from one object and "paint" it onto another object

- This requires separating texture and shape
- That's HARD, but we can cheat
- Assume we can capture shape by boundary and rough shading



Then, just add another constraint when sampling: similarity to underlying image at that spot

parmesan

+

=

rice

+

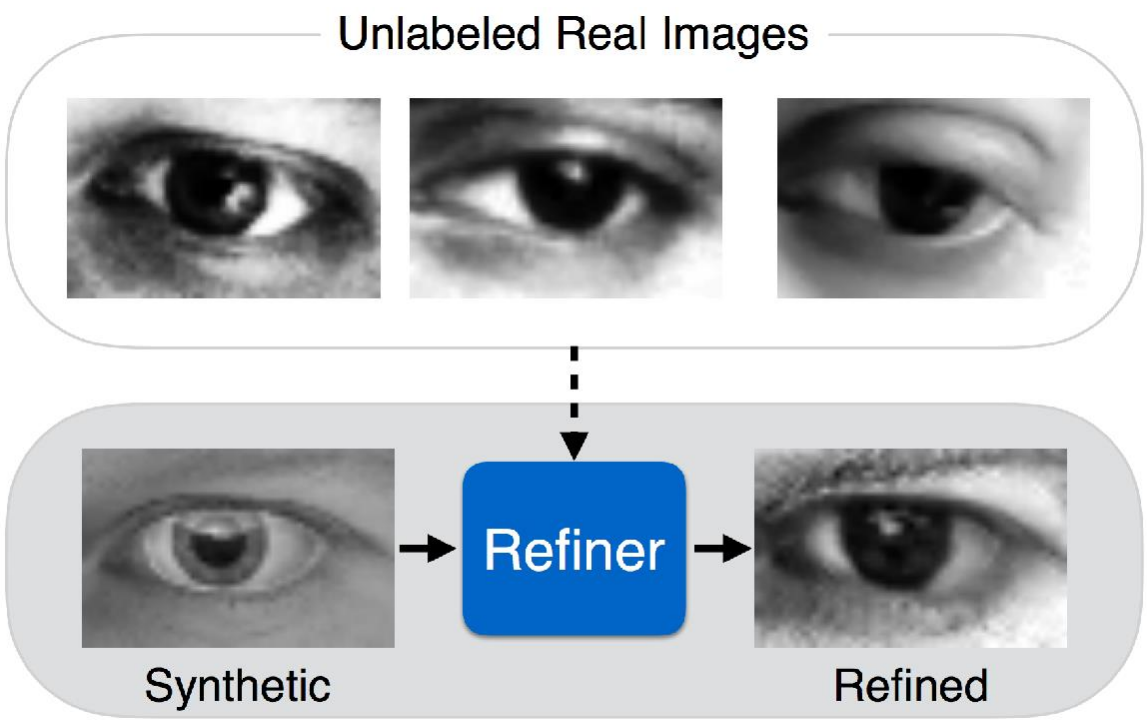
=

Efros

Plan for today

- Generative models: What are they?
- Techniques: Generative Adversarial Networks
  - Conditional GANs
  - Cycle-consistency loss
  - Dealing with sparse data
  - *Applications interleaved with techniques*
- Generation before GANs
- Generation for synthetic training data

Generating Training Data



Shrivastava et al., "Learning From Simulated and Unsupervised Images Through Adversarial Training", CVPR 2017

