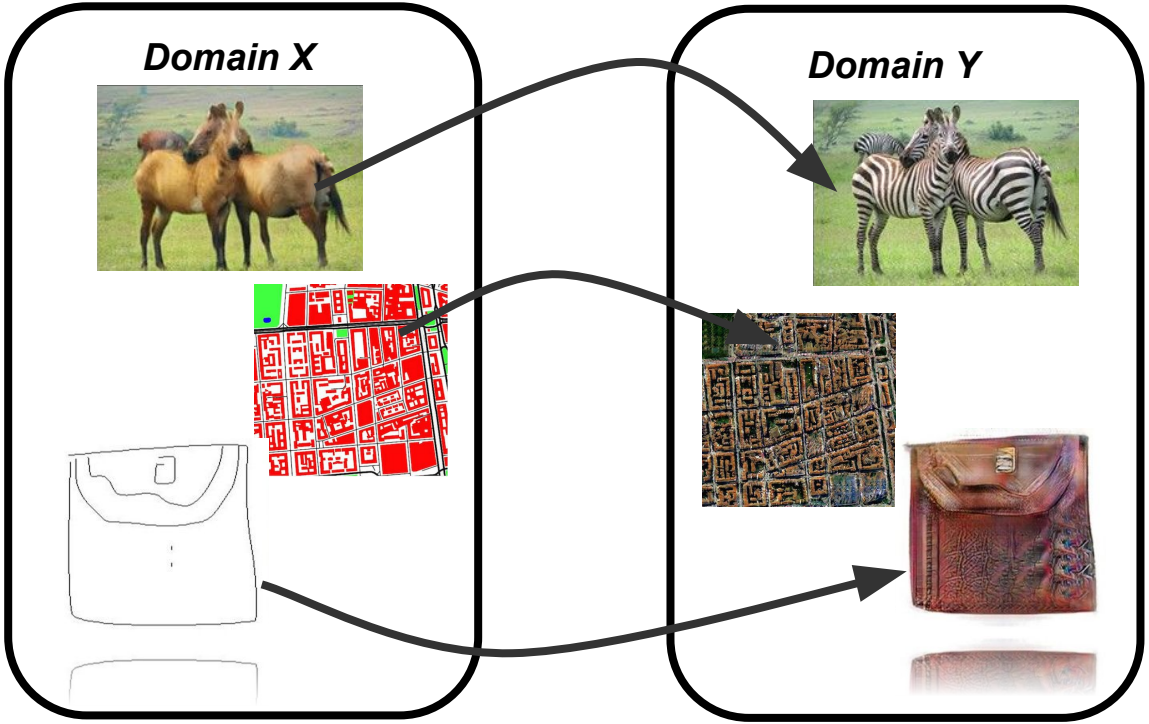
Domain Transfer

Weizmann Institute of Science
Kenny Green | Rotem Shaul | Chang Liu
July 2018

The World of Domain Transfer



Translates from a **source domain *X*** to **target domain *Y***

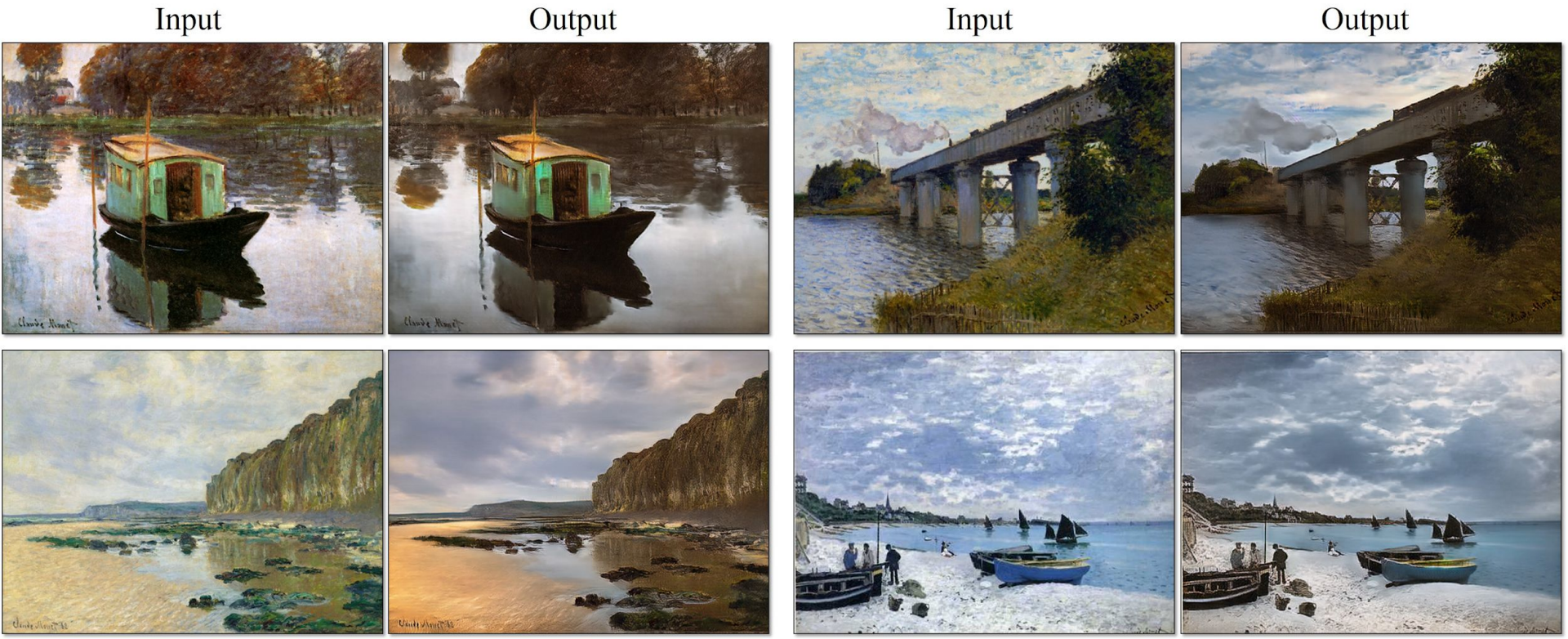
Learn a non-linear mapping function **G: X → Y**
where Y is indistinguishable from G(X)

Image-to-Image Translation with Conditional Adversarial Networks, Isola et al., Nov 2016 **[Pix2Pix]**
Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, Zhu et al., Mar 2017 **[CycleGAN]**

Applications: Style Transfer

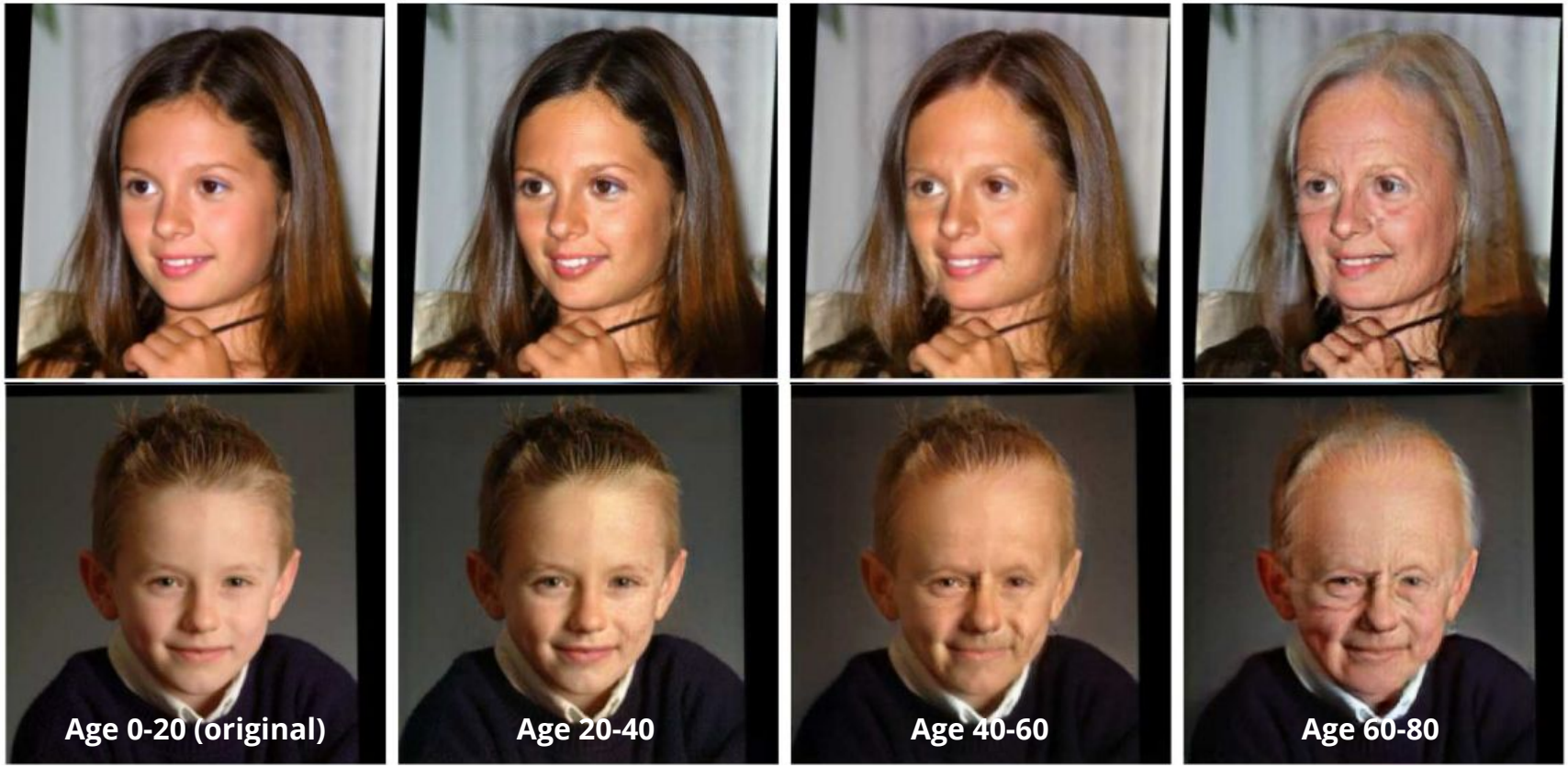
Implementing Neural Artistic Style Transfer, Josh Whittle, Apr 2017

Applications: Style Transfer



Images from [Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks](), Jun-Yan Zhu et al. ICCV 2017

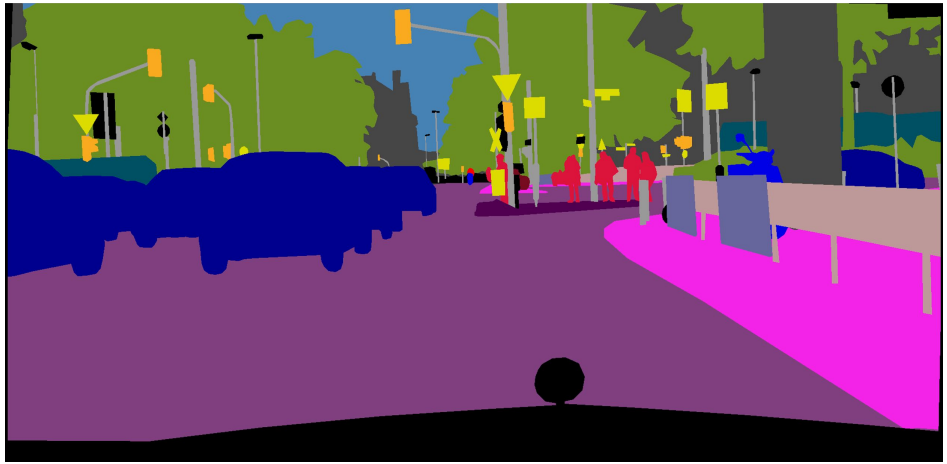
Applications: Face Aging



Generative Adversarial Style Transfer Networks for Face Aging, CVPR, June 2018

Applications: Generate Images from Segment Labels

Input labels



Synthesized image

High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs, Wang et al., Nov 2017

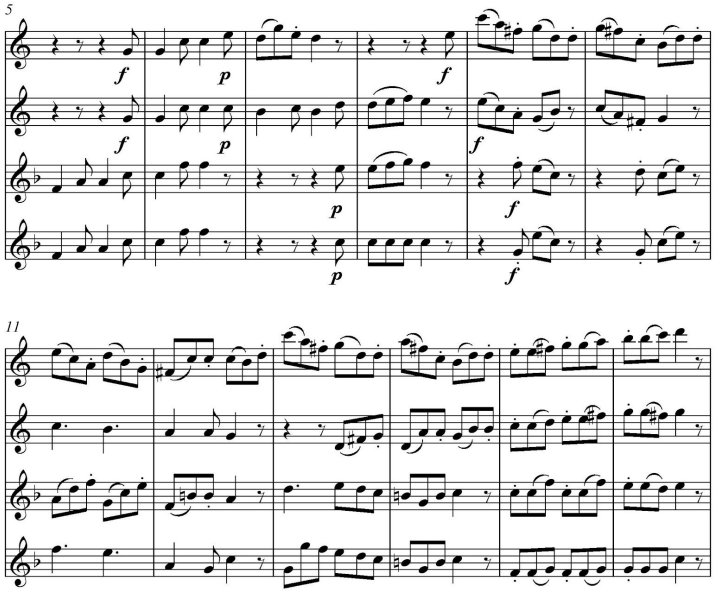
Application: Music Domain Translation

The following audio samples were transferred between:

- Symphony, Mozart
- String Quartet, Haydn
- Cantata (Chorus opera & Orchestra), Bach
- Organ, Bach
- Piano, Bach
- Harpsichord, Bach

- ❖ Electric Guitar, Charlie Christian
- ❖ Electric Guitar, Metallica
- ❖ Classical Guitar & Orchestra , Jazz
- ❖ Trumpet & Orchestra, Jazz
- ❖ Midi samples of Piano & Trumpet, Elvis Presley, Rihanna
- ❖ Music of Africa
- ❖ Whistling (Human)

- In training distribution
- ❖ Out of training distribution



link to demonstration: https://www.youtube.com/watch?v=vdxCqNWTpUs

A Universal Music Translation Network, Mor, Wolf, Polyak, Taigman, May 2018

AGENDA

Image to Image translation papers: as of mid-June 2018, there have been 30+ related papers since original Pix2Pix paper

Paired vs Unpaired Training

Recap on GANs

The Supervised Learning Approach

- Baseline (Pix2Pix)
- Multimodal Approach (BiCycleGAN)

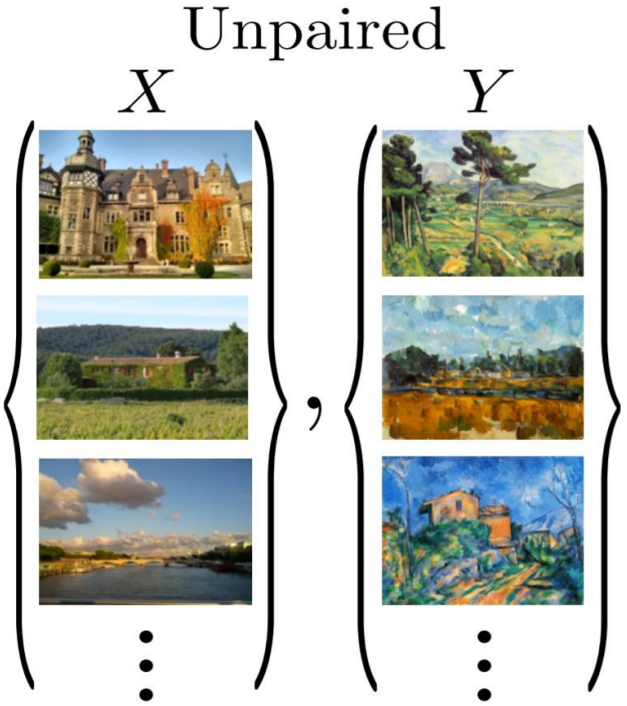
Unsupervised Learning Approach

- Baseline (CycleGAN)
- Adding Constraints (DistanceGAN)

Conclusion + Q&A

Paired vs Unpaired

Supervised Learning



Unsupervised Learning

Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, Zhu et al., Mar 2017 **[CycleGAN]**

