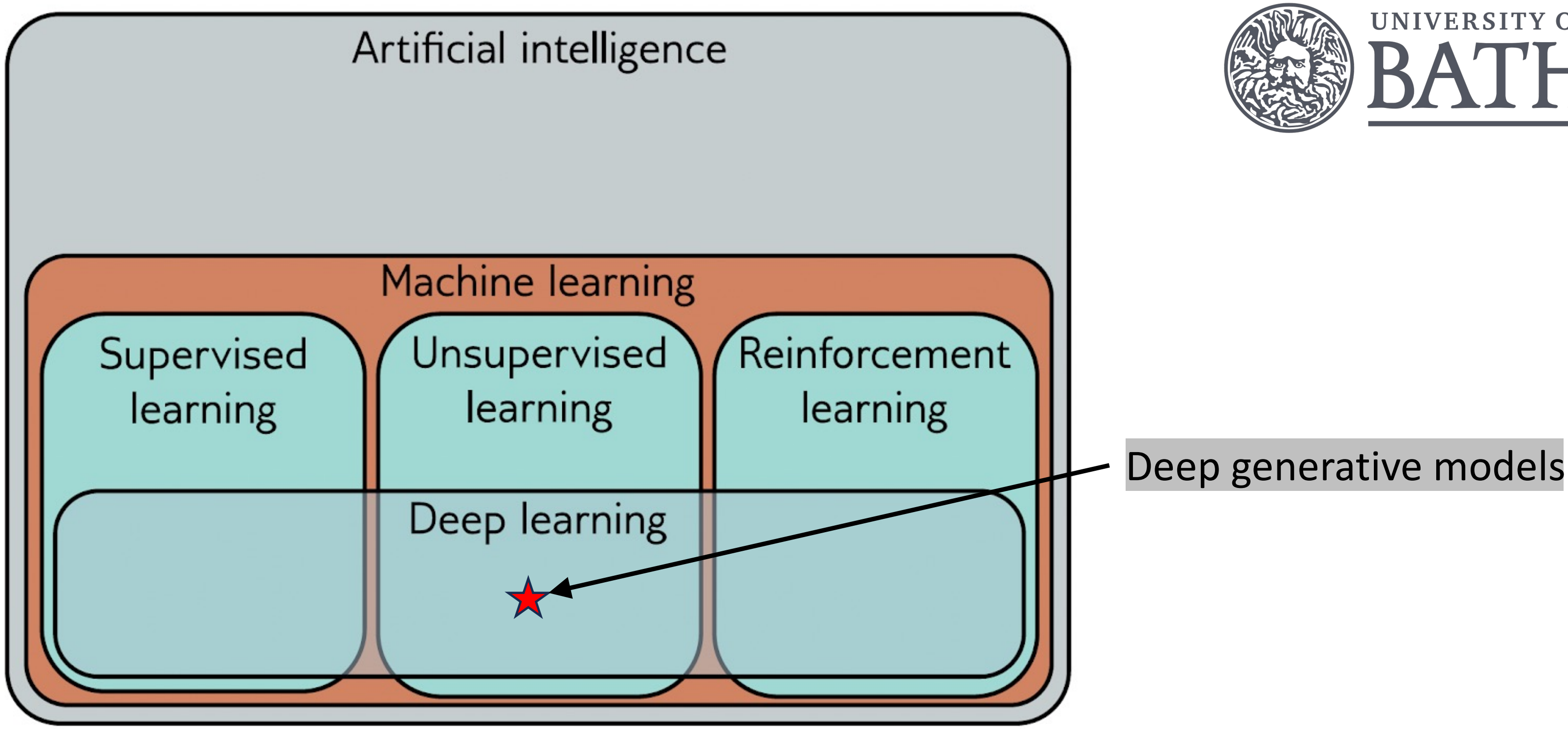
Week 07: Generative Adversarial Networks (GANs)

Dr. Hongping Cai
Department of Computer Science
University of Bath

Topic 1: Introduction to Deep Generative Models

UNIVERSITY OF BATH



- **Data**: X – data, Y – label
- **Goal**: Learn a mapping: Y = f(X)

Supervised

- **Data**: X – data
- **Goal**: Learn underlying structure of the data.

Unsupervised

Image from: Simon Prince's book: Understanding Deep Learning, 2023.

Discriminative models vs Generative models

- Discriminative models
  - Discriminate between different kind of data instance
  - Care about boundary between classes only
  - E.g., Linear regression, SVM, neural networks
  - Model p(Y|X)
- Generative models
  - Discover the distribution of training samples
  - Care about entire distribution – harder, but more powerful
  - E.g., Gaussian process, density estimation, HMM
  - Model p(X, Y), or just p(X) if no labels

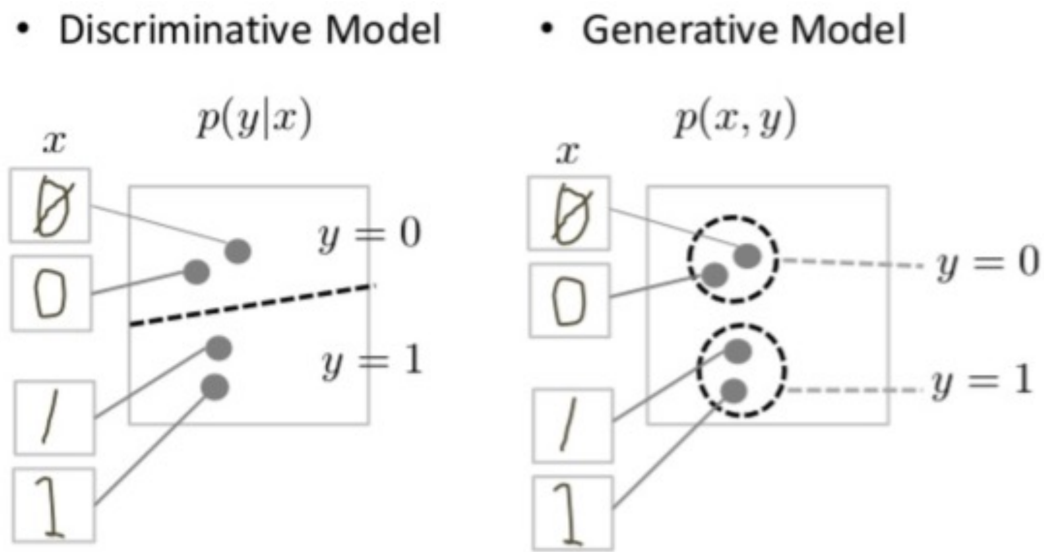


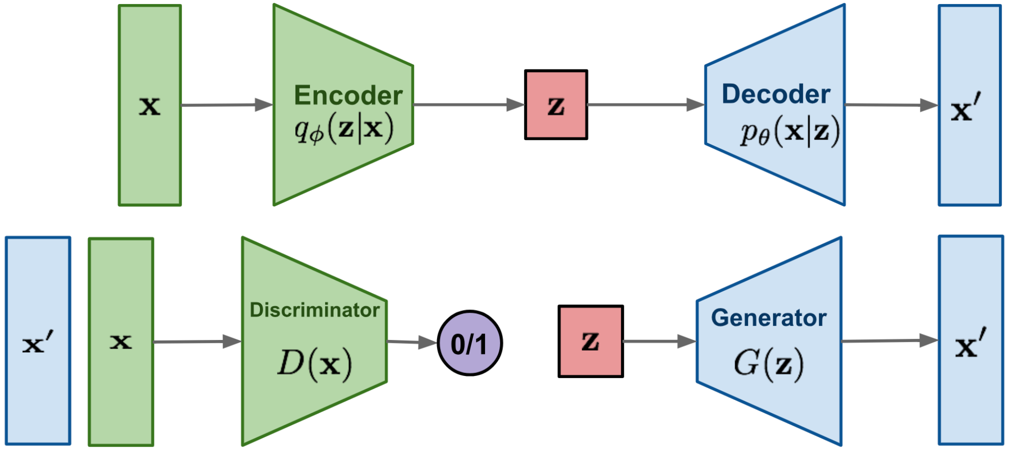
Image from: https://developers.google.com/machine-learning/gan/generative

Two Typical Deep Generative Models

Both are unsupervised methods. Neither of them explicitly learns $p(x)$

Variational Autoencoder (VAE)

Generative Adversarial Network (GAN)



Other deep generative models:

- **Augoregressive models, e.g., PixelRNN, PixeCNN**
- **Normalizing flows**
- **Diffusion models**

Image from: https://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html

Reference

- Lecture on Generative Adversarial Networks: https://developers.google.com/machine-learning/gan/generative Video lecture by Ava Soleimany: MIT course on Deep Generative Models, YouTube.
- Lecture by Rowel Atienza: https://docs.google.com/presentation/d/13fiFibqjl9ps_CktJzMNAvoZXOlzHQDu8eRSb3a227g/edit#slide=id.g389666c0c0_0_5

Topic 2: GANs

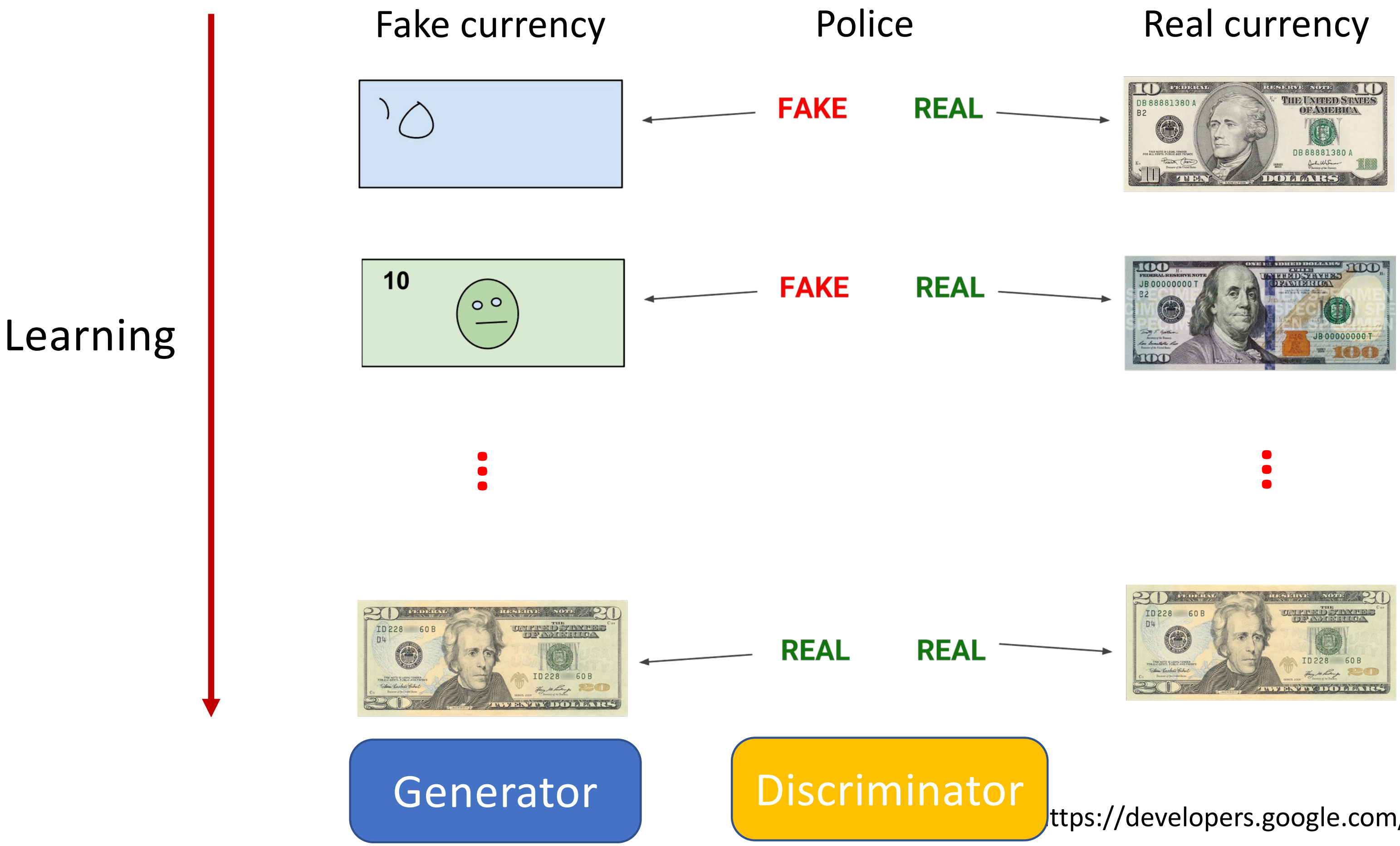
Generative Adversarial Networks (GANs)

- GAN [Goodfellow-NIPS2014] is an unsupervised machine learning technique.
- GAN is a generative model to generate data with similar characteristics as the real data.

GAN generated faces by training on the CELEBA-HQ dataset [Karras-ICLR2018]

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

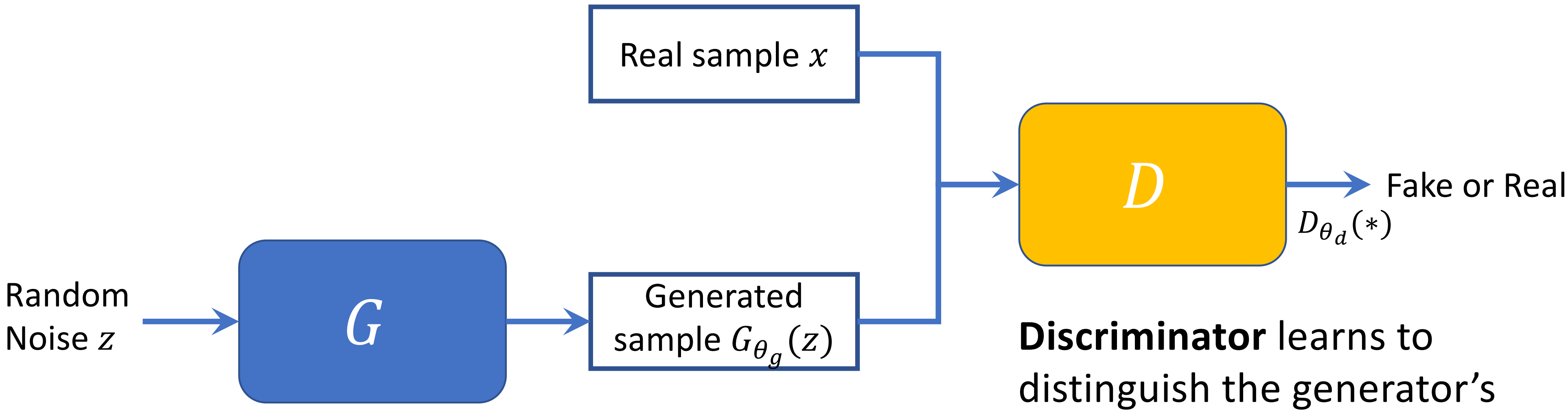
Intuition behind GANs



ttps://developers.google.com/machine-learning/gan/gan_structure

GAN Architecture

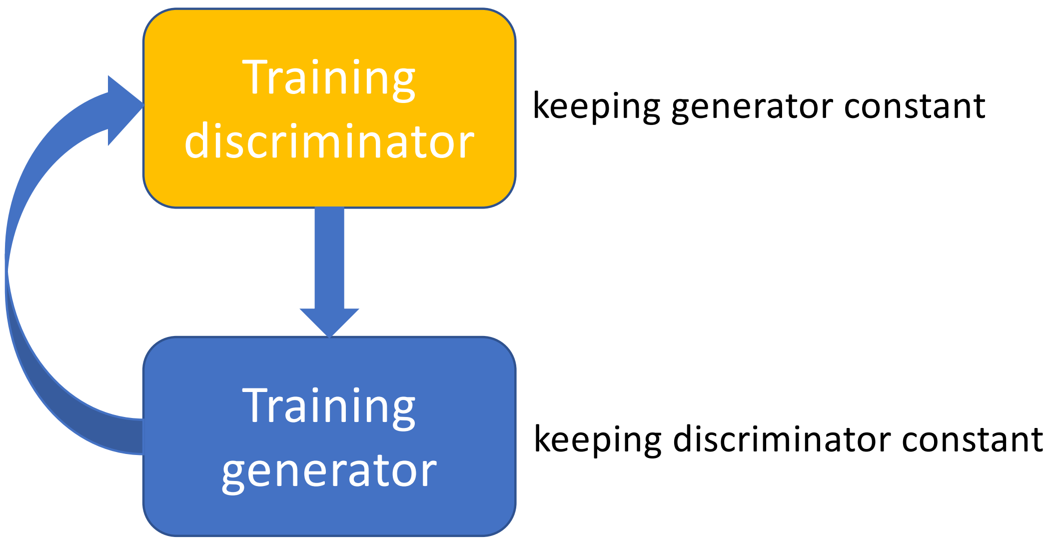
- A GAN has two neural networks competing with each other.



Generator learns to generate data that is similar to the real data.

Discriminator learns to distinguish the generator's fake data from real data.

