Indian Institute of Information Technology, Allahabad

Generative Adversarial Networks

By

Dr. Satish Kumar Singh & Dr. Shiv Ram Dubey
Computer Vision and Biometrics Lab
Department of Information Technology
Indian Institute of Information Technology, Allahabad

TEAM

Computer Vision and Biometrics Lab (CVBL)

Department of Information Technology

Indian Institute of Information Technology Allahabad

Course Instructors

Dr. Satish Kumar Singh, Associate Professor, IIIT Allahabad (Email: sk.singh@iiita.ac.in)

Dr. Shiv Ram Dubey, Assistant Professor, IIIT Allahabad (Email: srdubey@iiita.ac.in)

DISCLAIMER

The content (text, image, and graphics) used in this slide are adopted from many sources for academic purposes. Broadly, the sources have been given due credit appropriately. However, there is a chance of missing out some original primary sources. The authors of this material do not claim any copyright of such material.

Outline

- Unsupervised Learning
- Generative tasks
- Generative Adversarial Networks
- GAN Objective Function
- GAN Variants
- Image-to-Image Translation
- GANs: Recent Trends

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

Credit: cs231n, Stanford

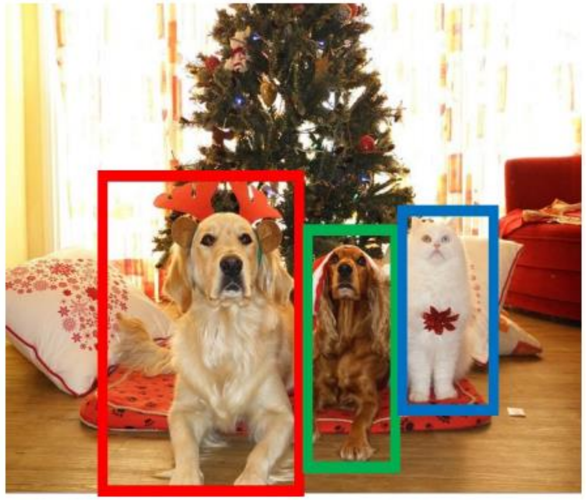
Supervised vs Unsupervised Learning

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

Credit: cs231n, Stanford

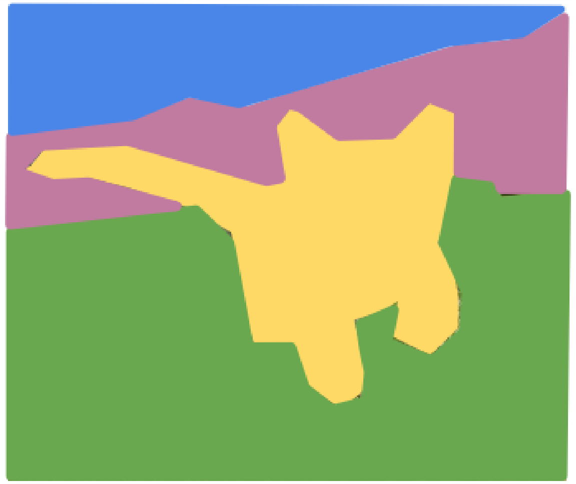
Supervised vs Unsupervised Learning

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

Credit: cs231n, Stanford

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.

Credit: cs231n, Stanford

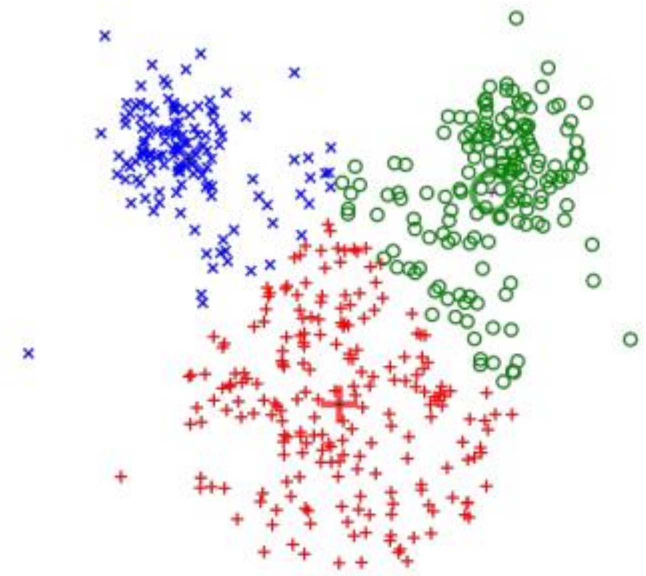
Supervised vs Unsupervised Learning

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

Credit: cs231n, Stanford

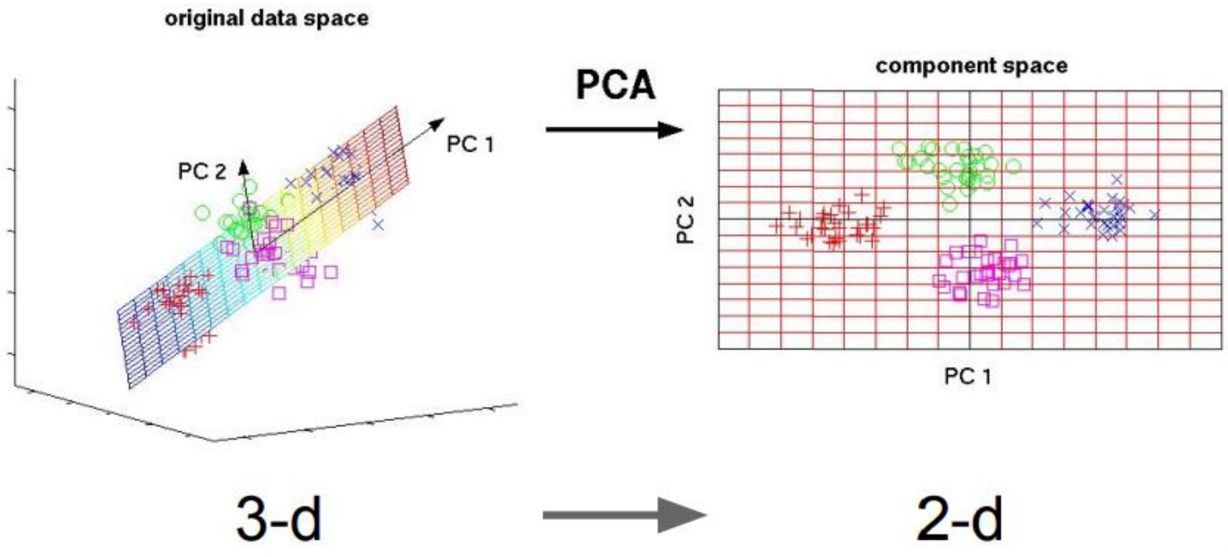
Supervised vs Unsupervised Learning

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

Credit: cs231n, Stanford

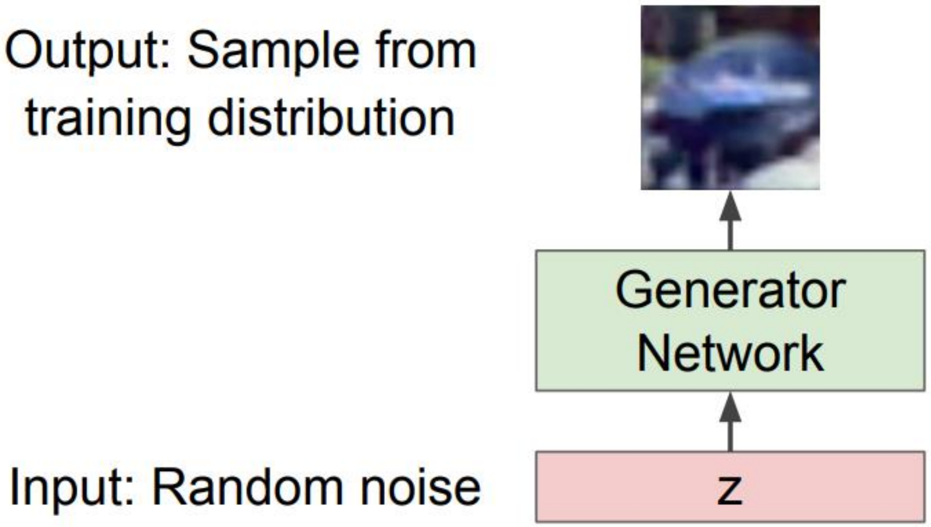
Supervised vs Unsupervised Learning

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.

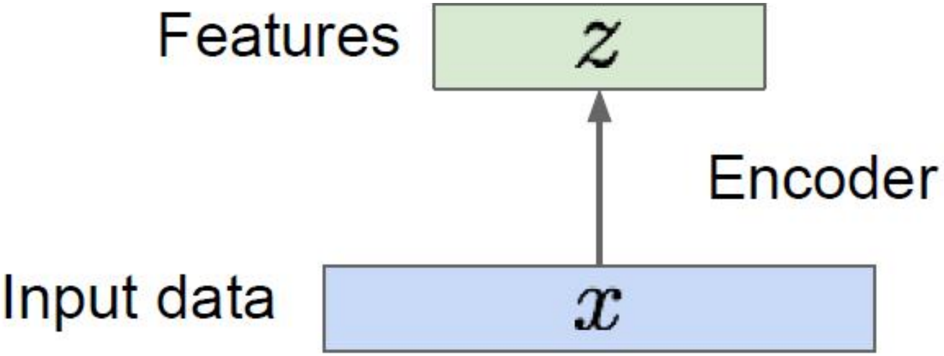


Generative Advarsarial Networks (Distribution learning)

Credit: cs231n, Stanford

Autoencoders

Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data

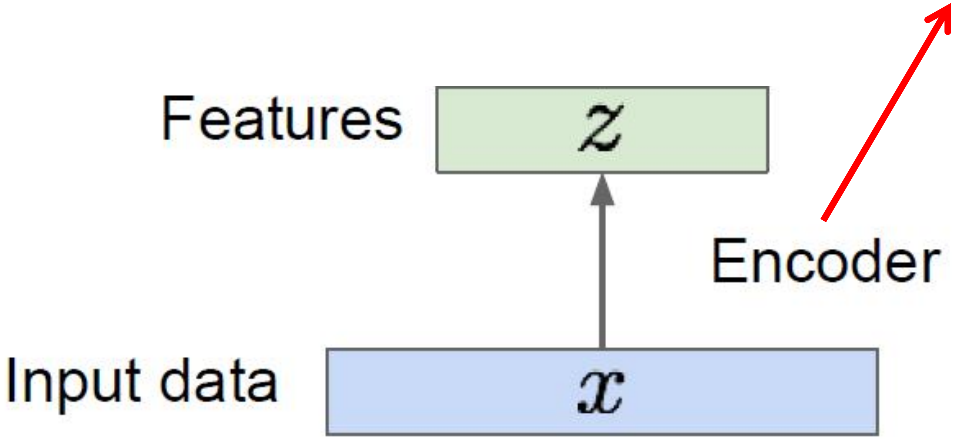


Credit: cs231n, Stanford

Autoencoders

Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data

Originally: Linear + nonlinearity (sigmoid)
Later: Deep, fully-connected
Later: ReLU CNN



Credit: cs231n, Stanford

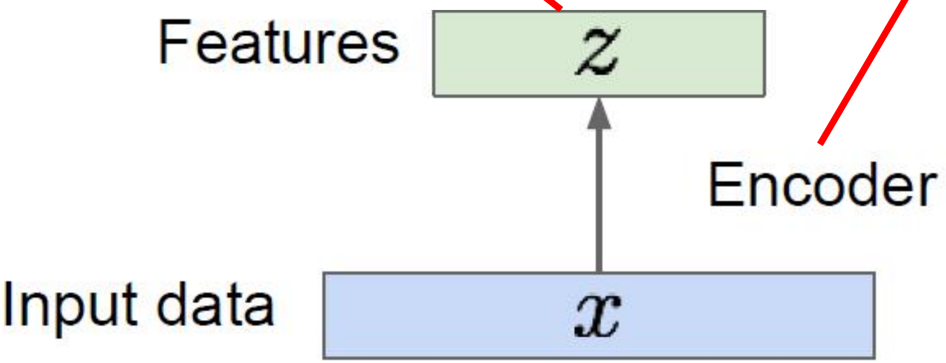
Autoencoders

Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data

Z usually smaller than X (Dimensionality Reduction)

Q: Why dimensionality reduction?

Originally: Linear + nonlinearity (sigmoid)
Later: Deep, fully-connected
Later: ReLU CNN



Credit: cs231n, Stanford

Autoencoders

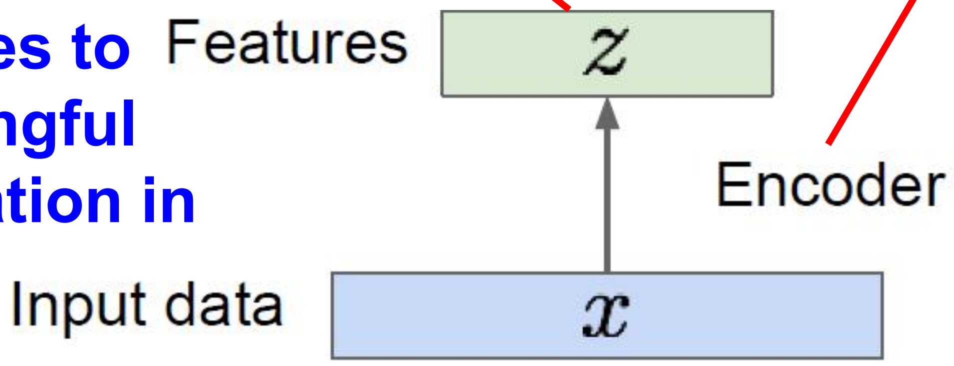
Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data

Z usually smaller than X (Dimensionality Reduction)

Q: Why dimensionality reduction?

A: Want features to capture meaningful factors of variation in data

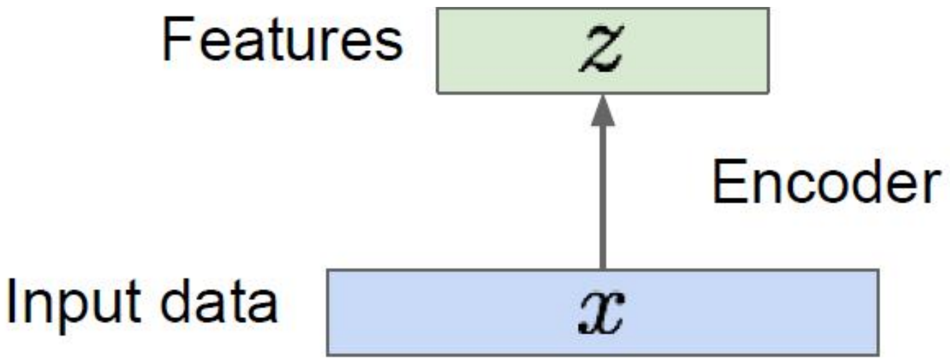
Originally: Linear + nonlinearity (sigmoid)
Later: Deep, fully-connected
Later: ReLU CNN



Credit: cs231n, Stanford

Autoencoders

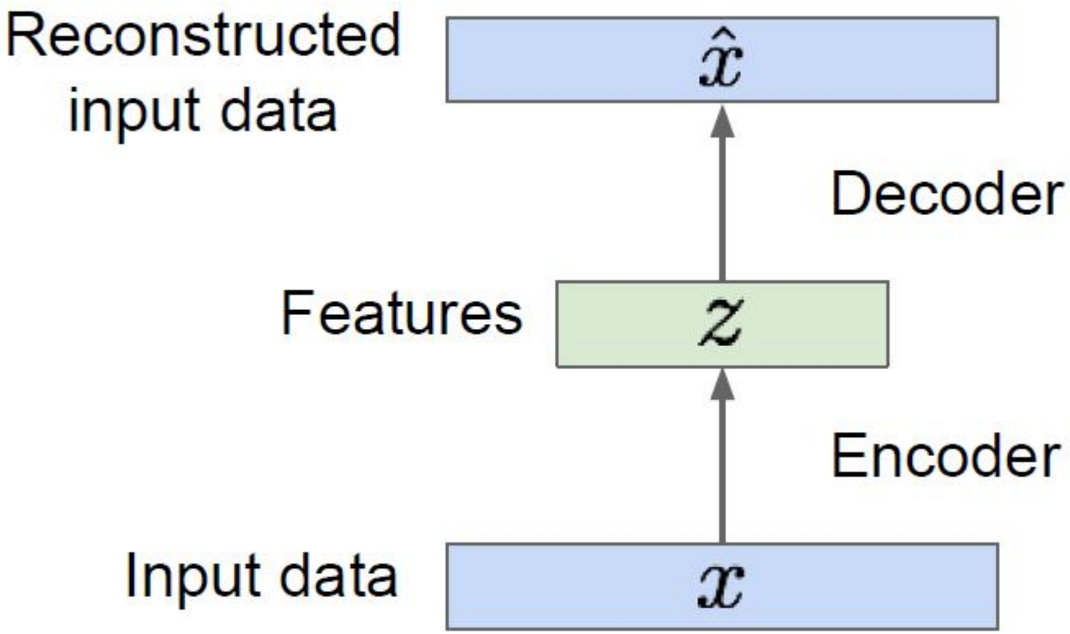
How to learn this feature representation?



Credit: cs231n, Stanford

Autoencoders

How to learn this feature representation?
Train such that features can be used to reconstruct original data “Autoencoding” - encoding itself

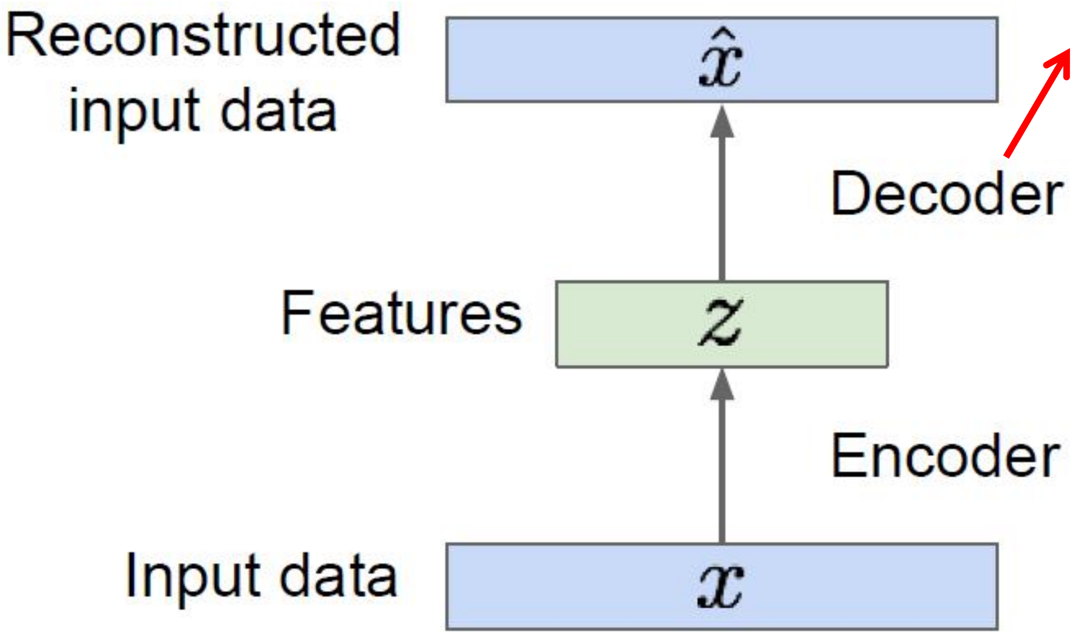


Credit: cs231n, Stanford

Autoencoders

How to learn this feature representation?

Train such that features can be used to reconstruct original data “Autoencoding” - encoding itself



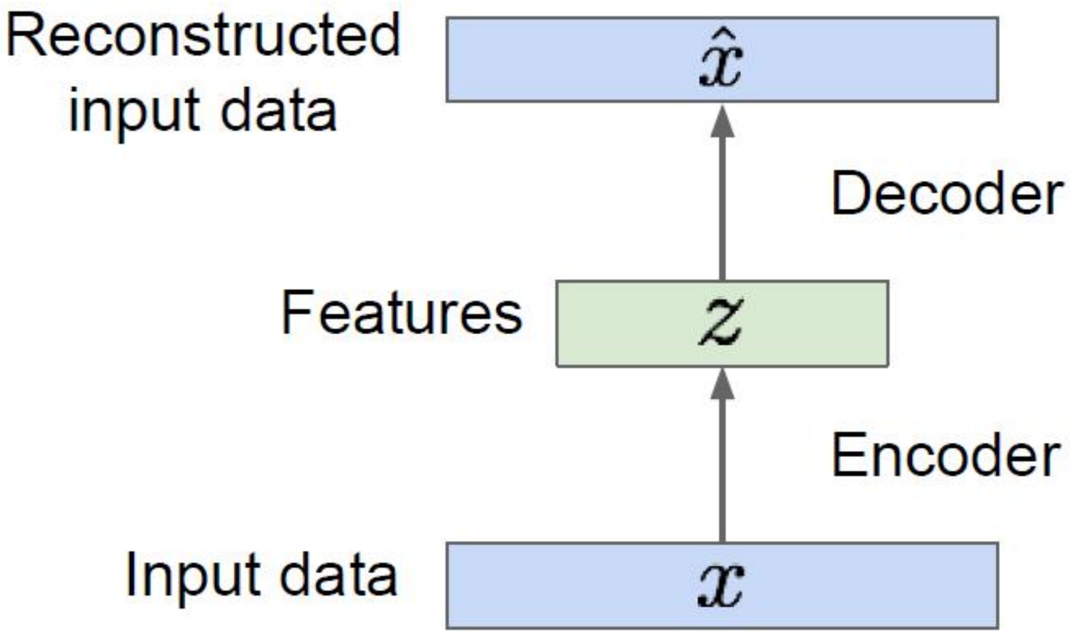
Originally: Linear + nonlinearity (sigmoid)
Later: Deep, fully-connected
Later: ReLU CNN

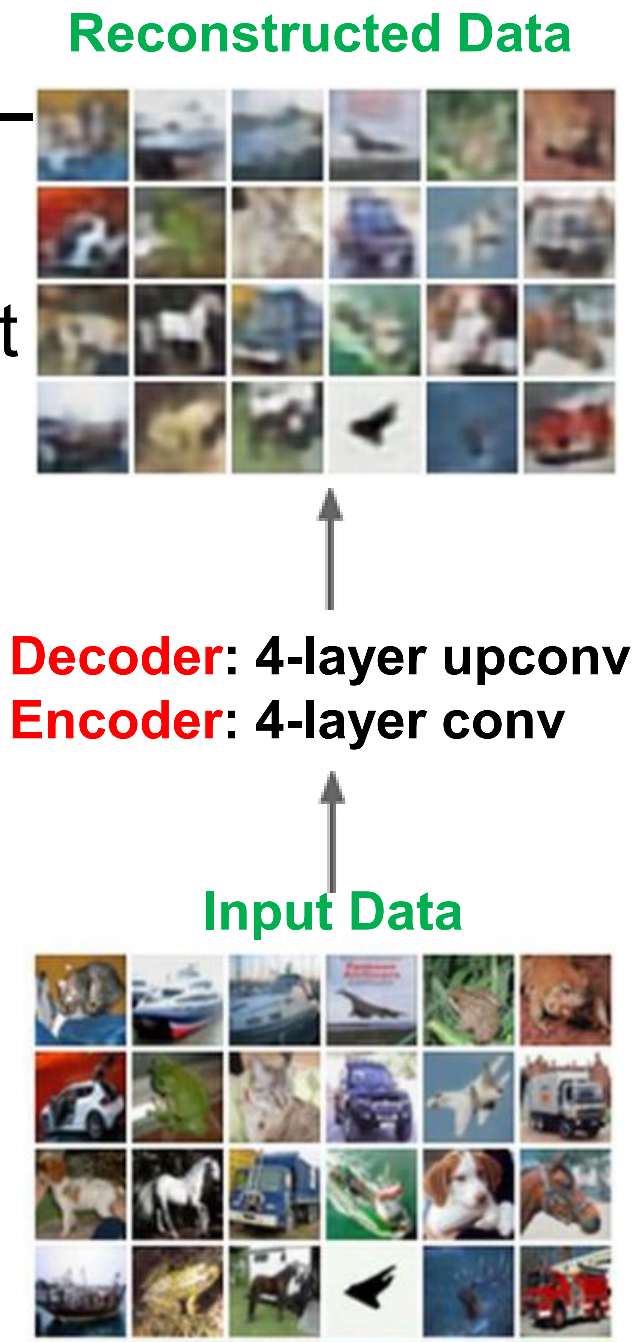
Credit: cs231n, Stanford

Autoencoders

How to learn this feature representation?