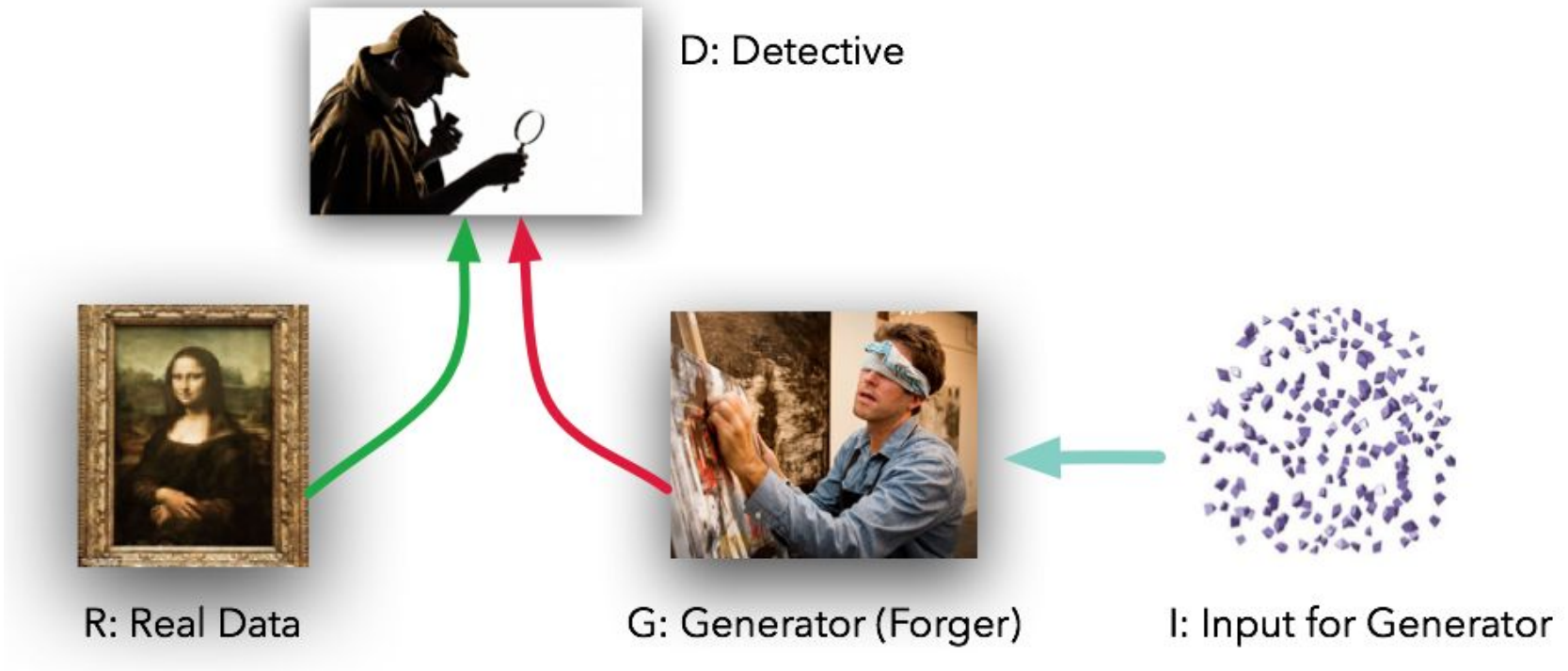
Domain Transfer: Why GANs?

Naive approach: CNNs minimise Euclidean distance between predicted and ground truth pixels
→ *blurry results*
→ *naturalness/realistic not a requirement*
(small subset of all possible generatable images)

Desire for high level goal: *"make output indistinguishable from reality"* → GAN: ideal solution!

"Changing lead into gold. Why do you ask?"

Recap: Conceptualising GANs



Picture from [Generative Adversarial Networks (GANs) in 50 lines of code](), DevNag, February 2017

Recap: Generative Adversarial Network (GAN)

Discriminator trained to discriminate between images *x* from sample space and images outside the *sample space*

Generator trained to generate images that *look like* its from the *sample space*

Train based on the output from D

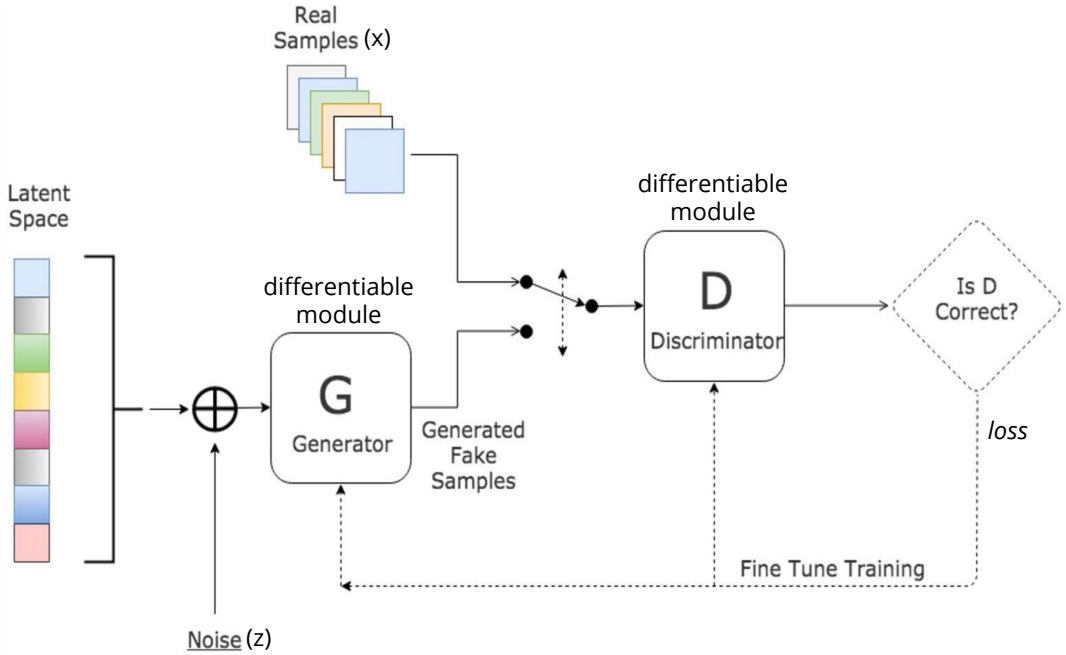


Figure reproduced from blog GAN: When Deep Learning Meets Game Theory, Ibrahim, January 2017

Recap: Training the Discriminator

Lock **generator**

Backpropagate error to update **discriminator** weights

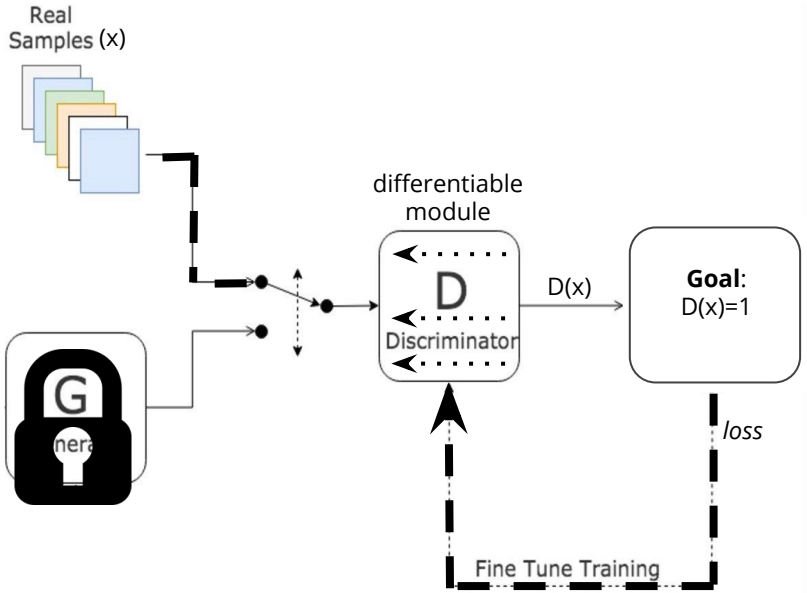
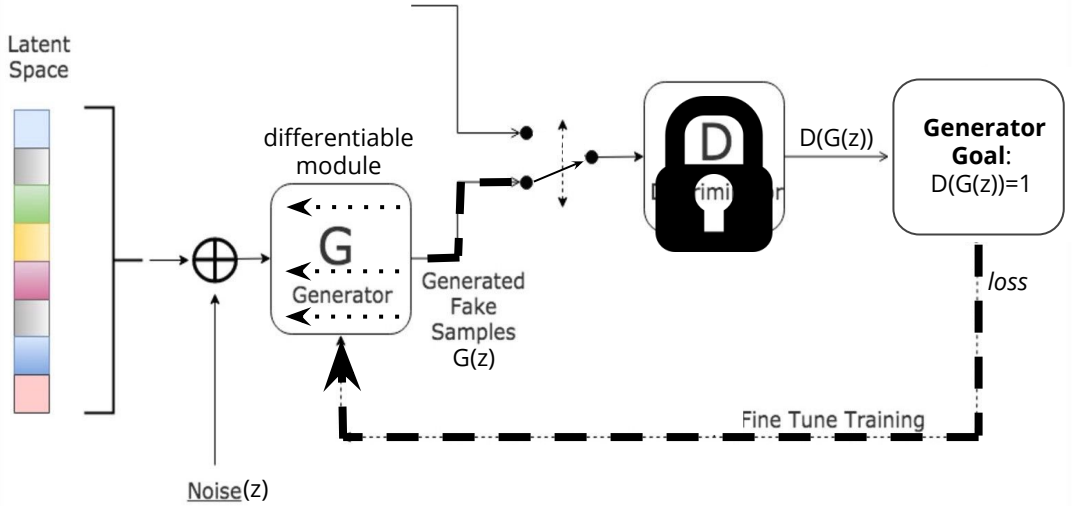


Figure partially reproduced from blog GAN: When Deep Learning Meets Game Theory, Ibrahim, January 2017

Recap: Training the Generator

Lock **discriminator**

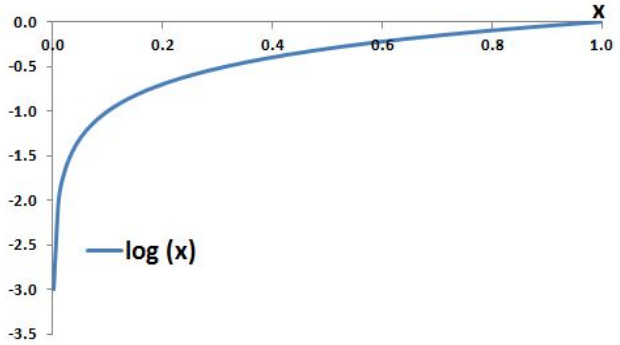
Backpropagate error to update **generator** weights



Visualisation of GAN training:
https://youtu.be/US2PsDhC658

Figure partially reproduced from blog **GAN: When Deep Learning Meets Game Theory**, Ibrahim, January 2017

Generalized GAN Loss Function



Loss fn of the discriminator

loss = 0 for a perfectly trained discriminator and highly +ve when untrained

log of the **probability that real image is rated highly**
negative=miss ... 0=perfect hit

Discriminator fn
0=fake ... 1=real

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\left[\log D(\boldsymbol{x})\right] - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}}\left[\log\left(1 - D\left(G(\boldsymbol{z})\right)\right)\right]$$

Real Sample

Generated Sample

log of the **probability that fake generated image is rated poorly**
negative=miss .... 0=perfect hit

Loss fn of the generator is the opposite of the discriminator

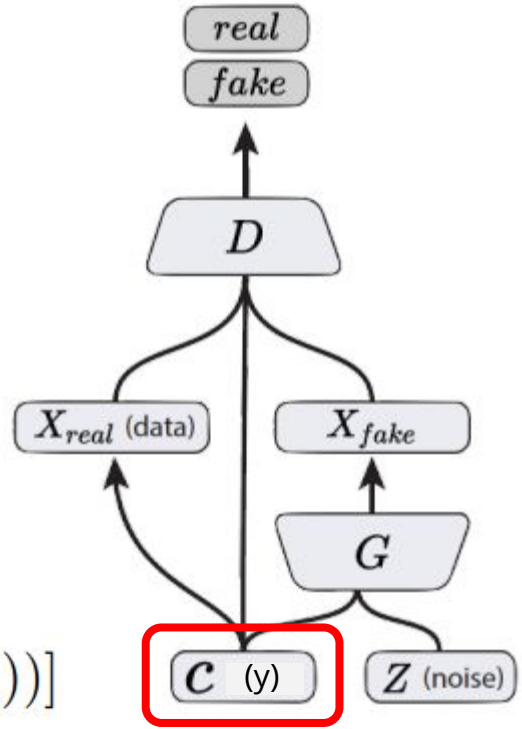
$$J^{(G)} = -J^{(D)}$$

Conditional GANS

Modification: condition generator & discriminator on additional information → better multi-modal learning

Provides **defined constraints** on generated output

Leads to many practical applications when **explicit supervision** is available



Objective loss fn

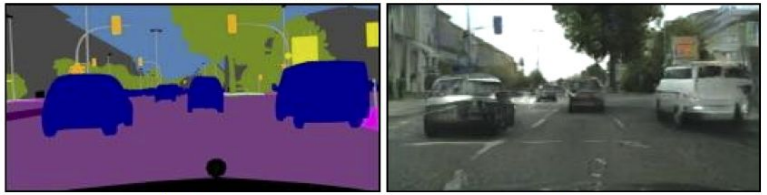
$$\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))]$$

Conditional GANs, Mirza & Osindero, Nov 2014

The Basic Principles of Domain Transfer

Pix2Pix ⟶ Image to Image Translation

Labels to Street Scene

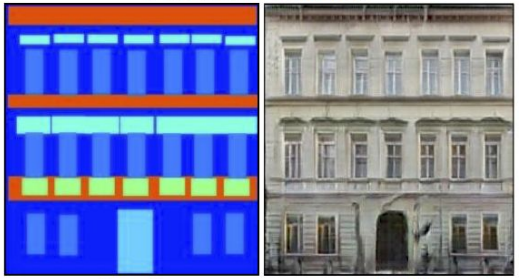


input output

Aerial to Map

input output

Labels to Facade



input output

Day to Night

input output

BW to Color

input output

Edges to Photo

input output

[Pix2Pix demo](#) online; [Image-to-Image Translation with Conditional Adversarial Networks](#), Isola et al., Nov 2016

Example: Learning to Color from Paired Datasets

Dataset:

INPUT (x) TARGET (y)

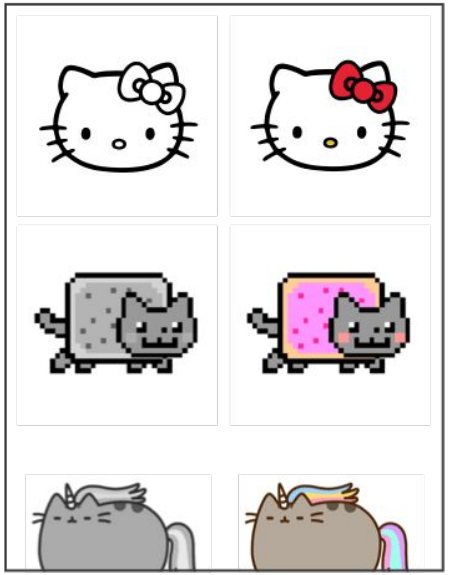
INPUT

OUTPUT

GROUND TRUTH

Consider the dataset:

INPUT TARGET



Pix2Pix - Objective Function

Objective loss fn

input image

the condition in CGAN

target training image

generated output image

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z))]$$

typical CGAN objective fn

input image

random vector

Additional L1 objective - to enforce low frequency correctness

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

tasks generator to be near ground truth using L1-norm distance

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

final objective fn

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

Pix2Pix Algorithm - Step 1: Train Discriminator

- Generator **generates output image**
- Discriminator looks at input/target pair & input/output pair → **produces guess on realism** on each
- **Weights of discriminator adjusted** based on classification error of both combined