Training GANs – Alternating training



Training Discriminator

- **Binary cross-entropy loss** for binary classification:

$$J(W) = -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)}\log f_W\left(x^{(n)}\right) + \left(1 - y^{(n)}\right)\log\left(1 - f_W\left(x^{(n)}\right)\right)\right)$$

actual
{0, 1}

prediction
[0,1]

Training Discriminator

- **Binary cross-entropy loss** for binary classification:

$$J(W) = -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)}\log f_W\left(x^{(n)}\right) + \left(1 - y^{(n)}\right)\log\left(1 - f_W\left(x^{(n)}\right)\right)\right)$$

actual
{0, 1}

prediction
[0,1]

$$J(W) =$$

For all positive samples ($y^{(n)} = 1$)

For all negative samples ($y^{(n)} = 0$)

Training Discriminator

- **Binary cross-entropy loss** for binary classification:

$$J(W) = -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)}\log f_W\left(x^{(n)}\right) + (1-y^{(n)})\log\left(1-f_W\left(x^{(n)}\right)\right)\right)$$

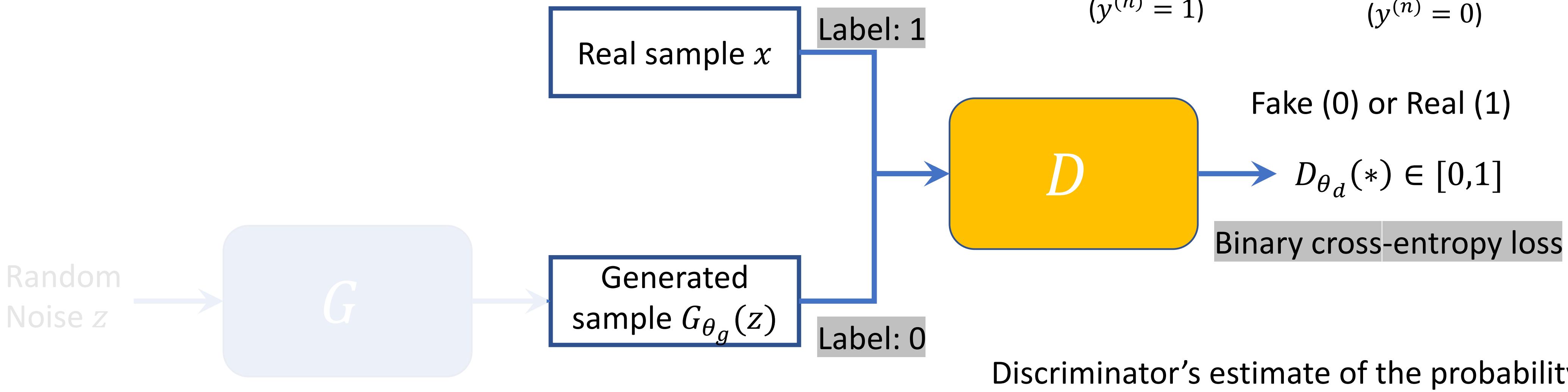
actual
{0, 1}

prediction
[0,1]

$$J(W) = \underbrace{-\frac{1}{N_1}\sum_{y^{(n)}=1}\log f_W\left(x^{(n)}\right)}_{\text{For all positive samples } (y^{(n)}=1)} \underbrace{-\frac{1}{N_0}\sum_{y^{(n)}=0}\log\left(1-f_W\left(x^{(n)}\right)\right)}_{\text{For all negative samples } (y^{(n)}=0)}$$

Binary cross-entropy loss: $-\underbrace{\frac{1}{N_1}\sum_{y^{(n)}=1}\log f_W(x^{(n)})}_{\text{For all positive samples } (y^{(n)}=1)}-\underbrace{\frac{1}{N_0}\sum_{y^{(n)}=0}\log(1-f_W(x^{(n)}))}_{\text{For all negative samples } (y^{(n)}=0)}$

Training Discriminator

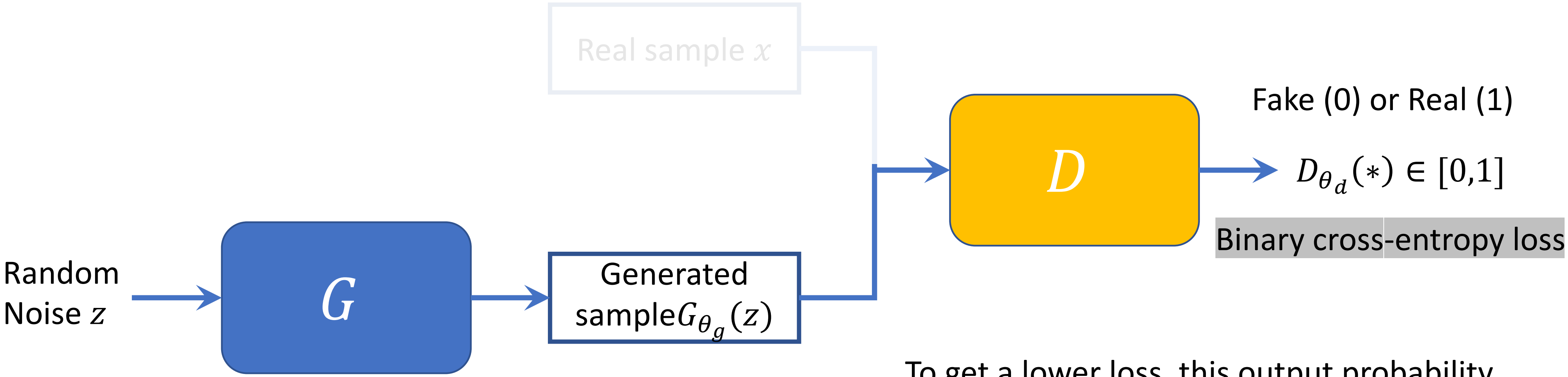


Discriminator's estimate of the probability that the real sample is real

- The discriminator loss: $\mathcal{L}_D = -E_x[\log D_{\theta_d}(x)] - E_z[\log(1 - D_{\theta_d}(G_{\theta_g}(z)))]$
- It penalizes the misclassification.

Discriminator's estimate of the probability that the generated sample is real

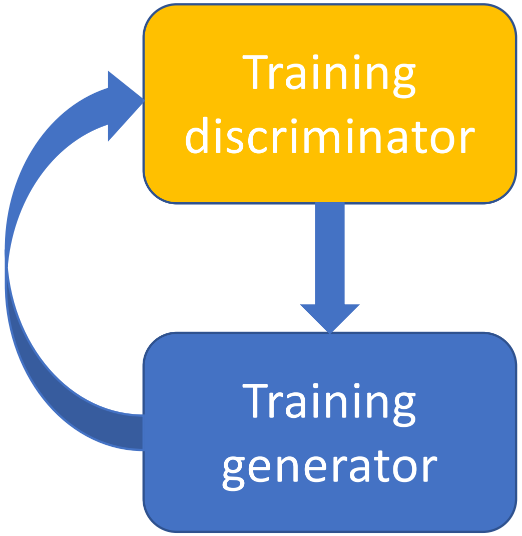
Training Generator



To get a lower loss, this output probability must be as close to 1 as possible.

- The generator loss: $\mathcal{L}_G = -E_z[\log \boxed{D_{\theta_d}(G_{\theta_g}(z))}]$
- It penalizes the generator for failing to fool the discriminator.
- During the back propagation, the discriminator's gradients are passed down to the Generator, however, its weight are frozen.

Training GANs – Alternating training

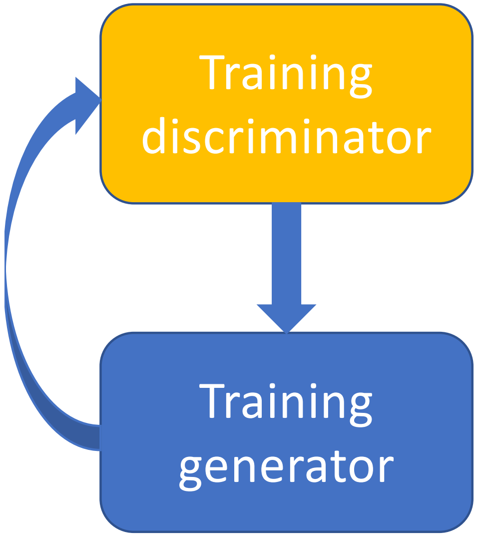


$$\mathcal{L}_D = -E_x[\log D_{\theta_d}(x)] - E_z[\log(1 - D_{\theta_d}(G_{\theta_g}(z)))]$$

$$\mathcal{L}_G = -E_z[\log D_{\theta_d}(G_{\theta_g}(z))]$$

Objective: $\min\limits_{\theta_g} \max\limits_{\theta_d} E_x[\log D_{\theta_d}(x)] + E_z[\log(1 - D_{\theta_d}(G_{\theta_g}(z)))]$

Training GANs – Alternating training



- Step 1: Sample minibatch of B training samples $x^{(1)}, x^{(2)}, \dots, x^{(B)}$
- Step 2: Sample minibatch of B random vectors $z^{(1)}, z^{(2)}, \dots, z^{(B)}$
- Step 3: Update the generator parameters θ_g by mini-batch gradient descent for one or more epochs.

$$\nabla_{\theta_g} \mathcal{L}_G = -\frac{1}{B} \nabla_{\theta_g} \sum_{i=1}^{B} \log D_{\theta_d}(G_{\theta_g}(z^{(i)}))$$

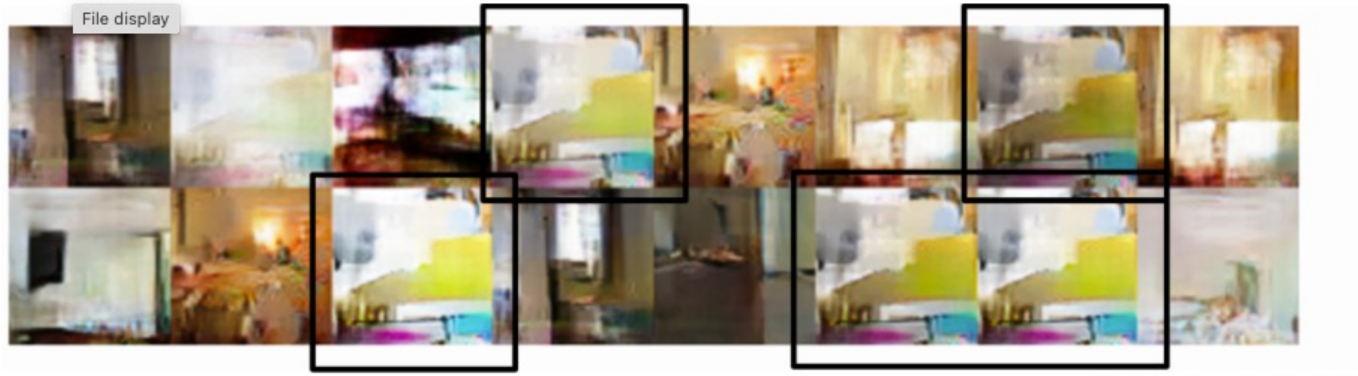
- Step 4: Update the discriminator parameters θ_d by mini-batch gradient descent for one or more epochs.

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

- Repeat Step 3 and Step 4 for fixed number of epochs

Challenges of training GANs

- **Mode Collapse**: the generator learns to produce a limited variety of outputs, or even the same output, regardless of the input z.



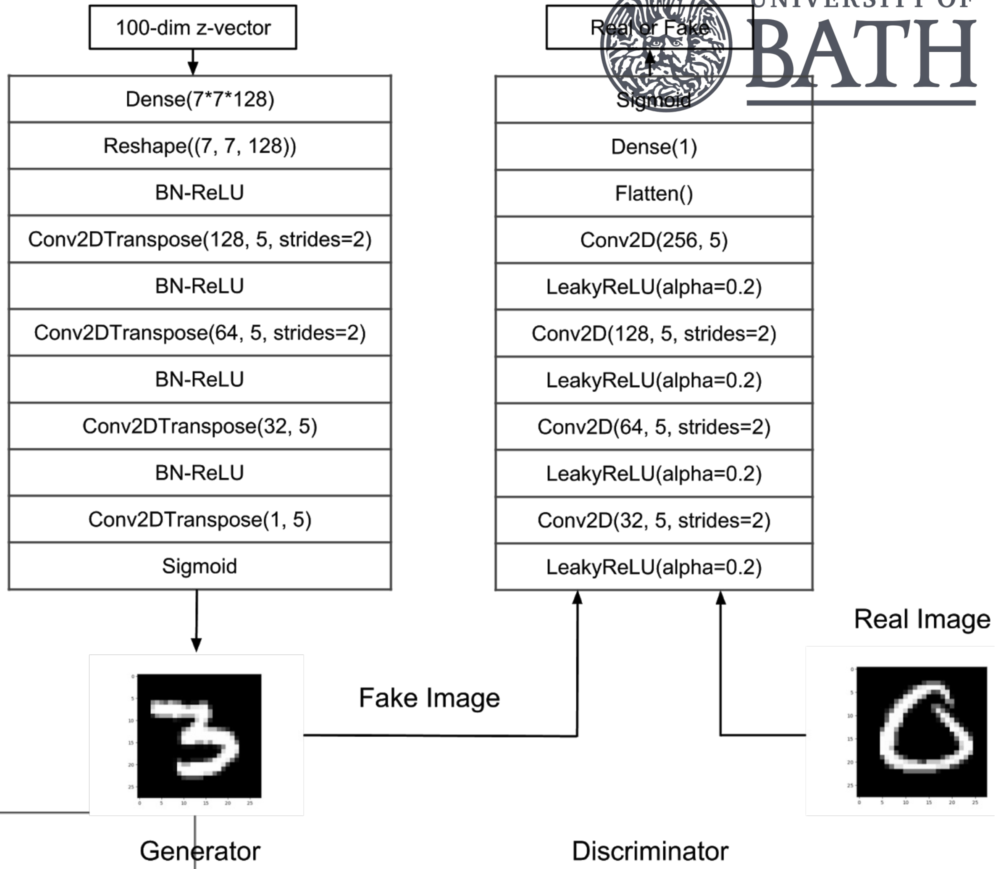
Arjovsky et al., 2017

- **Non-convergence**: the generator and discriminator continue to adapt in response to each other without reaching a stable equilibrium

More details: D Saxena et al. Generative Adversarial Networks (GANs): Challenges, Solutions, and Future Directions, 2020

DCGAN

- Deep Convolutional GAN [Radford-2016] is a GAN implemented using Deep CNN.



Architecture guidelines for stable Deep Convolutional GANs

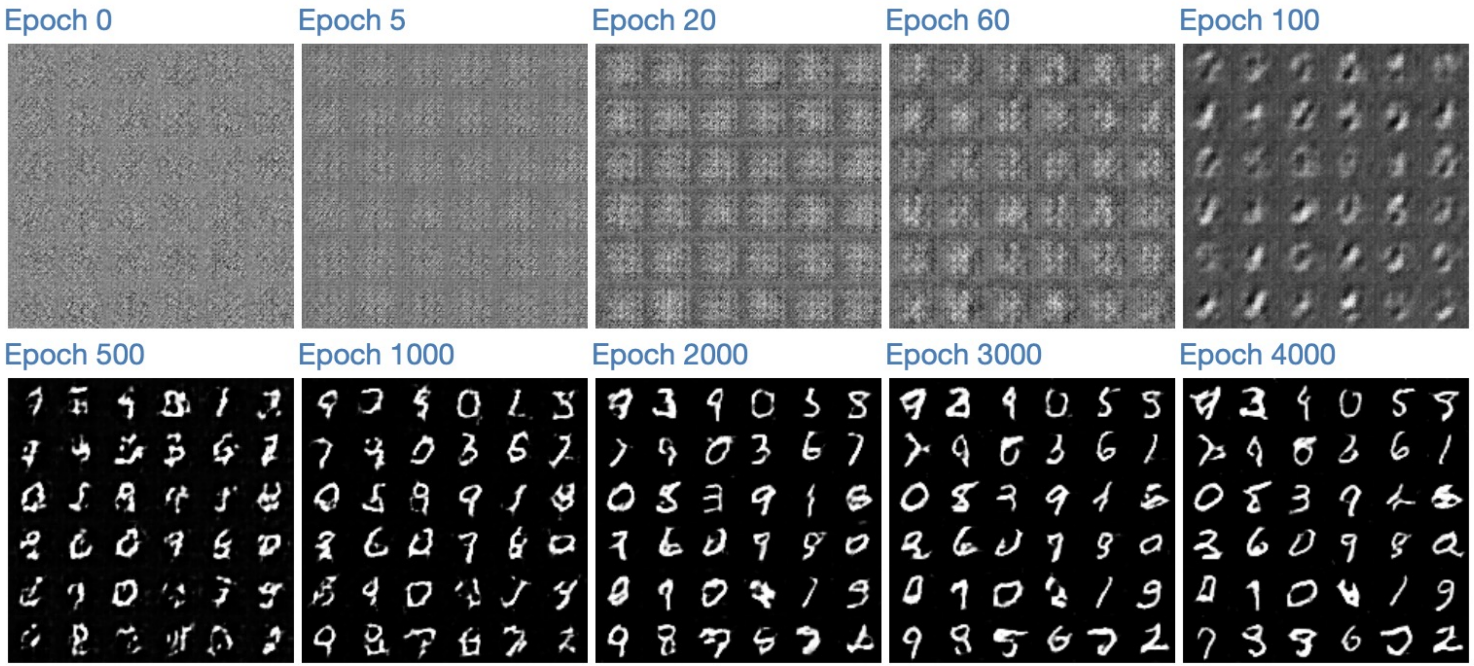
- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