Train such that features can be used to reconstruct original data "Autoencoding" - encoding itself



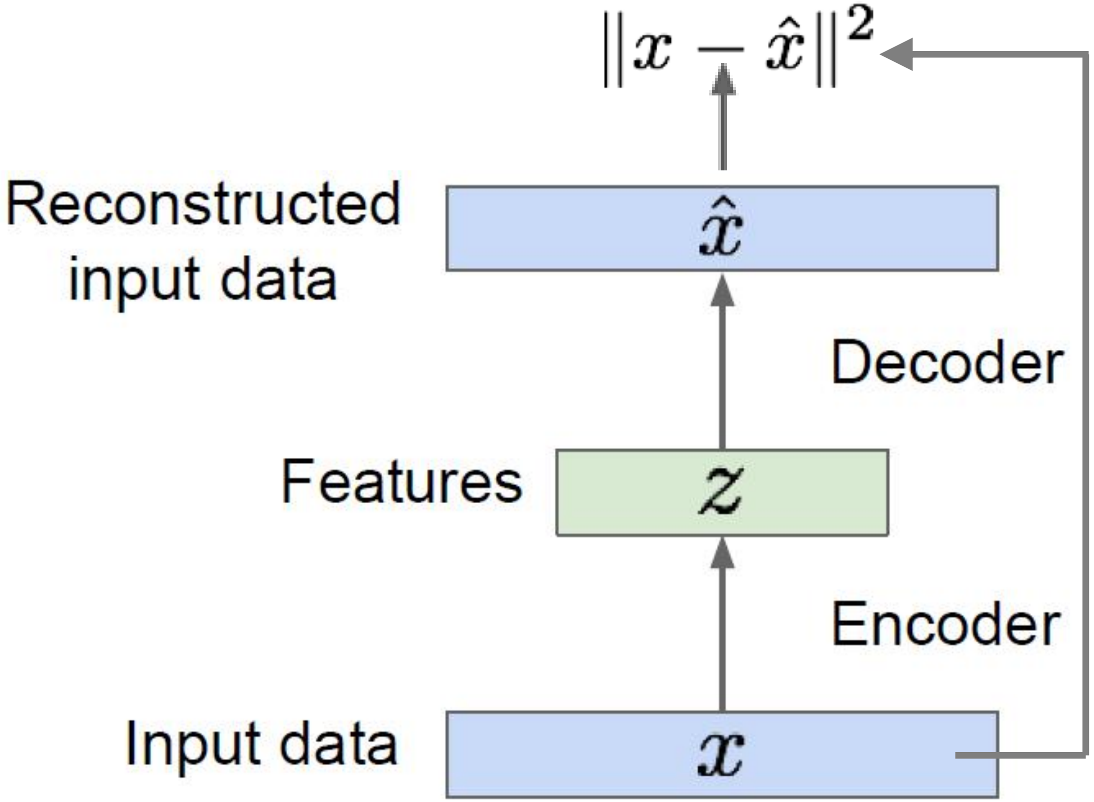


Credit: cs231n, Stanford

Autoencoders

Train such that features can be used to reconstruct original data

L2 Loss Function



Reconstructed Data

Decoder: 4-layer upconv
Encoder: 4-layer conv

Input Data

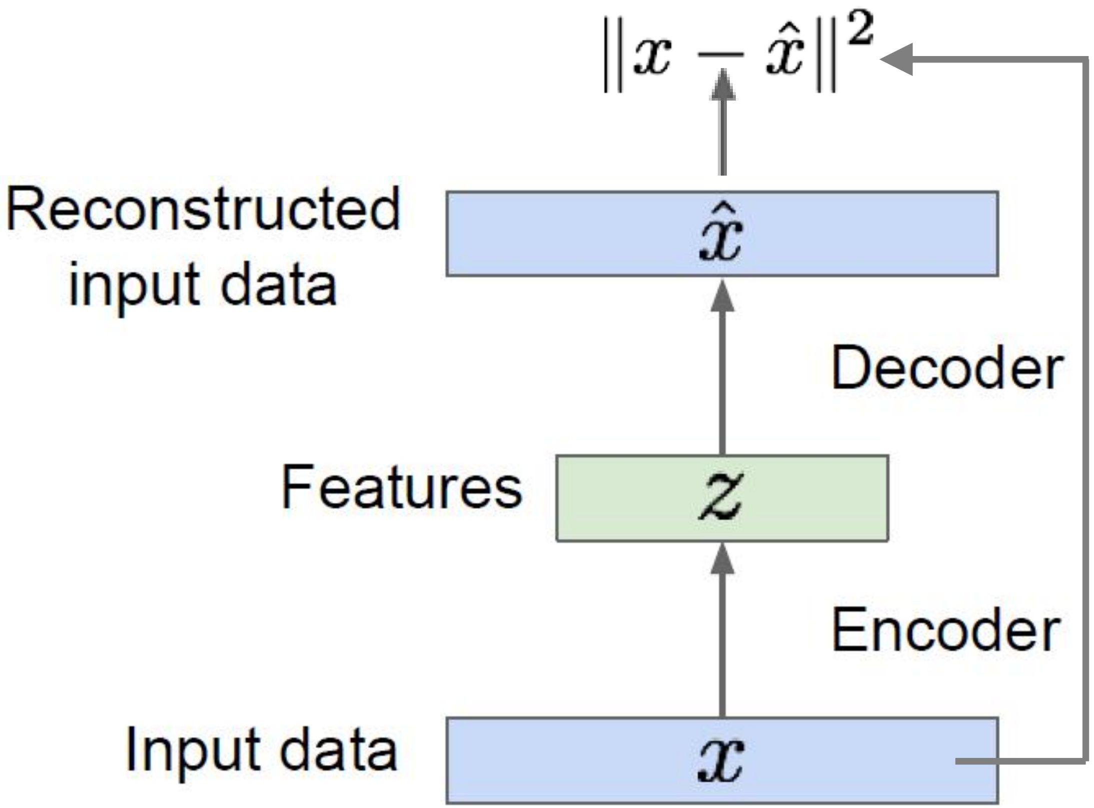
Credit: cs231n, Stanford

Autoencoders

Train such that features can be used to reconstruct original data

Doesn’t use labels!

L2 Loss Function



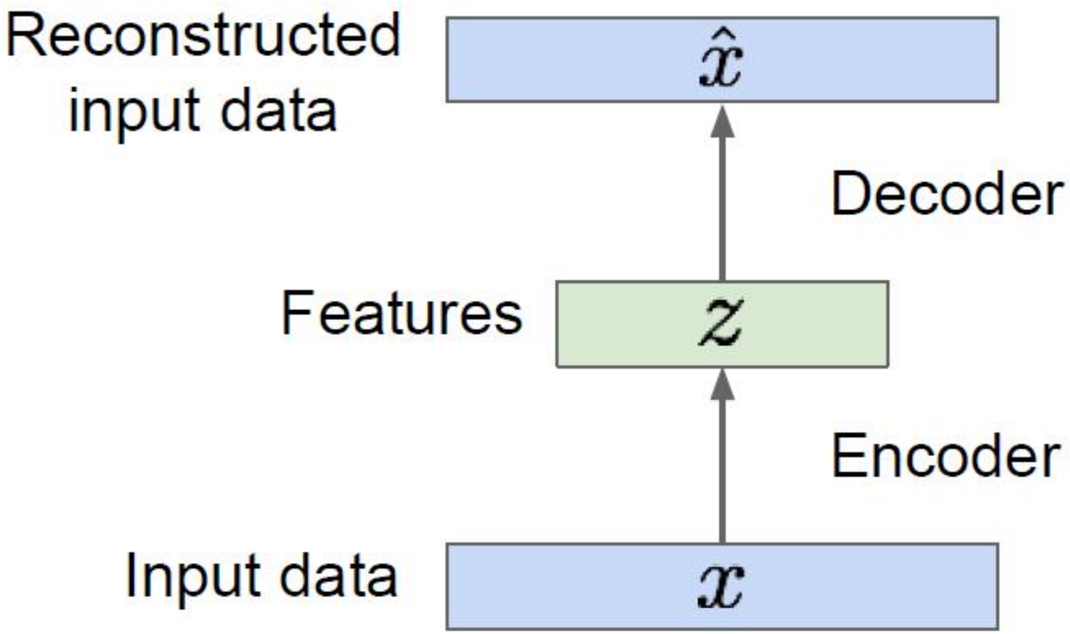
Reconstructed Data

Decoder: 4-layer upconv
Encoder: 4-layer conv

Input Data

Credit: cs231n, Stanford

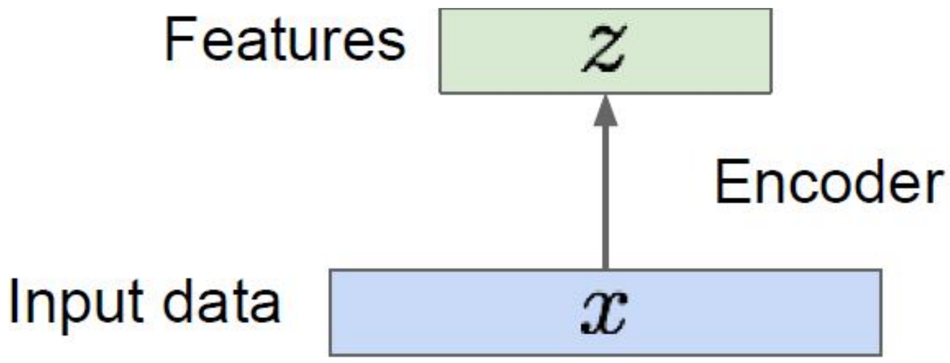
Autoencoders



After training, throw away decoder

Credit: cs231n, Stanford

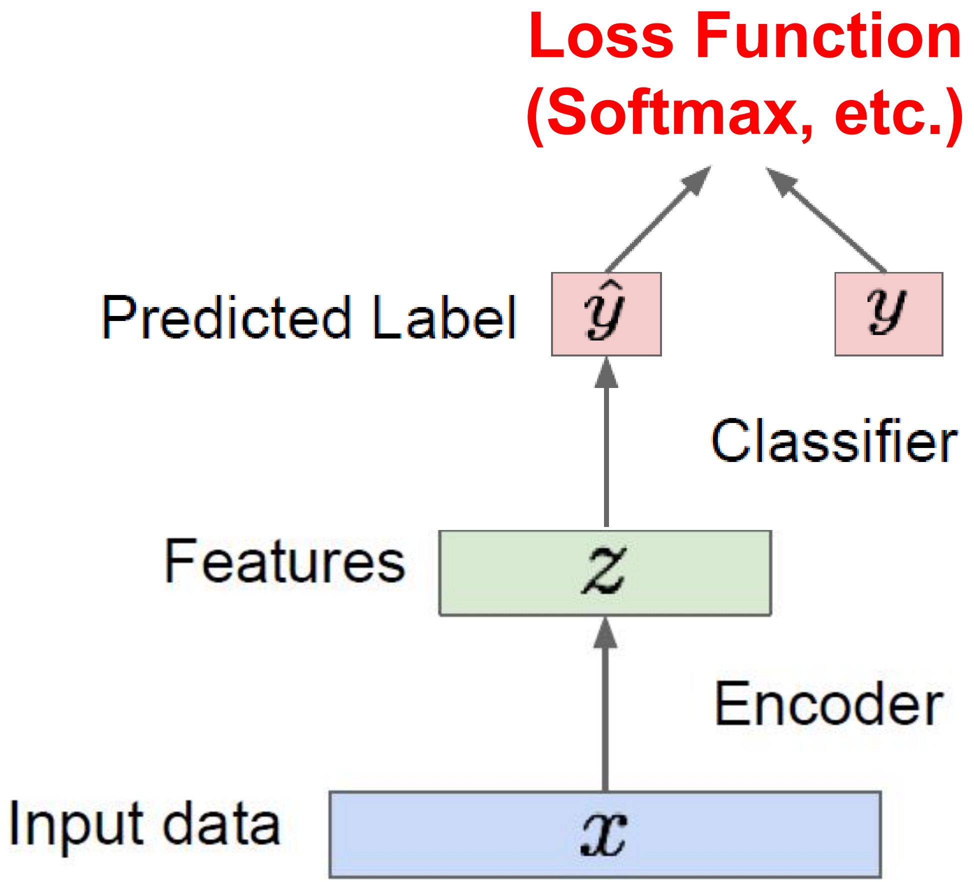
Autoencoders



Credit: cs231n, Stanford

Autoencoders

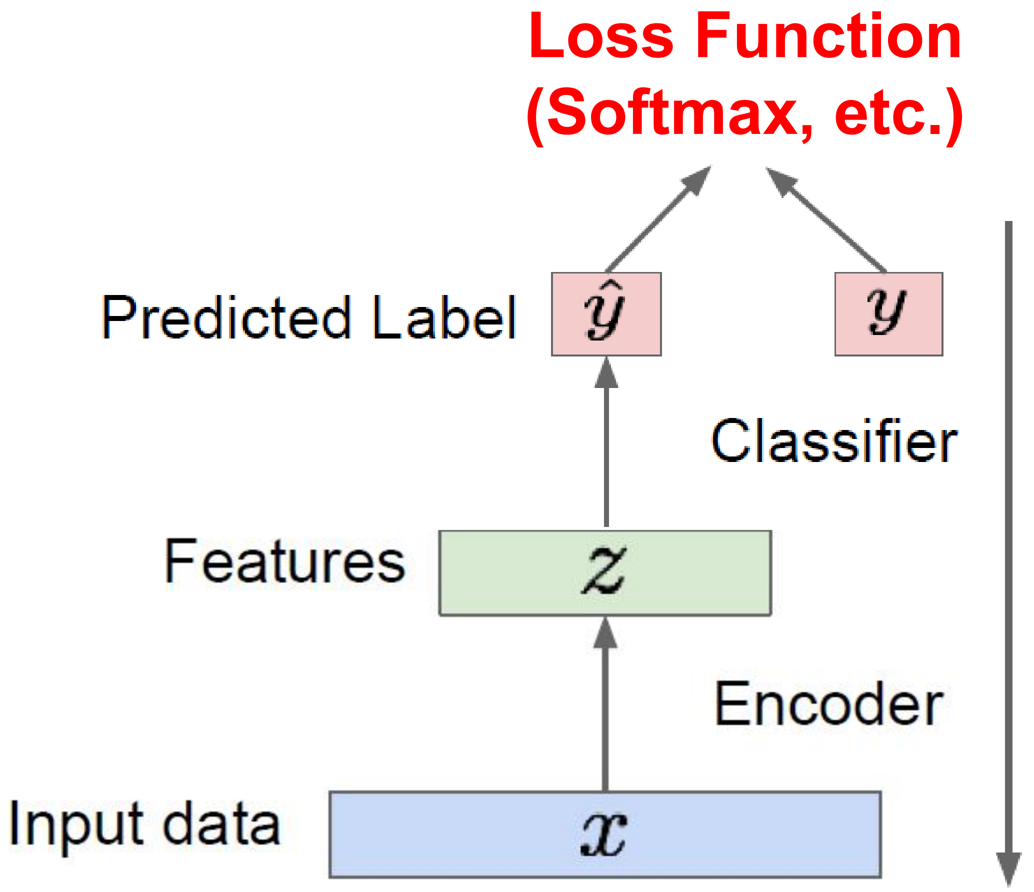
Encoder can be used to initialize a supervised model



Credit: cs231n, Stanford

Autoencoders

Encoder can be used to initialize a supervised model



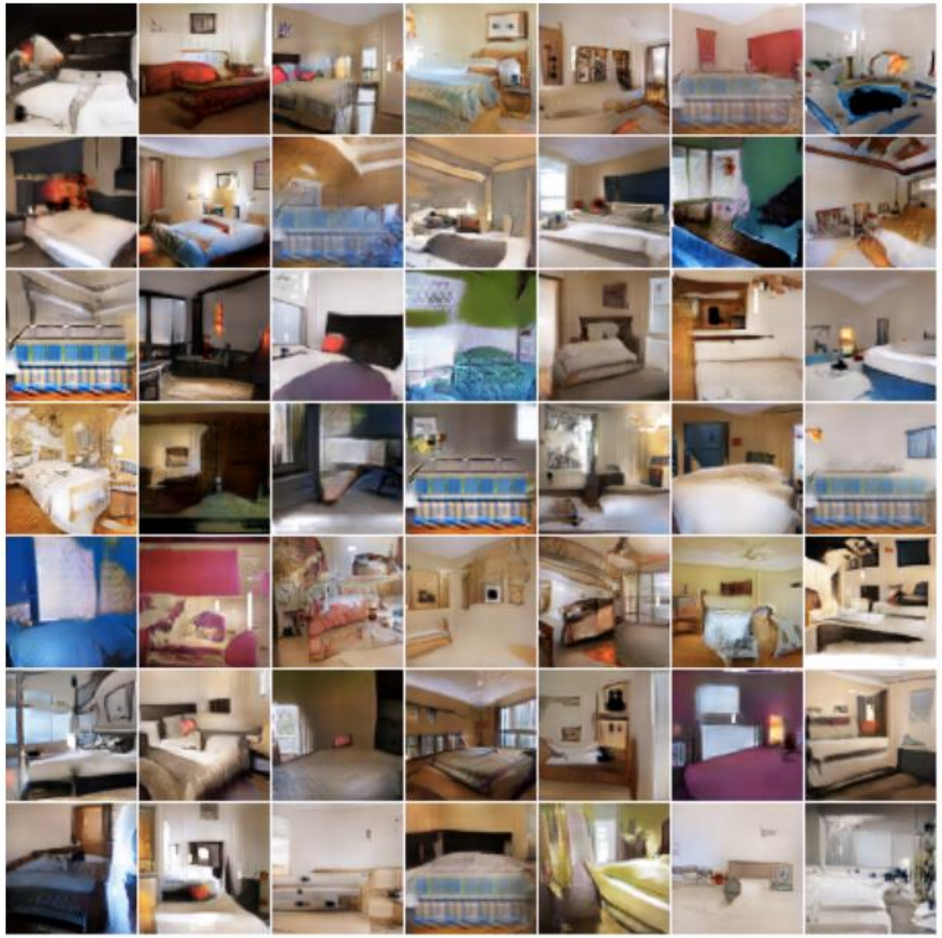
Fine-tune encoder jointly with classifier

Train for final task (sometimes with small data)

Credit: cs231n, Stanford

Generative tasks

- Generation (from scratch): learn to sample from the distribution represented by the training set
  - *Unsupervised learning* task



Generative tasks

- Generation conditioned on class label

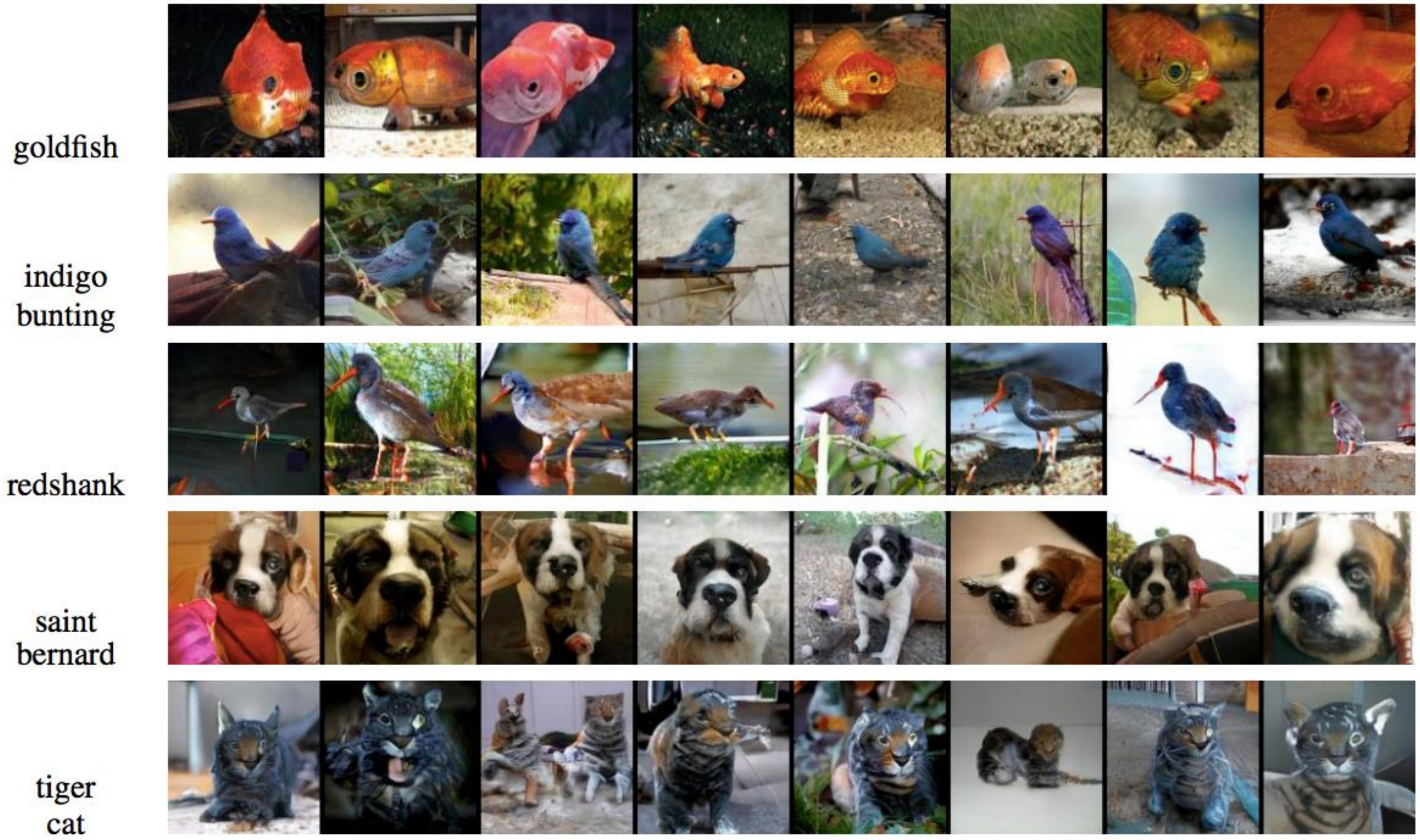
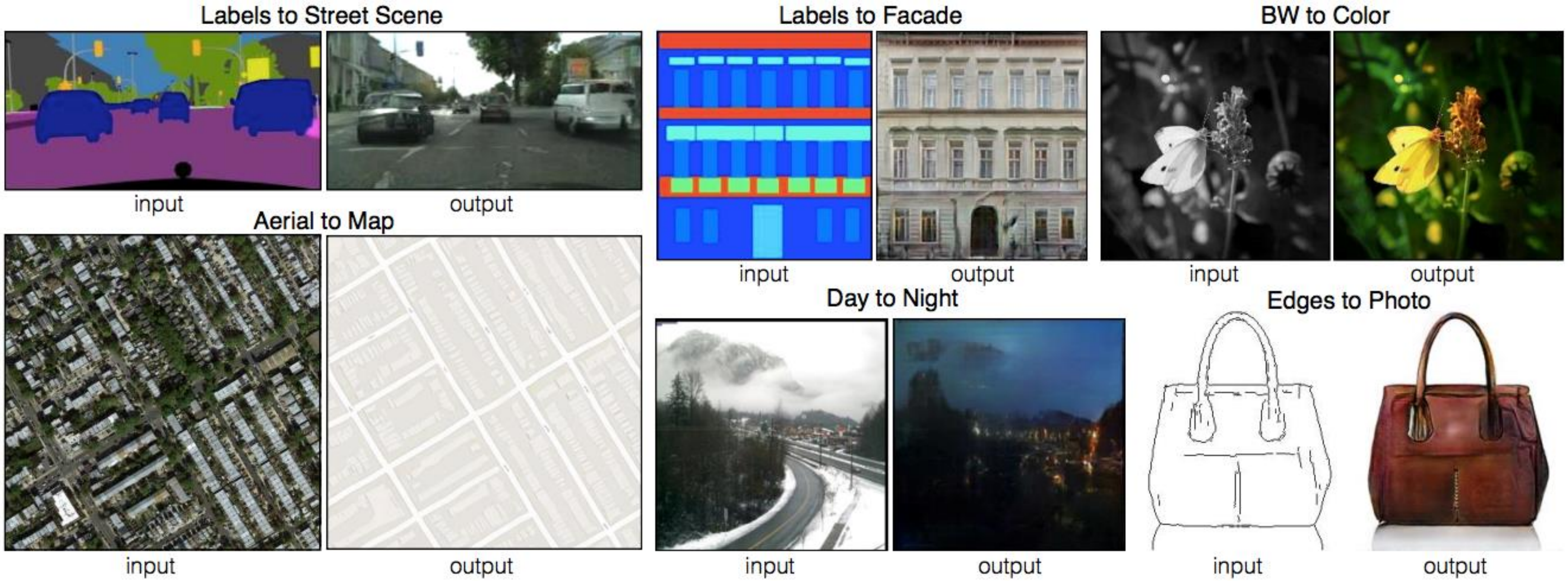