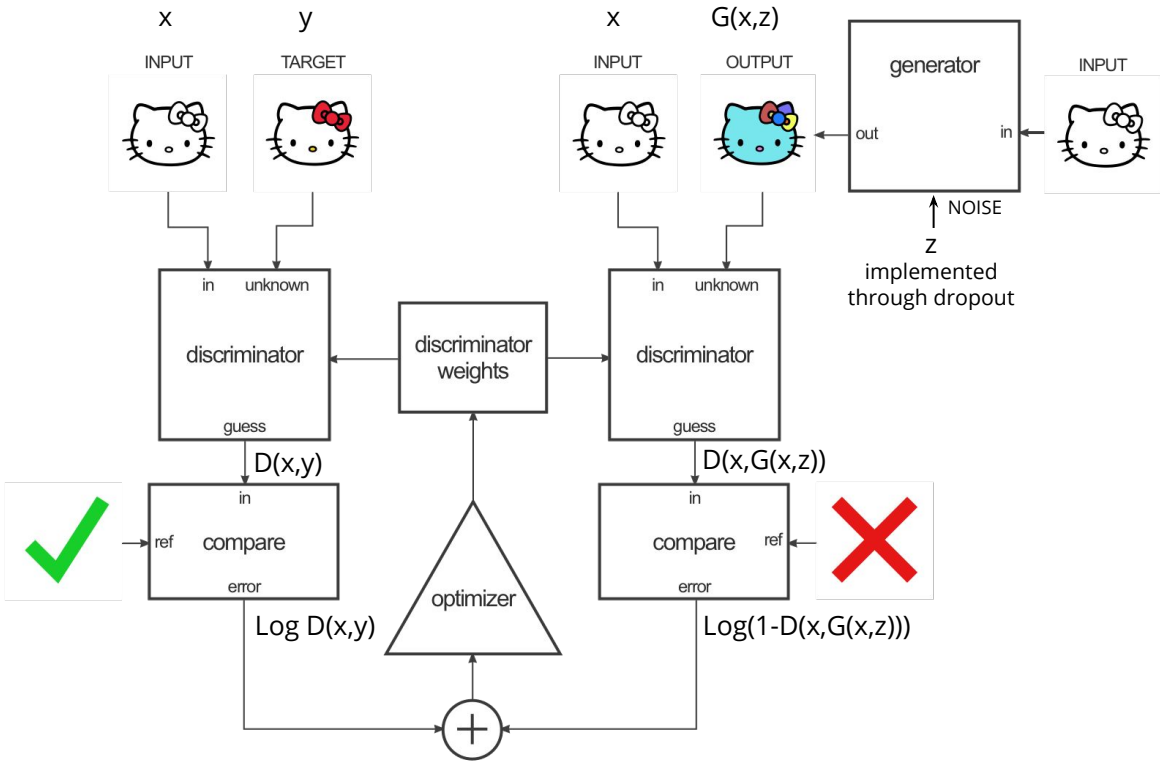
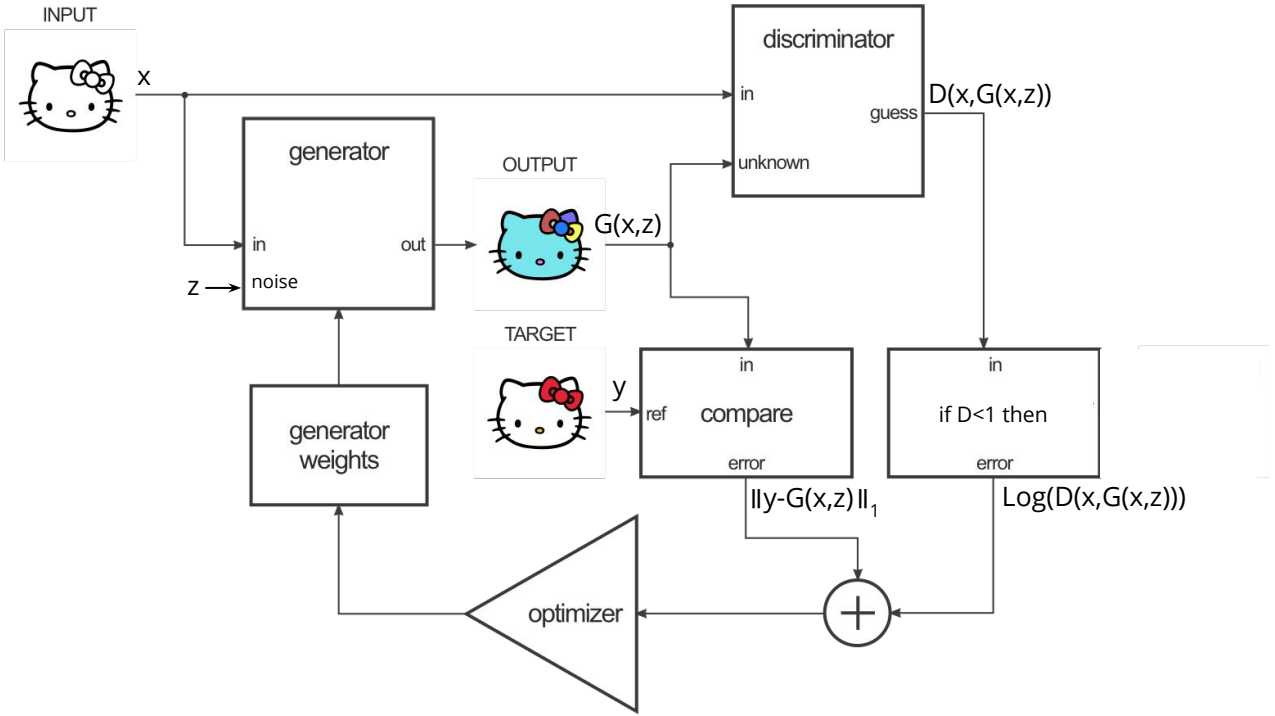


Image reproduced from [Image-to-Image Translation in TensorFlow](), Hesse, Jan 2017

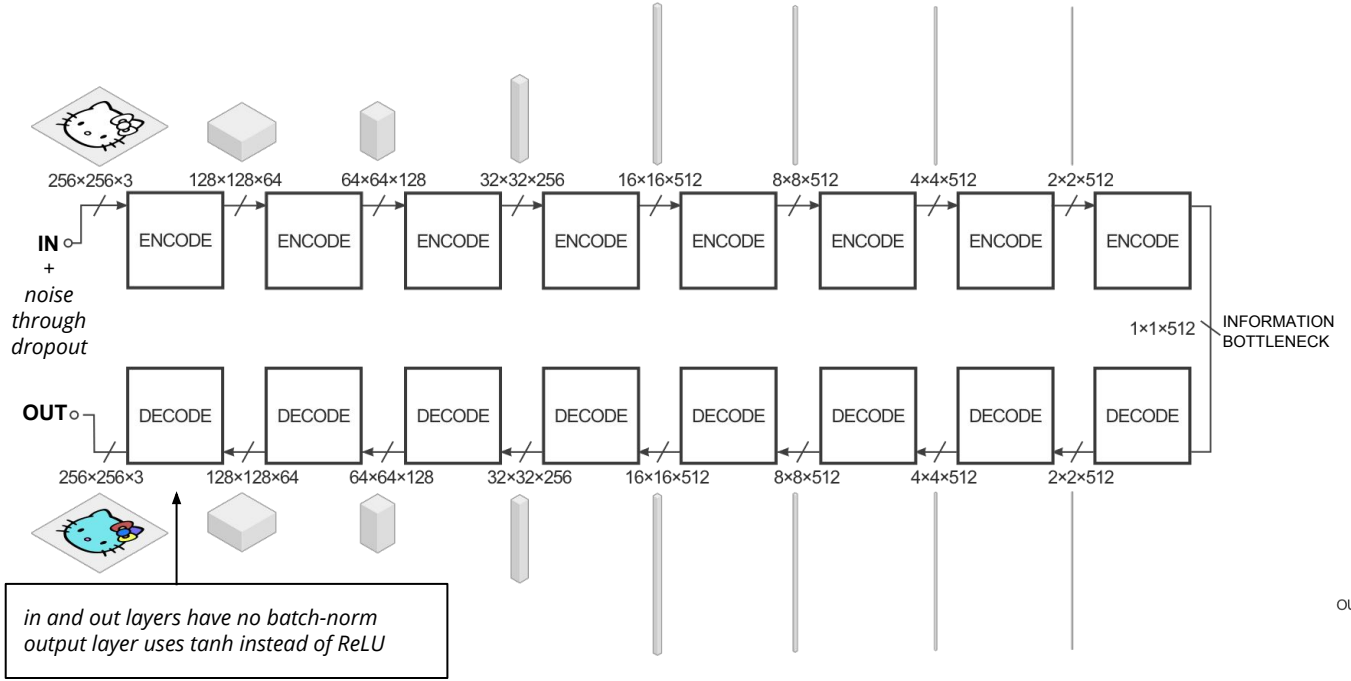
Pix2Pix Algorithm - Step 2: Train Generator

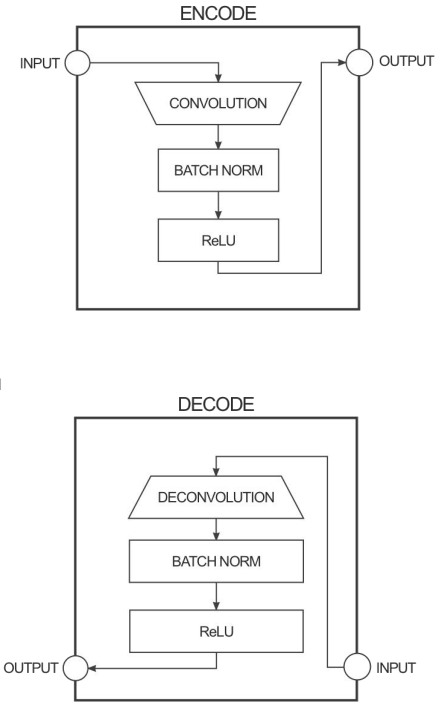
Generator's weights adjusted based on output of discriminator as well as difference between output & target image



Underlying image reproduced from [Image-to-Image Translation in TensorFlow](), Hesse, Jan 2017

Generator Architecture: based on DCGAN

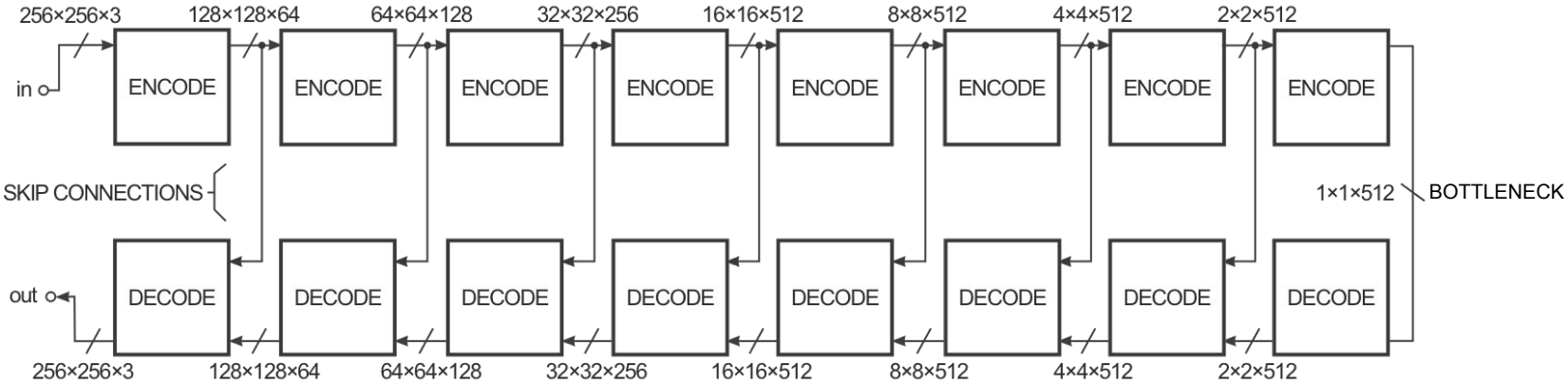




Unsupervised Representation Learning with Deep Convolutional GANs, Radford, Jan 2016

Image reproduced from Image-to-Image Translation in TensorFlow, Hesse, Jan 2017

Improve Performance: U-Net



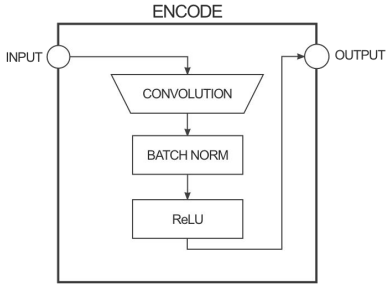
Skip connections **concatenates** all channels in layer *i* with those of at layer *n-i*

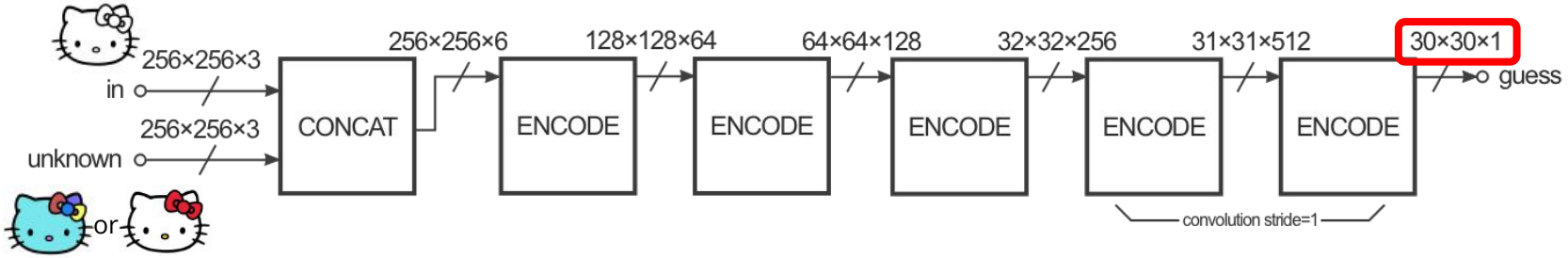
Great deal of **low-level information** shared by in + out layers (e.g. edges): pass image details from encode layers to decode layers enabling better training and image recovery

U-Net: CNN for Biomedical Image Segmentation, Ronneberger et al., May 2015

Image reproduced from Image-to-Image Translation in TensorFlow, Hesse, Jan 2017

Discriminator Architecture + PatchGAN





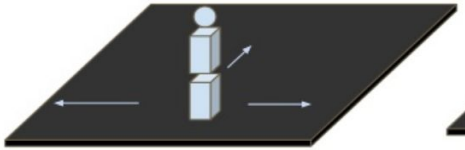
Patch is 30x30 matrix of pixel values (0~1) → each indicates how **believable** corresponding patch of unknown image is

Each pixel corresponds to overall believability of **70x70 patch** (patches overlap, receptive fields padded with zeros)

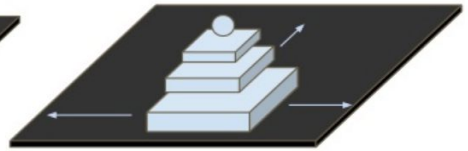
This discriminator is run convolutionally across image, **average all responses** to provide ultimate **D**

Image reproduced from [Image-to-Image Translation in TensorFlow](), Hesse, Jan 2017

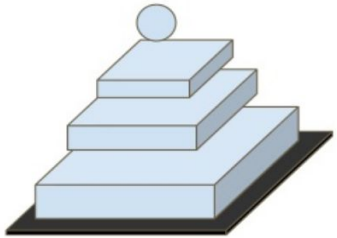
... PixelGAN vs PatchGAN vs ImageGAN



PixelGAN judges each pixel individually and independently



PatchGAN only penalises structure at the scale of local patches → focuses on higher frequency artifacts



ImageGAN takes into account features across the whole image

Original PatchGAN paper: [Precomputed real-time texture synthesis with markovian generative adversarial networks](), Li & Wand,, Apr 2016

Image reproduced from [Image-to-Image Translation with Conditional Adversarial Nets (UPC Reading Group) presentation](), Victor Garcia, Nov 2016

Measuring Results: AMT & FCN-Score

Subjective: amazon mechanical turk Artificial Artificial Intelligence

For each algorithm, 50 *turkers* presented with series of trials pitting real image against fake for 1 second

Turkers select which they think is real

First 10 trials: practise with feedback
Next 40 trials: real with no feedback

Objective: FCN Score (using **FCN-8s** architecture)

Pseudo-metric: Uses pre-trained semantic classifiers from real images (*e.g. streetview dataset*) to measure discriminability of the generated images

Intuition: if generated images are realistic, classifiers trained on the equivalent real images will be able to classify the synthesised image correctly as well

Fully Convolutional Networks for Semantic Segmentation, Long & Shelhamer, 2015

Results: PixelGAN vs PatchGAN vs ImageGAN

1×1

Encourages color diversity but no effect on spatial sharpness

16×16

Locally sharp results with tiling artifacts beyond the scale it can observe

286×286

Visually similar to 70×70, but lower FCN-score

70×70

Sharp outputs, even if incorrect, in the spatial and color dimensions

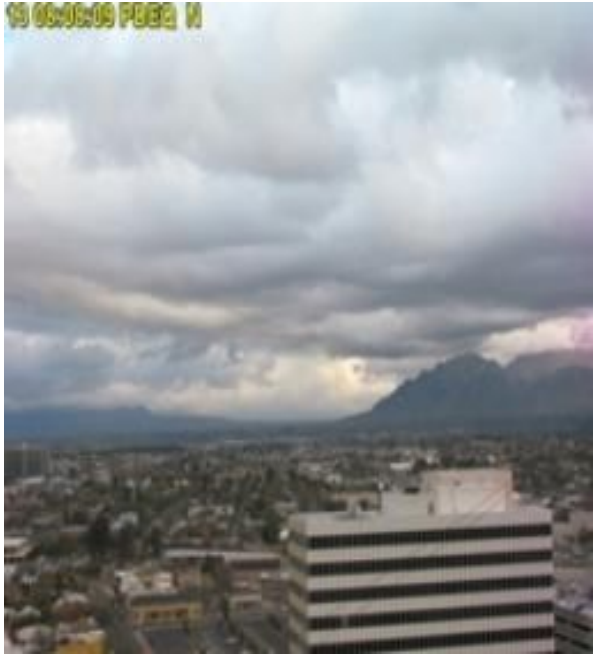
FCN-scores for various receptive field sizes of the discriminator