[Radford-2016] Radford et al, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", ICLR 2016

Img from: https://docs.google.com/presentation/d/13fiFibqjl9ps_CktJzMNAvoZXOlzHQDu8eRSb3a227g/edit#slide=id.p

DCGAN

MNIST dataset



Generated samples

Image from: http://www.timzhangyuxuan.com/project_dcgan/

DCGAN

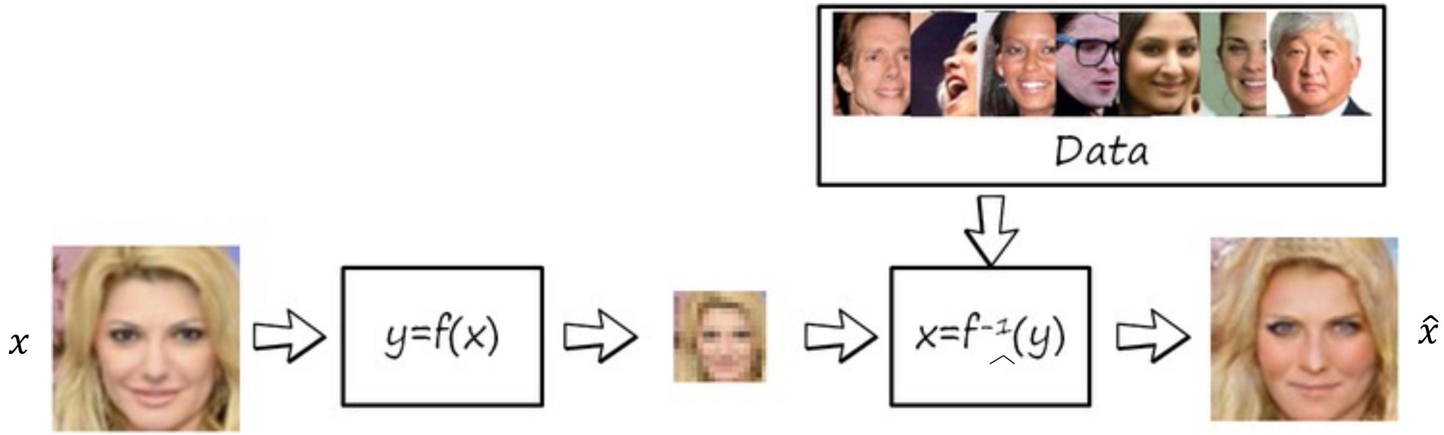
CelebA dataset

Generated samples

Image from: http://www.timzhangyuxuan.com/project_dcgan/

Super Resolution

- Data-driven image-to-image super resolution



Standard Objective: Minimize the MSE between x and $\hat{x}$ (**pixel-level**)

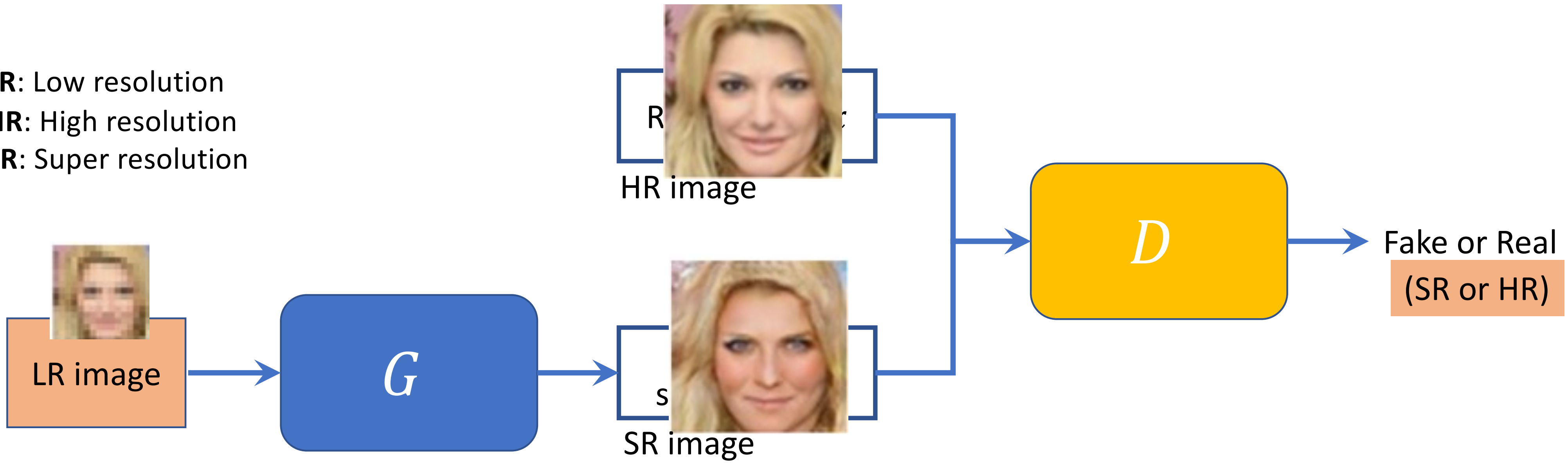
+ Benefit: Maximizing Peak Signal-to-Noise Ratio (PSNR)
- Drawback: Limit for capture perceptually relevance features, e.g., losing high textural details

Minimize the MSE between $\phi_{ij}(x)$ and $\phi_{ij}(\hat{x})$ (**feature-level**)

Ledig et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", CVPR 2017.

SRGAN [Ledig-2017]

LR: Low resolution
HR: High resolution
SR: Super resolution



Objective:

$$l^{SR} = \underbrace{l^{SR}_{X}}_{\text{content loss}} + \underbrace{10^{-3} l^{SR}_{Gen}}_{\text{adversarial loss}}$$

$$l^{SR}_{VGG/i,j} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2$$

$$l^{SR}_{Gen} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$

Ledig et al. “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network”, CVPR 2017.

SRGAN [Ledig-2017]

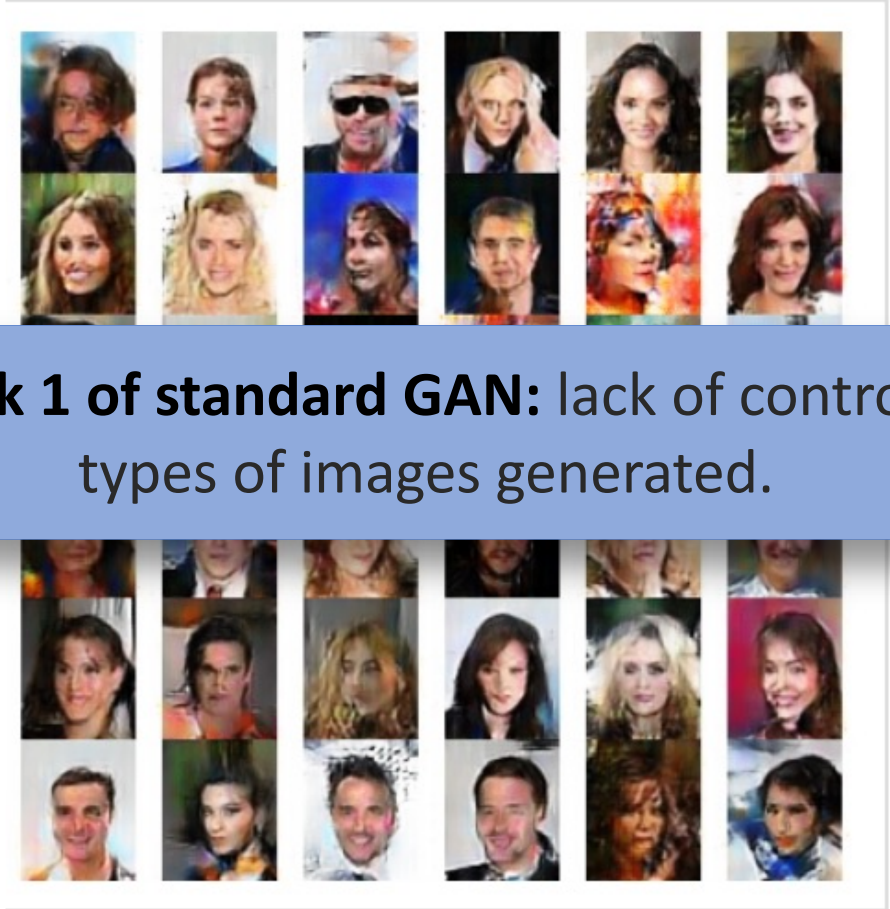
bicubic (21.59dB/0.6423)	SRResNet (23.53dB/0.7832)	SRGAN (21.15dB/0.6868)	original

Ledig et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", CVPR 2017.

Reference

- Lecture on Generative Adversarial Networks: https://developers.google.com/machine-learning/gan
- Video lecture by Ava Soleimany: MIT course on Deep Generative Models, YouTube.
- Lecture by Rowel Atienza: https://docs.google.com/presentation/d/13fiFibqjl9ps_CktJzMNAvoZXOlzHQDu8eRSb3a227g/edit#slide=id.g389666c0c0_0_5

Topic 3: GANs Variations (Conditional GAN)



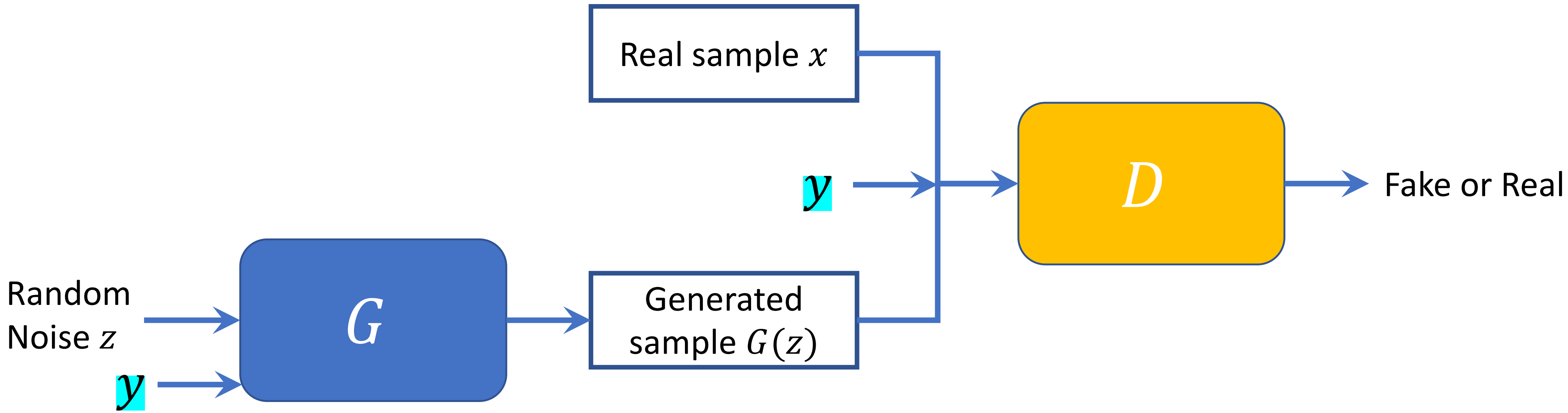
Drawback 1 of standard GAN: lack of control of the types of images generated.

Generated samples of GAN

Image from: http://www.timzhangyuxuan.com/project_dcgan/

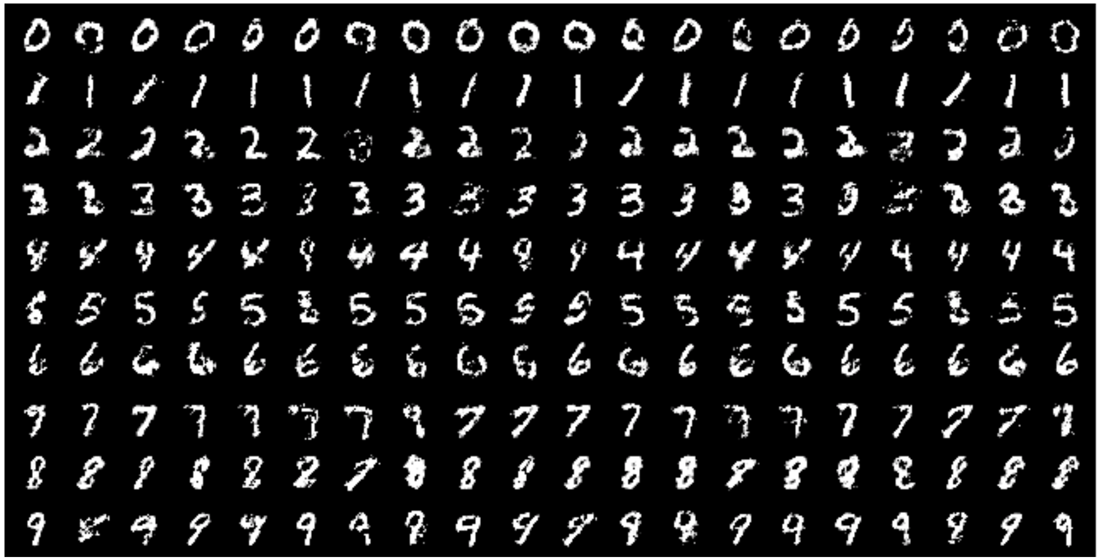
Conditional GAN (CGAN)

- CGAN [Mirza-2014] imposes a constraint (e.g., a label/category) to control the attribute of the generator output.
- CGANs model the conditional probability P(X | Y).



Objective: $\min_{\theta_g} \max_{\theta_d} E_x[\log D_{\theta_d}(x|y)] + E_z[\log(1 - D_{\theta_d}(G_{\theta_g}(z|y)))]$

Conditional GAN (CGAN)



Generated MNIST digits, each row conditioned
on one label [Mirza-2014]