Figure source

Generative tasks

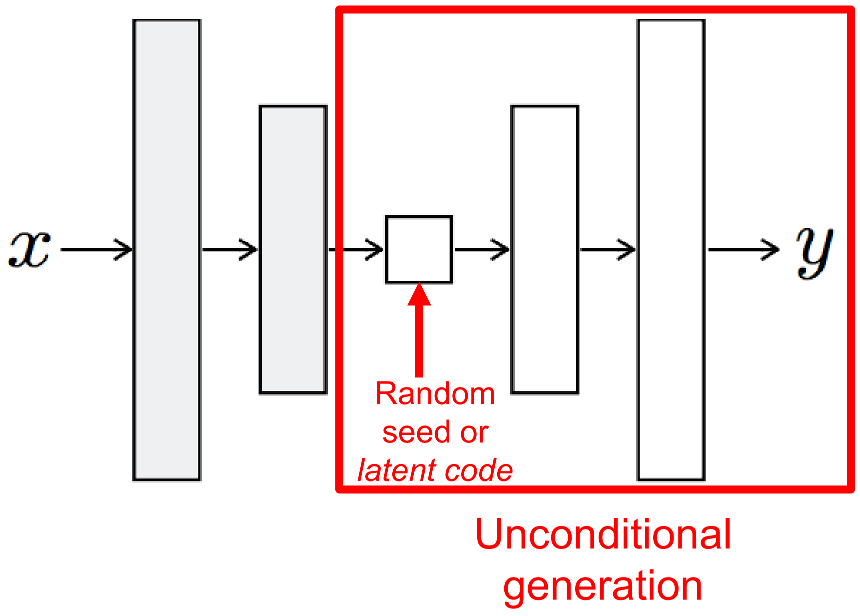
- Generation conditioned on image (image-to-image translation)



P. Isola, J.-Y. Zhu, T. Zhou, A. Efros, [Image-to-Image Translation with Conditional Adversarial Networks](), CVPR 2017

Designing a network for generative tasks

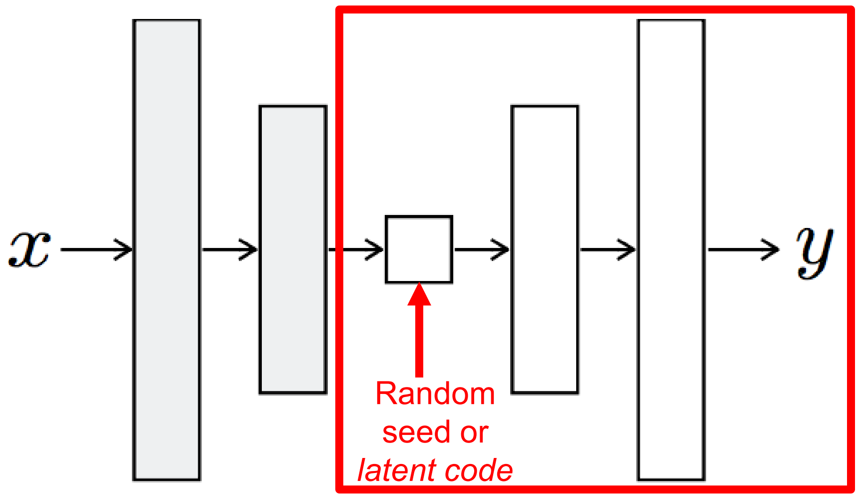
1. We need an architecture that can generate an image
   - Recall upsampling architectures for dense prediction



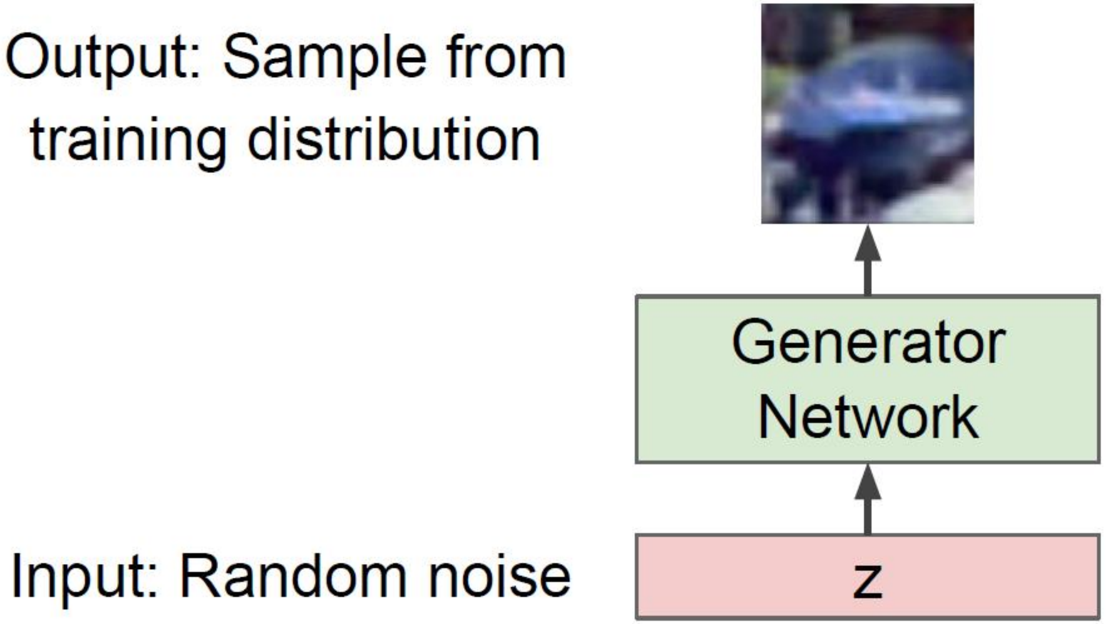
Designing a network for generative tasks

1. We need an architecture that can generate an image
   - Recall upsampling architectures for dense prediction

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



Unconditional generation



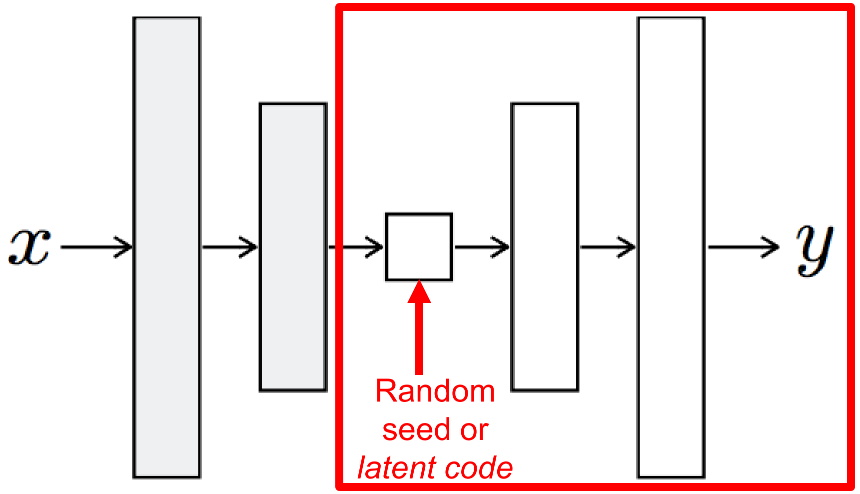
Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Credit: cs231n, Stanford

Designing a network for generative tasks

1. We need an architecture that can generate an image
   - Recall upsampling architectures for dense prediction

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

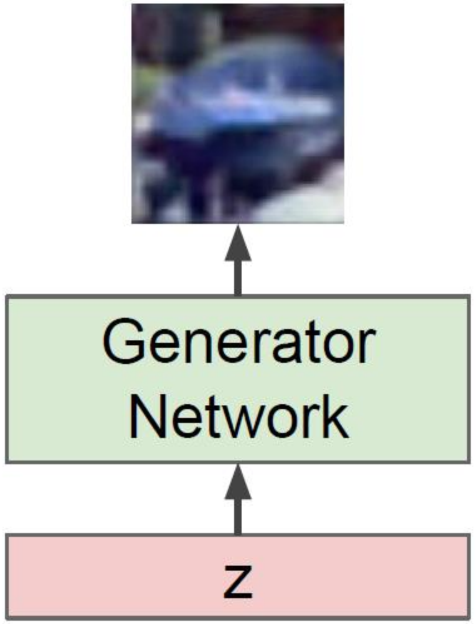


Unconditional generation

Output: Sample from training distribution

A neural network can be used to represent this complex transformation?

Input: Random noise



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

Credit: cs231n, Stanford

Designing a network for generative tasks

1. We need an architecture that can generate an image
    - Recall upsampling architectures for dense prediction

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

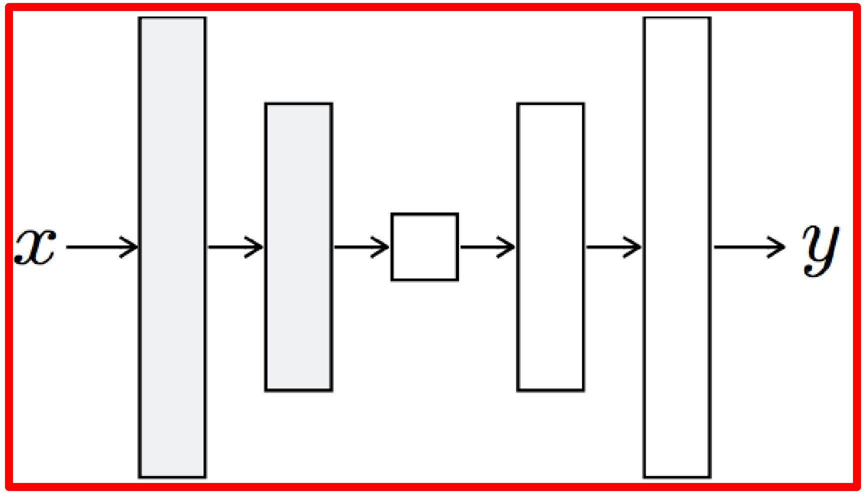
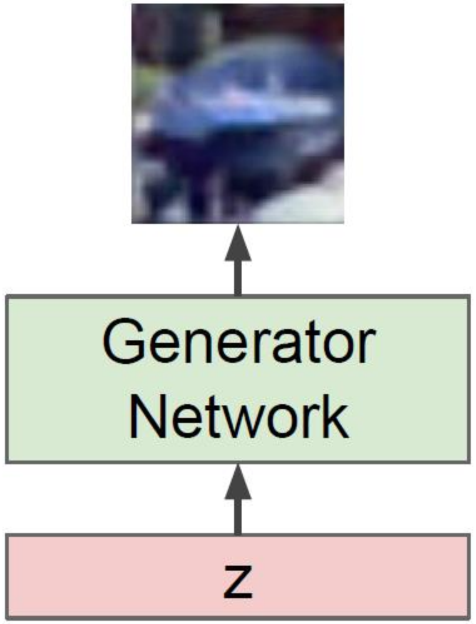


Image-to-image translation

Output: Sample from training distribution

A neural network can be used to represent this complex transformation?

Input: Random noise



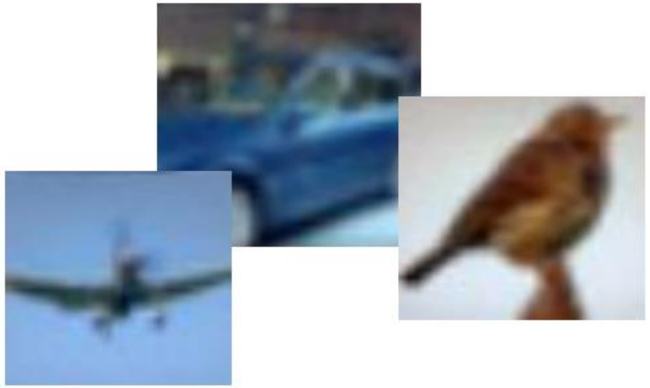
Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Credit: cs231n, Stanford

Designing a network for generative tasks

1. We need an architecture that can generate an image
   - Recall upsampling architectures for dense prediction
2. We need to design the right loss function

Learning to sample



Training data $x \sim p_{\text{data}}$

Generated samples $x \sim p_{\text{model}}$

We want to learn p_{model} that matches p_{data}

Adapted from [Stanford CS231n]

Generative adversarial networks

- Train two networks with opposing objectives:
  - **Generator:** learns to generate samples
  - **Discriminator:** learns to distinguish between generated and real samples

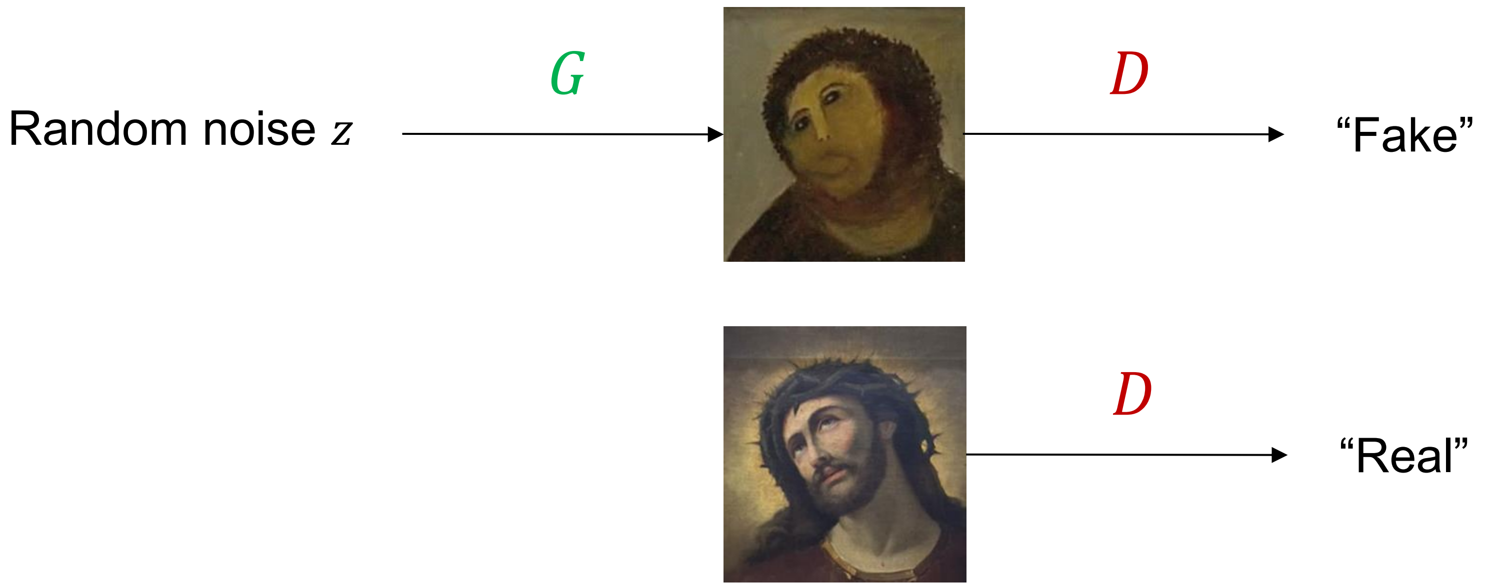


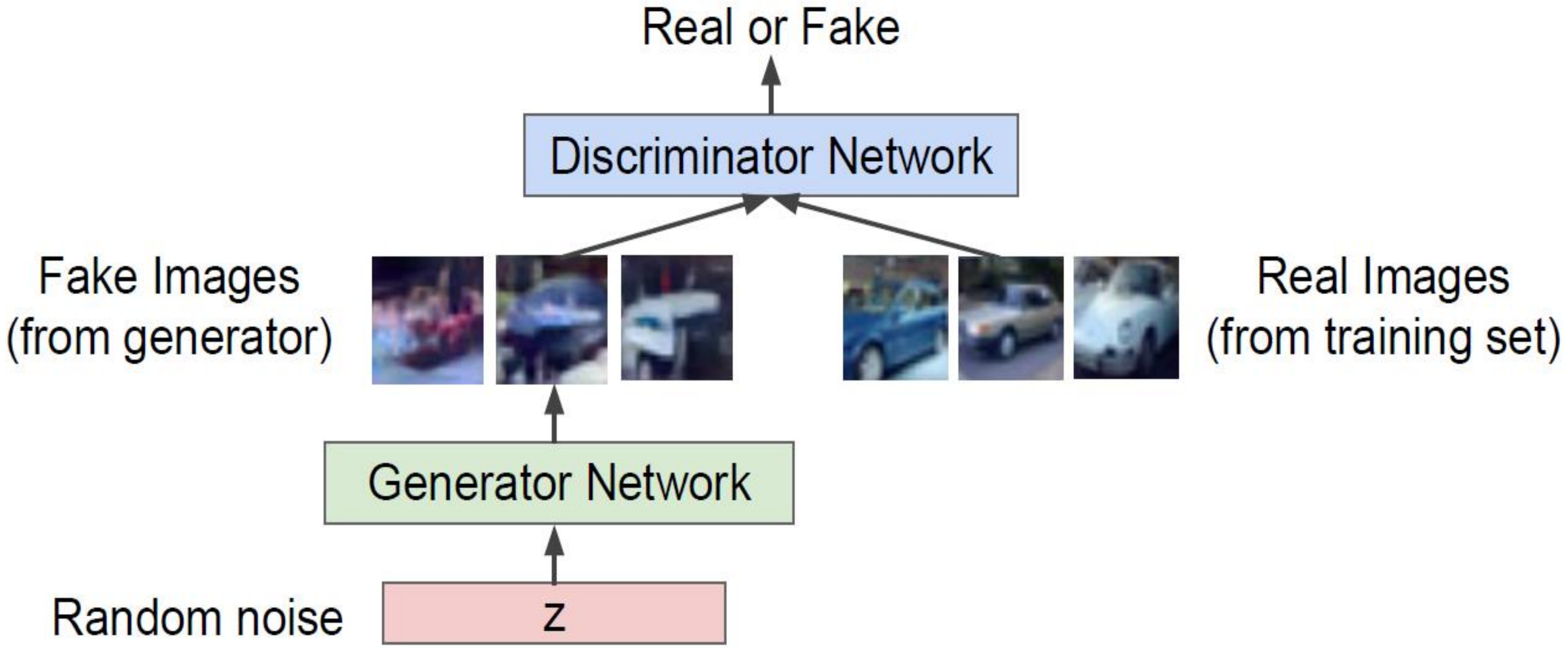
Figure adapted from F. Fleuret

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, NIPS 2014

Generative adversarial networks

Generator network: try to fool the discriminator by generating real-looking images

Discriminator network: try to distinguish between real and fake images



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

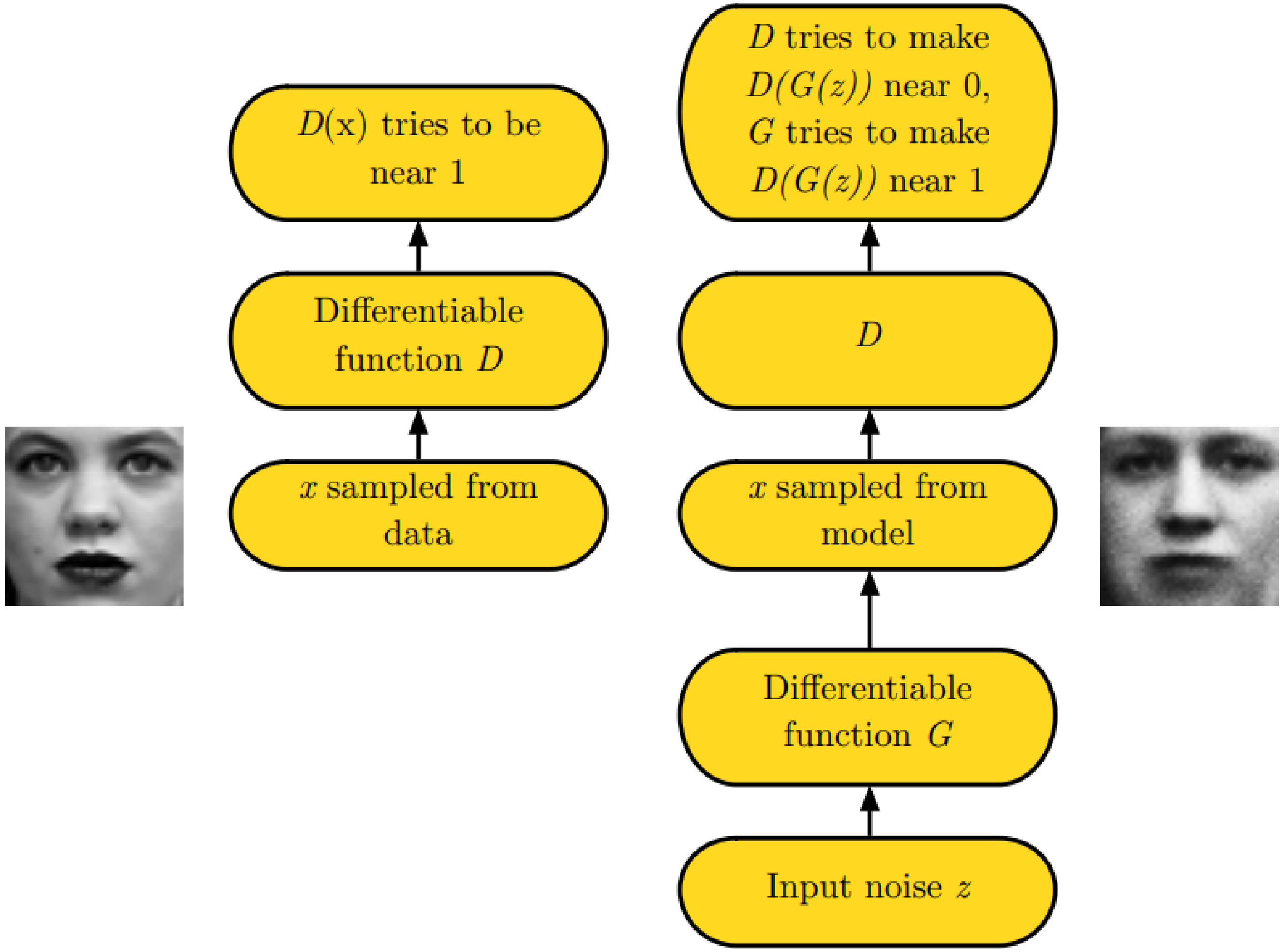
Fake and real images copyright Emily Denton et al. 2015.