Discriminator receptive field	Per-pixel acc.	Per-class acc.	Class IOU*
1×1	0.39	0.15	0.10
16×16	0.65	0.21	**0.17**
70×70	**0.66**	**0.23**	**0.17**
286×286	0.42	0.16	0.11

**intersection over union of bounding boxes evaluated on Cityscapes labels → photos*

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

Results: Day to Night



GROUND TRUTH

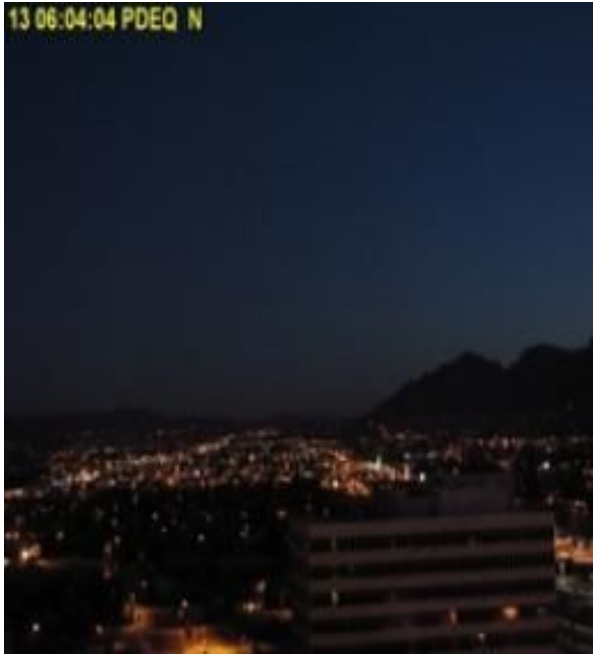


Image-to-Image Translation with Conditional Adversarial Networks, Isola et al., Nov 2017

Results: Semantic Segmentation

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

Results: Sketches to Handbags

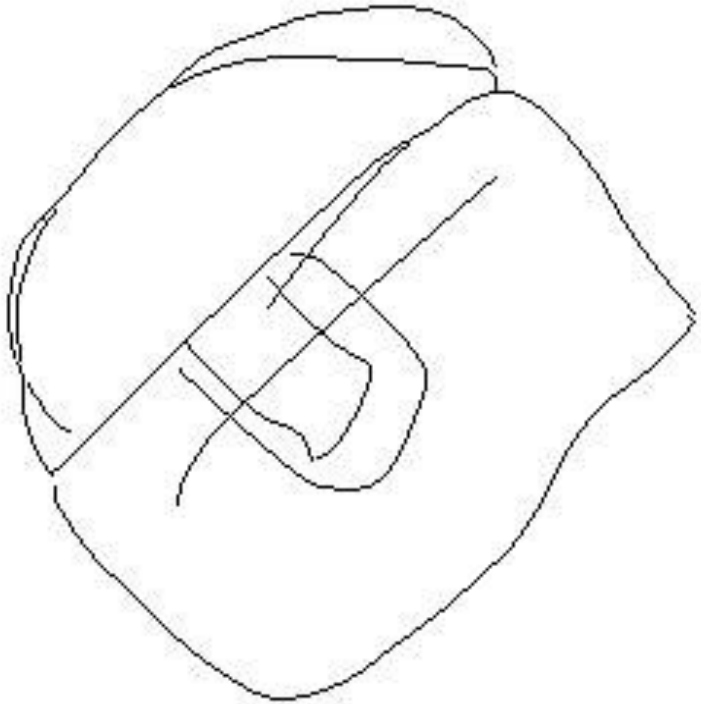
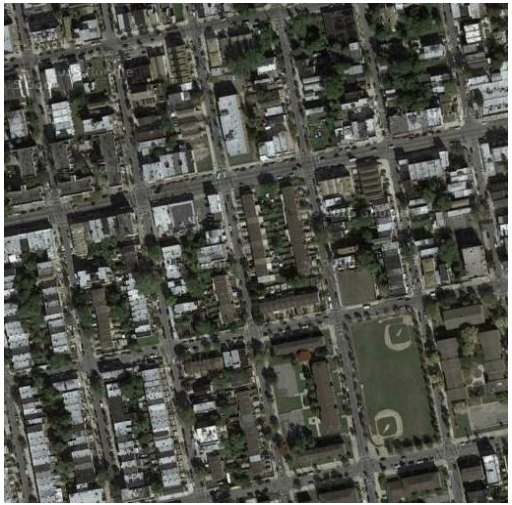


Image-to-Image Translation with Conditional Adversarial Networks, Isola et al., Nov 2017

Results: Generating Maps



GROUND TRUTH

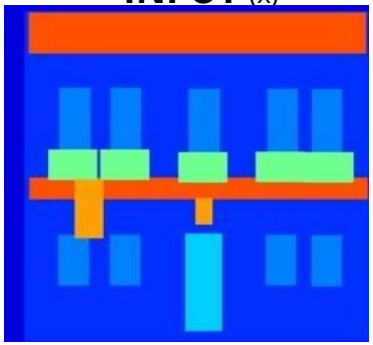
	Photo → Map	Map → Photo
Loss	% Turkers labeled *real*	% Turkers labeled *real*
L1	2.8% ± 1.0%	0.8% ± 0.3%
L1+cGAN	6.1% ± 1.3%	**18.9% ± 2.5%**

Table 4: AMT "real vs fake" test on maps↔aerial photos.

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

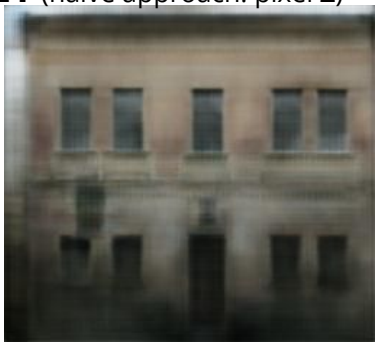
Results - Varied Objectives

INPUT (x)



GROUND TRUTH (y)

L1 (naive approach: pixel Δ)



GAN (didn't use a training image →not conditioned on y)

CGAN (conditioned on y)

L1+GAN

L1+CGAN

FCN Scores

Loss	Per-pixel acc.	Per-class acc.
L1	0.42	0.15
GAN	0.22	0.05
cGAN	0.57	0.22
L1+GAN	0.64	0.20
L1+cGAN	**0.66**	**0.23**
Ground truth	0.80	0.26

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

Results:
Varied Architecture

FCN Scores

Loss	Per-pixel acc.	Per-class acc.
Encoder-decoder (L1)	0.35	0.12
Encoder-decoder (L1+cGAN)	0.29	0.09
U-net (L1)	0.48	0.18
U-net (L1+cGAN)	**0.55**	**0.20**

INPUT (x)

GROUND TRUTH (y)

ENC-DEC+L1

ENC-DEC+L1+CGAN

U-Net+L1

U-Net+L1+CGAN

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

Supervised Multimodal Approach

MultiModal Approach - Introduction

- Im2Im translation is ambiguous - a **single** input image corresponds to **multiple** possible outputs.
- The aim is to learn a **distribution** of **possible** outputs, which we can **sample** from (more on this later)

Toward Multimodal Image-to-Image Translation, Zhu et al., Nov 2017

MultiModal Approach - Goals

Two main goals for the outputs:

- Perceptually **realistic** results
- **Diversity** in the results

Toward Multimodal Image-to-Image Translation, Zhu et al., Nov 2017

BiCycleGAN - The Setting

- Builds upon **pix2pix** framework
- Assumes **paired inputs** i.e. *supervised case*
- Combination of two earlier works - **cVAE-GAN and cLR-GAN**

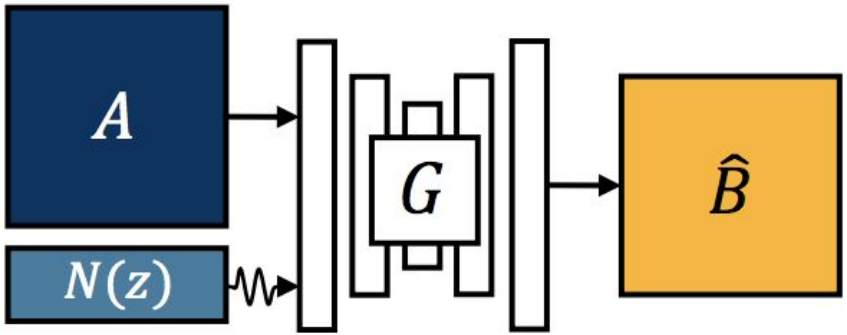
Toward Multimodal Image-to-Image Translation, Zhu et al., Nov 2017

BiCycleGAN - Latent Variables

Latent/hidden variables, usually denoted by **z** are inferred from observed data **x** and help explain the observed data

Latent space Z is simply the vector space where the latent vectors live

We would like to make our *generator* **stochastic** by adding a **sampled latent vector** to its input that will allow it to generate multiple (varied) outputs per input

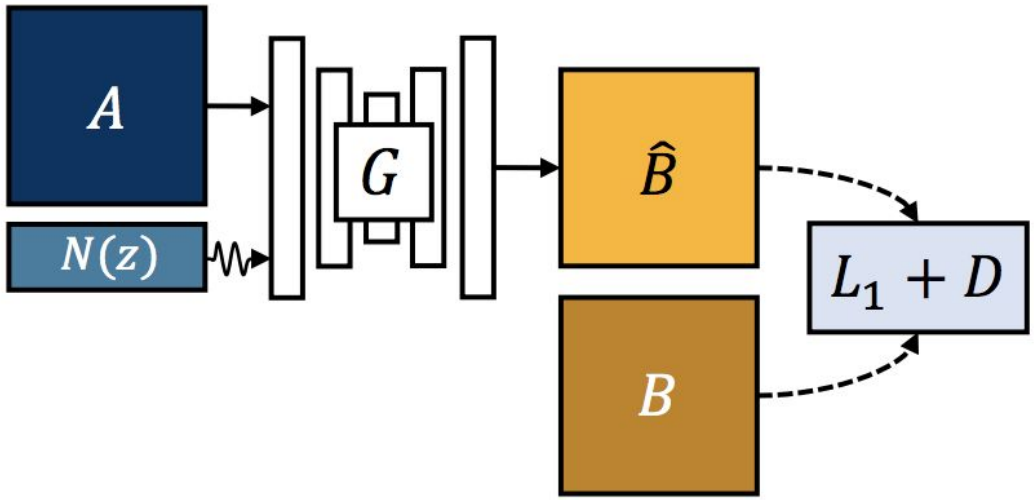


Toward Multimodal Image-to-Image Translation, Zhu et al., Nov 2017

Naive Solution - noise + pix2pix

A simple way to achieve variation from inputs, is a random vector **z** added to the generator input

Not effective: **z** does not contain meaningful information, network learns to ignore it & often leads to mode collapse



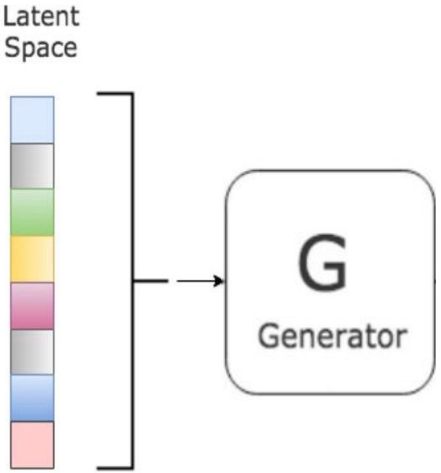
Toward Multimodal Image-to-Image Translation, Zhu et al., Nov 2017

The Missing Piece of GANs - Encoder to Latent Space

GANs are **generative** models and as such, can model well from the direction of a latent vector **z** to image **x** (**z -> x**)

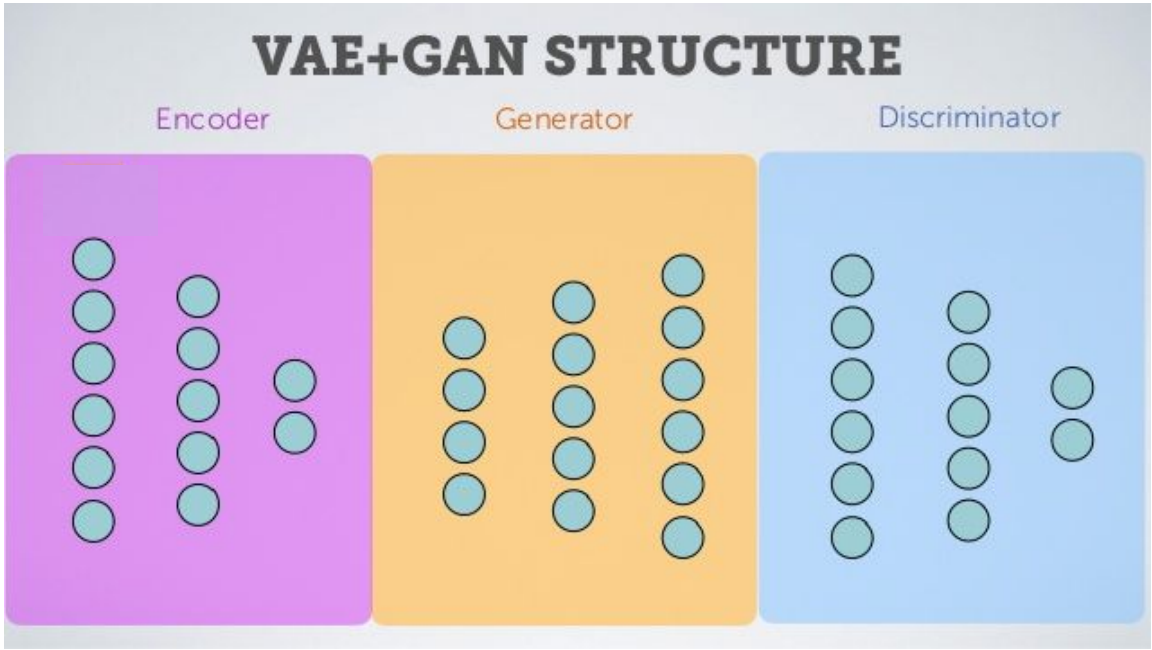
GANs lack the ability of **inference** which is going from **x -> z**

There are several ways of introducing a meaningful latent vector **z**



VAE-GAN

One way of doing both **inference** (encoding) and **generation** is combining VAE with GANs

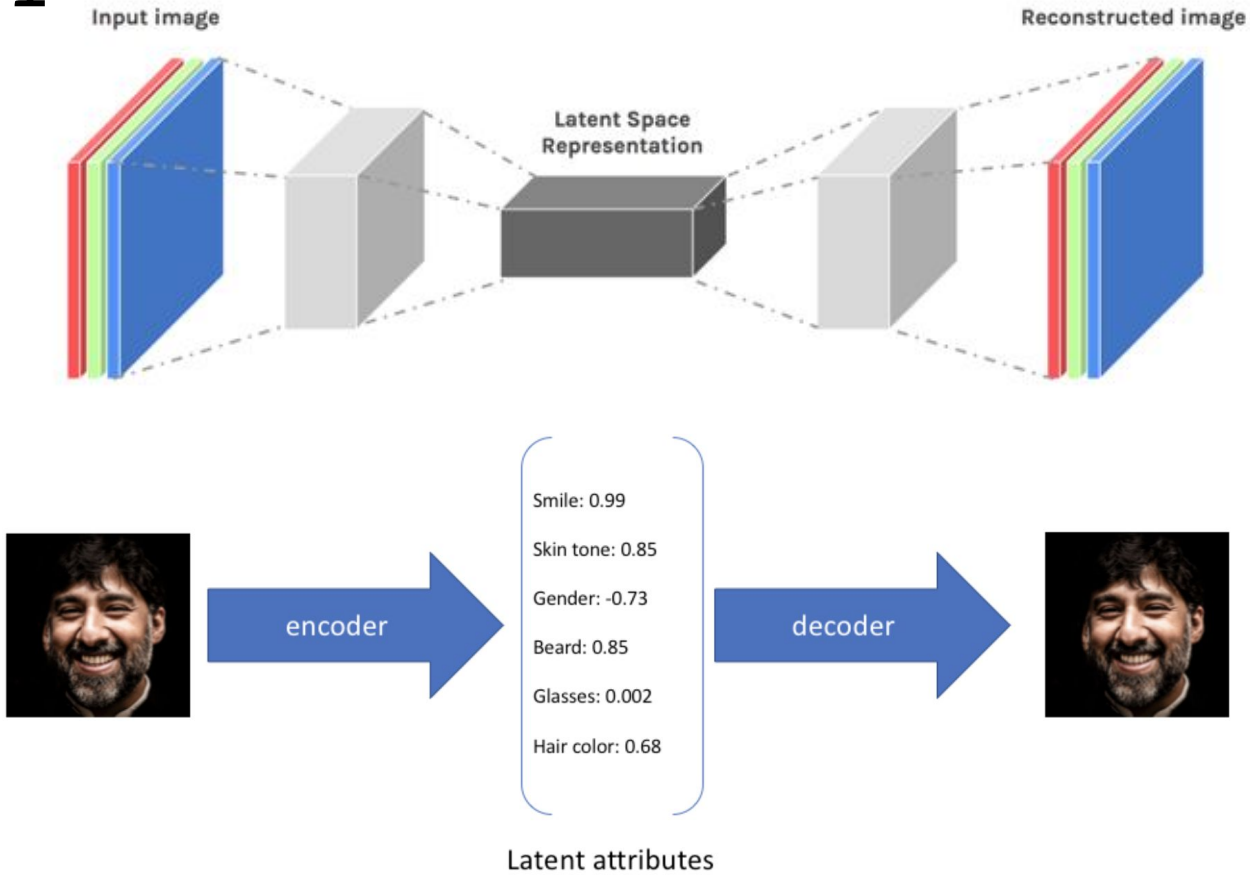


AutoEncoders 101

Learn a **meaningful** representation **z** given input **x** by using reconstruction error