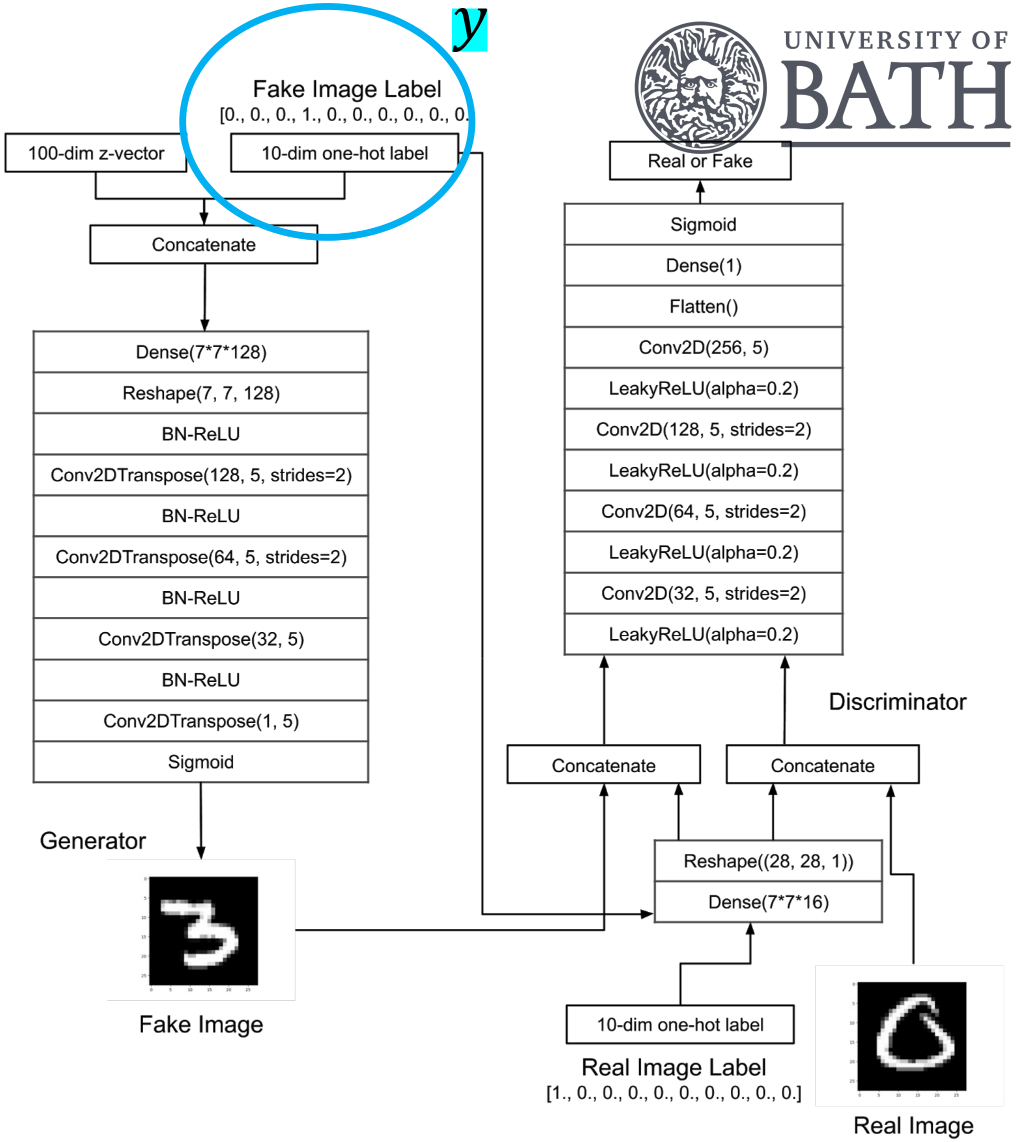


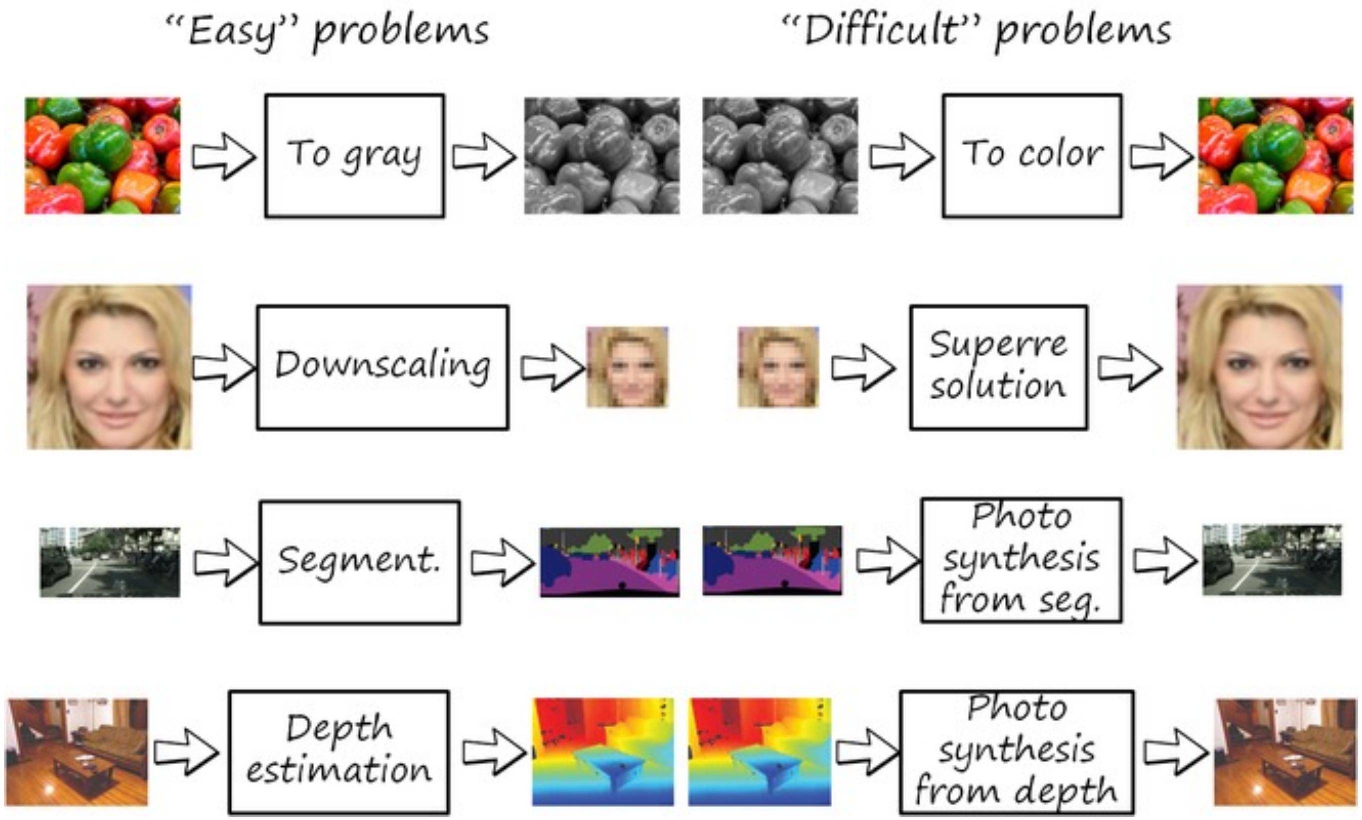
Image from:https://docs.google.com/presentation/d/13fiFibqjl9ps_CktJzMNAvoZXOlzHQDu8eRSb3a227g/edit#slide=id.p

Conditional GAN (CGAN)

- Category is only one type of conditions. Actually, the condition could be:
  - Image embeddings → image-to-image translation
  - Text embeddings → text-to-image synthesis
    - E.g., StackGAN [Zhang-2017]
  - Attribute embeddings → attribute-to-image synthesis
    - E.g., InfoGAN [Chen-2016]
  - …

Han Zhang et al. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks , ICCV 2017
Chen et al. "Infogan: Interpretable representation learning by information maximizing generative adversarial nets", NIPS 2016

Image-to-Image Translation



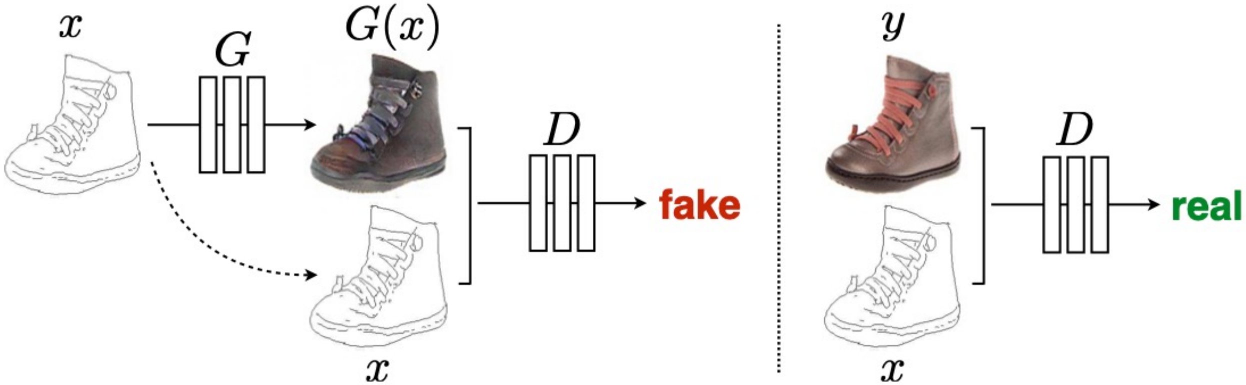
Pix2Pix

Paired image-to-image translation

Isola et al. Image-to-Image Translation with Conditional Adversarial Networks, CVPR 2017

Pix2Pix

- Pix2Pix [Isola-2017] is a type of CGAN, where the condition is the paired image.

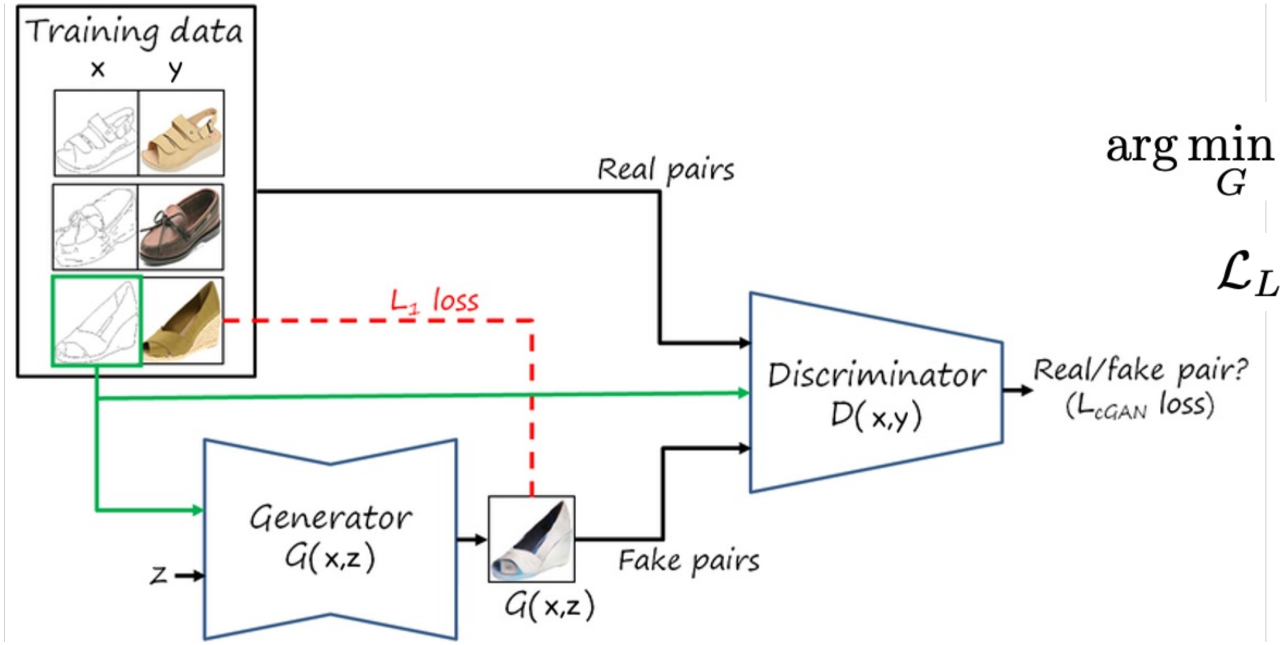


- Unlike the unconditional GAN, both the generator and discriminator observe the input edge map x.

Isola et al. Image-to-Image Translation with Conditional Adversarial Networks, CVPR 2017

Pix2Pix

- In addition to the the CGAN loss, a L1 loss that forces the generated image to remain as similar as possible to the ground-truth image.



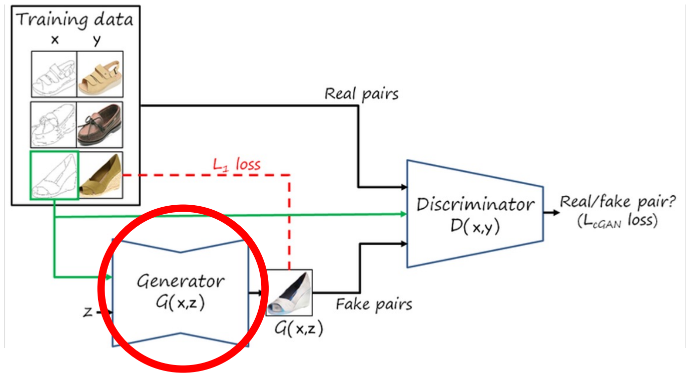
$$\arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$$

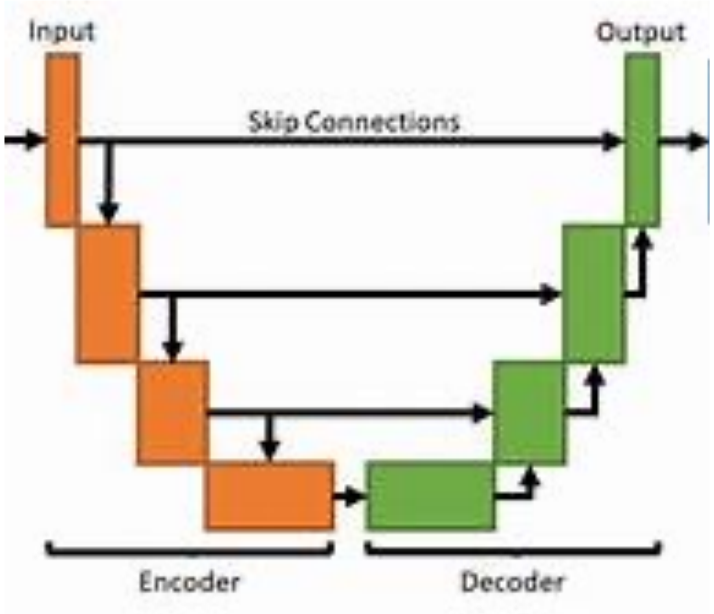
$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1]$$

Image from: https://www.lherranz.org/2018/08/07/imagetranslation/

Pix2Pix

- Generator: **U-Net + skips**

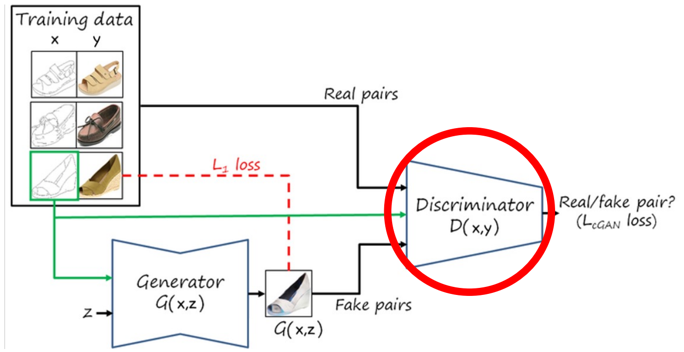


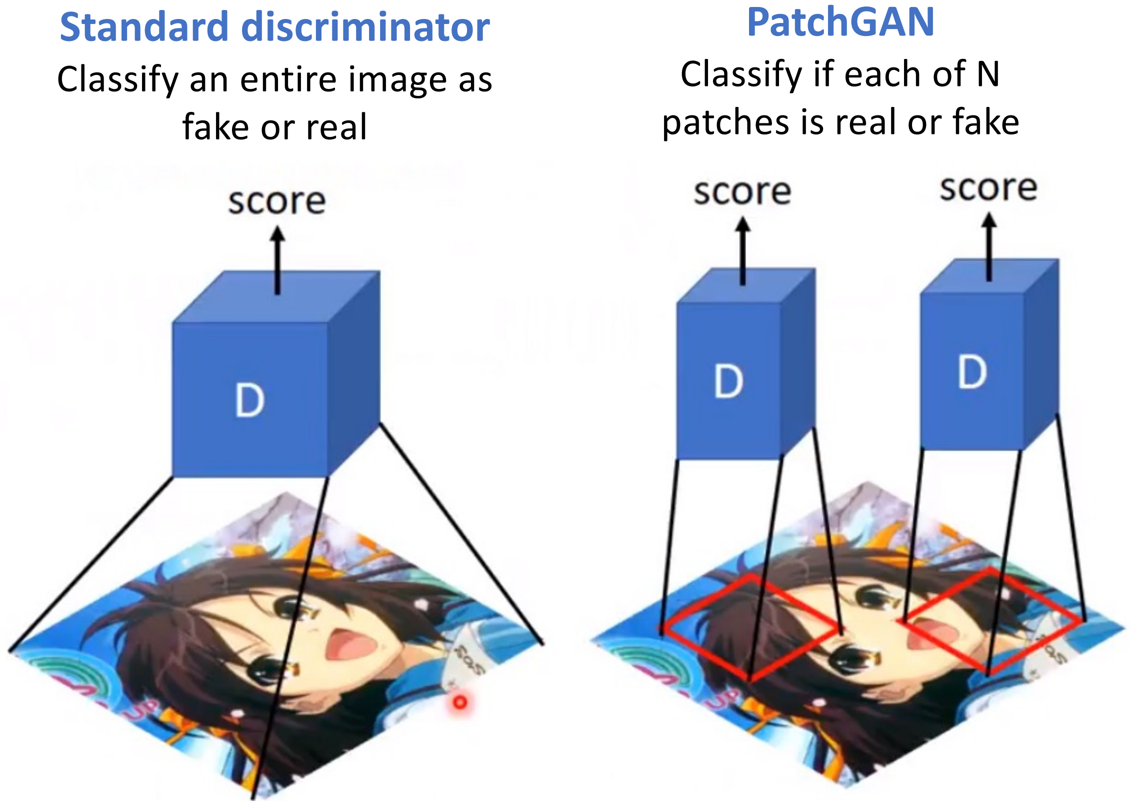


Isola et al. Image-to-Image Translation with Conditional Adversarial Networks, CVPR 2017