Credit: cs231n, Stanford

Generative adversarial networks

Generator network: try to fool the discriminator by generating real-looking images

Discriminator network: try to distinguish between real and fake images



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

Credit: cs231n, Stanford

GAN objective

- The discriminator $D(x)$ should output the probability that the sample x is real
  - That is, we want $D(x)$ to be close to 1 for real data and close to 0 for fake
- Expected conditional log likelihood for real and generated data:

$$\mathbb{E}_{x\sim p_{\text{data}}} \log D(x) \; + \; \mathbb{E}_{x\sim p_{\text{gen}}} \log\big(1 - D(x)\big)$$

$$= \mathbb{E}_{x\sim p_{\text{data}}} \log D(x) \; + \; \mathbb{E}_{z\sim p} \log(1 - D(G(z)))$$

We seed the generator with noise z drawn from a simple distribution p (Gaussian or uniform)

GAN objective

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

- The discriminator wants to correctly distinguish real and fake samples:

$$D^* = \arg\max_D V(G,D)$$

- The generator wants to fool the discriminator:

$$G^* = \arg\min_G V(G,D)$$

- Train the generator and discriminator jointly in a *minimax game*

GAN objective: Theoretical properties

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

- Assuming unlimited capacity for generator and discriminator and unlimited training data:
    - The objective $\min_G \max_D V(G, D)$ is equivalent to *Jensen-Shannon divergence* between p_{data} and p_{gen} and global optimum (*Nash equilibrium*) is given by $p_{\text{data}} = p_{\text{gen}}$

GAN objective: Theoretical properties

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

- Assuming unlimited capacity for generator and discriminator and unlimited training data:
  - The objective $\min_G \max_D V(G,D)$ is equivalent to *Jensen-Shannon divergence* between p_{data} and p_{gen} and global optimum (*Nash equilibrium*) is given by $p_{\text{data}} = p_{\text{gen}}$
  - If at each step, D is allowed to reach its optimum given G, and G is updated to decrease $V(G,D)$, then p_{gen} will eventually converge to p_{data}

GAN training

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

- Alternate between
  - *Gradient ascent* on discriminator:

$$D^* = \arg\max_D V(G,D)$$

  - *Gradient descent* on generator (minimize log-probability of discriminator being right):

$$G^* = \arg\min_G V(G,D)$$
$$= \arg\min_G \mathbb{E}_{z\sim p} \log(1 - D(G(z)))$$

  - In practice, do *gradient ascent* on generator (maximize log-probability of discriminator being wrong):

$$G^* = \arg\max_G \mathbb{E}_{z\sim p} \log(D(G(z)))$$

Non-saturating GAN loss (NSGAN)

$\min_{w_G} \mathbb{E}_{z\sim p} \log(1 - D(G(z)))$ vs. $\max_{w_G} \mathbb{E}_{z\sim p} \log(D(G(z)))$

Minimize log-probability of discriminator being right

Maximize log-probability of discriminator being wrong

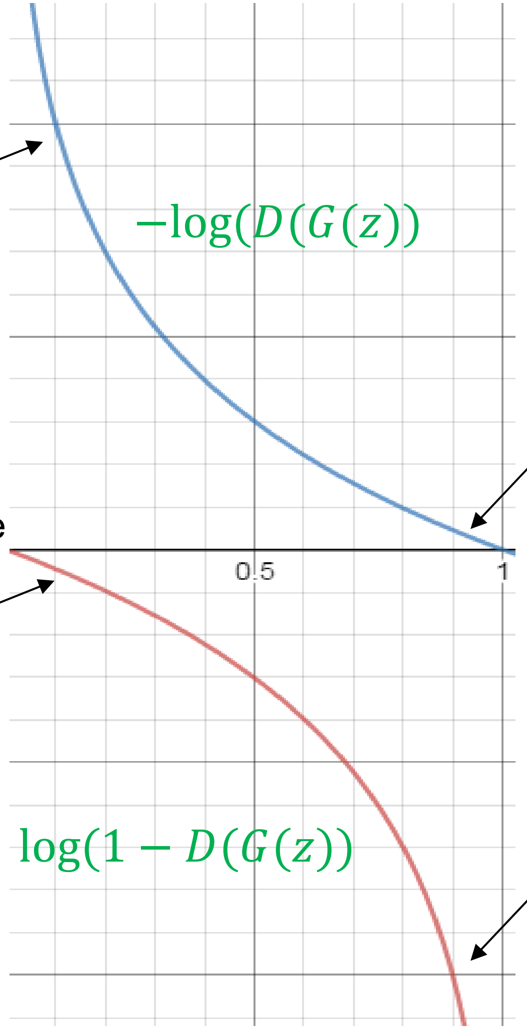
Non-saturating GAN loss (NSGAN)

$\min_{w_G} \mathbb{E}_{z\sim p} \log(1 - D(G(z)))$ vs. $\max_{w_G} \mathbb{E}_{z\sim p} \log(D(G(z)))$

Large gradients for
low-quality samples

Small gradients for
high-quality samples

Low discriminator score
(low-quality samples)



High discriminator score
(high-quality samples)

Want to learn
from confidently
rejected sample
but gradients
here are small

These samples
already fool the
discriminator so we
don't need large
gradients here

Figure source

NSGAN training algorithm

- Update discriminator:
  - Repeat for k steps:
    - Sample mini-batch of noise samples $z_1, \dots, z_m$ and mini-batch of real samples $x_1, \dots, x_m$
    - Update parameters of D by stochastic gradient ascent on

$$\frac{1}{m}\sum_m [\log D(x_m) + \log(1 - D(G(z_m)))]$$

- Update generator:
  - Sample mini-batch of noise samples $z_1, \dots, z_m$
  - Update parameters of G by stochastic gradient ascent on

$$\frac{1}{m}\sum_m \log D(G(z_m))$$

- Repeat until happy with results

NSGAN training algorithm

- Update discriminator:
  - Repeat for $\boxed{k \text{ steps}}$:
    - Sample mini-batch of noise samples $z_1, \ldots, z_m$ and mini-batch of real samples $x_1, \ldots, x_m$
    - Update parameters of D by stochastic gradient ascent on

$$\frac{1}{m}\sum_m [\log D(x_m) + \log(1 - D(G(z_m)))]$$

- Update generator:
  - Sample mini-batch of noise samples $z_1, \ldots, z_m$
  - Update parameters of G by stochastic gradient ascent on

$$\frac{1}{m}\sum_m \log D(G(z_m))$$

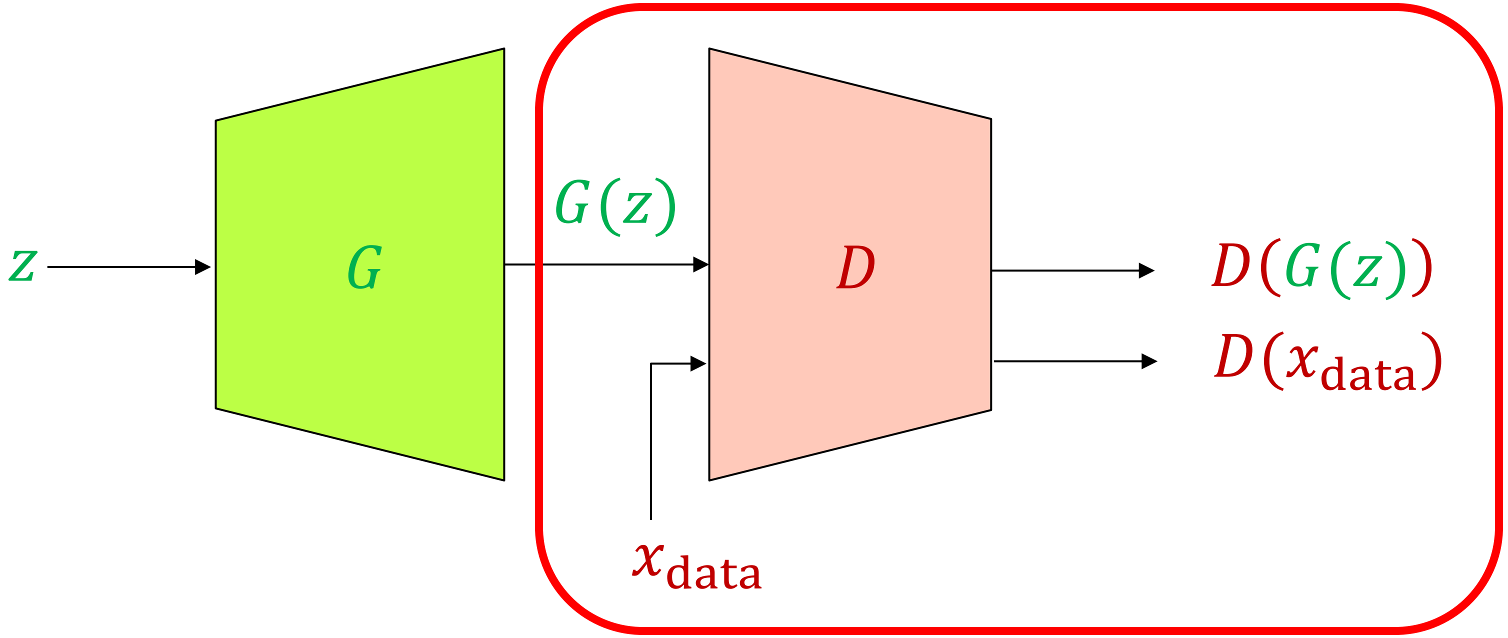
- Repeat until happy with results

Some find k=1 more stable, others use k > 1, no best rule.

Recent work (e.g. Wasserstein GAN) alleviates this problem, better stability!

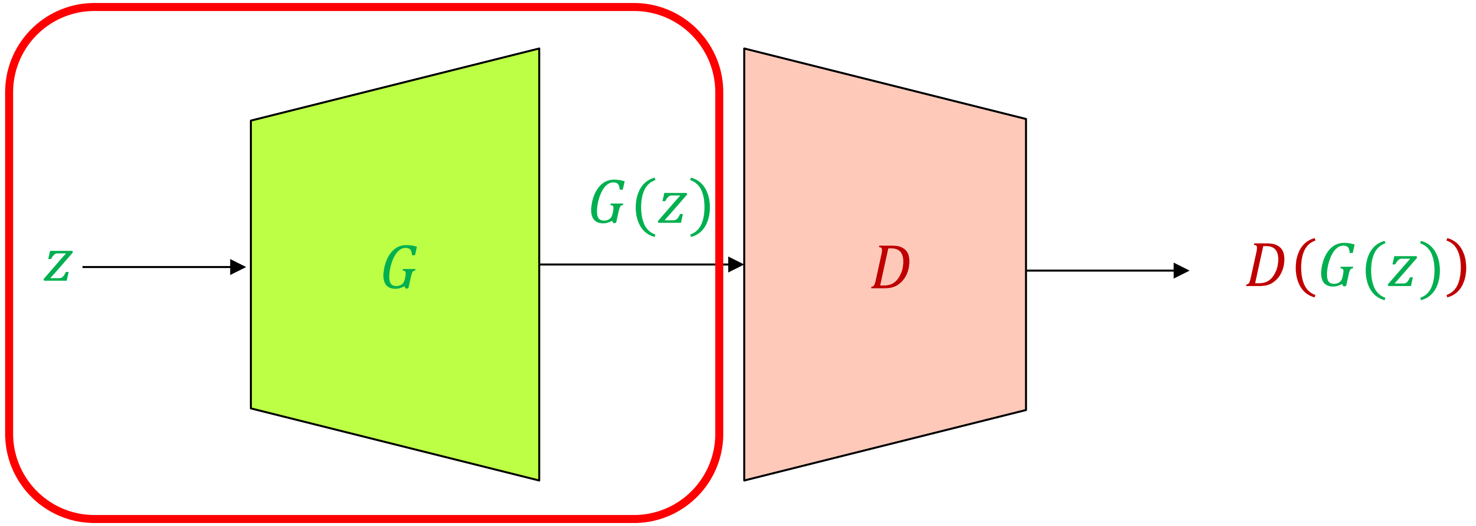
GAN: Conceptual picture

- Update discriminator: push $D(x_{\text{data}})$ close to 1 and $D(G(z))$ close to 0
  - The generator is a "black box" to the discriminator



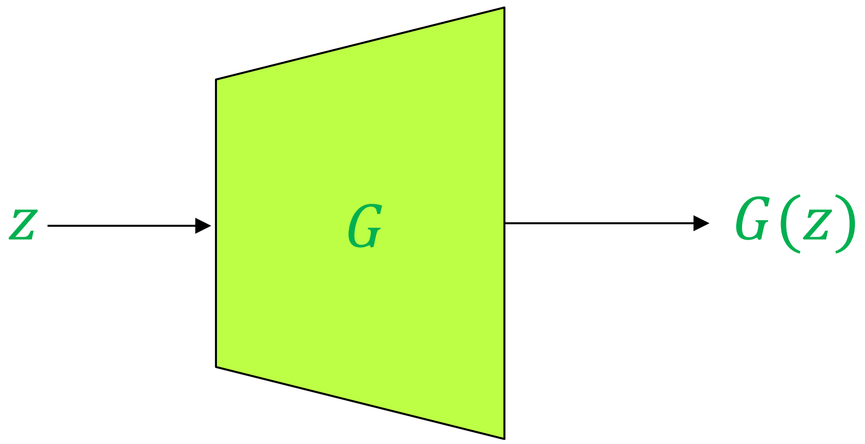
GAN: Conceptual picture

- Update generator: increase $D(G(z))$
  - Requires back-propagating through the composed generator-discriminator network (i.e., the discriminator cannot be a black box)
  - The generator is exposed to real data only via the output of the discriminator (and its gradients)

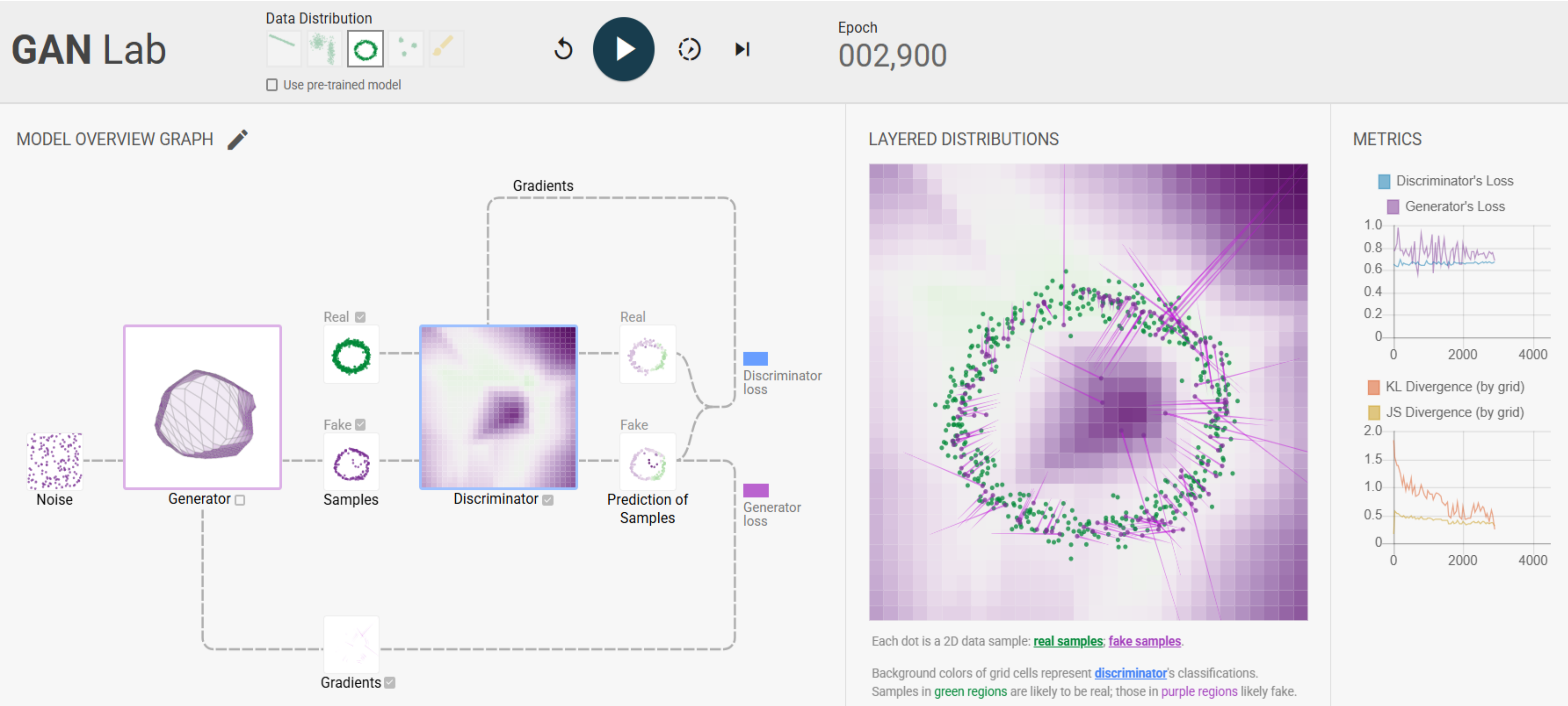


GAN: Conceptual picture

- Test time – the discriminator is discarded



GAN Demo



https://poloclub.github.io/ganlab/

Original GAN results

MNIST digits

Toronto Face Dataset

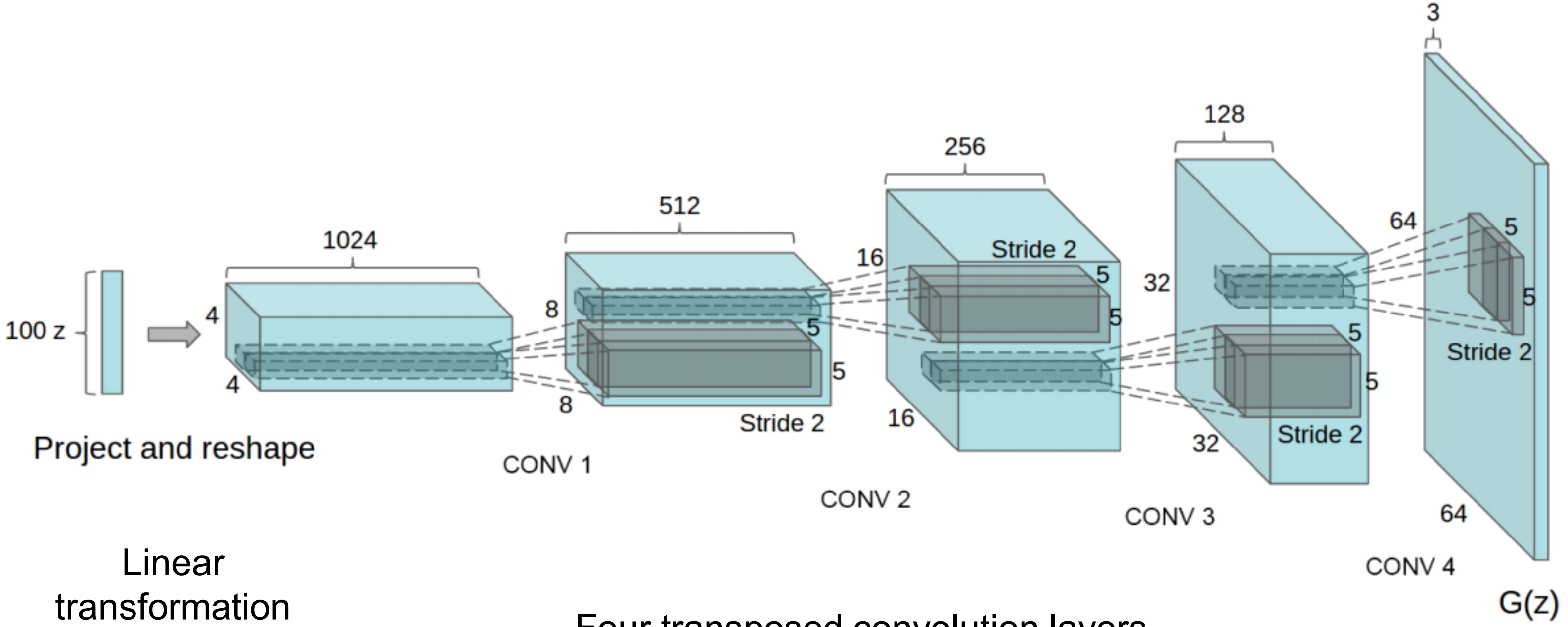
Nearest real image for
sample to the left

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, [Generative adversarial nets](), NIPS 2014

DCGAN

- Early, influential convolutional architecture for generator

Uniformly distributed input



Linear transformation

Four transposed convolution layers with ReLU activations

Tanh activations in the last layer

A. Radford, L. Metz, S. Chintala, [Unsupervised representation learning with deep convolutional generative adversarial networks](), ICLR 2016