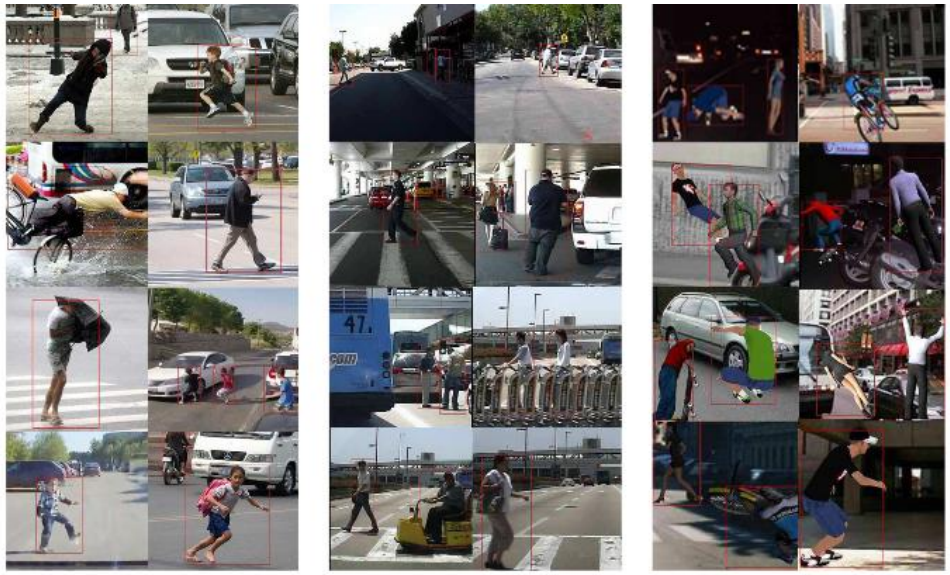
Generating Training Data

(a) Image examples from the Linemod dataset.

(b) Examples generated by our model, trained on Linemod.

Bousmalis et al., "Unsupervised Pixel-Level Domain Adaptation With Generative Adversarial Networks", CVPR 2017

Generating Training Data

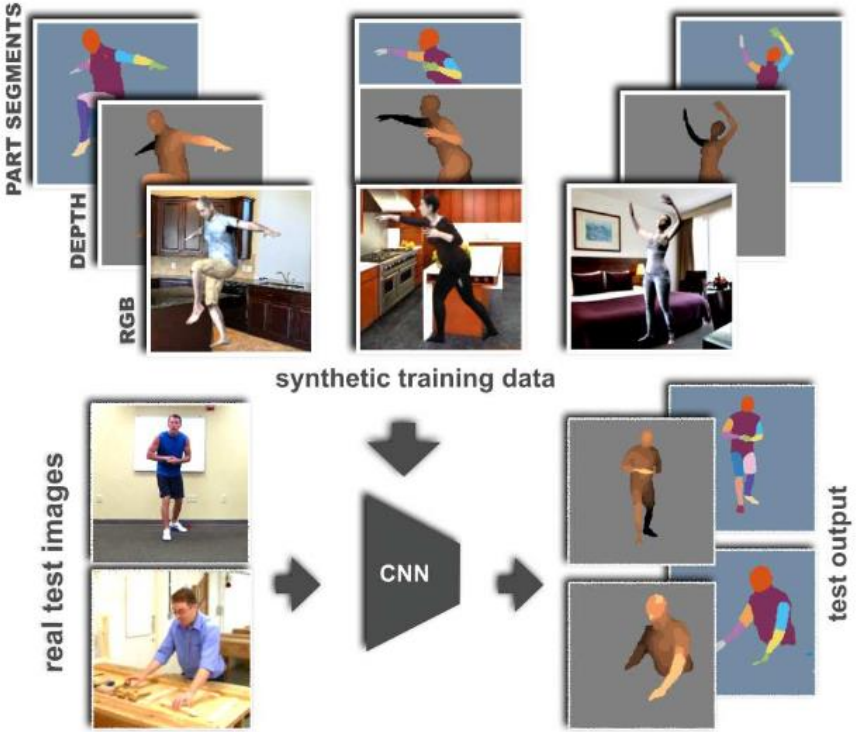


(a) (b) (c)

Figure 1: (a) Examples from our novel Precarious Pedestrian Dataset of dangerous, but rare pedestrian scenes.