Pix2Pix

- Discriminator: **PatchGAN**

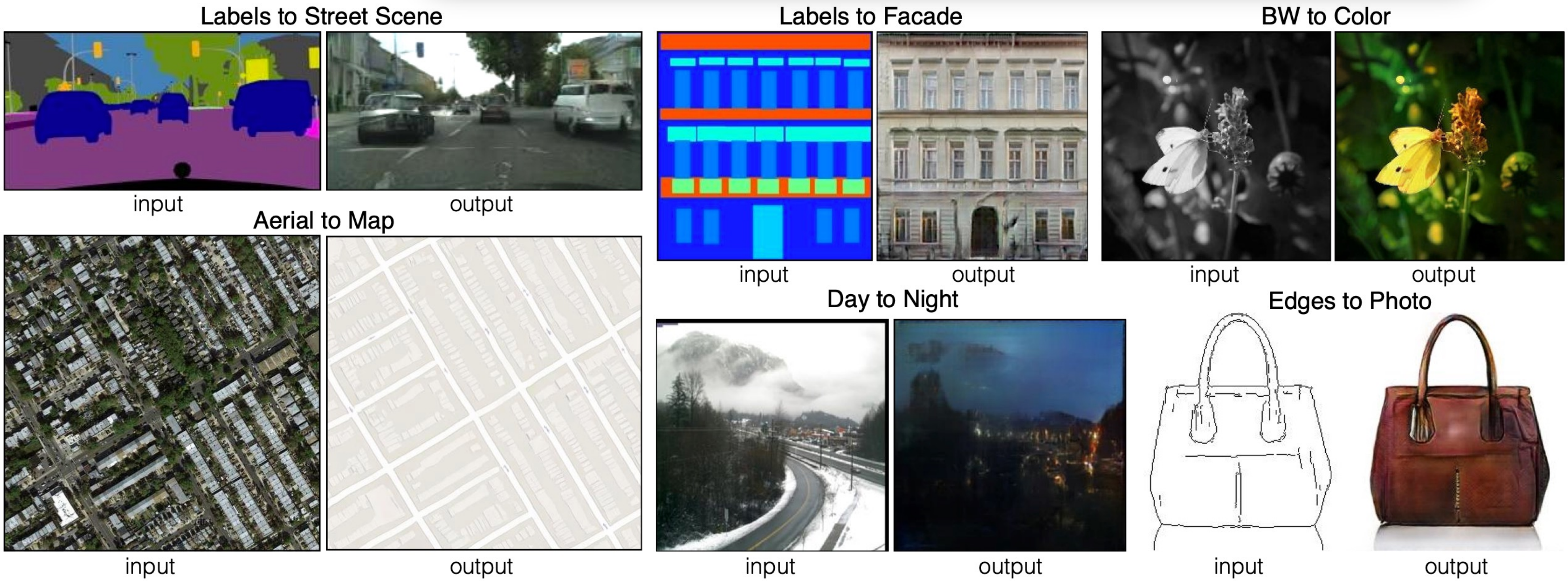




Isola et al. Image-to-Image Translation with Conditional Adversarial Networks, CVPR 2017

Pix2Pix

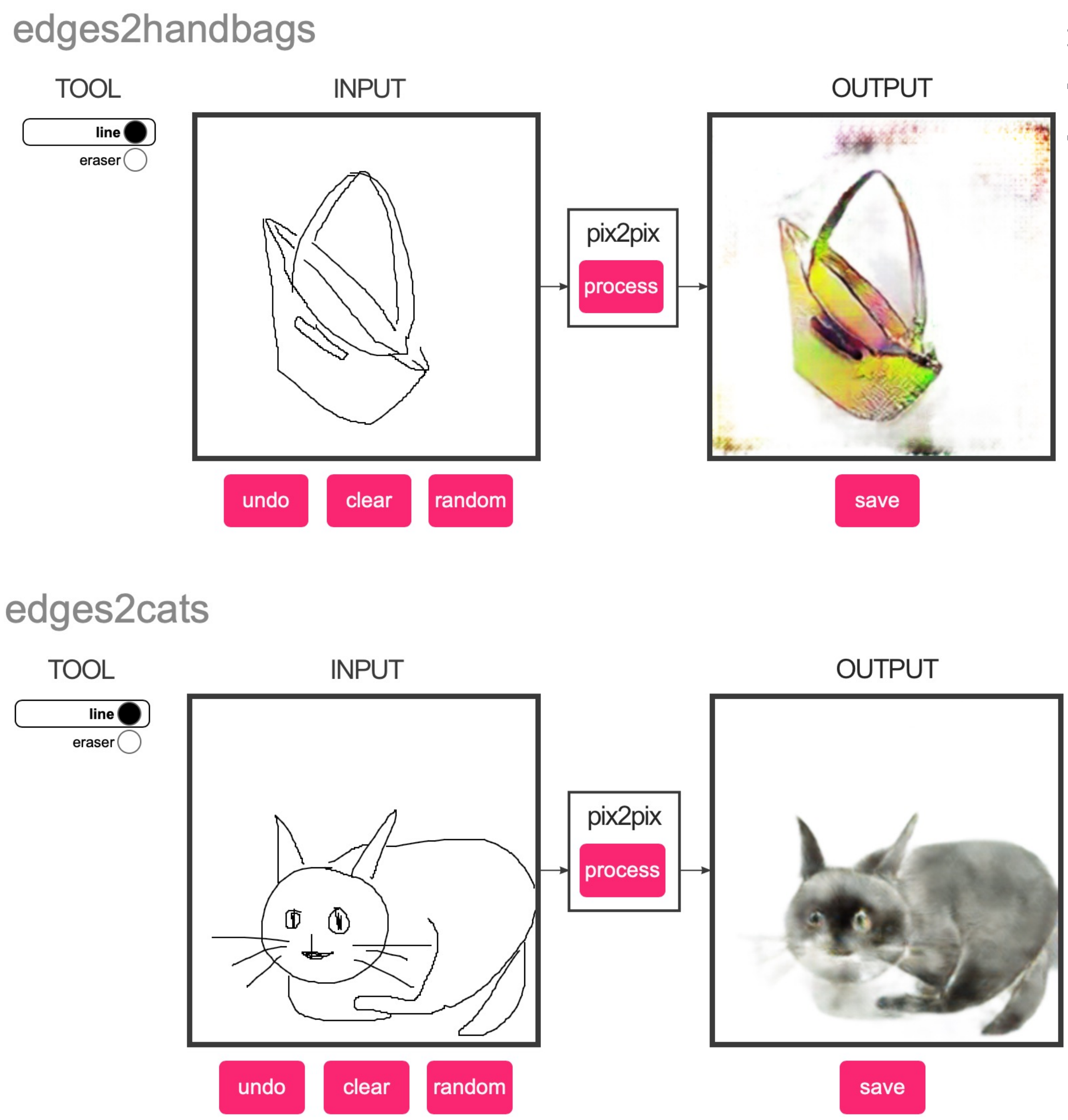
Pix2Pix is suitable for any tasks where paired image dataset can be created.



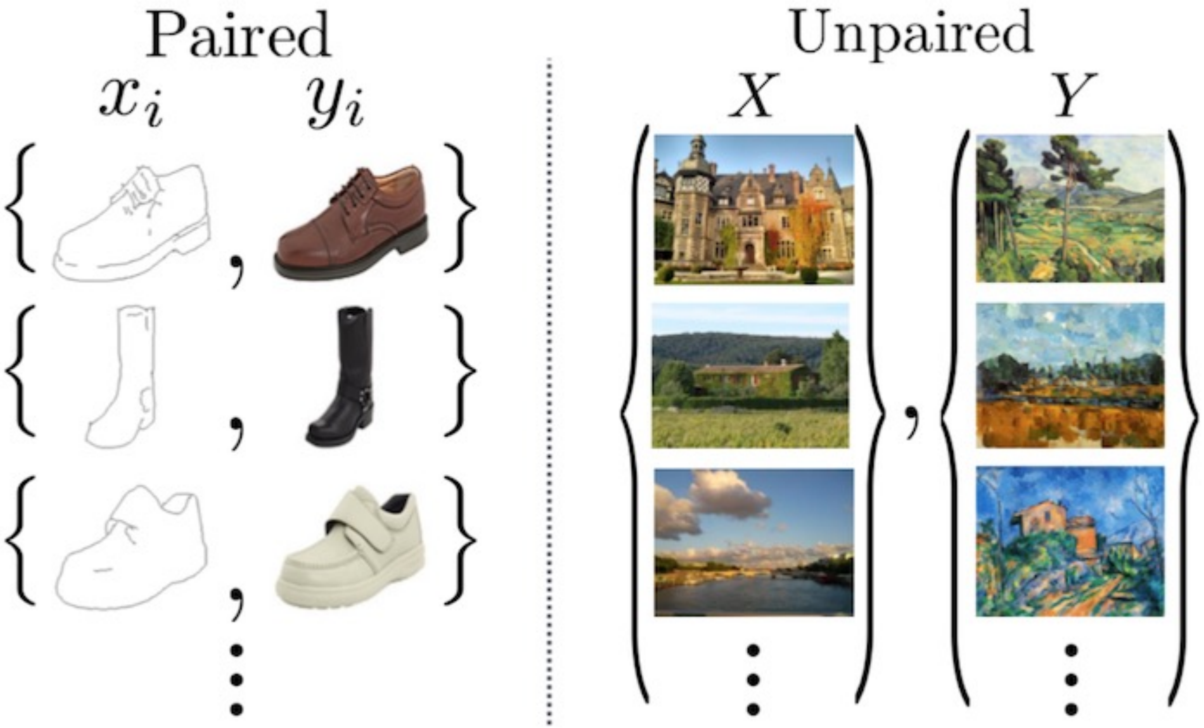
Isola et.al. “Image-to-Image Translation with Conditional Adversrial Networks”, CVPR 2017

Interactive demo of pix2pix

- Demo by Christopher Hesse: https://affinelayer.com/pixsrv/
- Tensorflow Code: https://github.com/affinelayer/pix2pix-tensorflow



But aligned pairs are not always available …



Zhu et al. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, ICCV 2017.

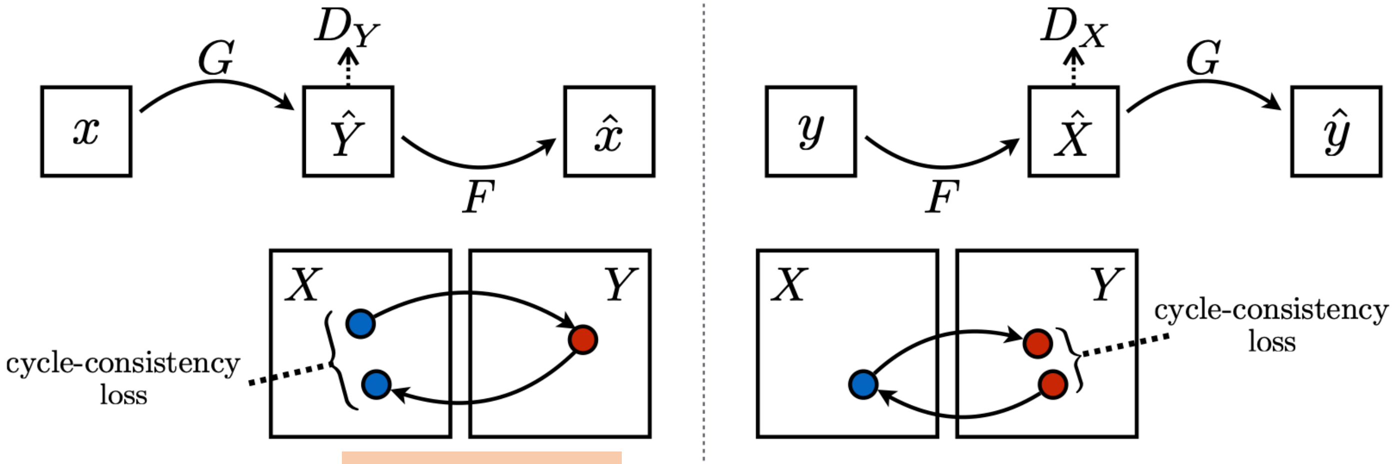
CycleGAN

A model for domain transfer

Zhu et al. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, ICCV 2017.

CycleGAN: Unpaired image-to-image translation

- CycleGAN[Zhu-ICCV2017] learns transformations across domains with unpaired data.



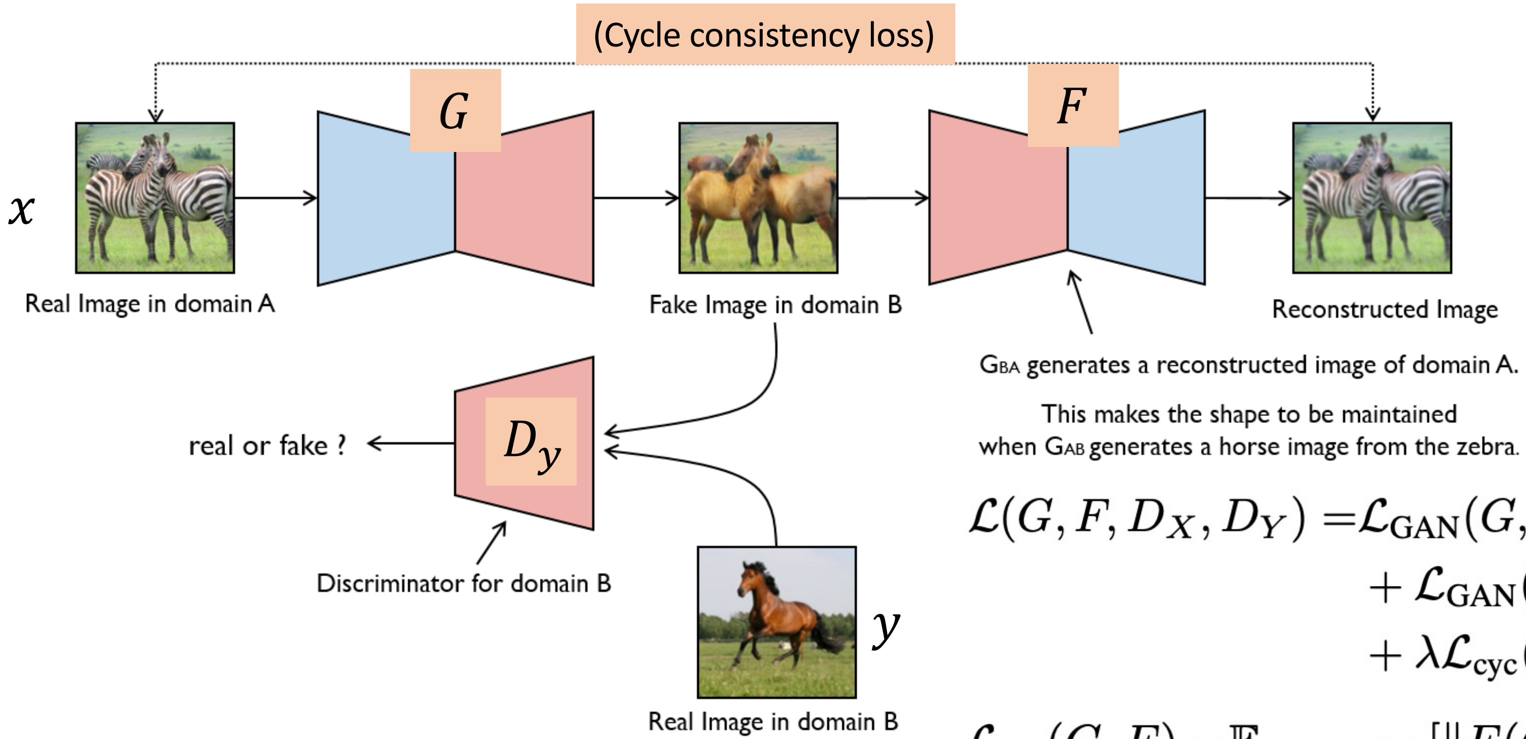
Two mapping:

$G: X \rightarrow Y$

$F: Y \rightarrow X$

Zhu et al. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, ICCV 2017.

Unpaired image-to-image translation: CycleGAN



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

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

Image from https://modelzoo.co/model/mnist-svhn-transfer

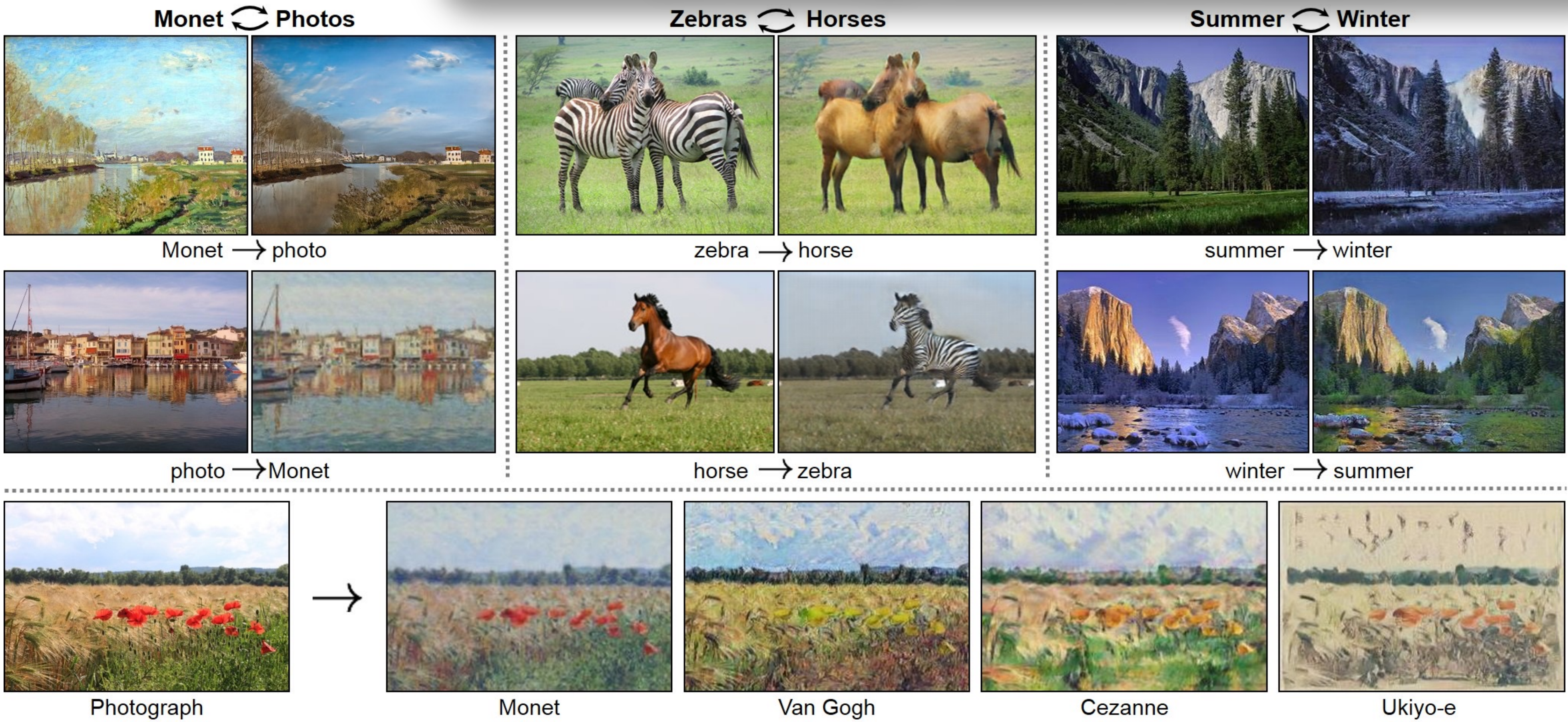
Zhu et al. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, ICCV 2017.