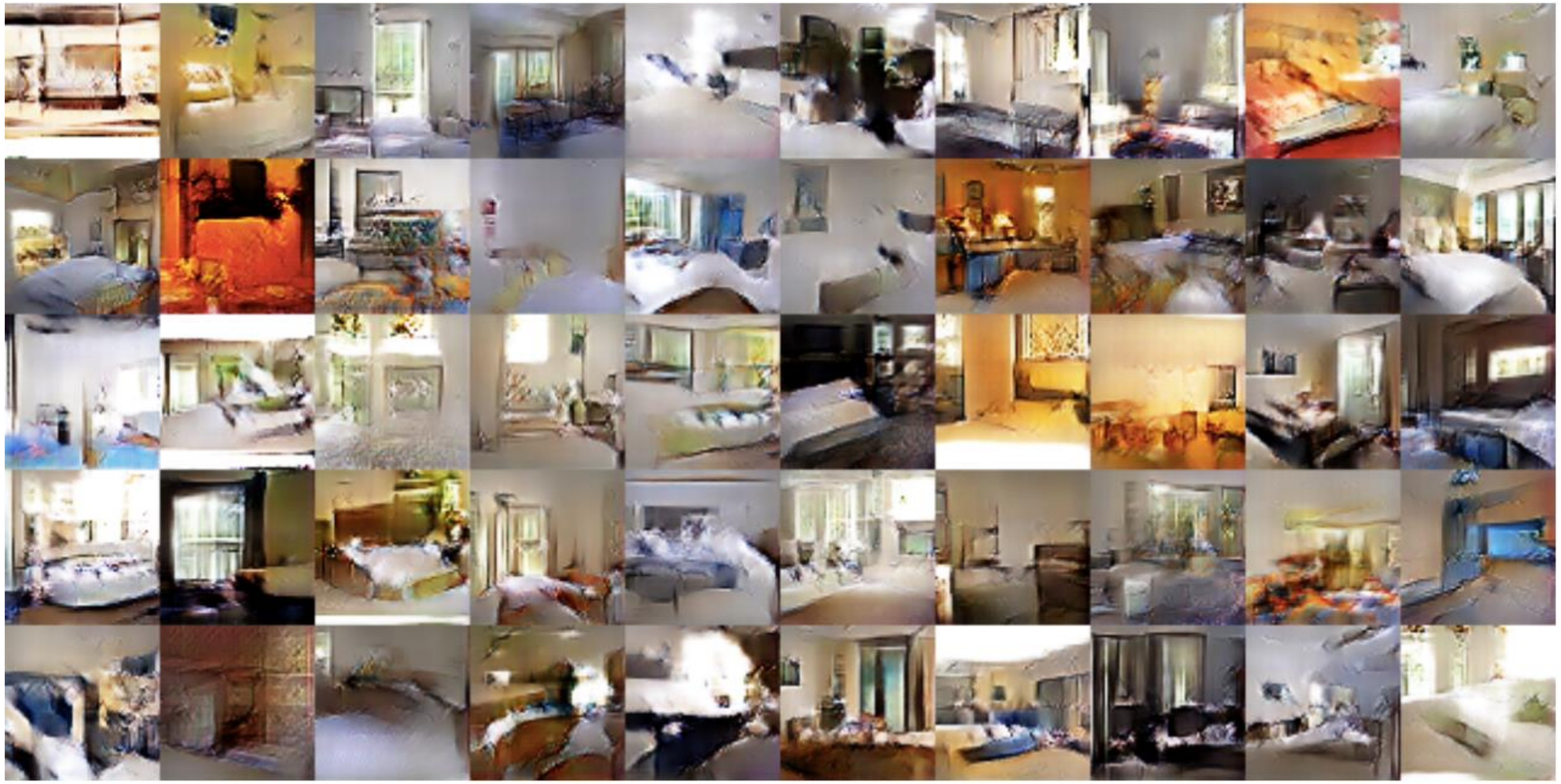
DCGAN

- Early, influential convolutional architecture for generator
- Discriminator architecture:
  - Don’t use pooling, only strided convolutions
  - Use Leaky ReLU activations (sparse gradients cause problems for training)
  - Use only one FC layer before the softmax output
  - Use batch normalization after most layers (in the generator also)

A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, ICLR 2016

DCGAN results

Generated bedrooms after one epoch

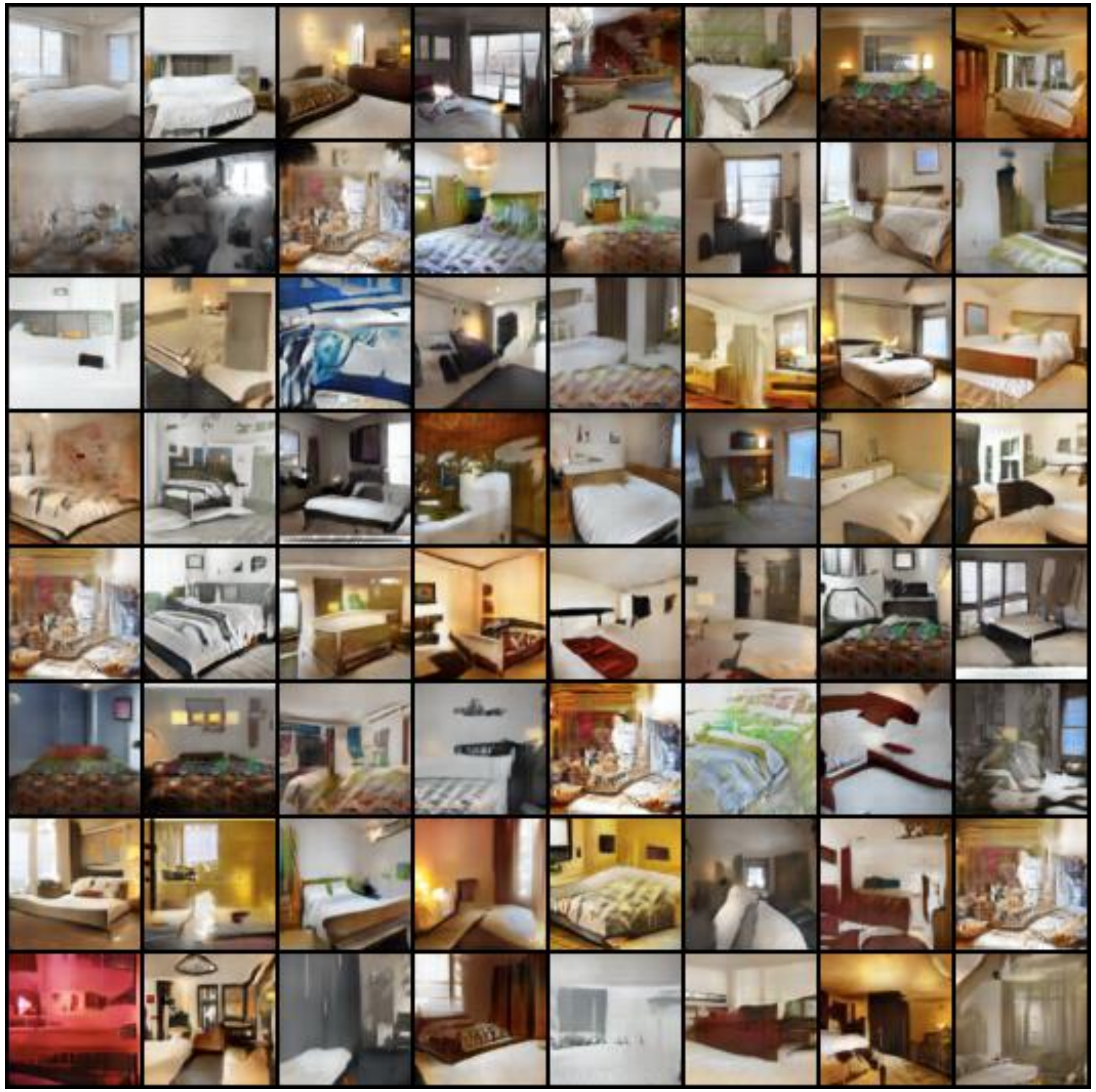


DCGAN results

Generated bedrooms after five epochs

DCGAN results

More bedrooms



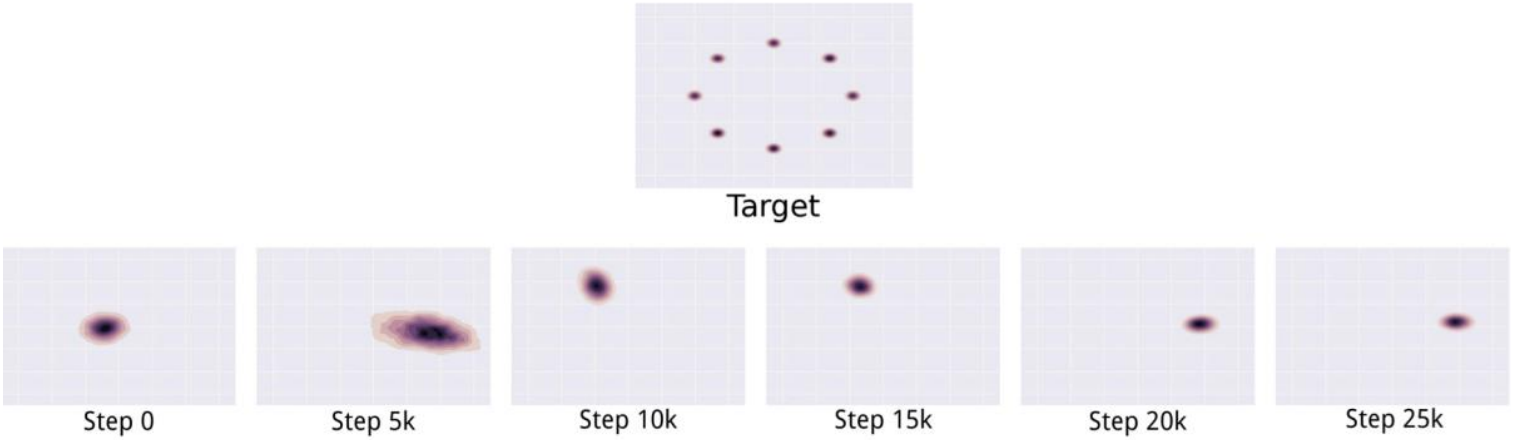
Source: F. Fleuret

Problems with GAN training

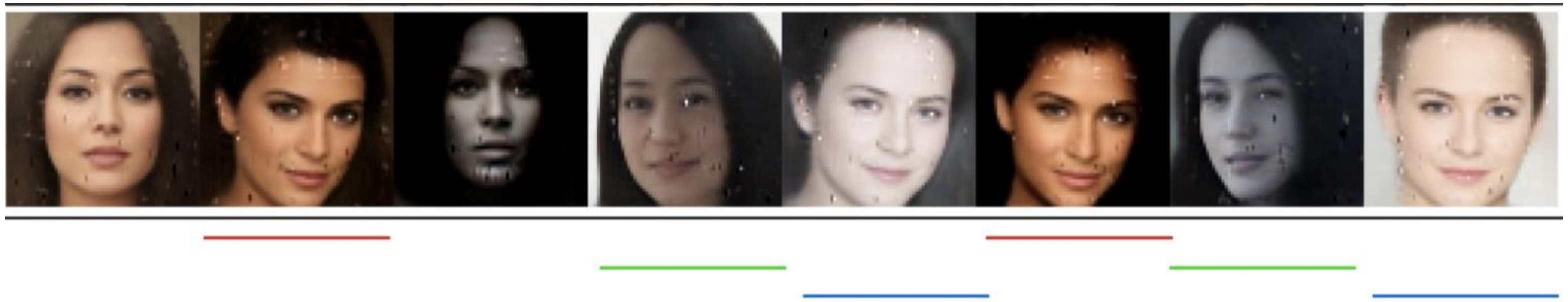
- Stability
  - Parameters can oscillate or diverge, generator loss does not correlate with sample quality
  - Behavior very sensitive to hyperparameter selection

Problems with GAN training

- Mode collapse
  - Generator ends up modeling only a small subset of the training data



Source



Source

Some popular GAN flavors

- WGAN and improved WGAN (WGAN-GP)
- LSGAN

Wasserstein GAN (WGAN)

- Motivated by *Wasserstein or Earth mover's distance*, which is an alternative to JS divergence for comparing distributions
  - In practice, use linear activation instead of sigmoid in the discriminator and drop the logs from the objective:

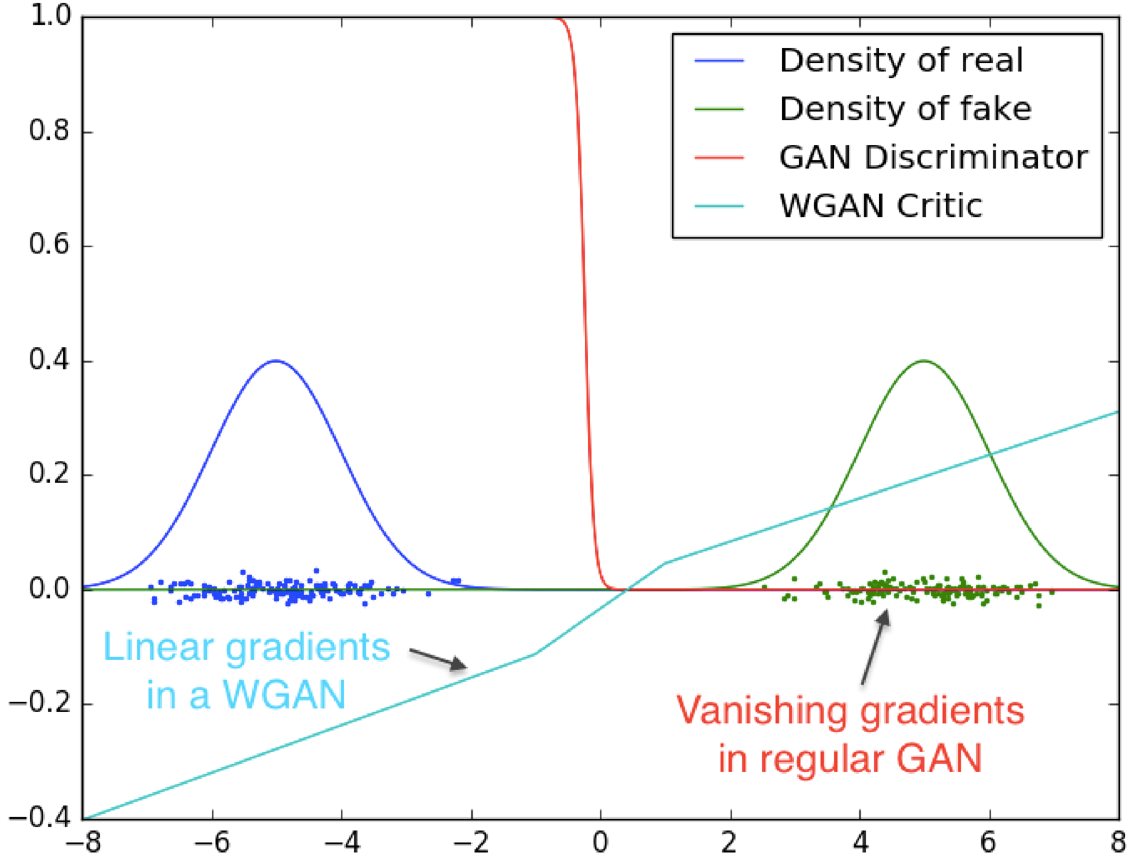
$$\min_G \max_D \left[\mathbb{E}_{x\sim p_{\text{data}}} D(x) - \mathbb{E}_{z\sim p} D(G(z))\right]$$

  - Due to theoretical considerations, important to ensure smoothness of discriminator
  - This paper's suggested method is clipping weights to fixed range $[-c, c]$

M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, ICML 2017

Wasserstein GAN (WGAN)

- Benefits (claimed)
  - Better gradients, more stable training

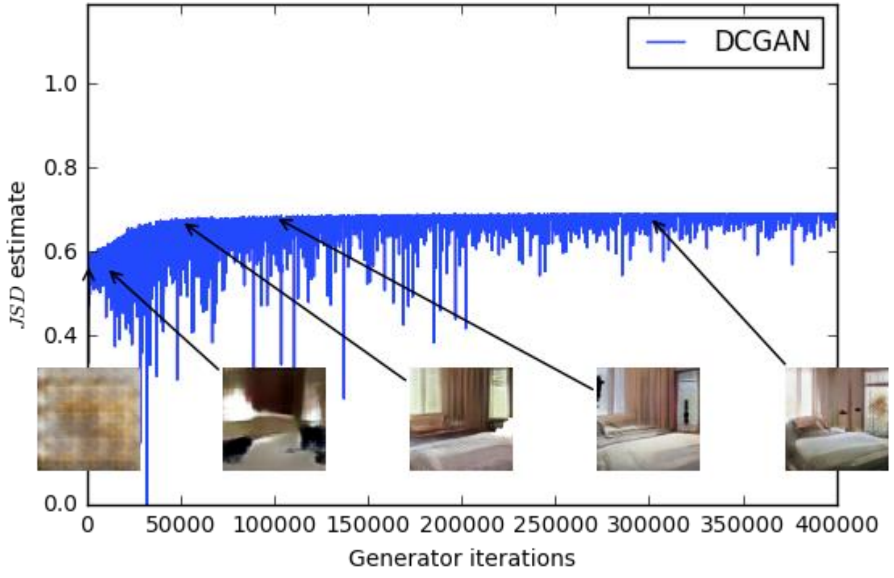


M. Arjovsky, S. Chintala, L. Bottou, [Wasserstein generative adversarial networks](), ICML 2017

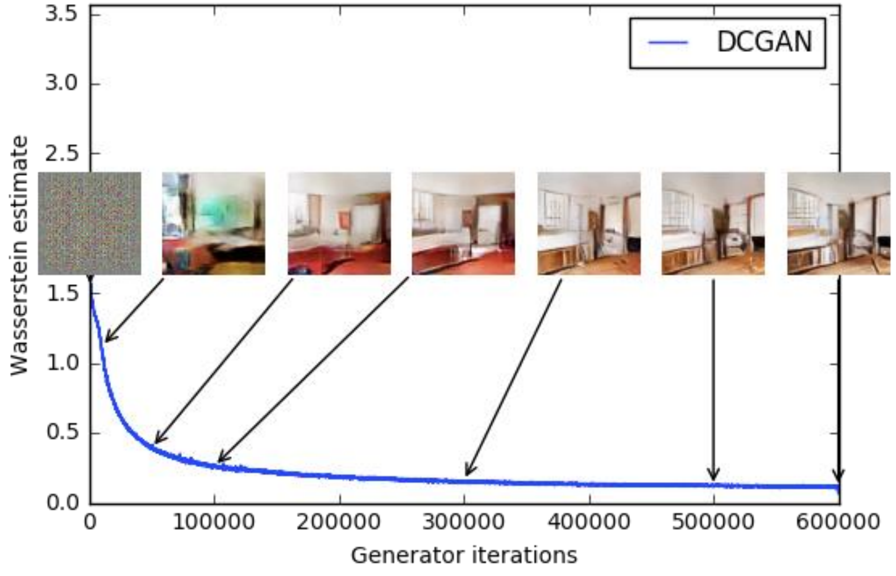
Wasserstein GAN (WGAN)

- Benefits (claimed)
  - Better gradients, more stable training
  - Objective function value is more meaningfully related to quality of generator output

Original GAN divergence



WGAN divergence



M. Arjovsky, S. Chintala, L. Bottou, [Wasserstein generative adversarial networks](), ICML 2017

Improved Wasserstein GAN (WGAN-GP)

- Weight clipping leads to problems with discriminator training
- Improved Wasserstein discriminator loss:

$$\mathbb{E}_{\tilde{x}\sim p_{\text{gen}}} D(\tilde{x}) - \mathbb{E}_{x\sim p_{\text{real}}} D(x)$$

$$+\,\lambda\, \mathbb{E}_{\hat{x}\sim p_{\hat{x}}}\left[\left(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1\right)^2\right]$$

Unit norm gradient penalty on points $\hat{x}$ obtained by interpolating real and generated samples

I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved training of Wasserstein GANs, NIPS 2017

Improved Wasserstein GAN: Results

I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, [Improved training of Wasserstein GANs](), NIPS 2017

Least Squares GAN (LSGAN)

- Use least squares cost for generator and discriminator
  - Equivalent to minimizing Pearson χ^2 divergence

$$D^* = \arg\min_D \left[\mathbb{E}_{x\sim p_{\text{data}}}(D(x)-1)^2 + \mathbb{E}_{z\sim p}(D(G(z)))^2\right]$$

Push discrim. response on real data close to 1

Push response on generated data close to 0

$$G^* = \arg\min_G \mathbb{E}_{z\sim p}\big(D\big(G(z)\big)-1\big)^2$$

Push response on generated data close to 1

X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, Least squares generative adversarial networks, ICCV 2017

Least Squares GAN (LSGAN)

- Benefits (claimed)
    - Higher-quality images

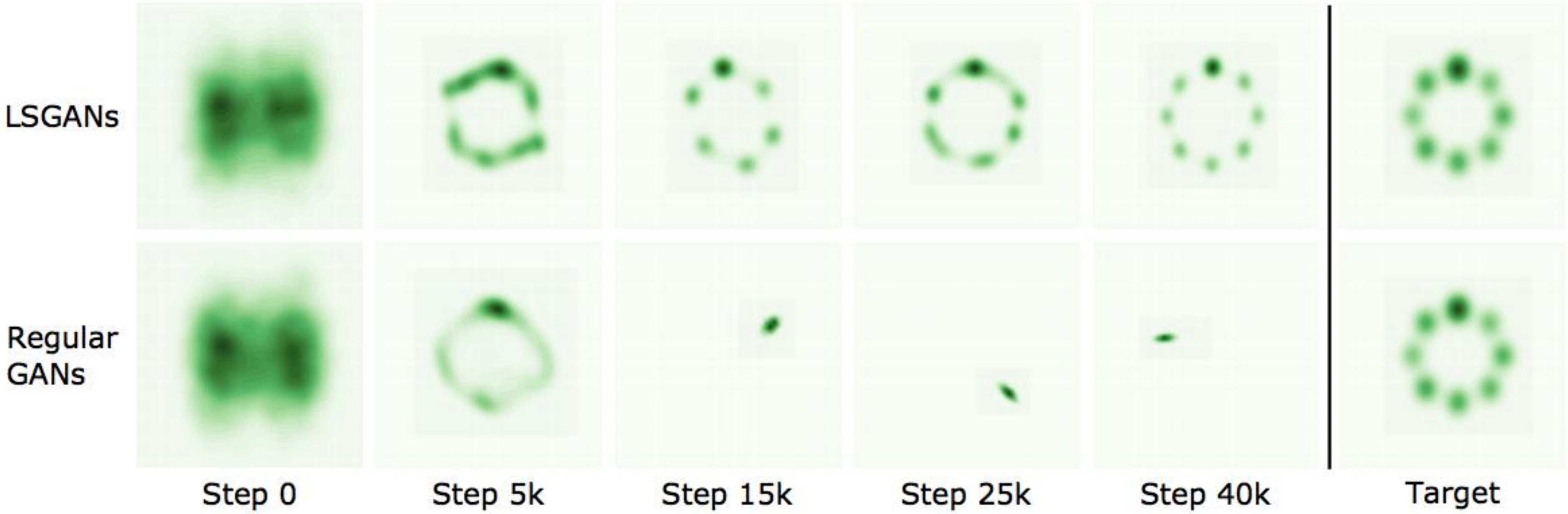
(a) Generated images (112 × 112) by LSGANs.