Huang and Ramanan, "Expecting the Unexpected: Training Detectors for Unusual Pedestrians With Adversarial Imposters", CV

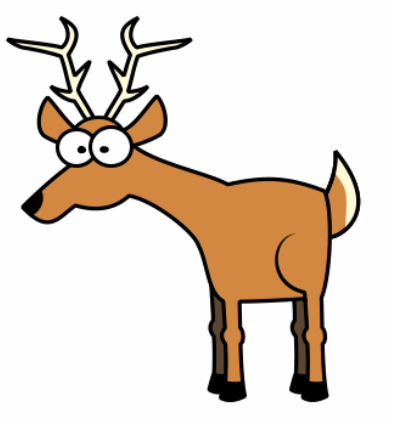
Generating Training Data



Varol et al., "Learning From Synthetic Humans", CVPR 2017

Recognizing atypical objects

- Computer vision models cannot recognize objects in unusual modalities well



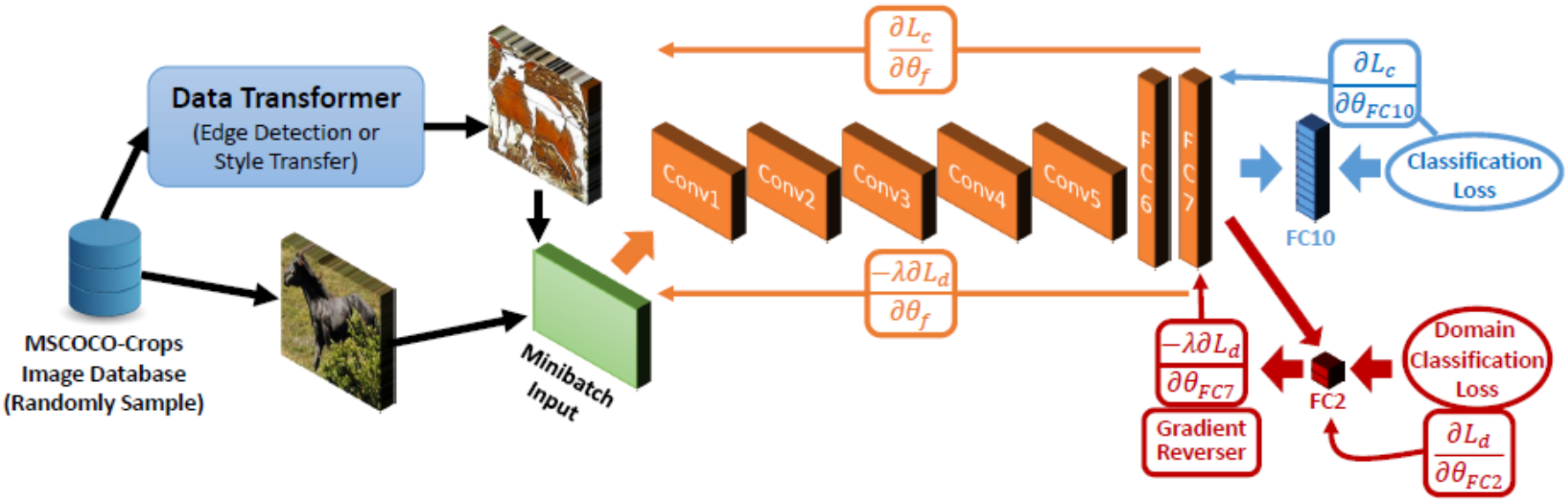
Thomas and Kovashka, in submission to ECCV 2018

Recognizing atypical objects

- Training a network with two source modalities helps it generalize to novel target modalities
- We can obtain good secondary modalities for free:
  - Automatic transformation of a photo to a cartoon/sketch (style transfer by Johnson et al. ECCV 2016)

Thomas and Kovashka, in submission to ECCV 2018

Recognizing atypical objects



Thomas and Kovashka, in submission to ECCV 2018

Recognizing atypical objects

- We outperform prior domain adaptation methods
  - Including a generous modification of Bousmalis et al. 2017 ~ our approach but with GAN instead of style transfer

	PACS: Art, Cartoons, Sketches (Li 2017)	Sketchy Database (Sangkloy 2016)
No Adaptation	0.388605333	0.093055
Ganin & Lempitsky 2015	0.537944333	0.112329
Castrejon et al. 2016	0.400783	0.085107
Ghifary et al. 2016	0.462287276	0.22547
Long et al. 2016	*0.566674*	*0.303163*
Bousmalis et al. 2017 + domain confusion	0.547297333	0.284765
Ours	**0.587813333**	**0.326453**
Supervised (upper bound)	0.904064333	0.822074

Thomas and Kovashka, in submission to ECCV 2018