CycleGAN

Video from: https://youtu.be/9reHvktowLY

CycleGAN

CycleGAN is suitable for stylistic transformations where paired image dataset is not feasible.



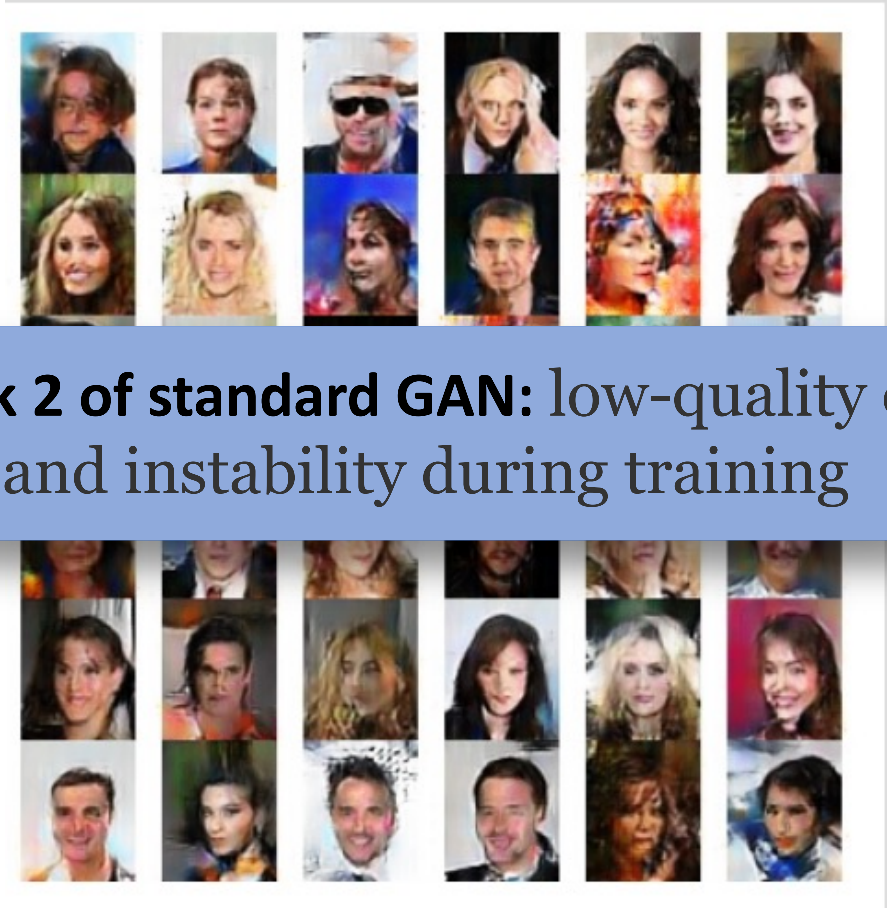
More results: https://junyanz.github.io/CycleGAN/
Code: https://github.com/junyanz/CycleGAN

Reference

- Lecture on Generative Adversarial Networks: https://developers.google.com/machine-learning/gan/applications
- YouTube Video by Ava Soleimany: MIT course on Deep Generative Models.
- Lecture by Rowel Atienza: https://docs.google.com/presentation/d/13fiFibqjl9ps_CktJzMNAvoZXOlzHQDu8eRSb3a227g/edit#slide=id.g389666c0c0_0_5
- https://junyanz.github.io/CycleGAN/
- Blog by Luis Herranz: https://www.lherranz.org/2018/08/07/imagetranslation/

Topic 4: GANs Variations (Progressive GAN and StyleGAN)

Drawback 2 of standard GAN: low-quality outputs and instability during training



Generated samples of GAN

Image from: http://www.timzhangyuxuan.com/project_dcgan/

Progression of GAN’s capability

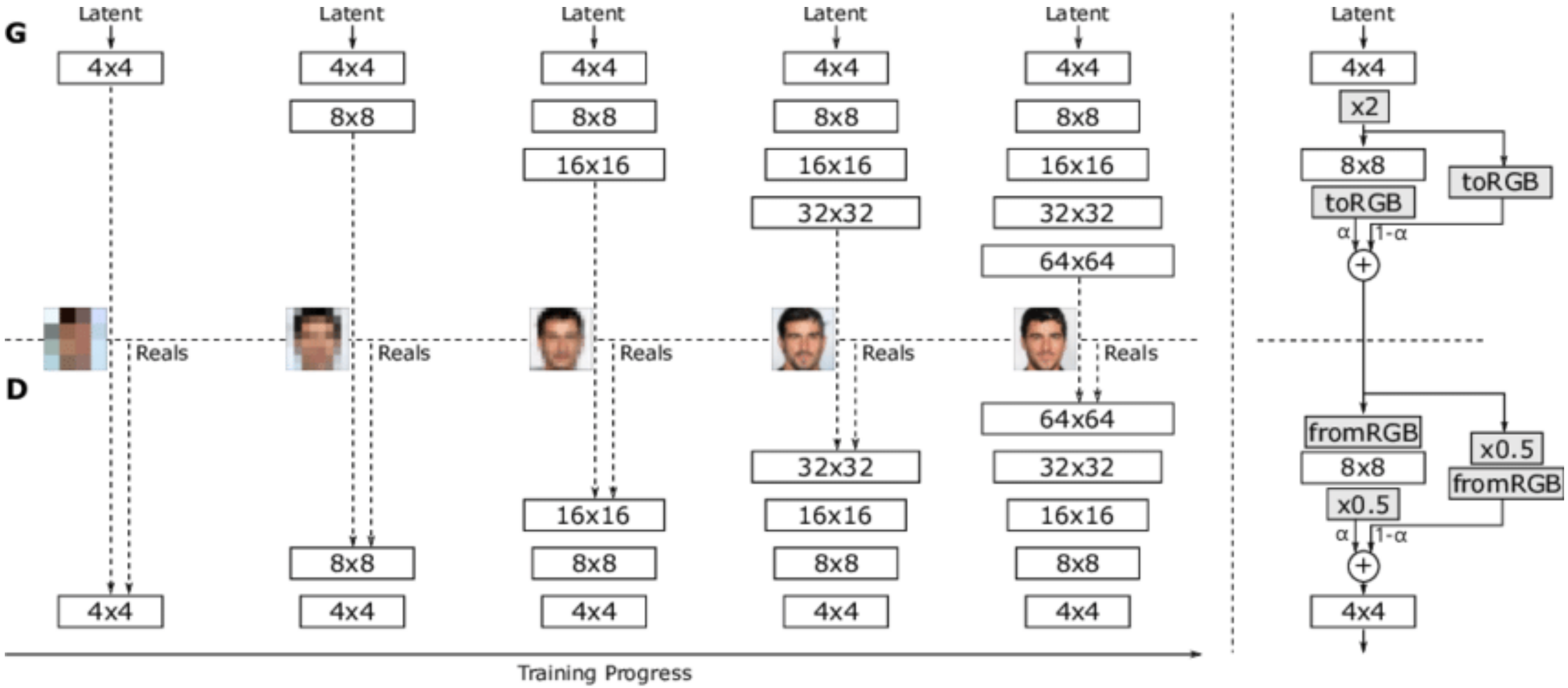
Generating high-resolution images has been considered difficult by GAN due to the unstable training.
But things changed since 2018 …

Progressive GAN

First discover large-scale structure of the image distribution,
then shift attention to increasingly finer scale details

Kerras et al. "Progressive growing of gans for improved quality, stability, and variation", ICLR 2018.

Progressive GAN [Karras-2018]



It starts with low-resolution images and then progressively increase the resolution by adding layers to the networks.

Kerras et al. “Progressive growing of gans for improved quality, stability, and variation”, ICLR 2018.

Progressive GAN [Karras-2018]

- Produce high resolution images
- Stabilize the training
- Speed up training

Kerras et al. “Progressive growing of gans for improved quality, stability, and variation”, ICLR 2018.

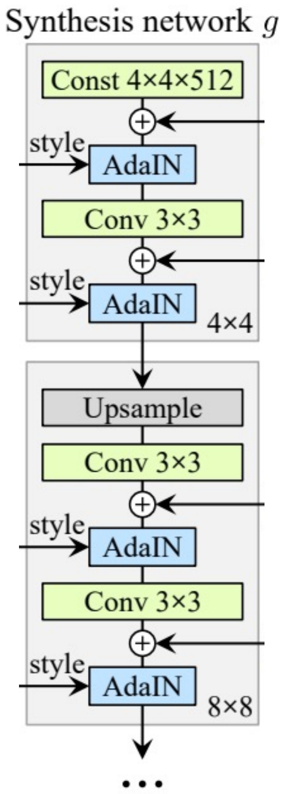
StyleGAN

Extension of progressive GAN, adding control of the generated image style.

Karras et al. “A style-based generator architecture for generative adversarial networks”, 2019.

StyleGAN [Kerras-2019]

Baseline Progressive GAN



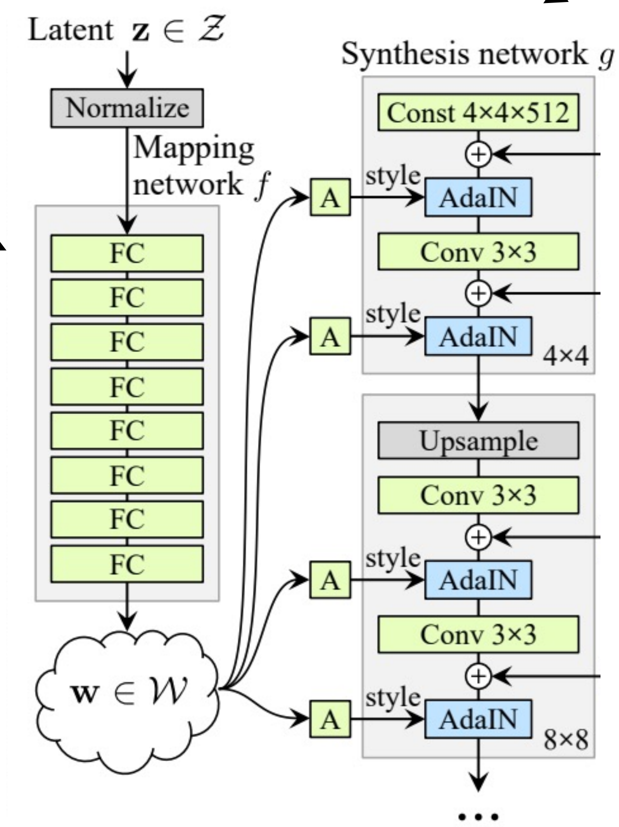
Style-based generator

Karras et al. “A style-based generator architecture for generative adversarial networks”, 2019.

StyleGAN [Kerras-2019]

Baseline Progressive GAN

Mapping network
outputs style vectors



Style-based generator

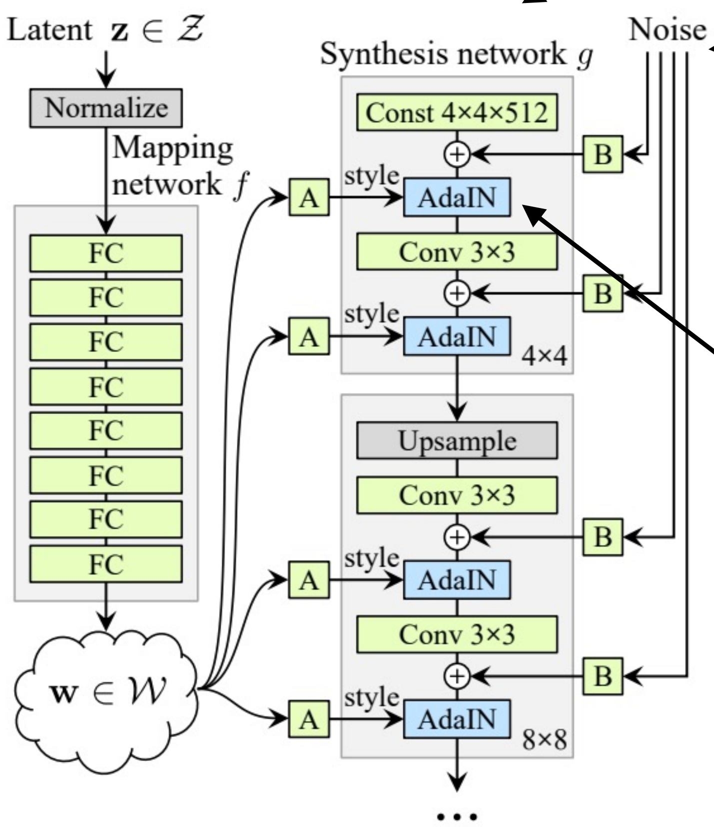
Karras et al. "A style-based generator architecture for generative adversarial networks", 2019.

StyleGAN [Kerras-2019]

Baseline Progressive GAN

Mapping network outputs style vectors

Adding noise: generate more stochastic details in the images.



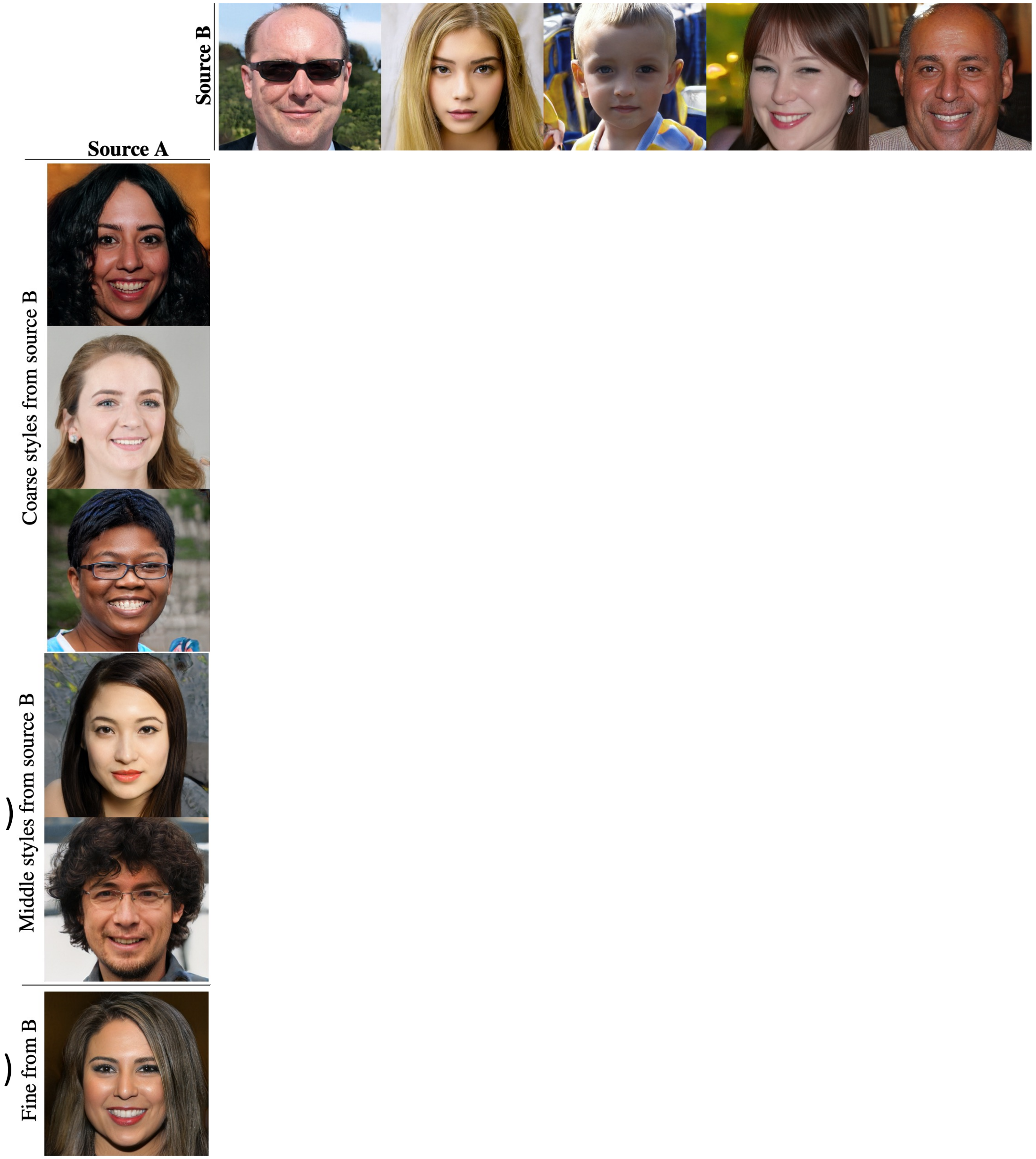
Style-based generator

Adaptive Instance Normalization (AdaIN) adds the style vector (y) to the normalized each feature map X_i.