(b) Generated images (112 × 112) by DCGANs.

X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, [Least squares generative adversarial networks](), ICCV 2017

Least Squares GAN (LSGAN)

- Benefits (claimed)
  - Higher-quality images
  - More stable and resistant to mode collapse



X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, [Least squares generative adversarial networks](), ICCV 2017

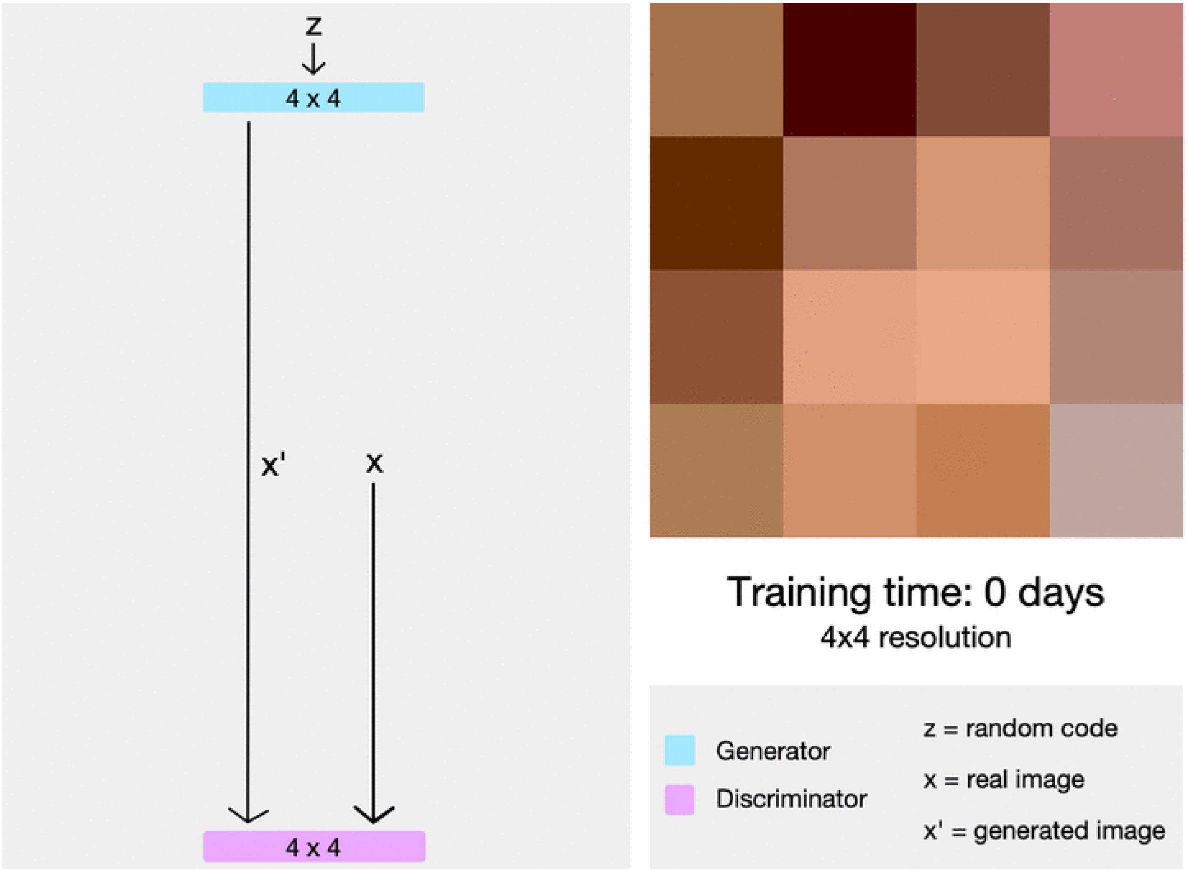
Progressive GANs

Realistic face images up to 1024 x 1024 resolution

T. Karras, T. Aila, S. Laine, J. Lehtinen. [Progressive Growing of GANs for Improved Quality, Stability, and Variation](). ICLR 2018

Progressive GANs

- Key idea: train lower-resolution models, gradually add layers corresponding to higher-resolution outputs



Source

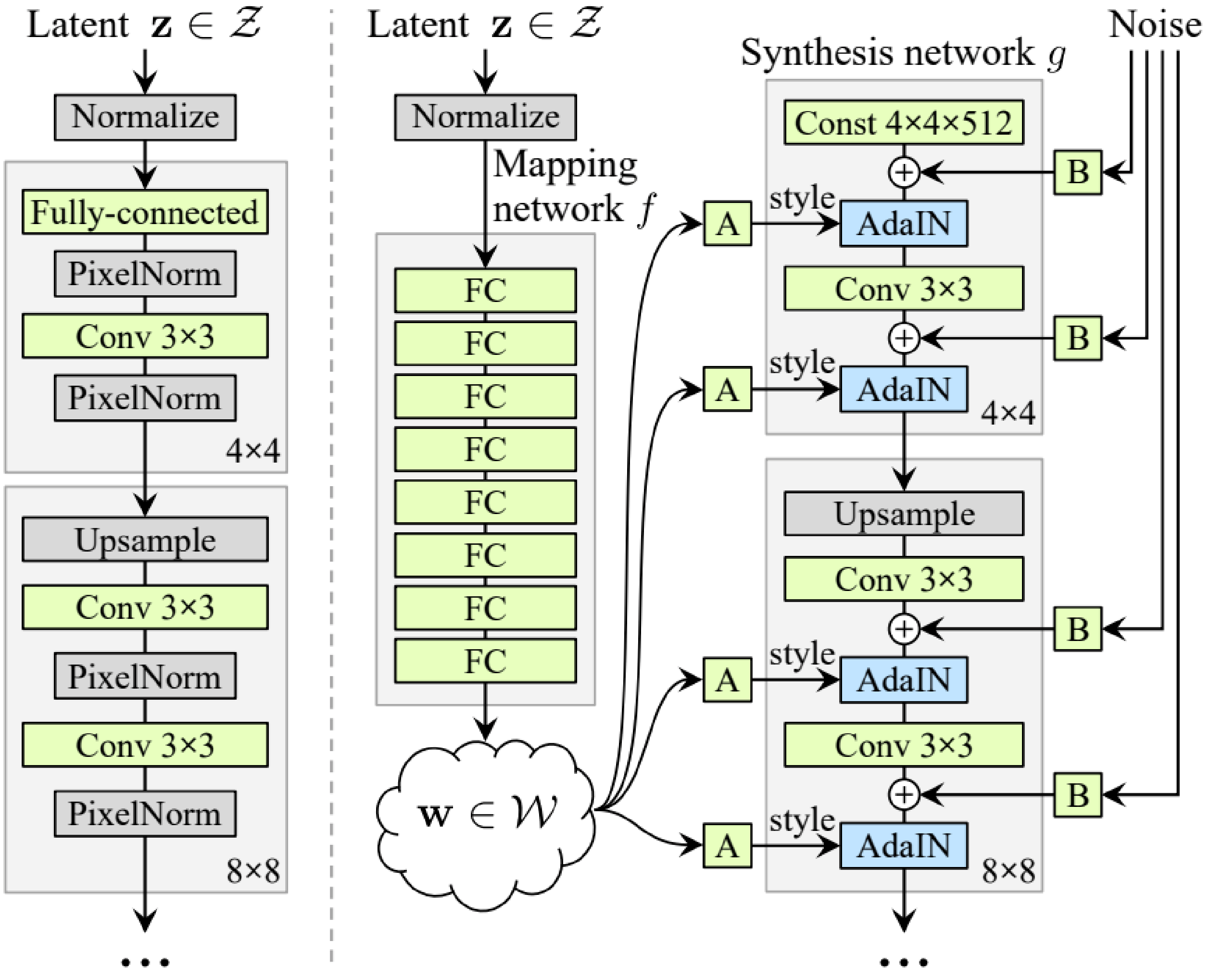
T. Karras, T. Aila, S. Laine, J. Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. ICLR 2018

Progressive GANs: Results

256 x 256 results for LSUN categories

StyleGAN

- Built on top of Progressive GAN
- Start with learned constant (instead of noise vector)
- Use a mapping network to produce a *style code* w using learned affine transformations A
- Use *adaptive instance normalization* (AdaIN): scale and bias each feature map using learned style values
- Add noise after each convolution and before nonlinearity (enables stochastic detail)



(a) Traditional

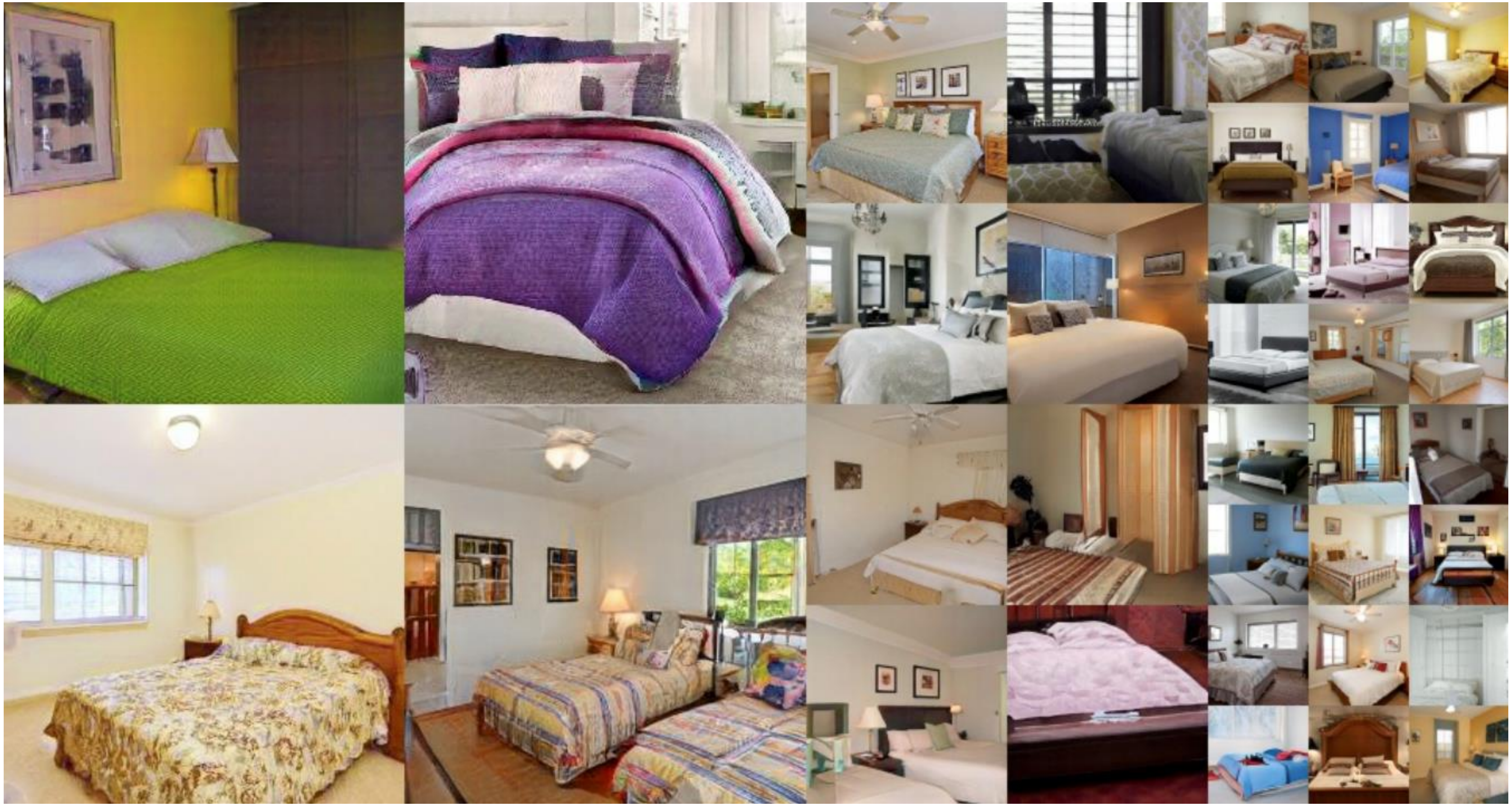
(b) Style-based generator

T. Karras, S. Laine, T. Aila. [A Style-Based Generator Architecture for Generative Adversarial Networks](). CVPR 2019

StyleGAN: Results

T. Karras, S. Laine, T. Aila. [A Style-Based Generator Architecture for Generative Adversarial Networks](). CVPR 2019

StyleGAN: Bedrooms



T. Karras, S. Laine, T. Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. CVPR 2019

StyleGAN: Cars

T. Karras, S. Laine, T. Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. CVPR 2019

Image-to-Image Translation

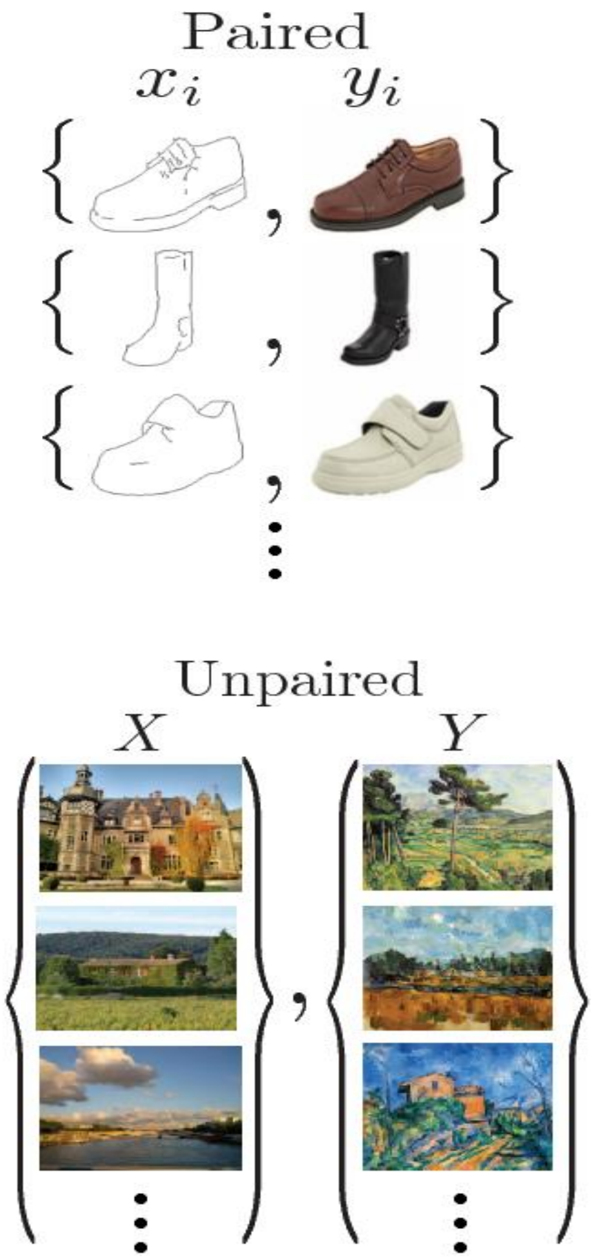
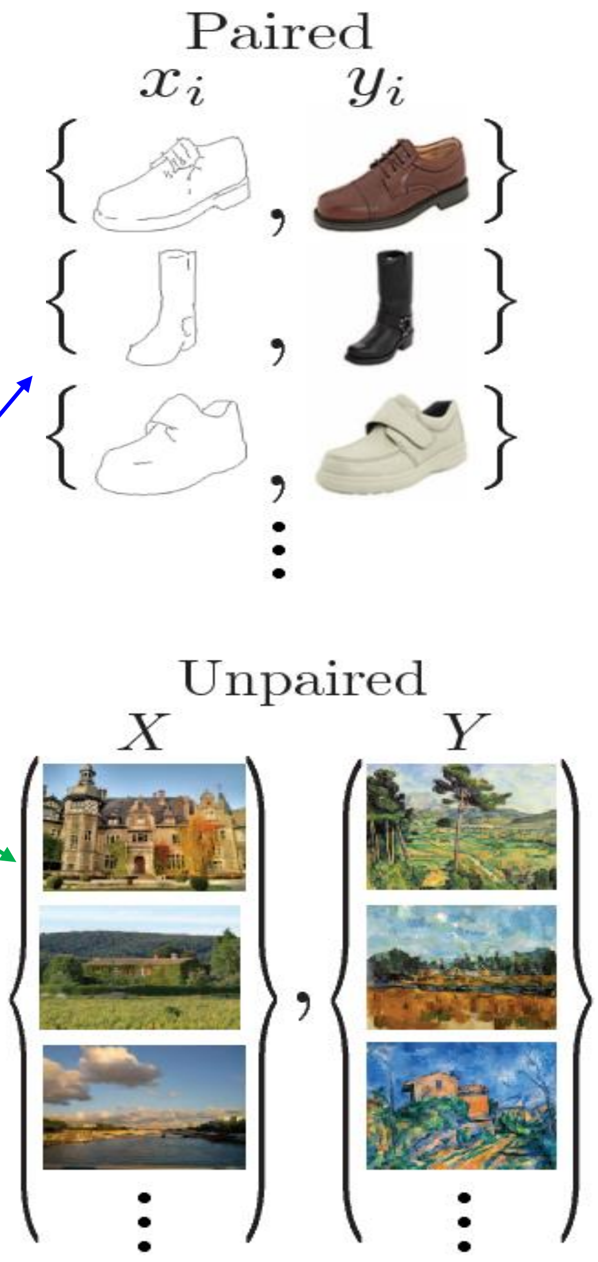


Image-to-Image Translation

- Conditional GAN (cGAN)
- Cycle-Consistent Adversarial Network (CycleGAN)
- Perceptual Cyclic-Synthesized Generative Adversarial Networks (PCSGAN)



1. Isola, Phillip, et al. Image-to-image translation with conditional adver*sarial networks. CVPR 2017.*
2. Zhu, Jun-Yan, et al. Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. *CVPR* 2017.
3. Babu, Kancharagunta Kishan, and Shiv Ram Dubey. PCSGAN: Perceptual Cyclic-Synthesized Generative Adversarial Networks for Thermal and NIR to Visible Image Transformation. *Neurocomputing,* 2020.

Image-to-Image Translation: GAN

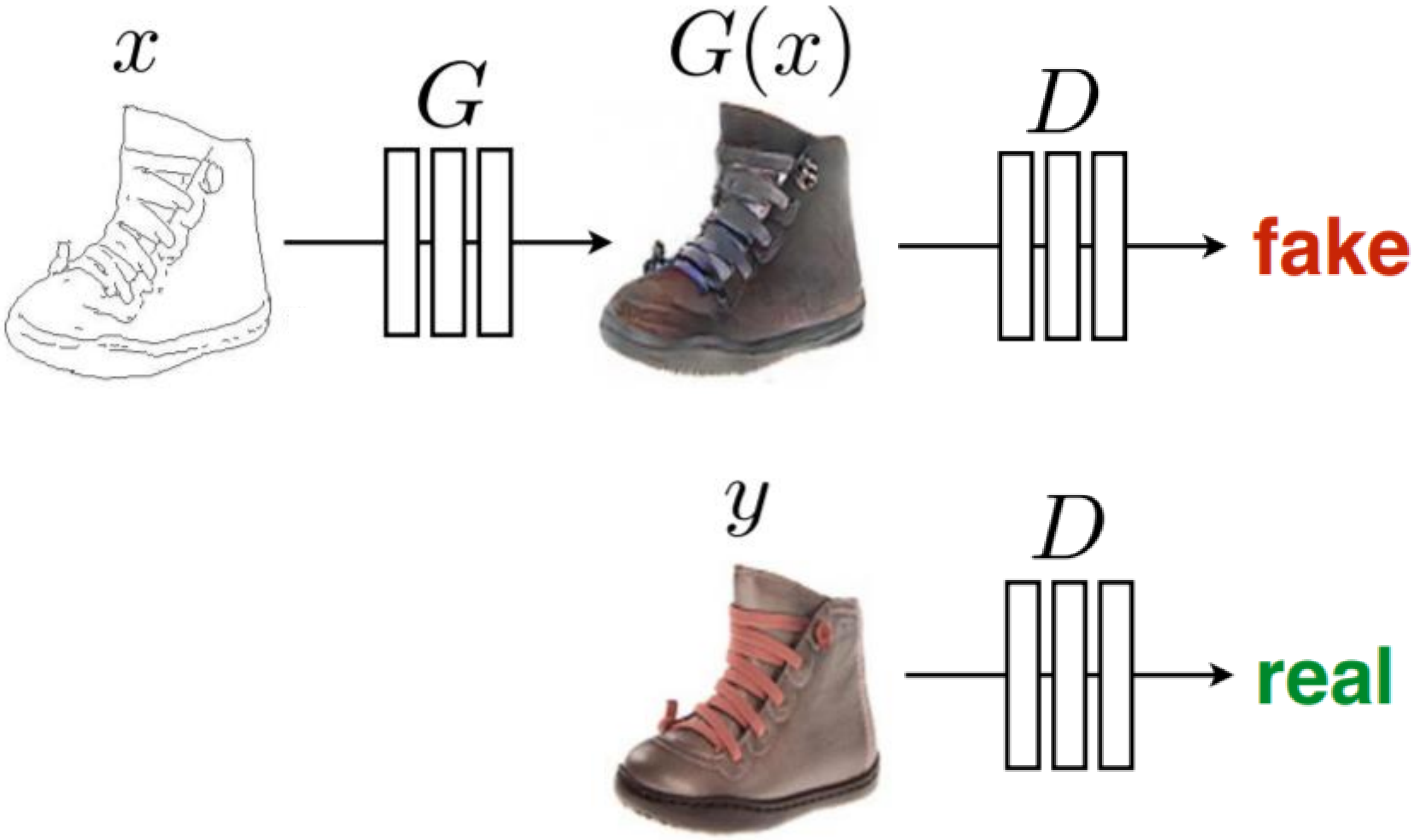
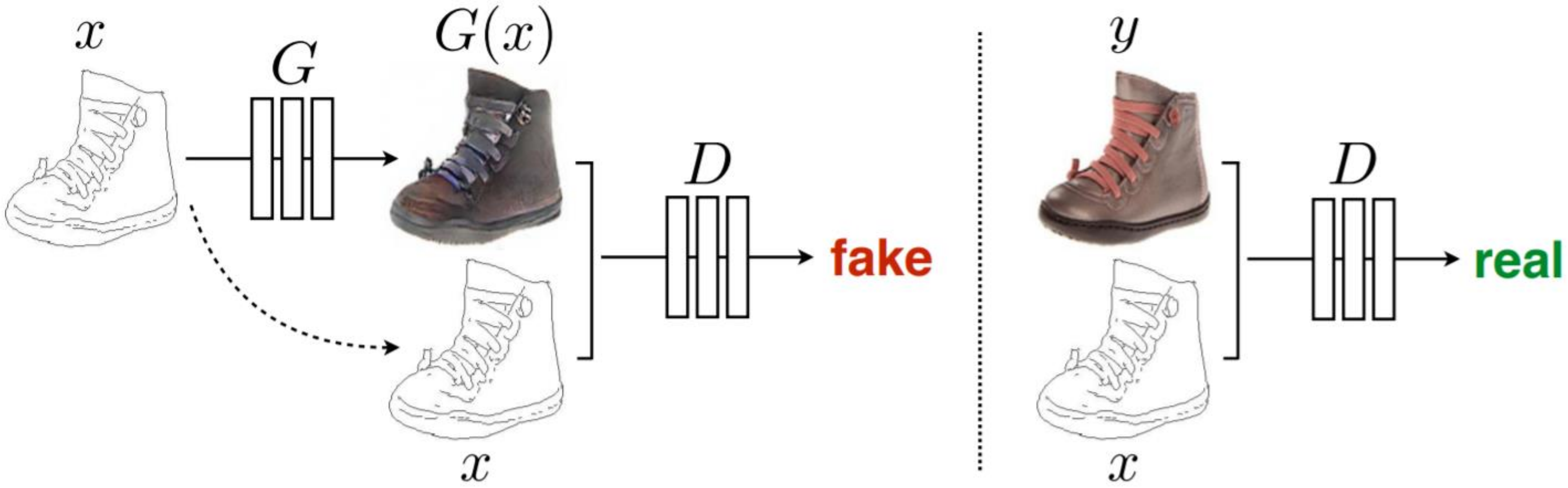
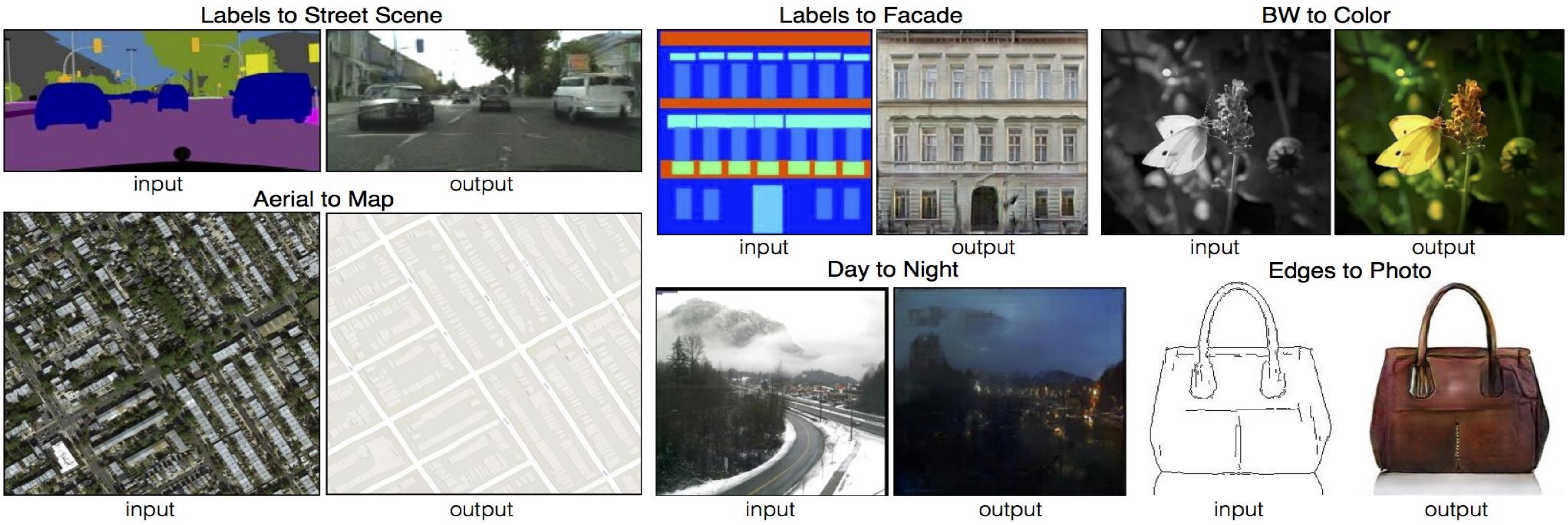