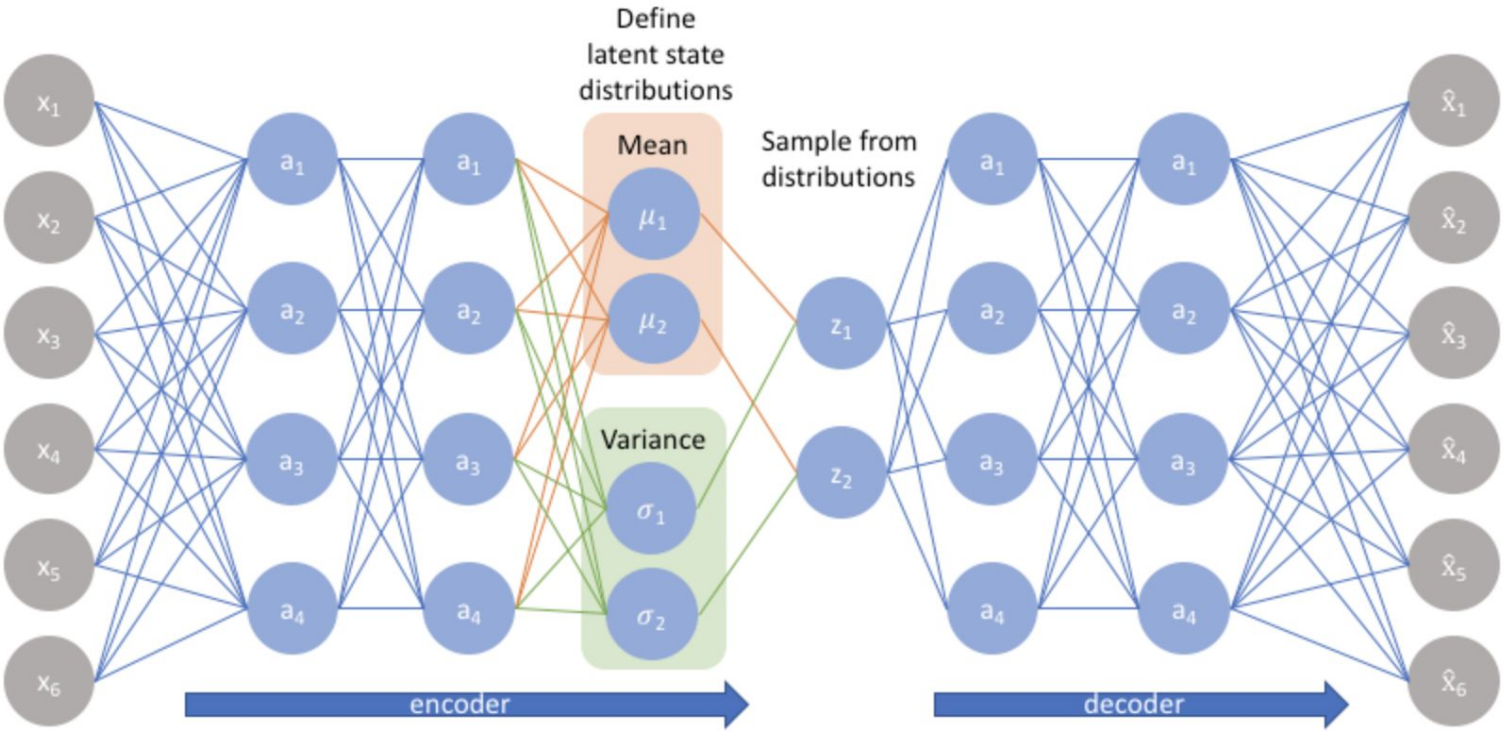
The encoder in this case is **deterministic**, we will get the same **z** for same **x** after training

Note: Not a generative model yet: We can not sample from z

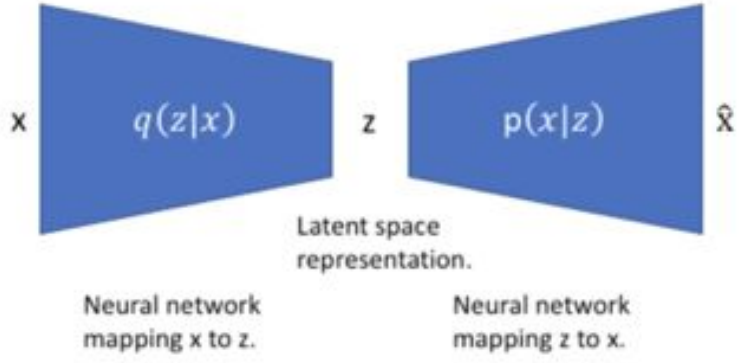


Variational AutoEncoders 101

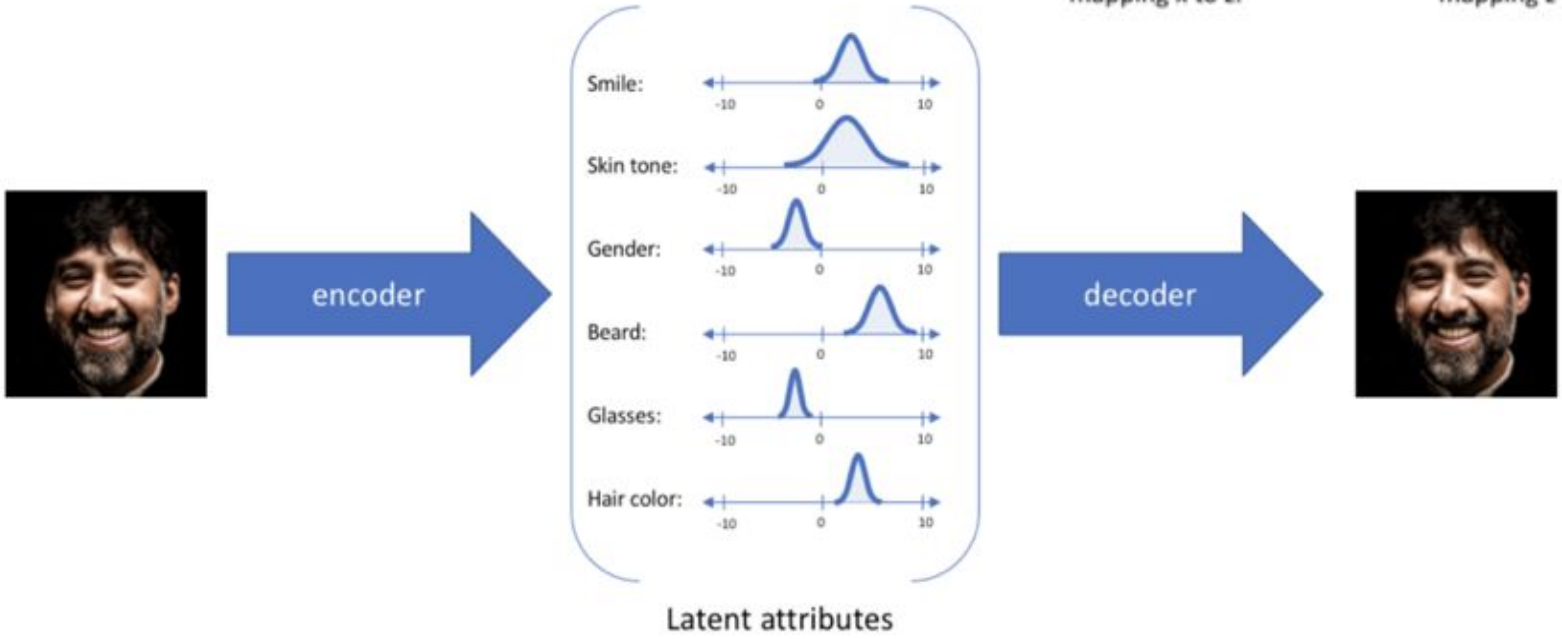
Explicitly learn **distribution function** per latent attribute, which we can sample from



Variational AutoEncoders 101

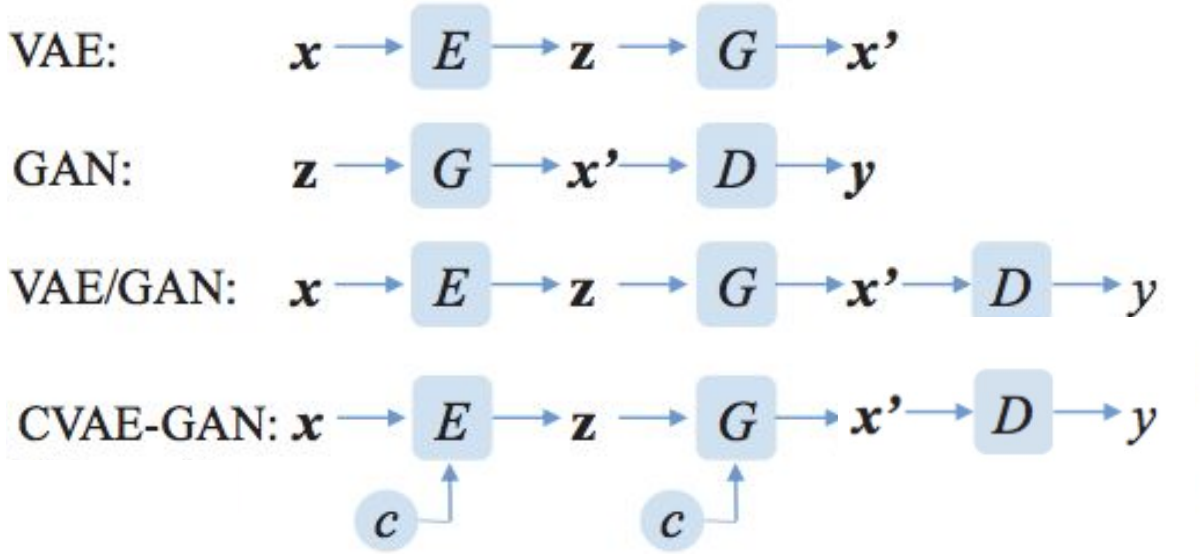


The encoder models distribution **q (z | x)**



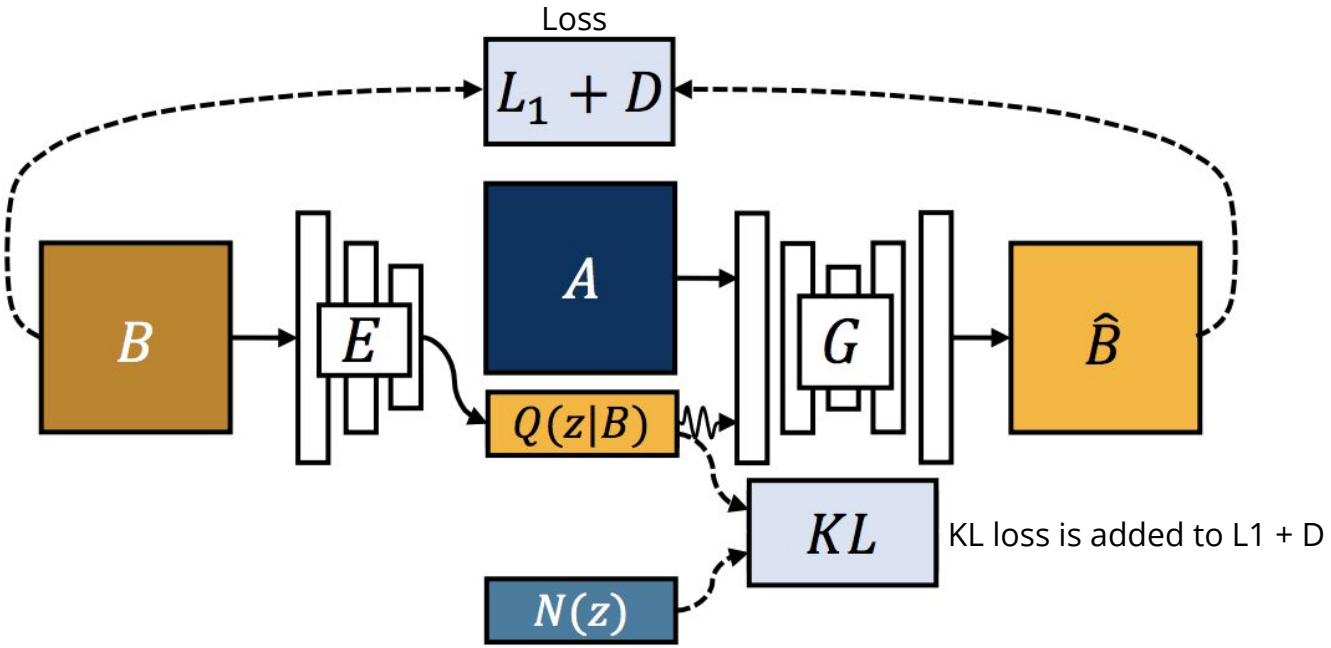
cVAE-GAN

Main idea: combine both models and treat the Decoder of VAE and Generator of the GAN as the same. Notice that we can achieve **multiple meaningful z's** from a **single x** using the **encoder** of VAE



cVAE-GAN

During training we give the VAE the ground truth image **B** as input, to sample a meaningful **z**



cVAE-GAN - loss functions

E(B) is the encoder, which models the distribution function **Q(z|B)**

So: GAN loss + we sample a **z** from the encoder and input it to generator

$$\mathcal{L}_{\text{GAN}}^{\text{VAE}} = \mathbb{E}_{\mathbf{A},\mathbf{B}\sim p(\mathbf{A},\mathbf{B})}[\log(D(\mathbf{A},\mathbf{B}))] + \mathbb{E}_{\mathbf{A},\mathbf{B}\sim p(\mathbf{A},\mathbf{B}),\mathbf{z}\sim E(\mathbf{B})}[\log(1 - D(\mathbf{A}, G(\mathbf{A},\mathbf{z})))]$$

We regularize the encoder by making it similar as possible to a unit gaussian

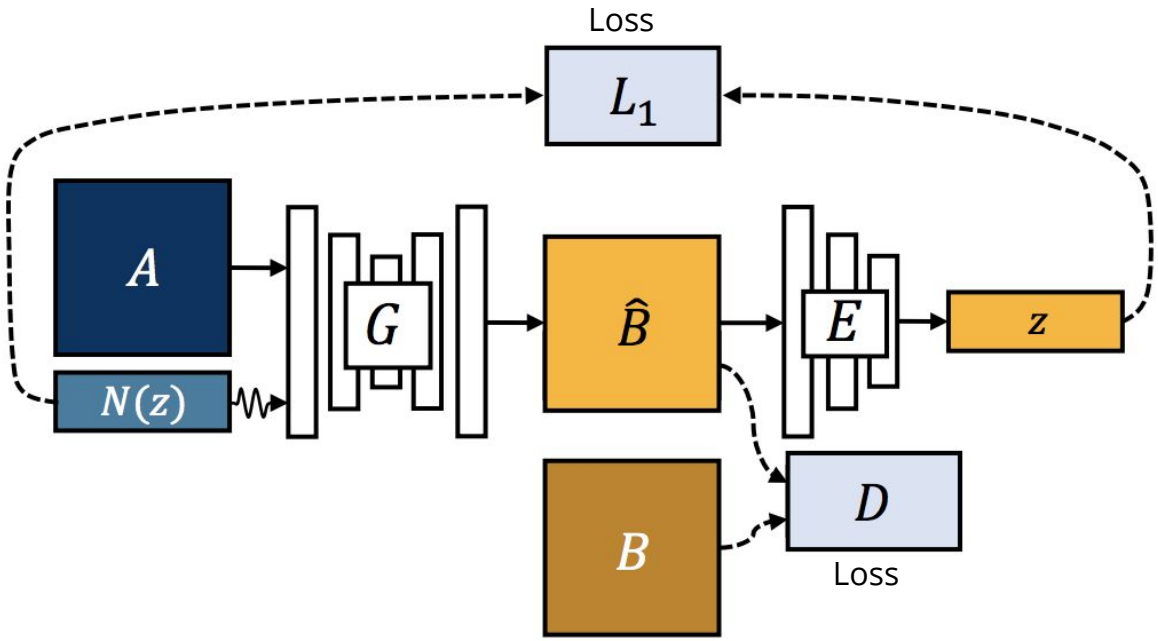
$$\mathcal{L}_{\text{KL}}(E) = \mathbb{E}_{\mathbf{B}\sim p(\mathbf{B})}[\mathcal{D}_{\text{KL}}(E(\mathbf{B})||\,\mathcal{N}(0, I))],$$