$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i}$$

Karras et al. "A style-based generator architecture for generative adversarial networks", 2019.

StyleGAN



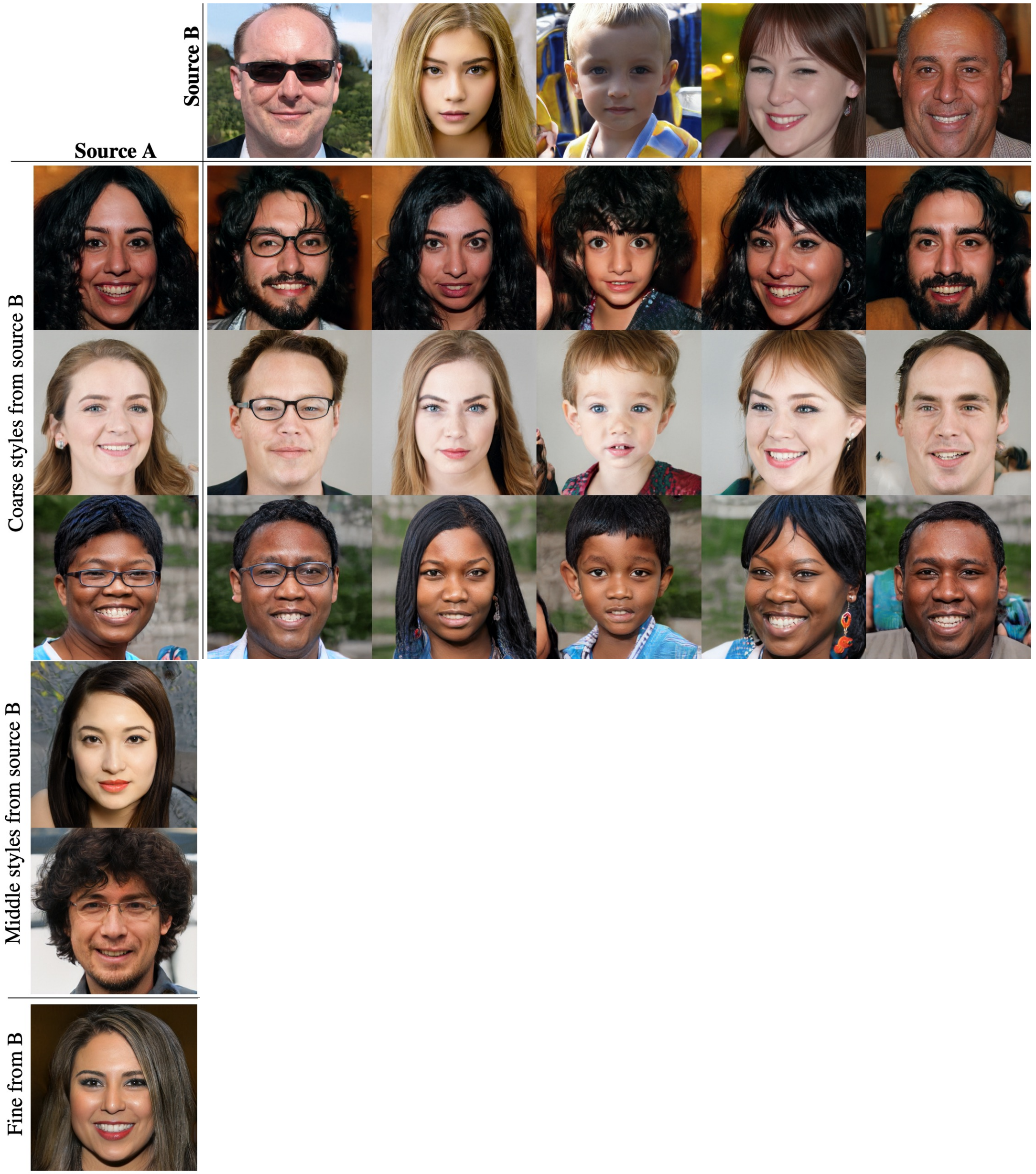
Coarse resolutions (4^2– 8^2)

Middle resolutions ($16^2 - 32^2$)

Fine resolutions ($64^2 - 1024^2$)

UNIVERSITY OF BATH

StyleGAN



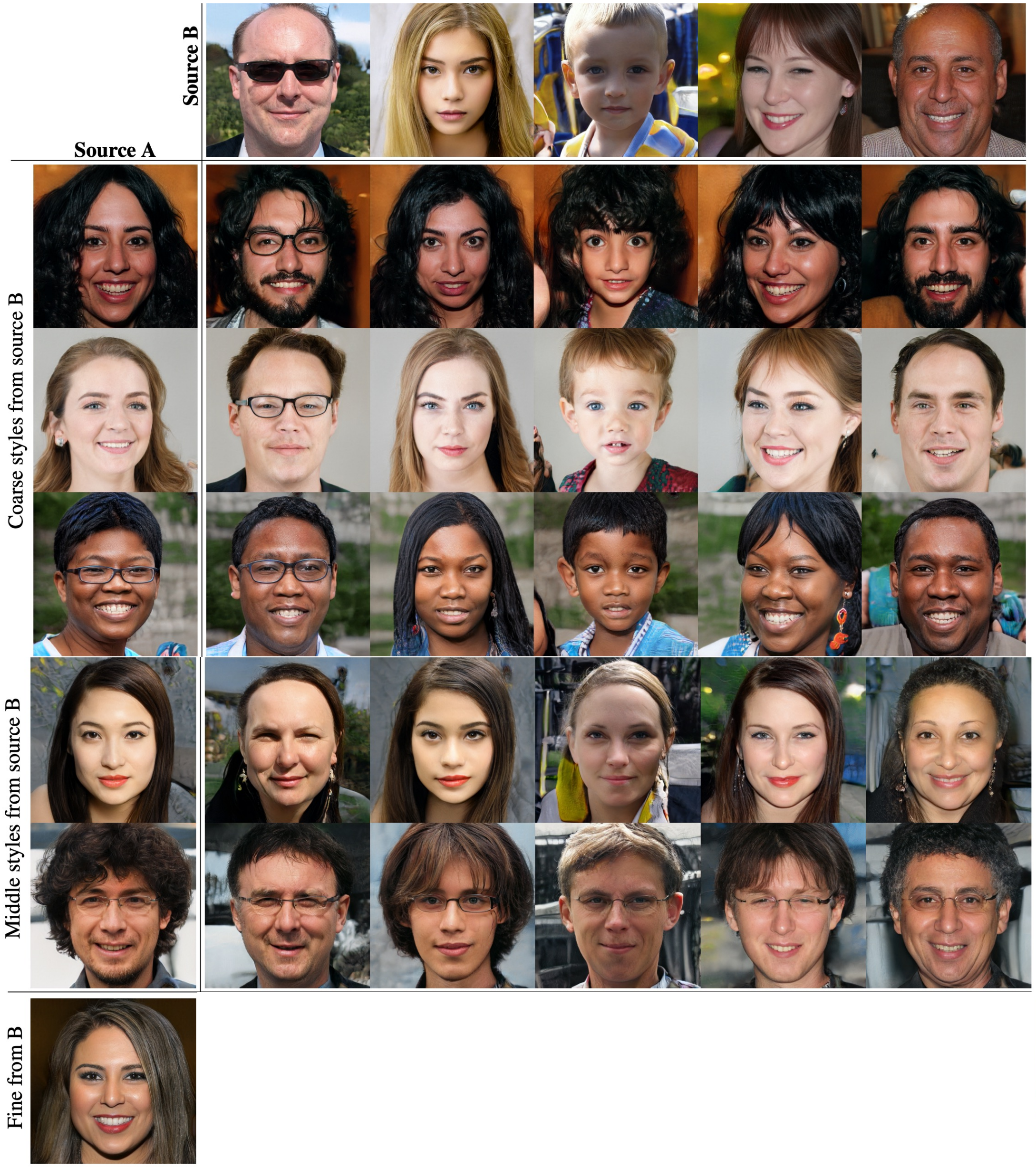
Coarse resolutions ($4^2 – 8^2$)

Middle resolutions ($16^2 – 32^2$)

Fine resolutions ($64^2 – 1024^2$)

Pose, hair style, face shape, gender, age, glasses

StyleGAN



Coarse resolutions ($4^2 - 8^2$)

Middle resolutions ($16^2 - 32^2$)

Fine resolutions ($64^2 - 1024^2$)

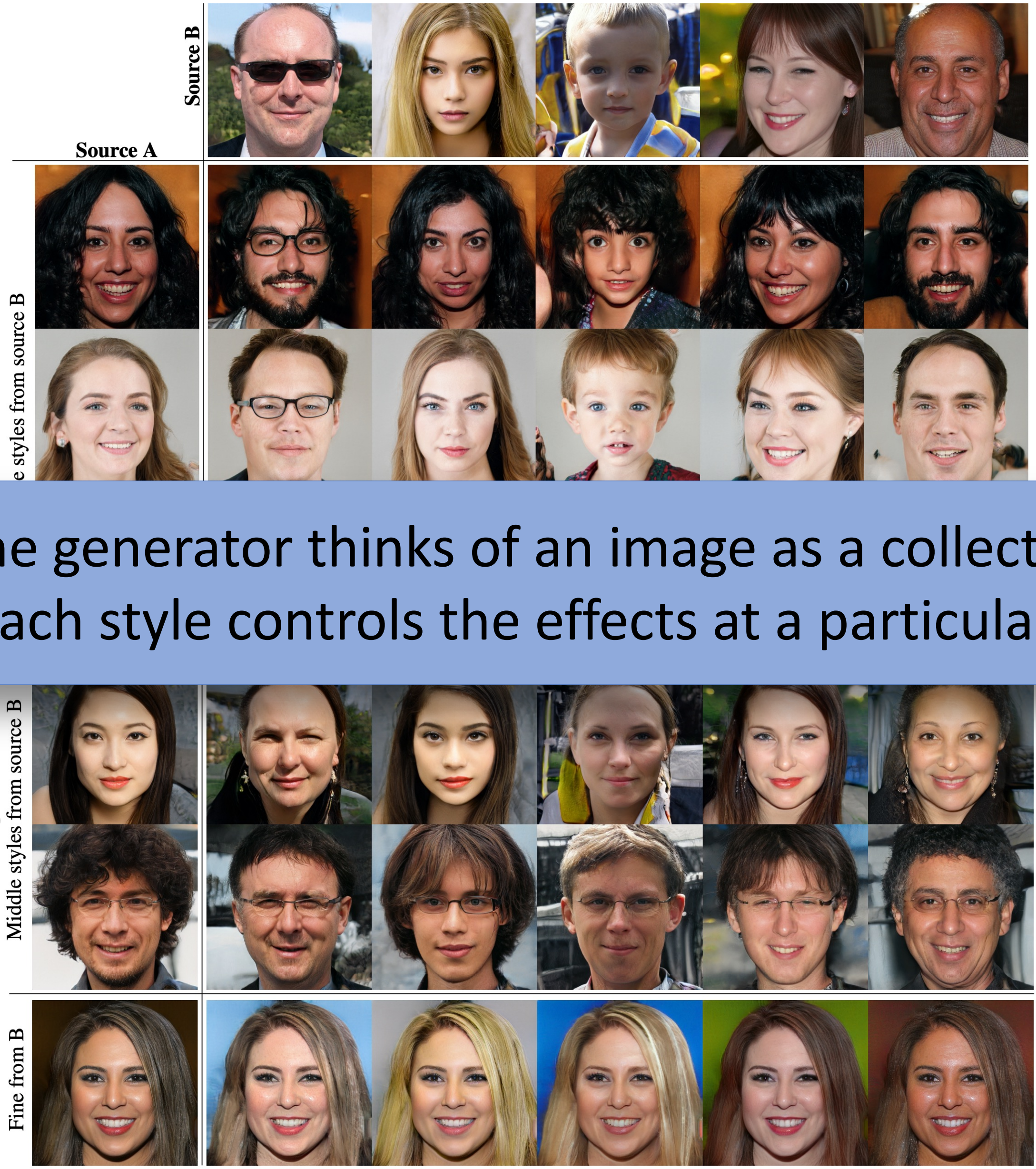
UNIVERSITY OF BATH

Pose, hair style, face shape, gender, age, glasses

Facial features, eyes

UNIVERSITY OF BATH

StyleGAN



Coarse resolutions (4^2– 8^2)

Middle resolutions ($16^2 - 32^2$)

Fine resolutions ($64^2 - 1024^2$)

Pose, hair style, face shape, gender, age, glasses

Facial features, eyes

Color scheme

In StyleGAN, the generator thinks of an image as a collection of "style", where each style controls the effects at a particular scale.

Reference

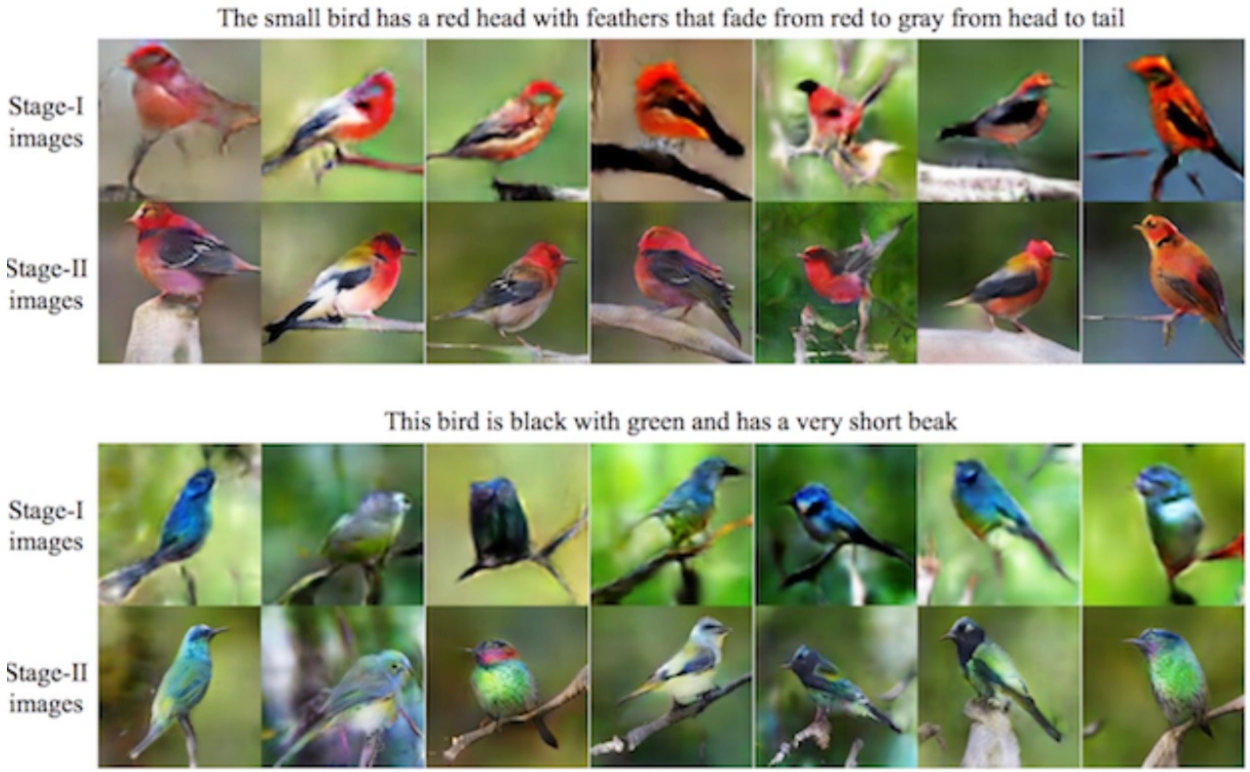
- Lecture on Generative Adversarial Networks: https://developers.google.com/machine-learning/gan/applications
- YouTube Video by Ava Soleimany: MIT course on Deep Generative Models.
- Lecture by Rowel Atienza: https://docs.google.com/presentation/d/13fiFibqjl9ps_CktJzMNAvoZXOlzHQDu8eRSb3a227g/edit#slide=id.g389666c0c0_0_5
- Blog by Joson Brownlee: https://machinelearningmastery.com/introduction-to-style-generative-adversarial-network-stylegan/