Image-to-Image Translation: Conditional GAN



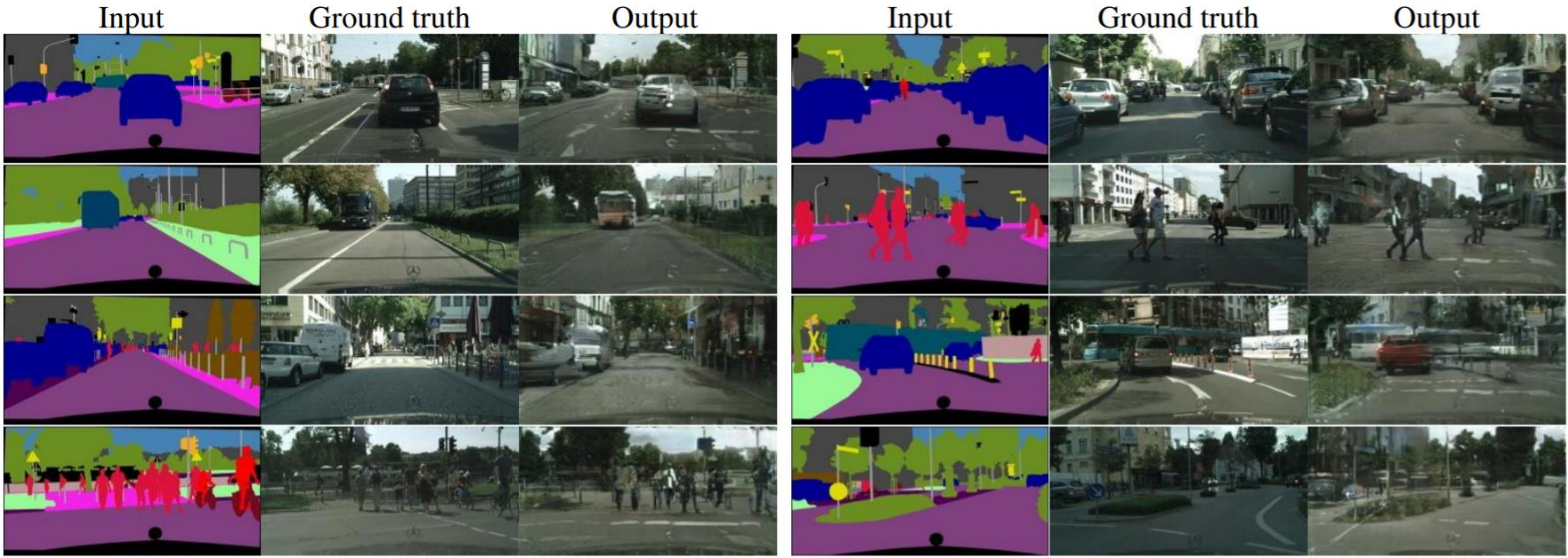
Isola, Phillip, et al. Image-to-image translation with conditional adver*sarial networks. CVPR 2017.*

Image-to-Image Translation: Conditional GAN



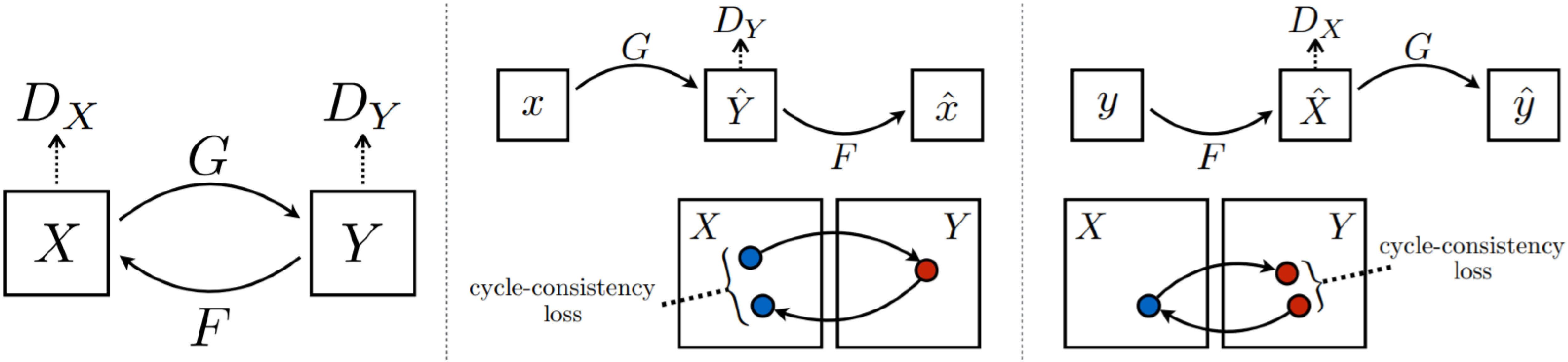
Isola, Phillip, et al. Image-to-image translation with conditional adver*sarial networks. CVPR 2017.*

Image-to-Image Translation: Conditional GAN



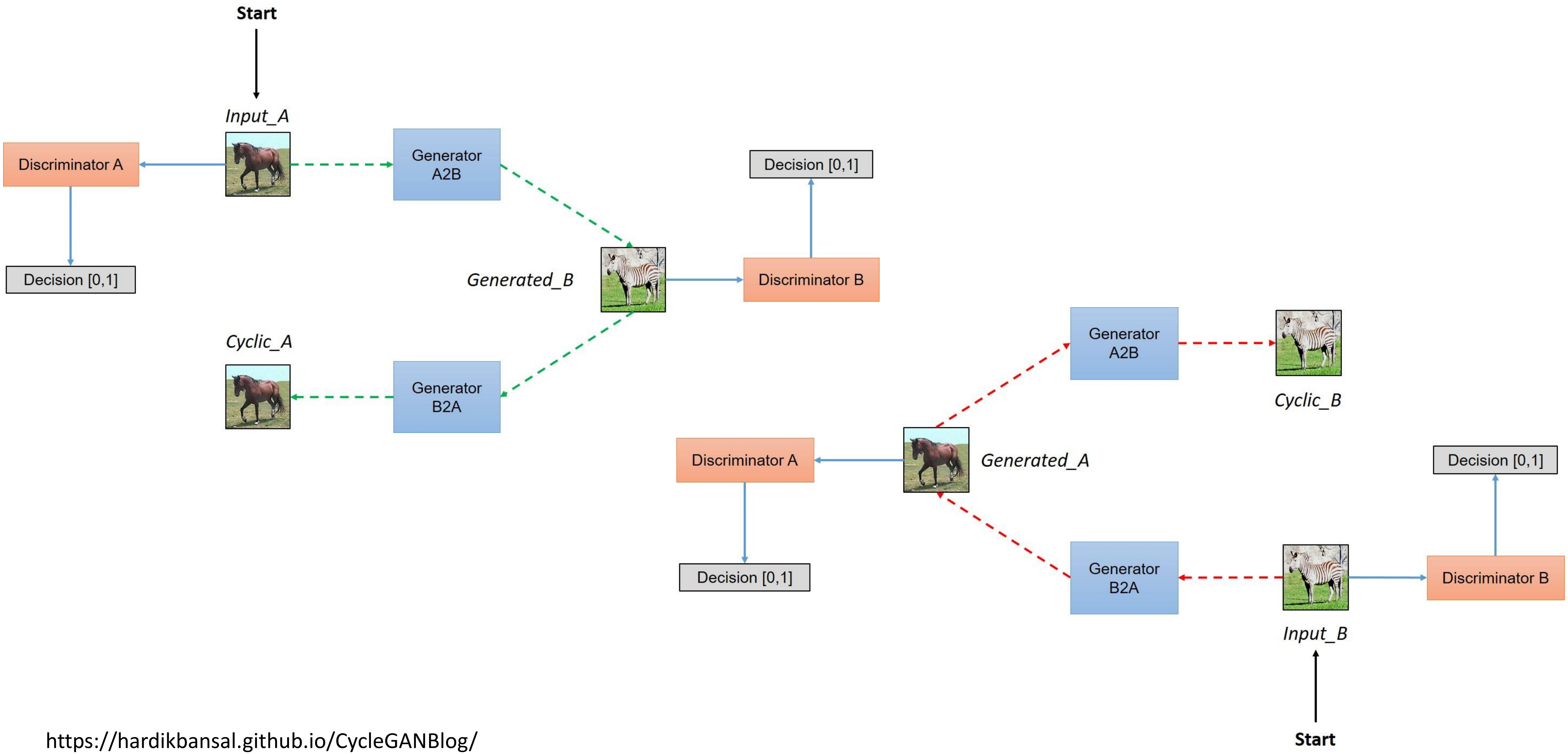
Isola, Phillip, et al. Image-to-image translation with conditional adver*sarial networks. CVPR 2017.*

Image-to-Image Translation: Cycle GAN



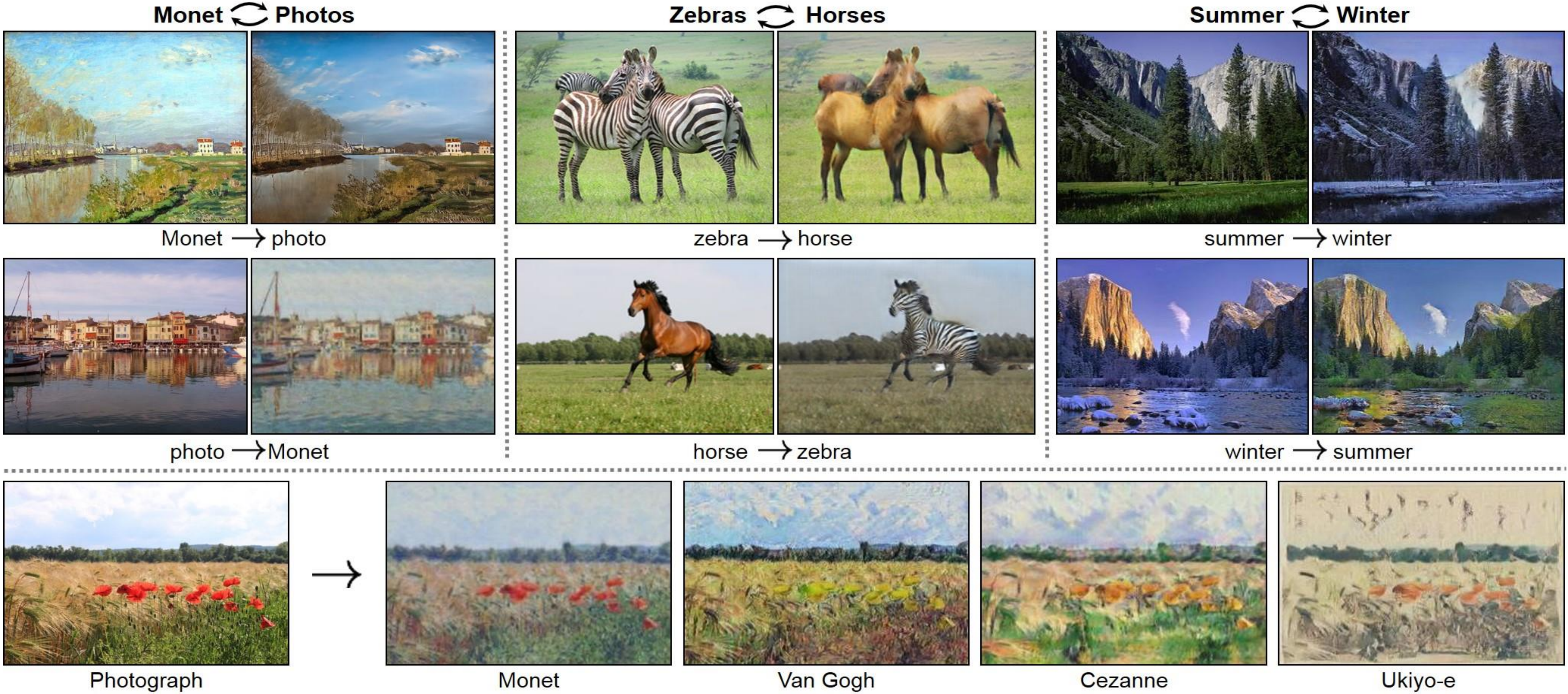
Zhu, Jun-Yan, et al. Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. *CVPR* 2017.

Image-to-Image Translation: Cycle GAN



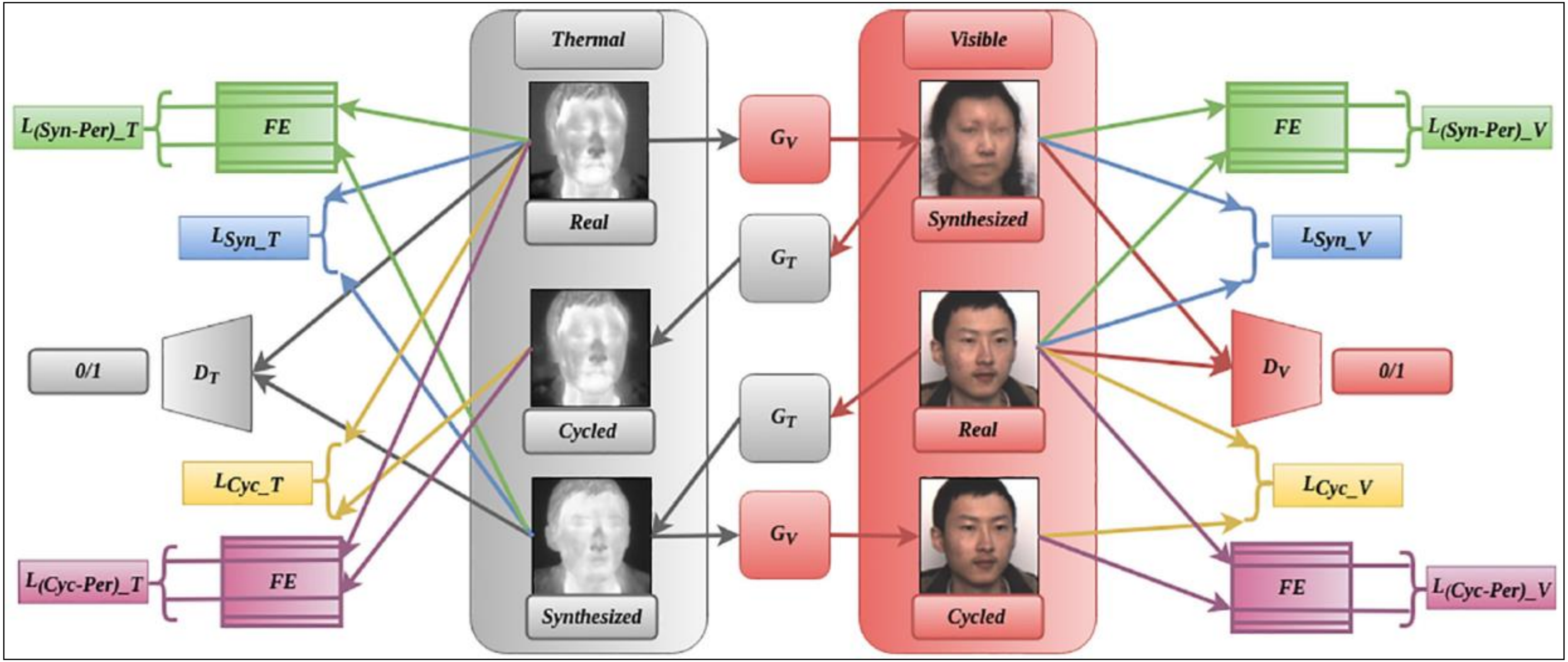
https://hardikbansal.github.io/CycleGANBlog/

Image-to-Image Translation: Cycle GAN



Zhu, Jun-Yan, et al. Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. *CVPR* 2017.

Image-to-Image Translation: PCSGAN



K.K. Babu and S.R. Dubey. PCSGAN: Perceptual Cyclic-Synthesized Generative Adversarial Networks for Thermal and NIR to Visible Image Transformation. *Neurocomputing*, 2020.

Image-to-Image Translation: PCSGAN

1st Column – Input Image
2nd Column – Pix2Pix
3rd Column – DualGAN
4th Column – CycleGAN
5th Column – PCSGAN
6th Column – Target Image

K.K. Babu and S.R. Dubey. PCSGAN: Perceptual Cyclic-Synthesized Generative Adversarial Networks for Thermal and NIR to Visible Image Transformation. *Neurocomputing*, 2020.

Image-to-Image Translation: PCSGAN

Methods	Metrics				
	SSIM	MSE	PSNR	LPIPS	MSSIM
Pix2pix	0.7555	74.6082	29.4587	0.089	0.7624
DualGAN	0.7638	75.4379	29.4201	0.099	0.7989
CycleGAN	0.7648	76.1482	29.351	0.088	0.7687
PS2GAN	0.8087	67.869	29.9676	0.064	0.8242
PAN	0.8125	69.0331	29.84	0.069	0.8281
PCSGAN (Ours)	**0.8275**	**64.6442**	**30.1686**	**0.059**	**0.8411**

Results comparison over the WHU-IIP face dataset.

SSIM - Structural Similarity Index Measure

MSE - Mean Square Error

PSNR - Peak Signal Noise to Ratio

LPIPS - Learned Perceptual Image Patch Similarity

MSSIM - Multi-scale SSIM

K.K. Babu and S.R. Dubey. PCSGAN: Perceptual Cyclic-Synthesized Generative Adversarial Networks for Thermal and NIR to Visible Image Transformation. *Neurocomputing,* 2020.