In total we have

$$G^*, E^* = \arg\min_{G,E}\max_{D} \quad \mathcal{L}_{\text{GAN}}^{\text{VAE}}(G, D, E) + \lambda\mathcal{L}_1^{\text{VAE}}(G, E) + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}(E).$$

cLR-GAN

- Sample random latent code **z**, add to gen input
- Train an encoder by making sure **E(G(A,z)) ~ z**
- The encoder in this case produces a **point estimate** not a **distribution** as before



cLR-GAN - loss functions

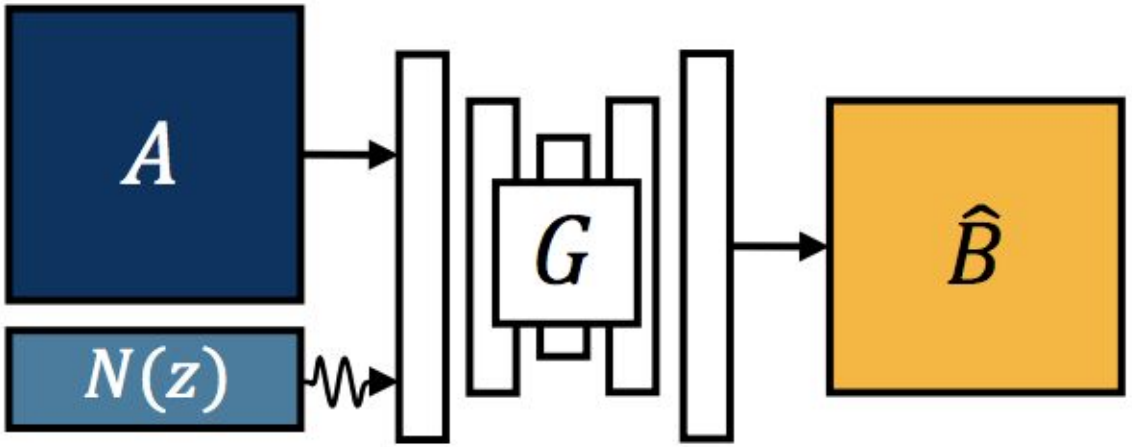
Sample **z** from prior, add to generator as input

$$\mathcal{L}_1^{\text{latent}}(G, E) = \mathbb{E}_{\mathbf{A}\sim p(\mathbf{A}),\mathbf{z}\sim p(\mathbf{z})}||\mathbf{z} - E(G(\mathbf{A}, \mathbf{z}))||_1$$

Use regular adversarial loss to make sure the generator is realistic

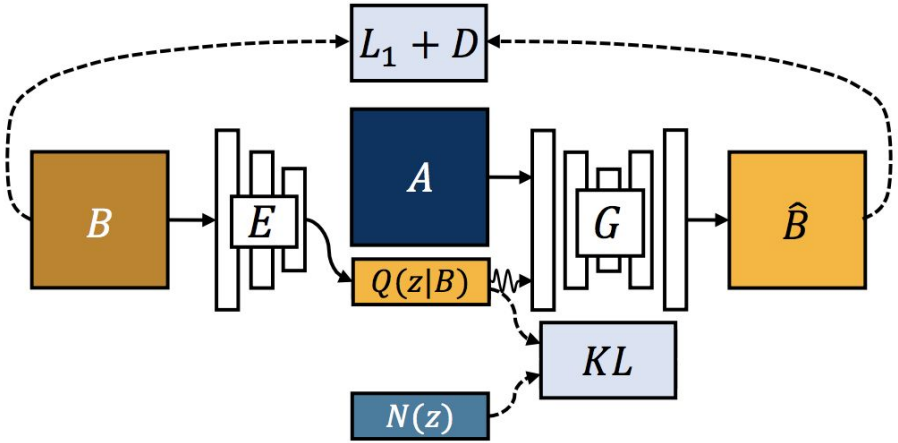
$$G^*, E^* = \arg\min_{G,E}\max_{D} \quad \mathcal{L}_{\text{GAN}}(G, D) + \lambda_{\text{latent}}\mathcal{L}_1^{\text{latent}}(G, E)$$

BiCycleGAN - Test Time of all models

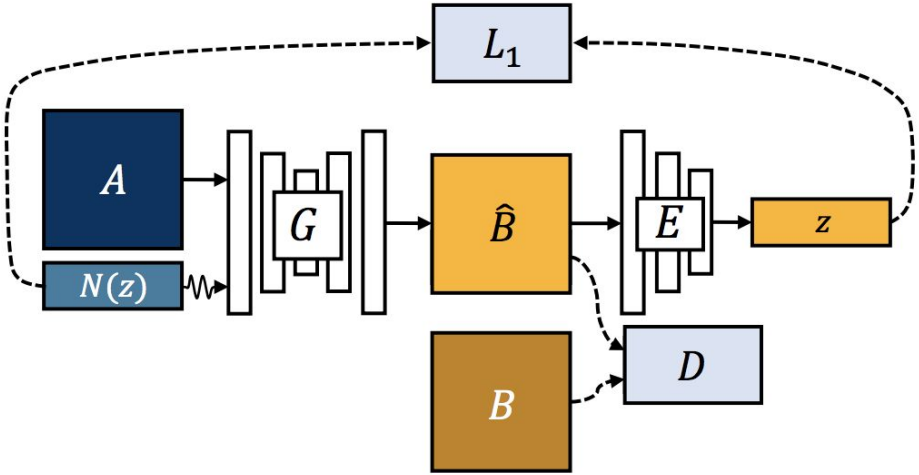


BiCycleGAN - Hybrid Model Motivation

In **cVAE-GAN** (B→ z → B'): the encoding is learned from **real data** (the ground truth), but at **test time** we sample from a **prior** distribution. This may cause unrealistic results.



In **cLR-GAN** (z → B' → z'): the latent space is **easily** sampled from a **simple prior** distribution, but we do **not** compare the input/output pairs directly.



Toward Multimodal Image-to-Image Translation, Zhu et al., Nov 2017

BiCycleGAN - Hybrid model

Main Idea: Combine **both cycles** (B→ z → B′ → z′)

The loss function is the combination of the two we've seen before:

$$G^*, E^* = \arg\min_{G,E} \max_{D} \quad \mathcal{L}_{\text{GAN}}^{\text{VAE}}(G, D, E) + \lambda \mathcal{L}_1^{\text{VAE}}(G, E)$$
$$+ \mathcal{L}_{\text{GAN}}(G, D) + \lambda_{\text{latent}} \mathcal{L}_1^{\text{latent}}(G, E) + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}(E)$$

Toward Multimodal Image-to-Image Translation, Zhu et al., Nov 2017

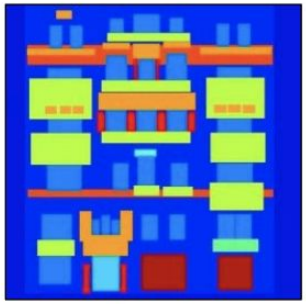
BiCycleGAN - Network & Training Details

Builds on same architecture as earlier...

- **Generator**: a U-Net is used
- **Discriminator**: uses PatchGANs
- **Encoder** has two variants:
  - CNN with few downsampling conv layers
  - ResNet architecture

Toward Multimodal Image-to-Image Translation, Zhu et al., Nov 2017

Qualitative Results



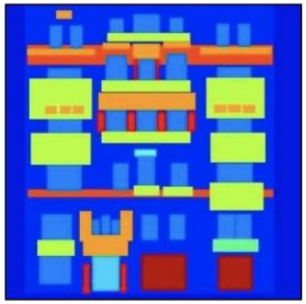
Input

Ground truth

pix2pix+noise

Toward Multimodal Image-to-Image Translation, Zhu et al., Nov 2017

Qualitative Results



Input

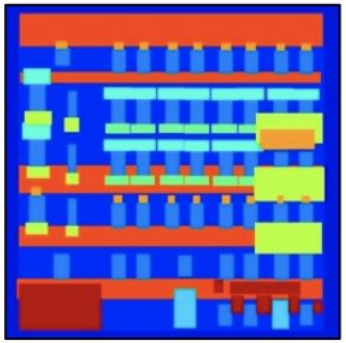
Ground truth

Toward Multimodal Image-to-Image Translation, Zhu et al., Nov 2017

Qualitative Results

Toward Multimodal Image-to-Image Translation, Zhu et al., Nov 2017

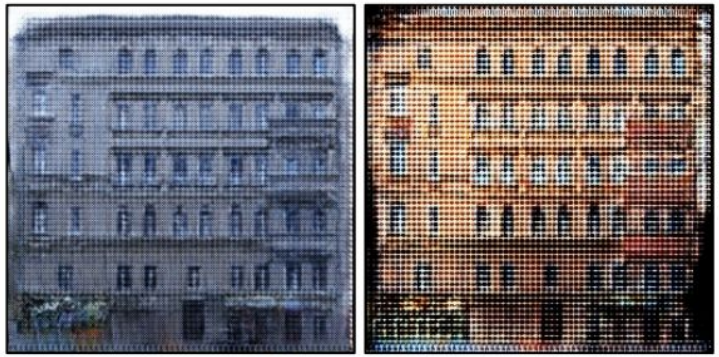
Qualitative Results - Latent Code Length



Input label

|z| = 2

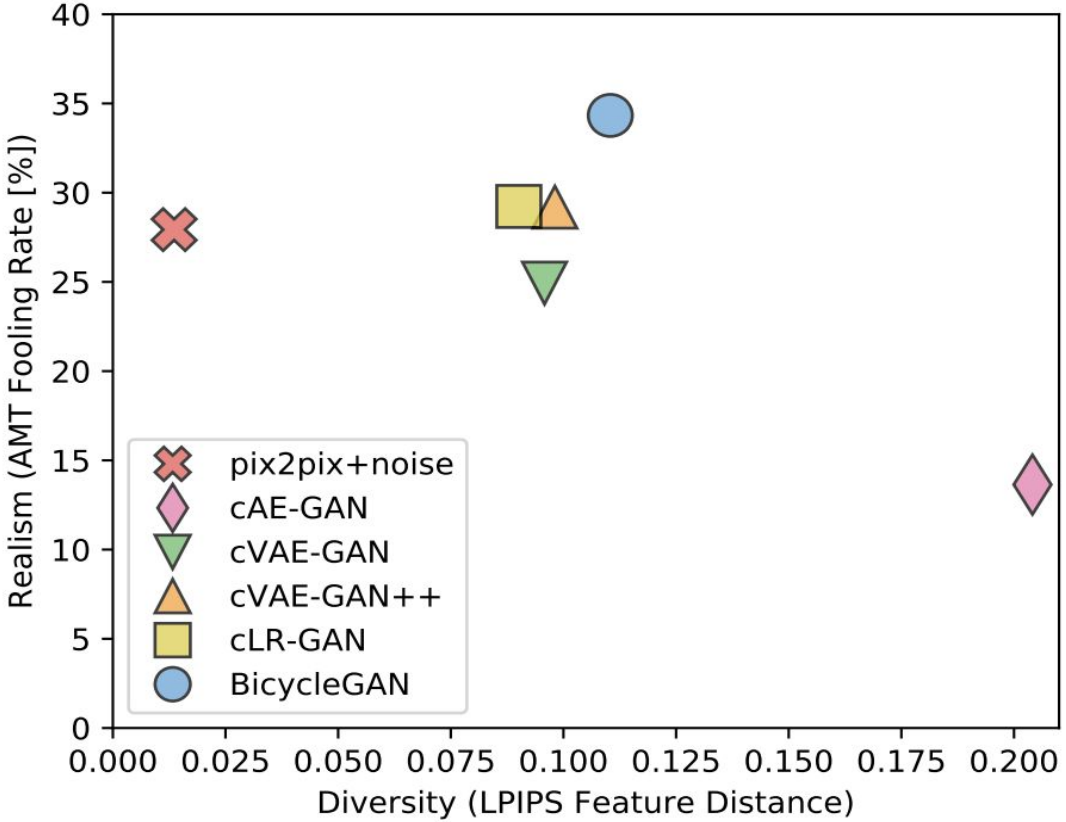
|z| = 8



|z| = 256

Toward Multimodal Image-to-Image Translation, Zhu et al., Nov 2017

Quantitative Results - Realism & Diversity



Measured on Google Maps → Satellite images task

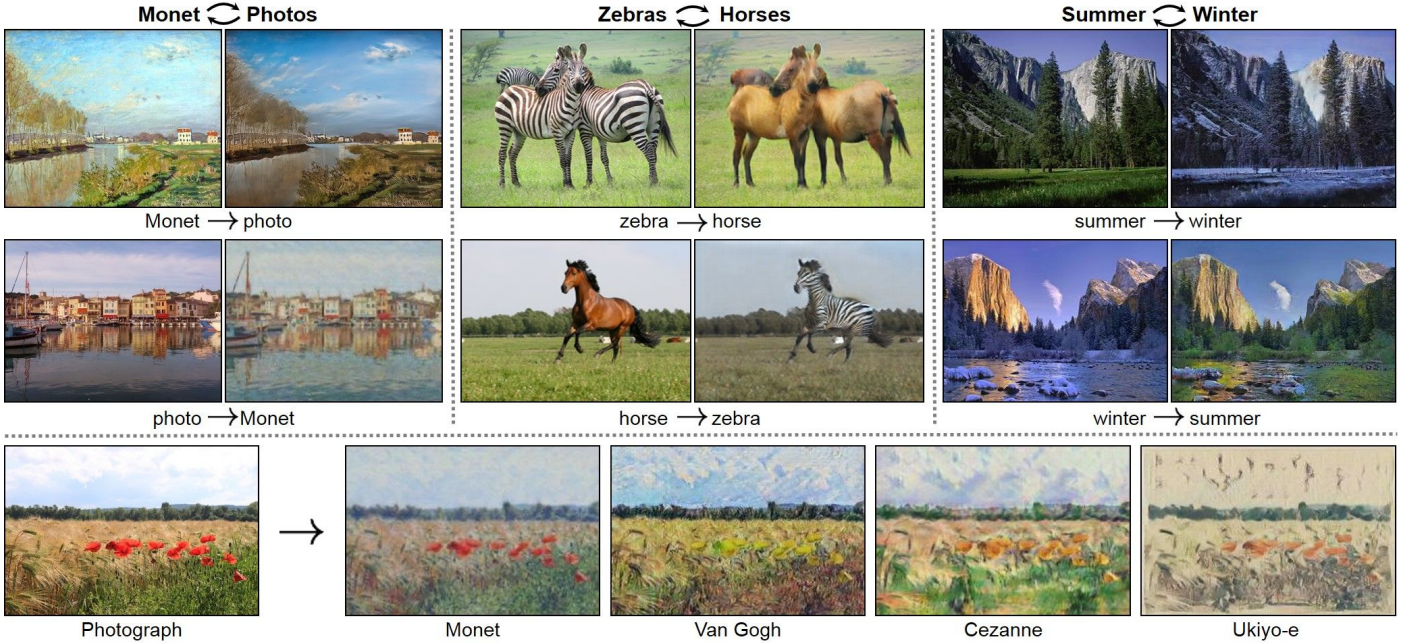
Toward Multimodal Image-to-Image Translation, Zhu et al., Nov 2017

Break

Unsupervised Domain Transfer

Unpaired I2I Translation

GAN model which can **transfer images from one to domain to another** without need of training pairs



Images from [Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks](), Jun-Yan Zhu et al. ICCV 2017

Unpaired Images - Unsupervised Learning

Goal: beyond paired datasets → unsupervised learning

Reasoning: obtaining paired training data can be difficult and sometimes not well defined (**e.g summer ↔ winter**)