Topic 5: More GANs Applications

Text-to-Image Synthesis

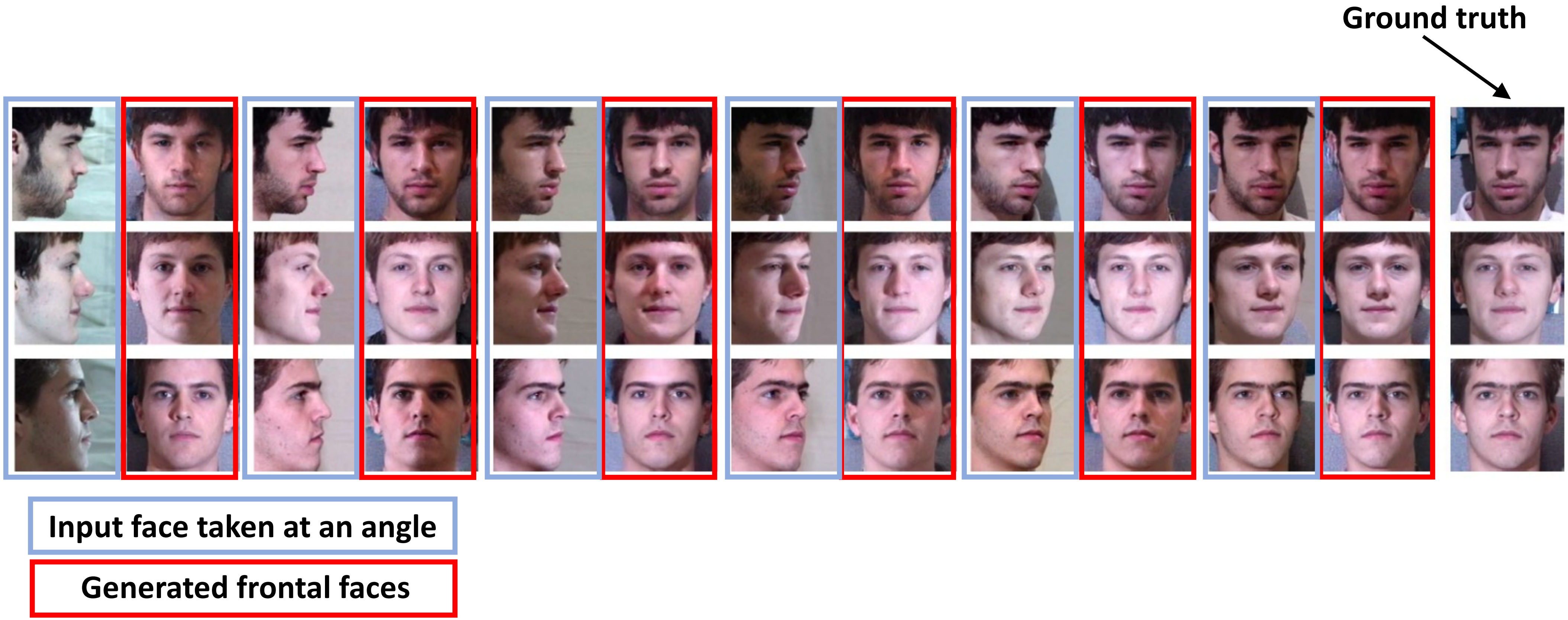
Input Text: A stained glass window with
an image of a blue strawberry

StackGAN [Zhang-ICCV17]



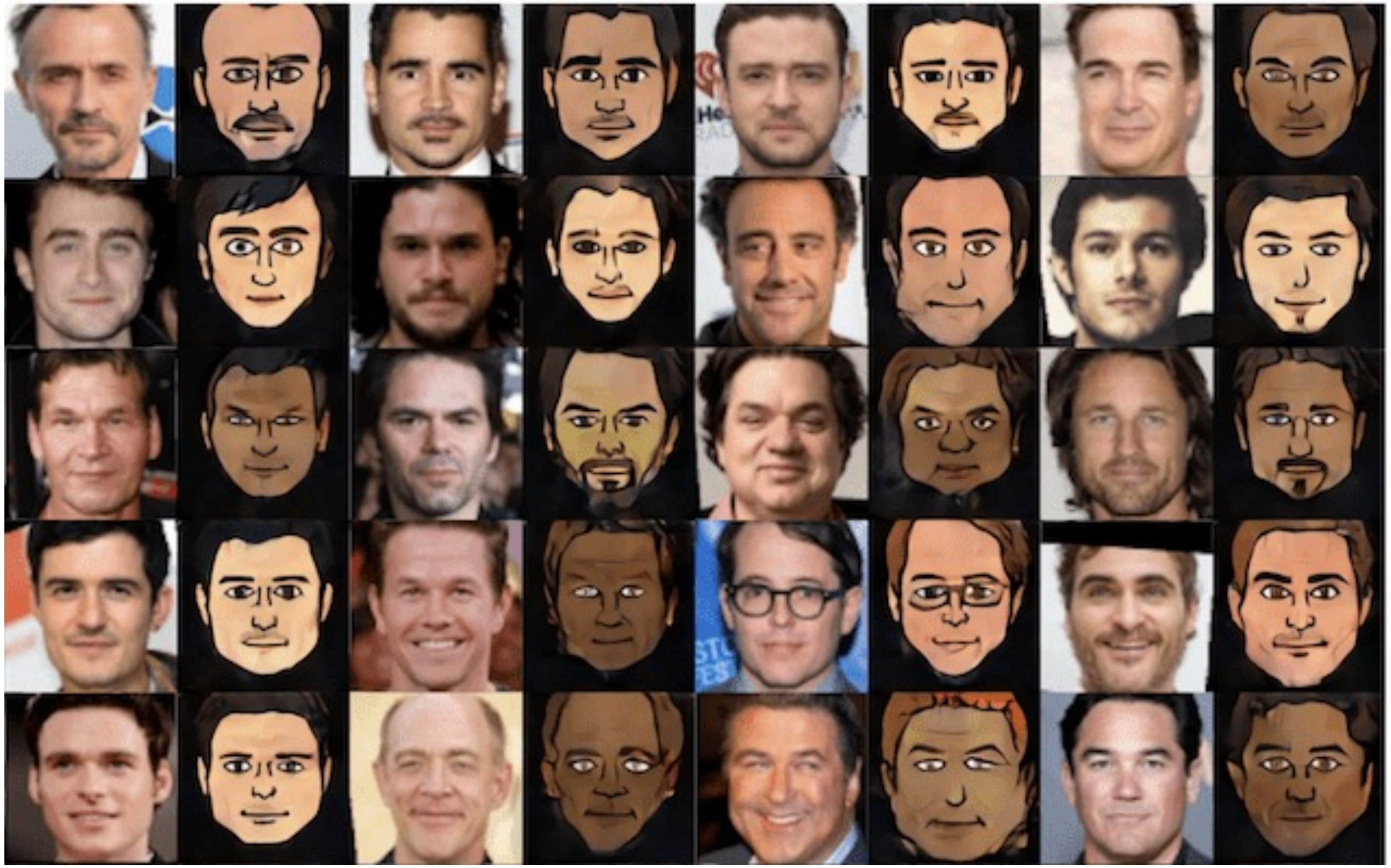
Reed et al. "Generative adversarial text to image synthesis". ICML 2016
Han Zhang et al. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks , ICCV 2017.

Face Frontal View Generation



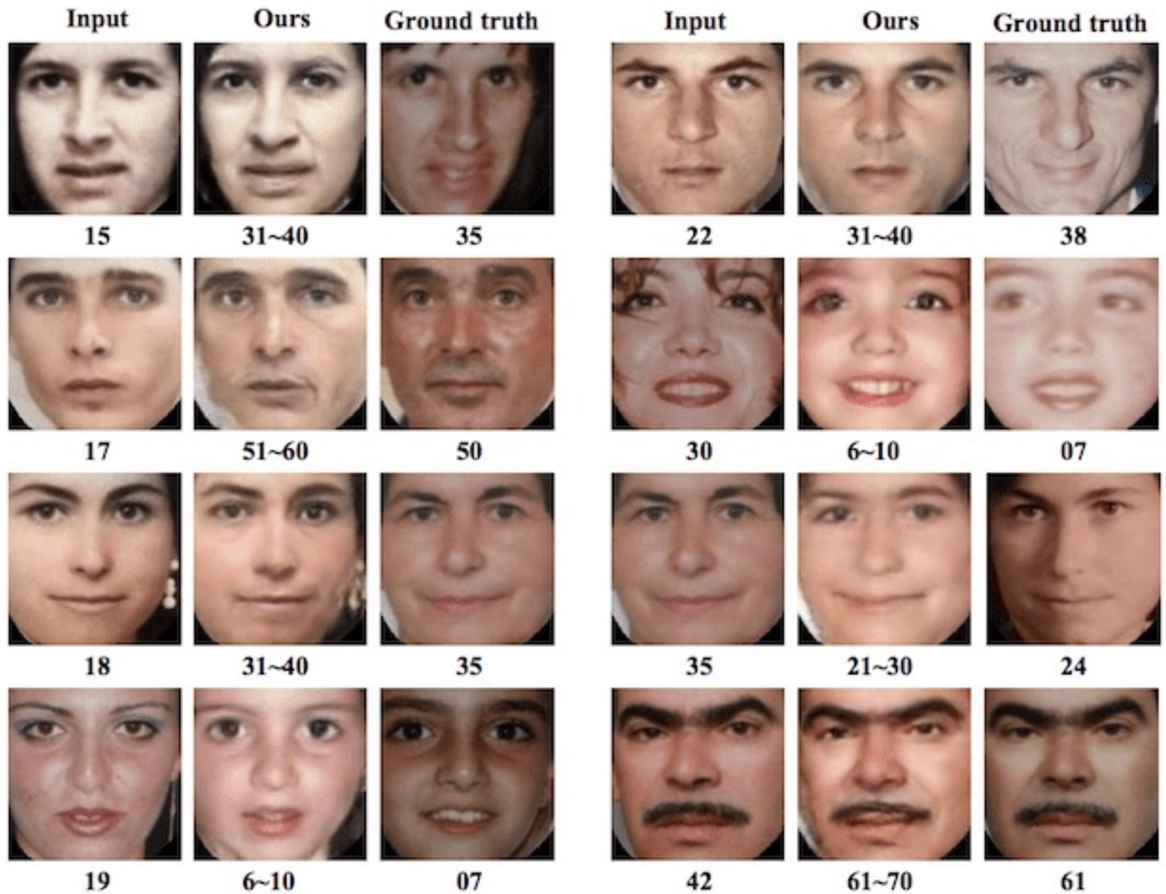
Rui Huang, et al. in their 2017 paper titled “Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis” ICCV 2017

Photos to Emojis



Taigman, et al. Unsupervised cross-domain image generation, 2016.

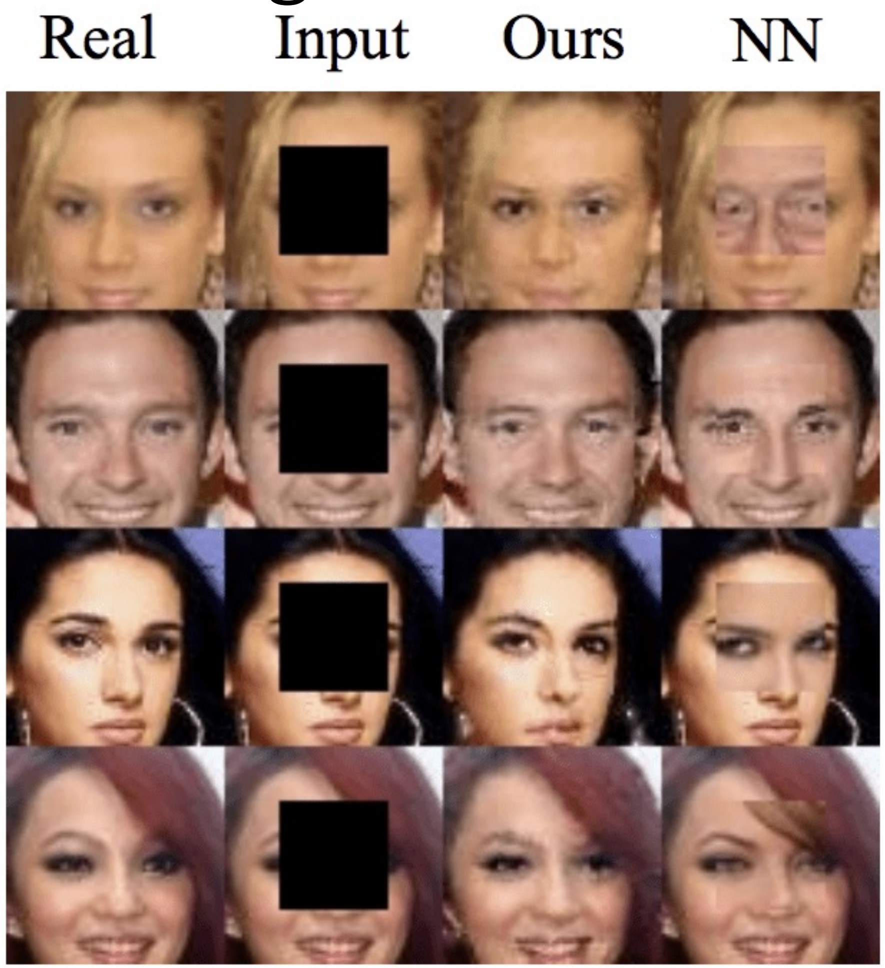
Face Aging



ANtipov et. al. Face Aging With Conditional Generative Adversarial Networks, 2017

Zhang et. al. Age Progression/Regression by Conditional Adversarial Autoencoder, 2017.

Photo Inpainting



Yeh et. al. “Semantic Image Inpainting with Deep Generative Models”, CVPR 2017

3D Object Generation

Wu et. al. “Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling”, 2016

Video prediction

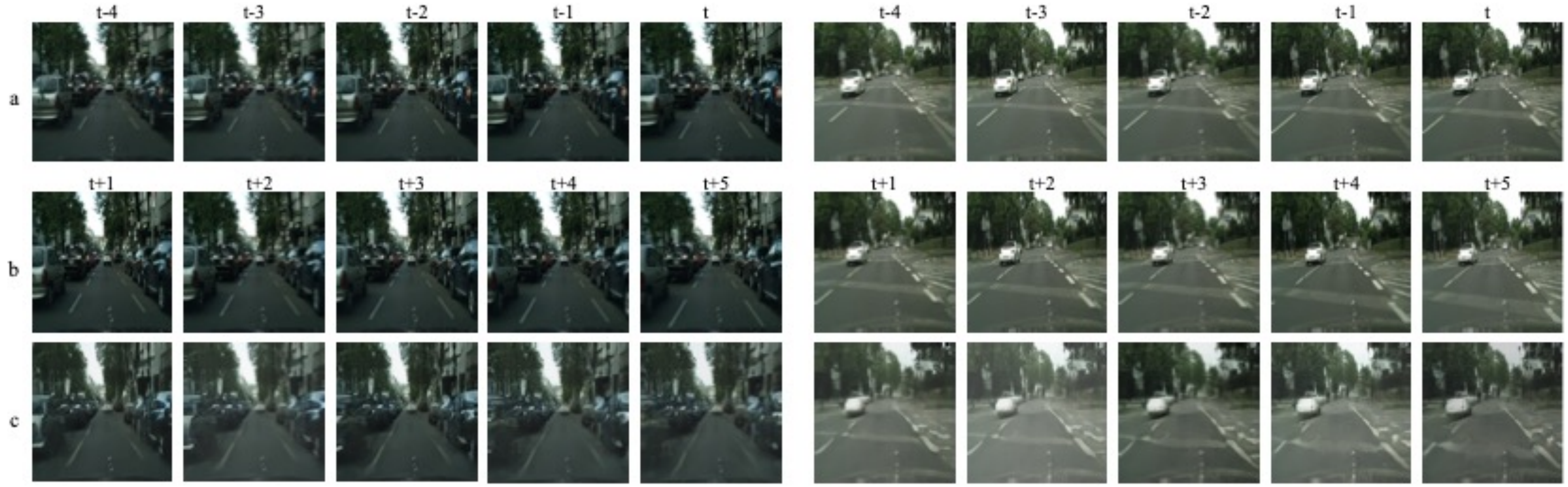


Figure 6: Prediction results for the Cityscapes test sequences. a: Input, b: Ground Truth, c: FutureGAN (ours)

Aigner et. al. FutureGAN: Anticipating the Future Frames of Video Sequences using Spatio-Temporal 3d Convolutions in Progressively Growing GANs. 2018.

GAN is a new discovered land.

Looking for treasures has never been stopped …

Reference

- Blog by Jason Brownlee: 18 Impressive Applications of GANs. https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/
- Blog by Martin Isaksson: Five GANs for Better Image Processing. https://towardsdatascience.com/five-gans-for-better-image-processing-fabab88b370b
- Noshory: https://github.com/nashory/gans-awesome-applications