GAN: Other Applications: Generate Cartoon Characters

Generate Cartoon Characters

Example of GAN-Generated Anime Character Faces.
Taken from Towards the Automatic Anime Characters Creation with Generative Adversarial Networks, 2017.

https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/

GAN: Other Applications: Text-to-Image Translation

Text-to-Image Translation (text2image)

Example of Textual Descriptions and GAN-Generated Photographs of Birds
Taken from StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks, 2016.

https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/

GAN: Other Applications: Face Frontal View Generation

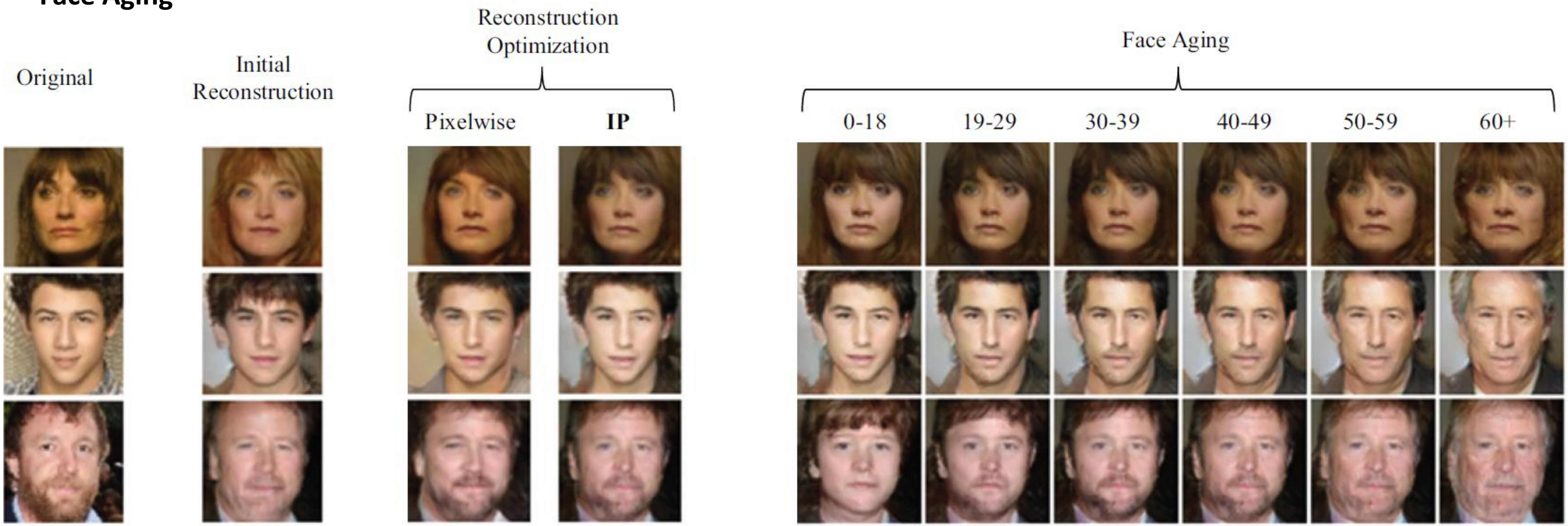
Face Frontal View Generation

Example of GAN-based Face Frontal View Photo Generation
Taken from Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis, 2017.

https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/

GAN: Other Applications: Face Aging

Face Aging



Example of Photographs of Faces Generated With a GAN With Different Apparent Ages.
Taken from Face Aging With Conditional Generative Adversarial Networks, 2017.

https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/

GAN: Other Applications: De-raining

De-raining

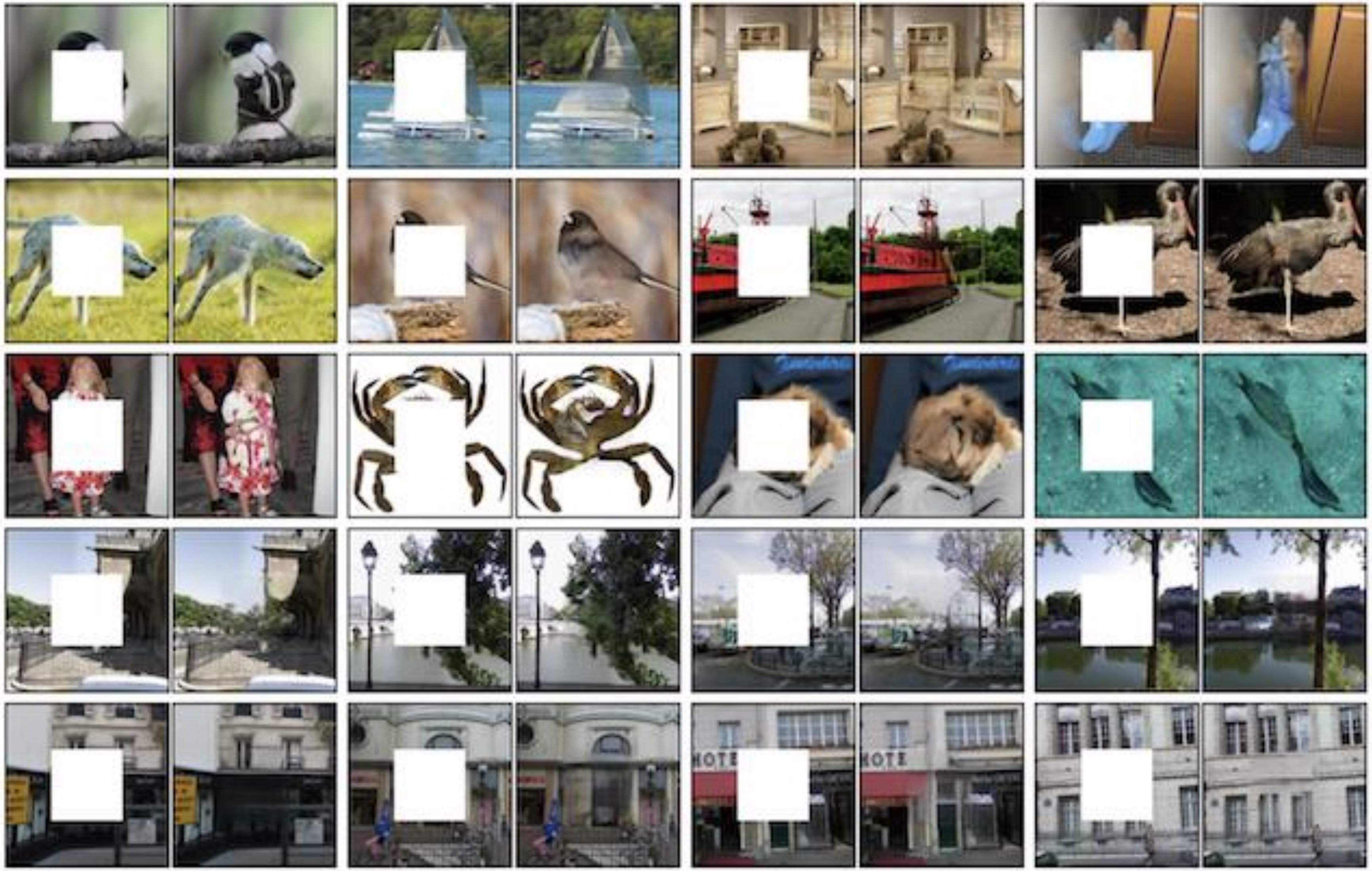
Example of Using a GAN to Remove Rain From Photographs
Taken from Image De-raining Using a Conditional Generative Adversarial Network

(a) (b) (c) (d)

https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/

GAN: Other Applications: Photo Inpainting

Photo Inpainting



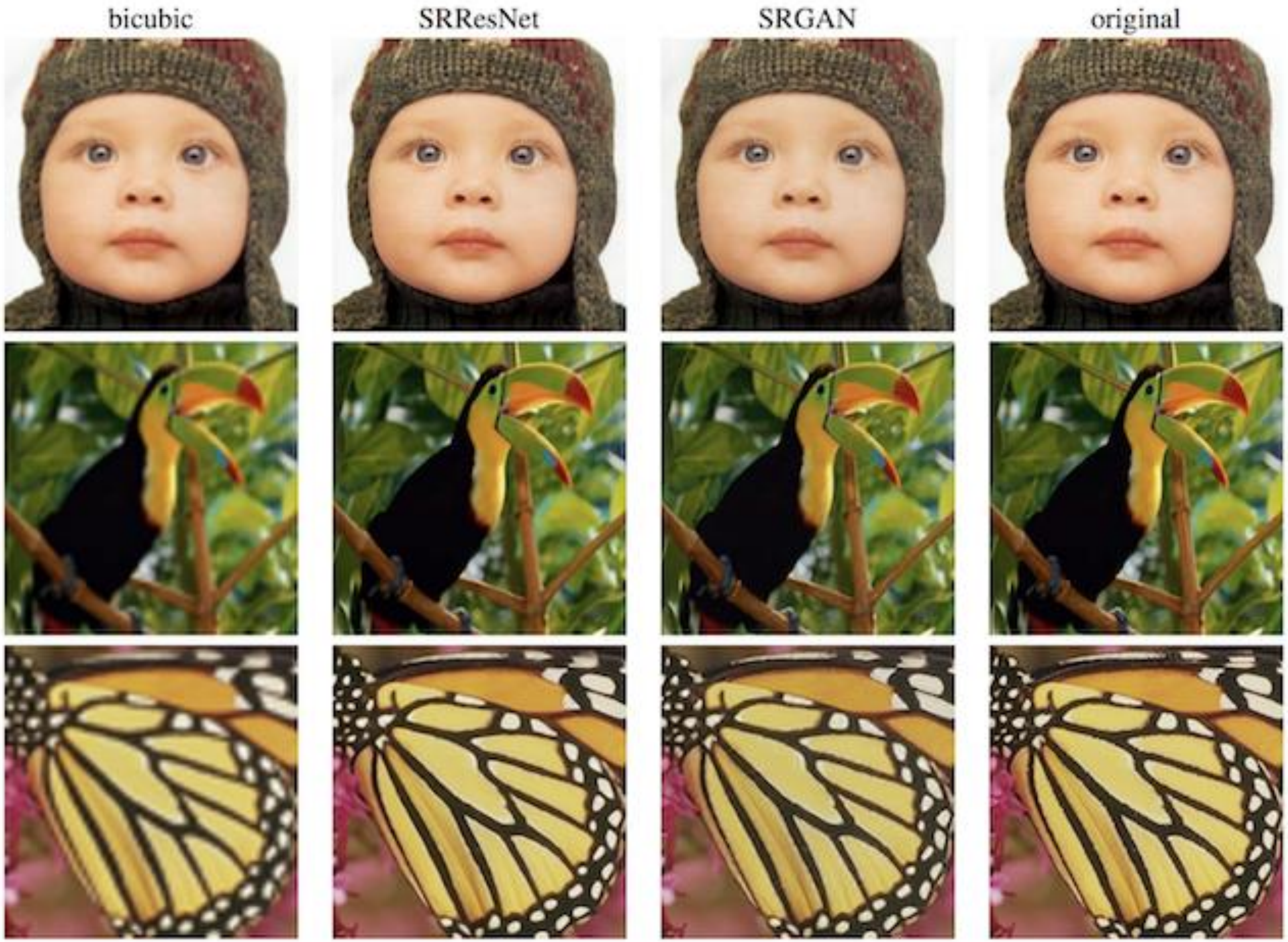
Example of GAN-Generated Photograph Inpainting Using Context Encoders.
Taken from Context Encoders: Feature Learning by Inpainting describe the use of GANs, specifically Context Encoders, 2016.

https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/

GAN: Other Applications: Super Resolution

Super Resolution

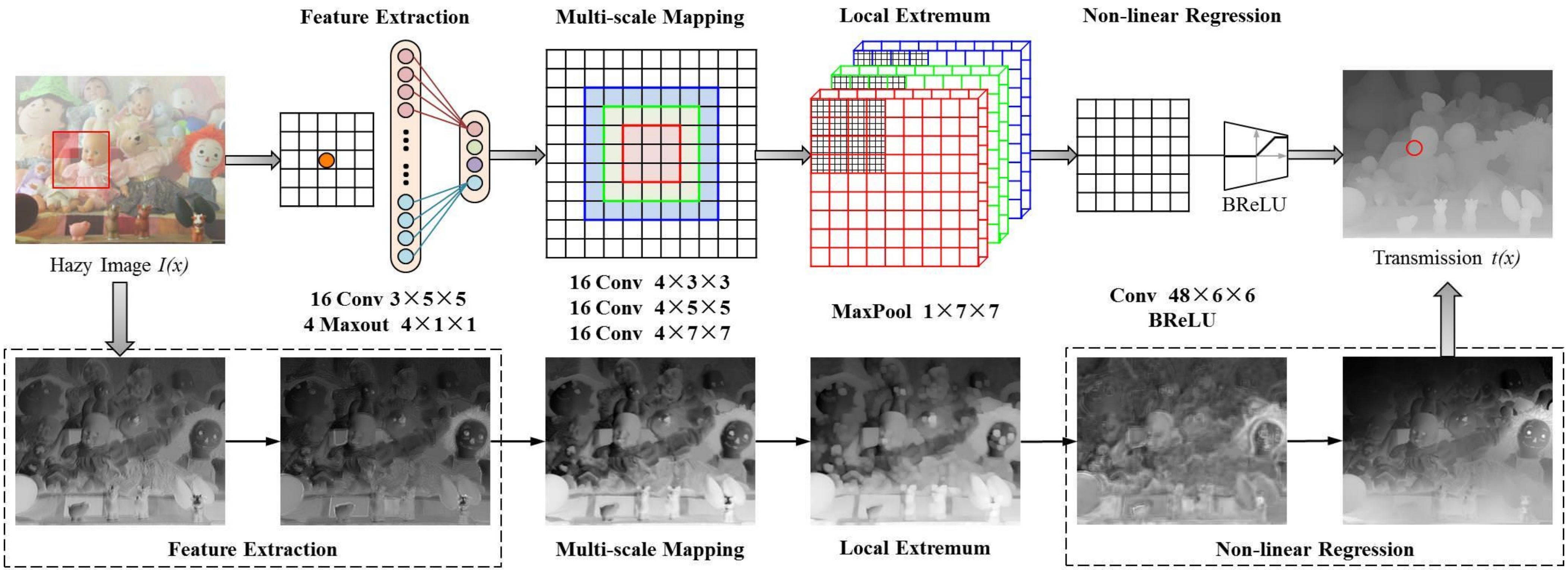
Example of GAN-Generated Images With Super Resolution.
Taken from Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, 2016.



https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/

GAN: Other Applications: Dehazing

DehazeNet: An End-to-End System for Single Image Haze Removal



https://caibolun.github.io/DehazeNet/

"The GAN Zoo"

- 3D-ED-GAN - Shape Inpainting using 3D Generative Adversarial Network and Recurrent Convolutional Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling (github)
- 3D-IWGAN - Improved Adversarial Systems for 3D Object Generation and Reconstruction (github)
- 3D-RecGAN - 3D Object Reconstruction from a Single Depth View with Adversarial Learning (github)
- ABC-GAN - ABC-GAN: Adaptive Blur and Control for improved training stability of Generative Adversarial Networks (github)
- ABC-GAN - GANs for LIFE: Generative Adversarial Networks for Likelihood Free Inference
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- ACtuAL - ACtuAL: Actor-Critic Under Adversarial Learning
- AdaGAN - AdaGAN: Boosting Generative Models
- AdvGAN - Generating adversarial examples with adversarial networks
- AE-GAN - AE-GAN: adversarial eliminating with GAN
- AEGAN - Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
- AffGAN - Amortised MAP Inference for Image Super-resolution
- AL-CGAN - Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
- ALI - Adversarially Learned Inference (github)
- AlignGAN - AlignGAN: Learning to Align Cross-Domain Images with Conditional Generative Adversarial Networks
- AM-GAN - Activation Maximization Generative Adversarial Nets
- AnoGAN - Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
- APE-GAN - APE-GAN: Adversarial Perturbation Elimination with GAN
- ARAE - Adversarially Regularized Autoencoders for Generating Discrete Structures (github)
- ARDA - Adversarial Representation Learning for Domain Adaptation
- ARIGAN - ARIGAN: Synthetic Arabidopsis Plants using Generative Adversarial Network
- ArtGAN - ArtGAN: Artwork Synthesis with Conditional Categorial GANs
- AttGAN - Arbitrary Facial Attribute Editing: Only Change What You Want
- AttnGAN - AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks
- b-GAN - Generative Adversarial Nets from a Density Ratio Estimation Perspective
- Bayesian GAN - Deep and Hierarchical Implicit Models
- Bayesian GAN - Bayesian GAN (github)
- BCGAN - Bayesian Conditional Generative Adverserial Networks
- BCGAN - Bidirectional Conditional Generative Adversarial networks
- BEGAN - BEGAN: Boundary Equilibrium Generative Adversarial Networks
- BGAN - Binary Generative Adversarial Networks for Image Retrieval (github)
- BicycleGAN - Toward Multimodal Image-to-Image Translation (github)
- BiGAN - Adversarial Feature Learning
- BS-GAN - Boundary-Seeking Generative Adversarial Networks
- C-GAN - Face Aging with Contextual Generative Adversarial Nets
- C-RNN-GAN - C-RNN-GAN: Continuous recurrent neural networks with adversarial training (github)
- CA-GAN - Composition-aided Sketch-realistic Portrait Generation
- CaloGAN - CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks (github)
- CAN - CAN: Creative Adversarial Networks, Generating Art by Learning About Styles and Deviating from Style Norms
- CapsuleGAN - CapsuleGAN: Generative Adversarial Capsule Network
- CatGAN - Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks
- CatGAN - CatGAN: Coupled Adversarial Transfer for Domain Generation
- CausalGAN - CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training
- CC-GAN - Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks (github)
- CDcGAN - Simultaneously Color-Depth Super-Resolution with Conditional Generative Adversarial Network
- CFG-GAN - Composite Functional Gradient Learning of Generative Adversarial Models
- CGAN - Conditional Generative Adversarial Nets
- CGAN - Controllable Generative Adversarial Network
- Chekhov GAN - An Online Learning Approach to Generative Adversarial Networks
- CipherGAN - Unsupervised Cipher Cracking Using Discrete GANs
- CM-GAN - CM-GANs: Cross-modal Generative Adversarial Networks for Common Representation Learning
- CoAtt-GAN - Are You Talking to Me? Reasoned Visual Dialog Generation through Adversarial Learning
- CoGAN - Coupled Generative Adversarial Networks
- ComboGAN - ComboGAN: Unrestrained Scalability for Image Domain Translation (github)

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

"The GAN Zoo"

- ConceptGAN - Learning Compositional Visual Concepts with Mutual Consistency
- Conditional cycleGAN - Conditional CycleGAN for Attribute Guided Face Image Generation
- constrast-GAN - Generative Semantic Manipulation with Contrasting GAN
- Context-RNN-GAN - Contextual RNN-GANs for Abstract Reasoning Diagram Generation
- Coulomb GAN - Coulomb GANs: Provably Optimal Nash Equilibria via Potential Fields
- Cover-GAN - Generative Steganography with Kerckhoffs' Principle based on Generative Adversarial Networks
- Cramèr GAN - The Cramer Distance as a Solution to Biased Wasserstein Gradients
- Cross-GAN - Crossing Generative Adversarial Networks for Cross-View Person Re-identification
- crVAE-GAN - Channel-Recurrent Variational Autoencoders
- CS-GAN - Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
- CVAE-GAN - CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training
- CycleGAN - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks (github)
- D-GAN - Differential Generative Adversarial Networks: Synthesizing Non-linear Facial Variations with Limited Numl Training Data
- D2GAN - Dual Discriminator Generative Adversarial Nets
- DA-GAN - DA-GAN: Instance-level Image Translation by Deep Attention Generative Adversarial Networks (with Supplementary Materials)
- DAGAN - Data Augmentation Generative Adversarial Networks
- DAN - Distributional Adversarial Networks
- DCGAN - Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks (githut
- DeblurGAN - DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks (github)
- Defense-GAN - Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models
- DeliGAN - DeLiGAN : Generative Adversarial Networks for Diverse and Limited Data (github)
- DF-GAN - Learning Disentangling and Fusing Networks for Face Completion Under Structured Occlusions
- DiscoGAN - Learning to Discover Cross-Domain Relations with Generative Adversarial Networks
- DistanceGAN - One-Sided Unsupervised Domain Mapping
- DM-GAN - Dual Motion GAN for Future-Flow Embedded Video Prediction
- DNA-GAN - DNA-GAN: Learning Disentangled Representations from Multi-Attribute Images
- dp-GAN - Differentially Private Releasing via Deep Generative Model
- DP-GAN - DP-GAN: Diversity-Promoting Generative Adversarial Network for Generating Informative and Diversified Text
- DPGAN - Differentially Private Generative Adversarial Network
- DR-GAN - Representation Learning by Rotating Your Faces
- DRAGAN - How to Train Your DRAGAN (github)
- DRPAN - Discriminative Region Proposal Adversarial Networks for High-Quality Image-to-Image Translation
- DSP-GAN - Depth Structure Preserving Scene Image Generation
- DTN - Unsupervised Cross-Domain Image Generation
- DualGAN - DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
- Dualing GAN - Dualing GANs
- Dynamics Transfer GAN - Dynamics Transfer GAN: Generating Video by Transferring Arbitrary Temporal Dynamics from a Source Video to a Single Target Image
- EBGAN - Energy-based Generative Adversarial Network
- ecGAN - eCommerceGAN : A Generative Adversarial Network for E-commerce
- ED//GAN - Stabilizing Training of Generative Adversarial Networks through Regularization
- EGAN - Enhanced Experience Replay Generation for Efficient Reinforcement Learning
- EnergyWGAN - Energy-relaxed Wassertein GANs (EnergyWGAN): Towards More Stable and High Resolution Image Generation
- ExGAN - Eye In-Painting with Exemplar Generative Adversarial Networks
- ExposureGAN - Exposure: A White-Box Photo Post-Processing Framework (github)
- ExprGAN - ExprGAN: Facial Expression Editing with Controllable Expression Intensity
- f-CLSWGAN - Feature Generating Networks for Zero-Shot Learning
- f-GAN - f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization
- FF-GAN - Towards Large-Pose Face Frontalization in the Wild
- FIGAN - Frame Interpolation with Multi-Scale Deep Loss Functions and Generative Adversarial Networks
- Fila-GAN - Synthesizing Filamentary Structured Images with GANs
- First Order GAN - First Order Generative Adversarial Networks
- Fisher GAN - Fisher GAN
- Flow-GAN - Flow-GAN: Bridging implicit and prescribed learning in generative models
- FSEGAN - Exploring Speech Enhancement with Generative Adversarial Networks for Robust Speech Recognition
- FTGAN - Hierarchical Video Generation from Orthogonal Information: Optical Flow and Texture
- FusedGAN - Semi-supervised FusedGAN for Conditional Image Generation

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

“The GAN Zoo”

- FusionGAN - Learning to Fuse Music Genres with Generative Adversarial Dual Learning
- G2-GAN - Geometry Guided Adversarial Facial Expression Synthesis
- GAGAN - GAGAN: Geometry-Aware Generative Adverserial Networks
- GAMN - Generative Adversarial Mapping Networks
- GAN - Generative Adversarial Networks (github)
- GAN-ATV - A Novel Approach to Artistic Textual Visualization via GAN
- GAN-CLS - Generative Adversarial Text to Image Synthesis (github)
- GAN-RS - Towards Qualitative Advancement of Underwater Machine Vision with Generative Adversarial Networks
- GAN-sep - GANs for Biological Image Synthesis (github)
- GAN-VFS - Generative Adversarial Network-based Synthesis of Visible Faces from Polarimetric Thermal Faces
- GANCS - Deep Generative Adversarial Networks for Compressed Sensing Automates MRI
- GANDI - Guiding the search in continuous state-action spaces by learning an action sampling distribution from off-target samples
- GANG - GANGs: Generative Adversarial Network Games
- GANosaic - GANosaic: Mosaic Creation with Generative Texture Manifolds
- GAP - Context-Aware Generative Adversarial Privacy
- GAWWN - Learning What and Where to Draw (github)
- GC-GAN - Geometry-Contrastive Generative Adversarial Network for Facial Expression Synthesis
- GeneGAN - GeneGAN: Learning Object Transfiguration and Attribute Subspace from Unpaired Data (github)
- GeoGAN - Generating Instance Segmentation Annotation by Geometry-guided GAN
- Geometric GAN - Geometric GAN
- GLCA-GAN - Global and Local Consistent Age Generative Adversarial Networks
- GMAN - Generative Multi-Adversarial Networks
- GMM-GAN - Towards Understanding the Dynamics of Generative Adversarial Networks
- GoGAN - Gang of GANs: Generative Adversarial Networks with Maximum Margin Ranking
- GP-GAN - GP-GAN: Towards Realistic High-Resolution Image Blending (github)
- GP-GAN - GP-GAN: Gender Preserving GAN for Synthesizing Faces from Landmarks
- GPU - A generative adversarial framework for positive-unlabeled classification
- GRAN - Generating images with recurrent adversarial networks (github)

And Many More ……………………

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

GANs: Things to Remember

Take game-theoretic approach: learn to generate from training distribution through 2-player game

Pros:
- Beautiful, state-of-the-art samples!

Cons:
- Trickier / more unstable to train
- Can’t solve inference queries such as p(x), p(z|x)

Active areas of research:
- Better loss functions, more stable training (Wasserstein GAN, LSGAN, many others)
- Conditional GANs, GANs for all kinds of applications

Thank you