Unpaired Images - Formulation

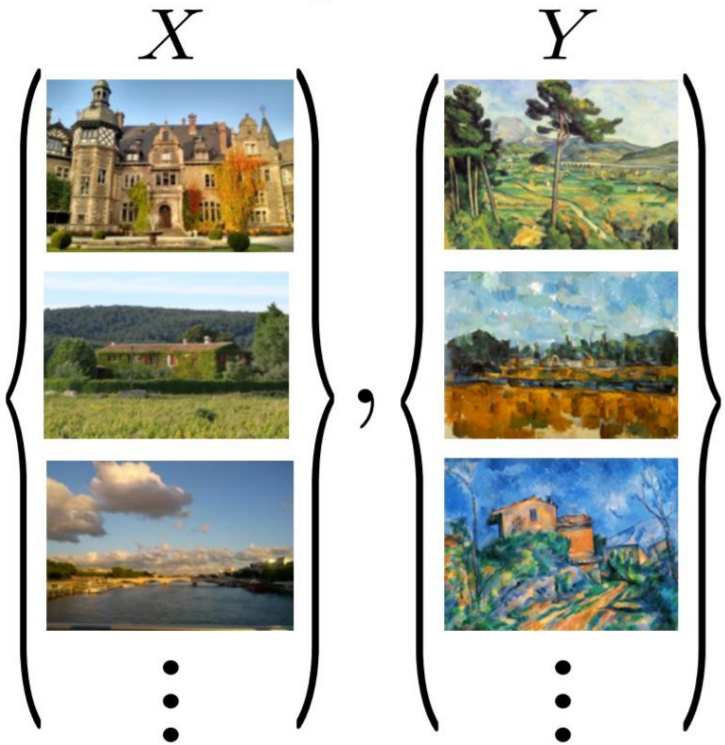
Input: two different data sets, representing **two domains:** X (e.g. landscapes) and Y (e.g. Monet paintings)

Learn two domain transfer functions **G: X → Y** & **F: Y → X**; **two generators** are needed

For training, need **two discriminators**: $\mathbf{D_X}$ & $\mathbf{D_Y}$ indicating whether images *fit* target distribution (X or Y)

→ need **two** GANs with adversarial loss

→ these GANs are connected to ensure cycle consistency



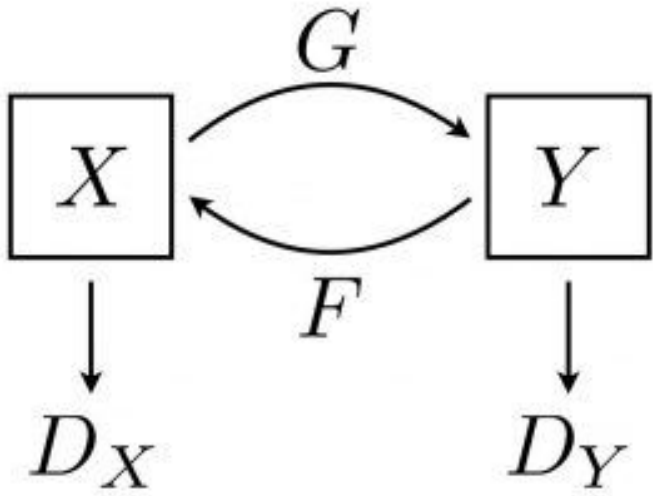
Images from Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, Jun-Yan Zhu et al. ICCV 2017

CycleGAN: Introduce Cycle Consistency

Motivation: no guarantee that input **x** (from domain x) & output **y** (with characteristics of domain y) will have any meaningful relationship

Also, in practice, training objective can often lead to **mode collapse**

To ensure a meaningful relationship between the input/output pairs & regularise mappings → **introduce cycle consistency goal**



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

Cycle Consistency Loss Fn

translating generated image from domain **y** back to original domain **x**

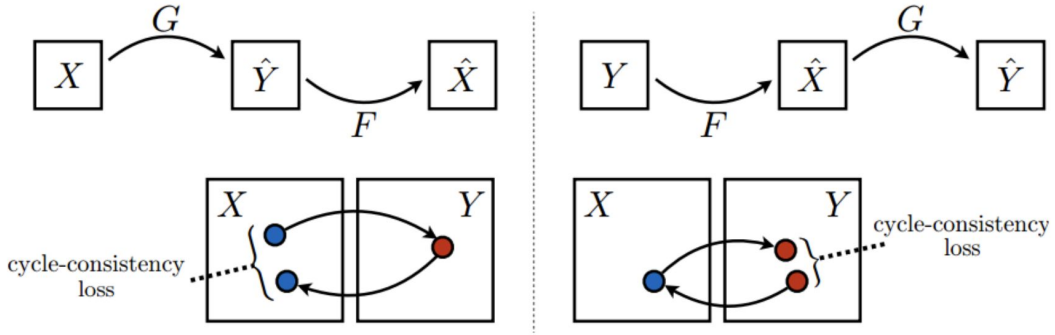
$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]$$

image **x** translated to domain **y** by function **G**

L1-norm difference between **generated image and conversion back to original domain**, versus **original**

Goal

F(G(x)) ~ x && G(F(y)) ~ y



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

Adversarial Loss Functions

For mapping function **G: X → Y** and its discriminator **D_Y** the objective fn is:

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

*where **G** tries to generate images **G(x)** that look similar to images from domain **Y**, while **D_Y** aims to distinguish between samples **G(x)** and real samples **y***

For the **inverse** mapping function **F: Y → X** and its discriminator **D_X** the objective fn is:

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

*where **F** tries to generate images **F(y)** that look similar to images from domain **X**, while **D_X** aims to distinguish between samples **F(y)** and real samples **x***

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

Full loss function

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

where λ controls the relative importance of the two objectives.

Goal is to solve:

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

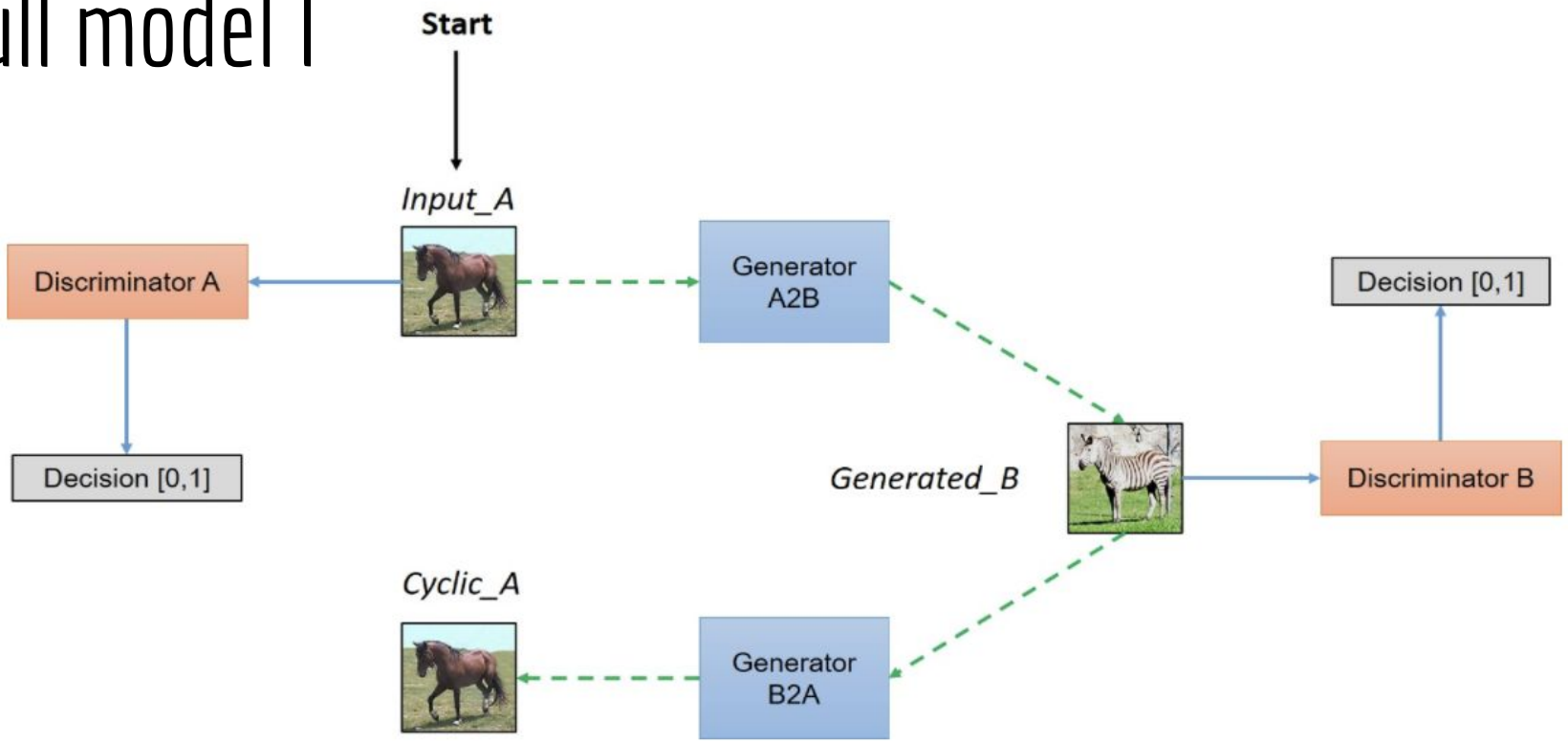
Model can be viewed as training two *autoencoders*

Learn autoencoder: F ◦ G : X → X
jointly with another: G ◦ F : Y → Y

These autoencoders each have special internal structure: *map an image to itself via an intermediate representation that is a translation of the image into another domain*

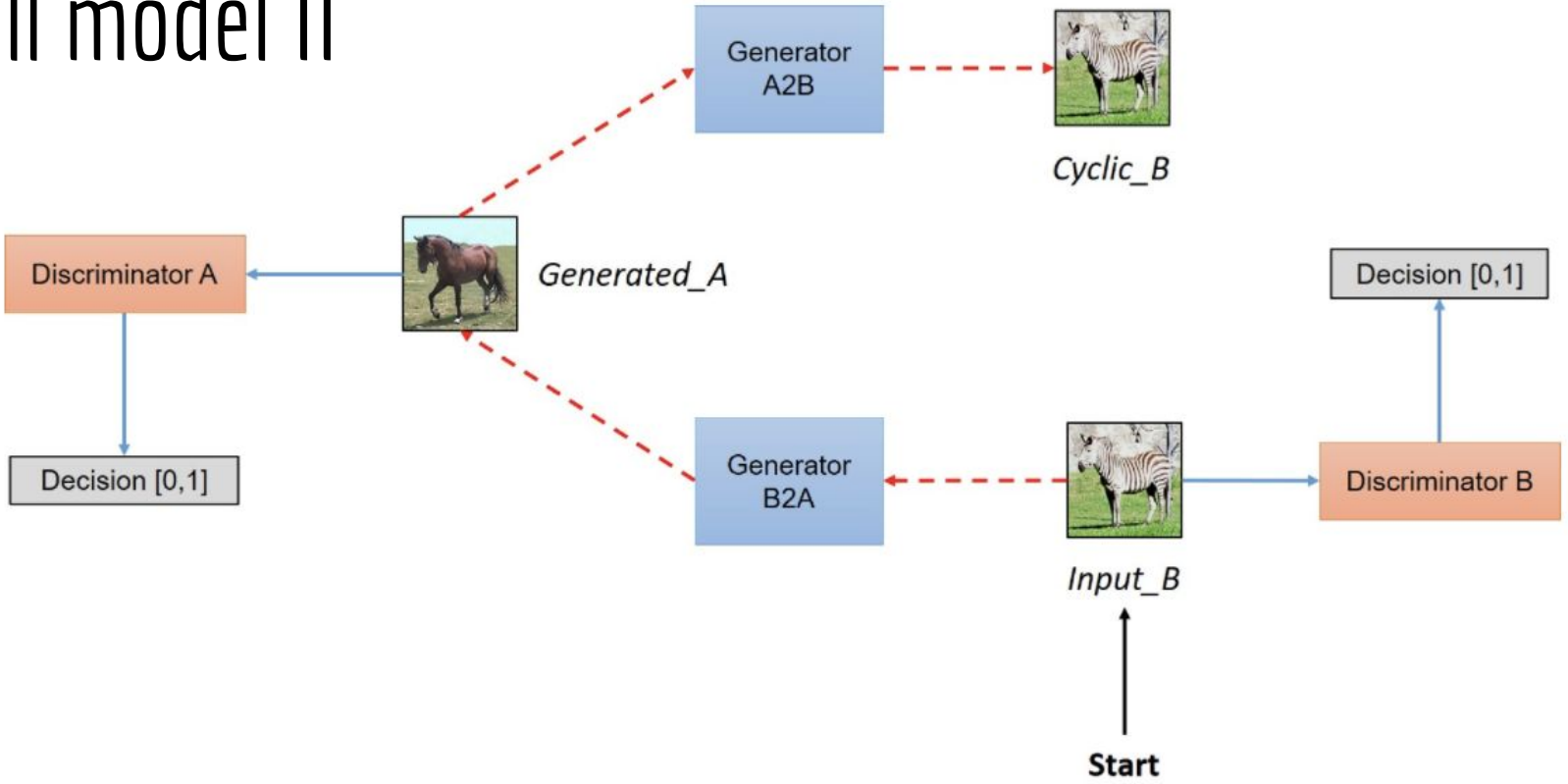
Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, Jun-Yan Zhu et al. ICCV 2017

Full model I



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

Full model II



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

Architecture

The **discriminator** is a **PatchGAN** as we've seen before

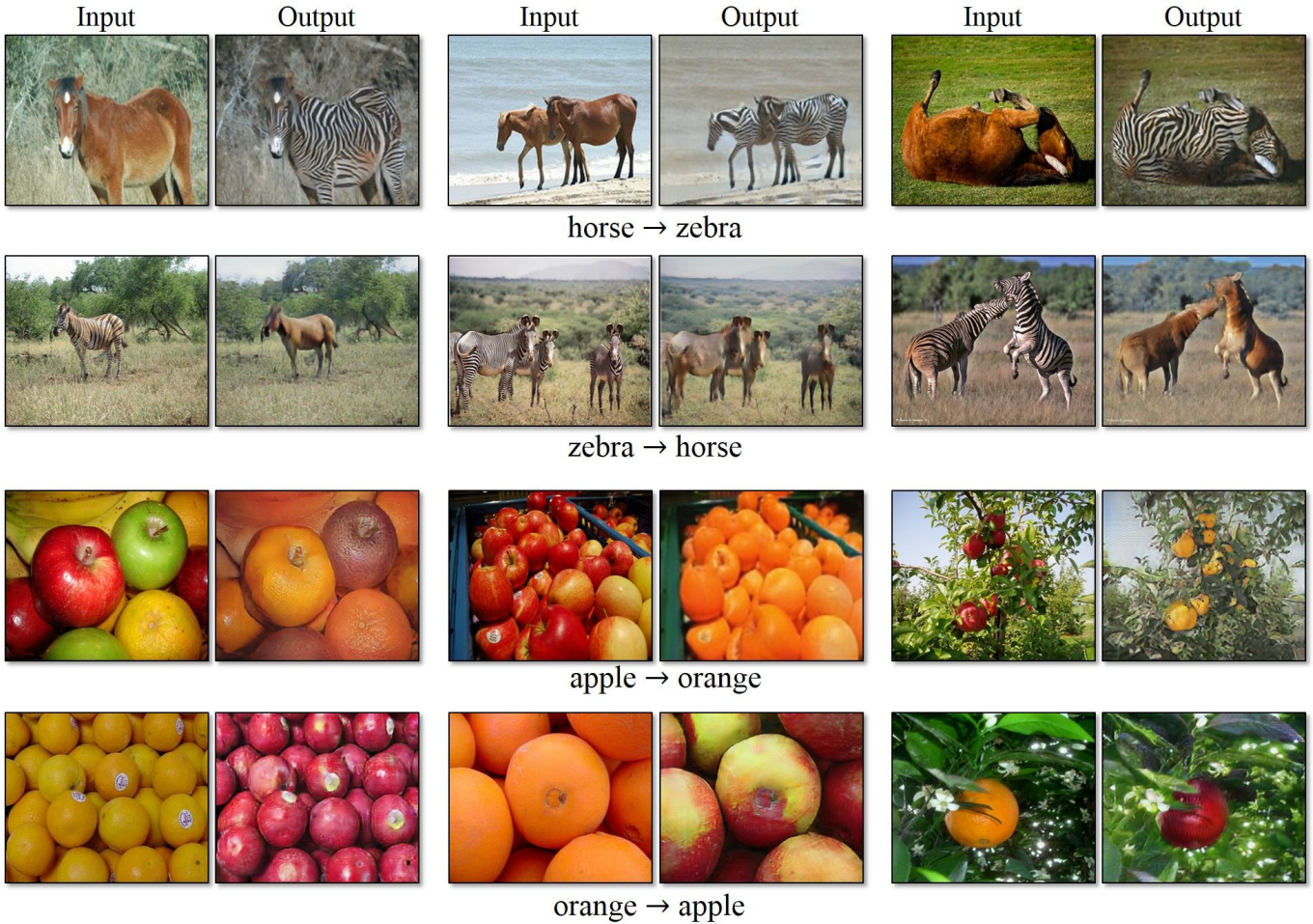
The **generator** uses **encode/decoder** scheme with few **residual blocks** with skip connections

Cyclicality is strongly enforced: **λ** is set to a high number

Results: style transfer

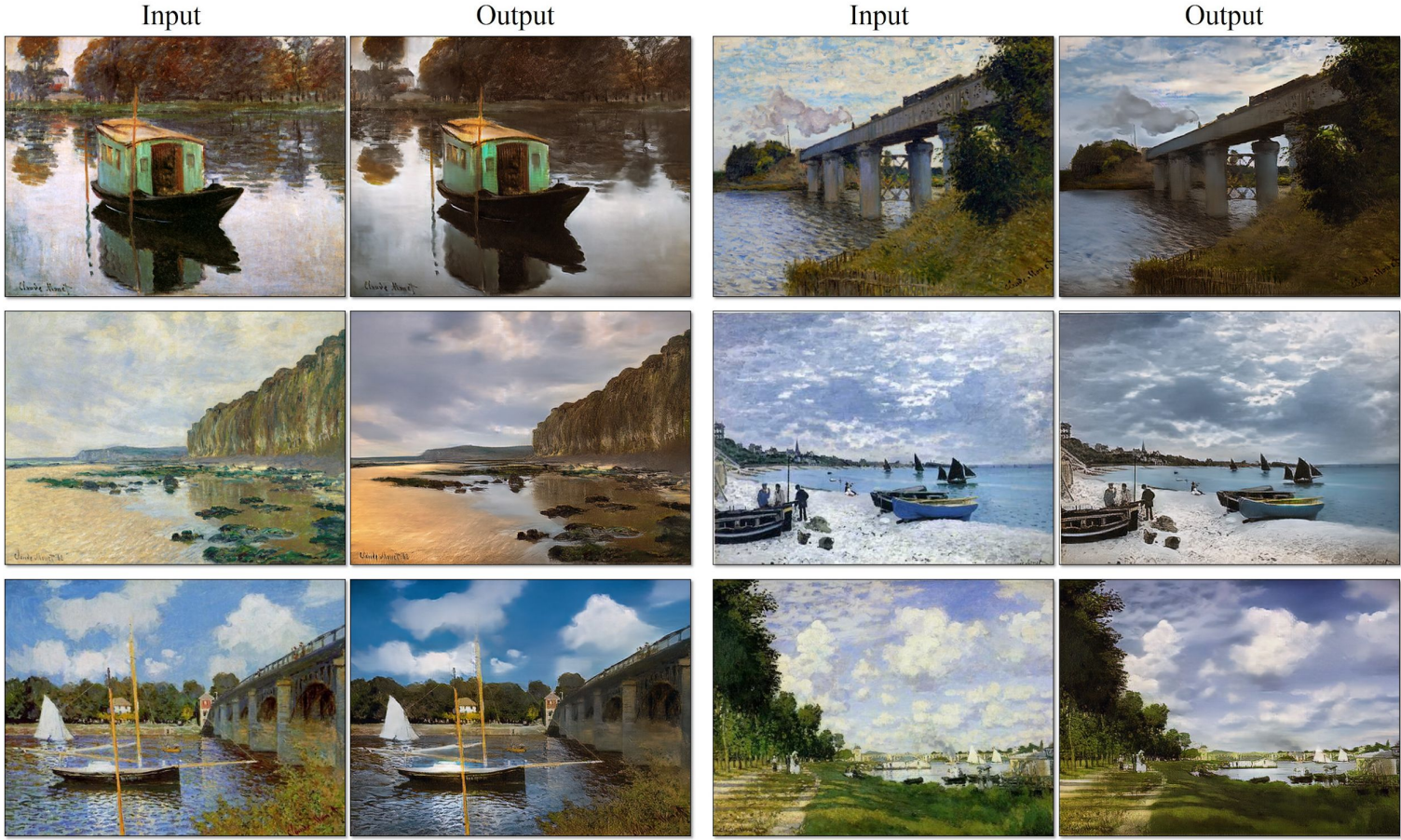
Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, Jun-Yan Zhu et al. ICCV 2017

Results: object trans-figuration



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

Results: Monet to photos



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

Ancient Map of Jerusalem ⟶ Satellite Image

After 45 hours of training on nVidia 1070:

Image from [Jack Clark Twitter Feed.](#) Jun 2017

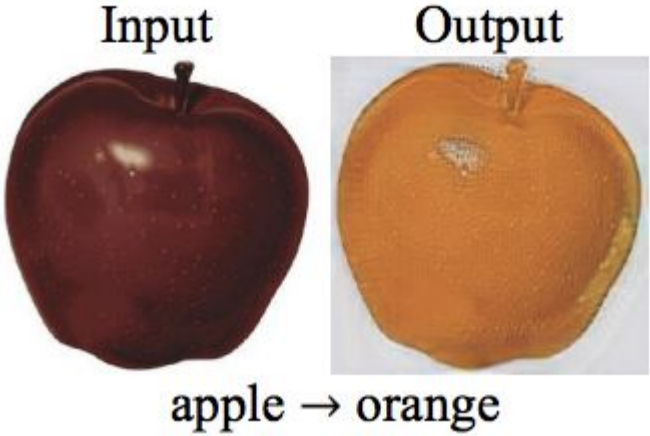
Limitations

The good: CycleGAN works well for texture or color changes

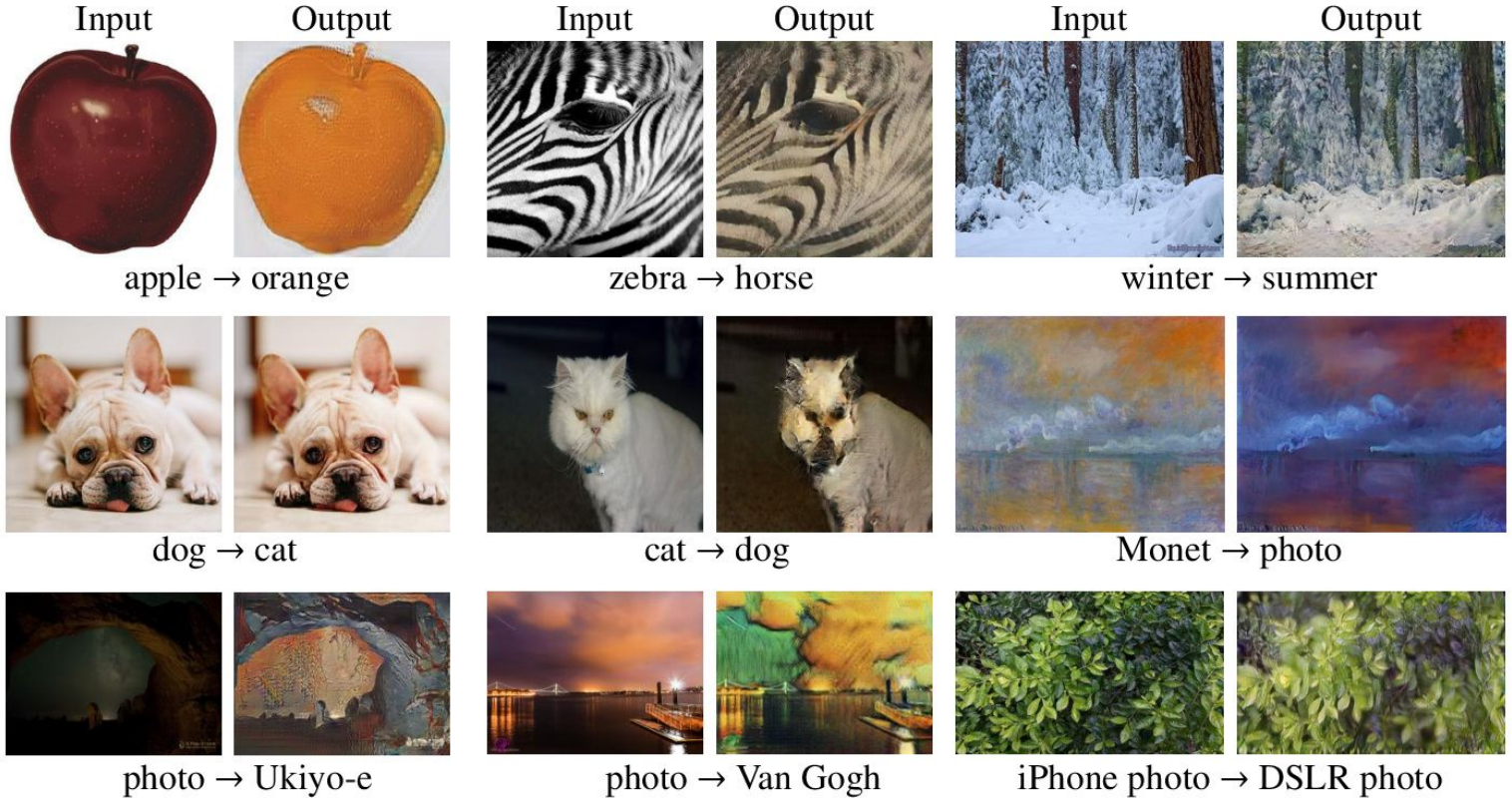
The bad: Geometric changes (e.g apple <-> orange) have little success

Dataset characteristic can also cause confusion, e.g ImageNet does not contain images of a man riding a horse/zebra

Still a gap between results from paired training data and those achieved by the unpaired method



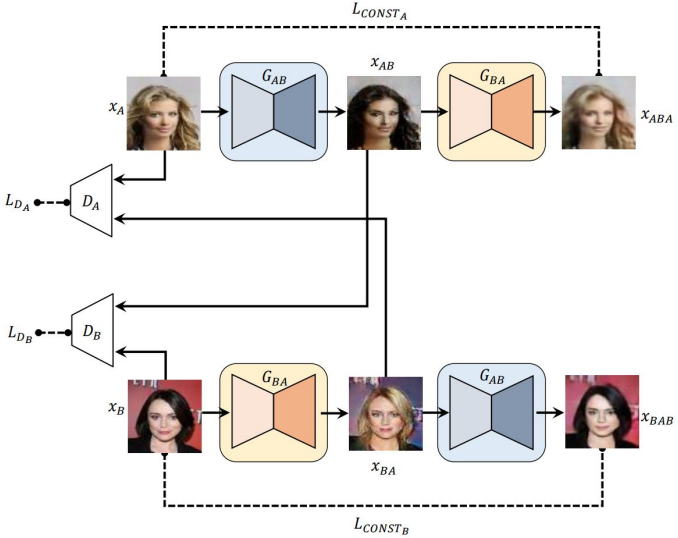
Results: Failures

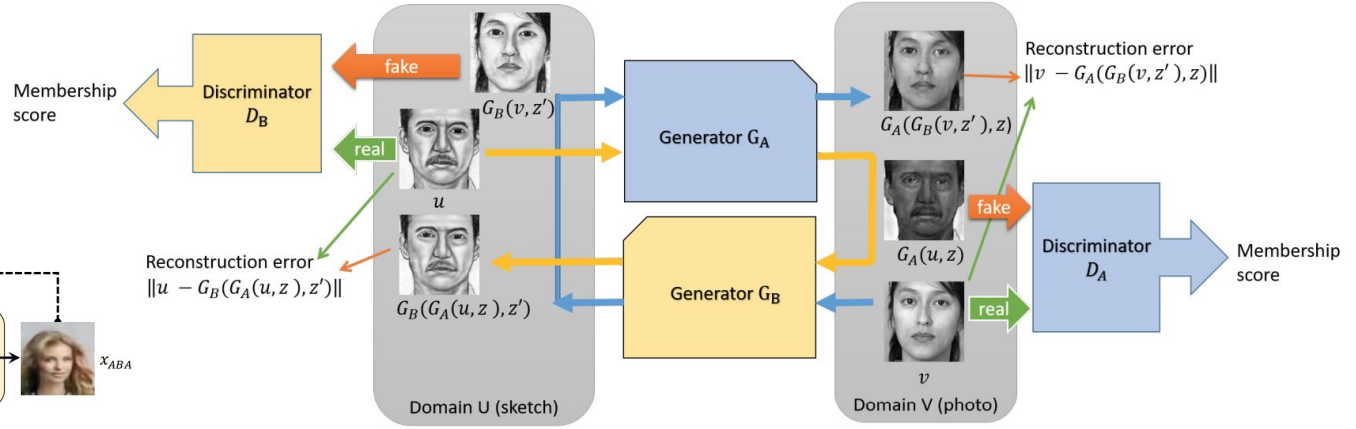


Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, Jun-Yan Zhu et al. ICCV 2017

Unsupervised: Adding Constraints

Unsupervised: More than CycleGAN



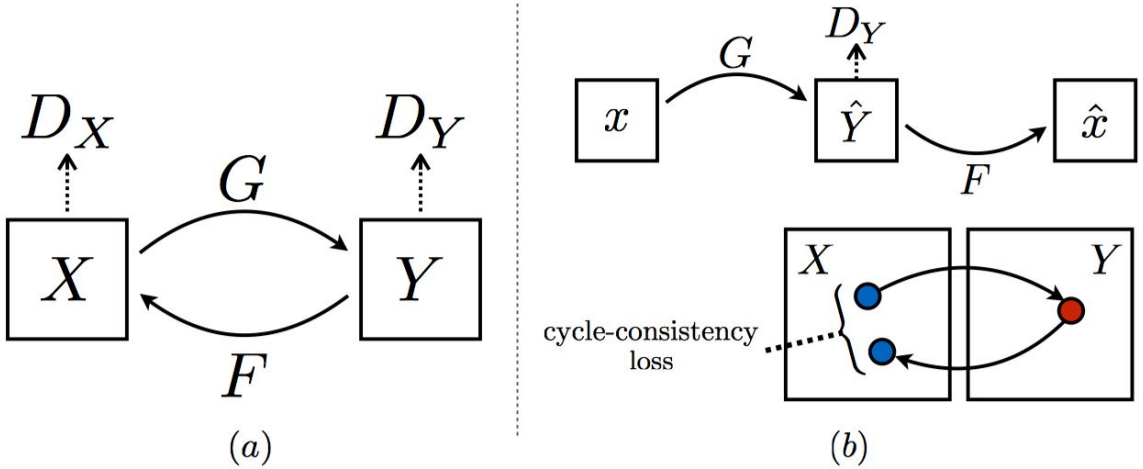


Same Core Idea Underlied

DualGAN: DualGAN: Unsupervised Dual Learning for Image-to-Image Translation, Zili Yi et al. ICCV 2017

DiscoGAN: Learning to Discover Cross-Domain Relations with Generative Adversarial Networks, Taeksoo Kim et al. ICML 2017

Cycle Consistency (Again)



Adversarial Constraints:

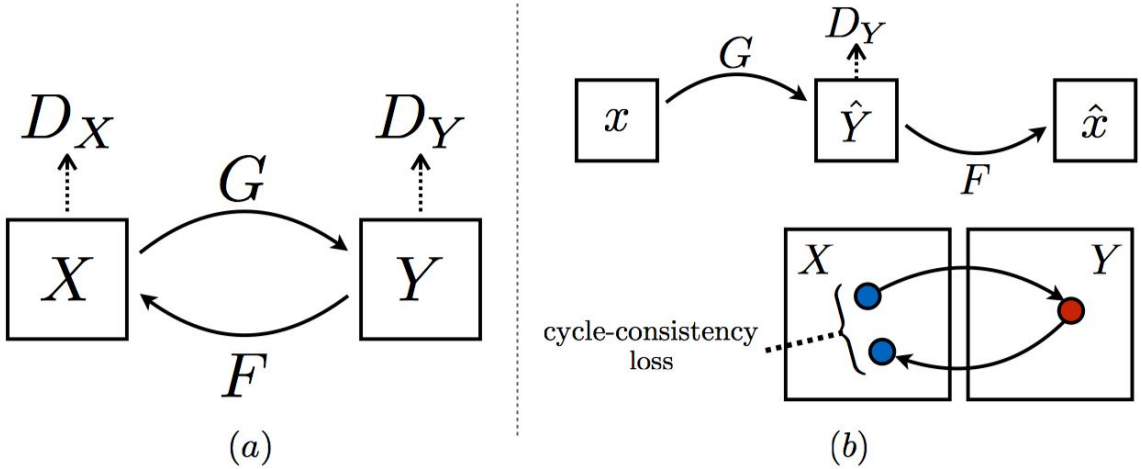
- ❏ Weak
- ❏ Large mapping space

Circularity Constraints:

- ❏ Stronger
- ❏ Smaller mapping space

Images from [Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks](), Jun-Yan Zhu et al. ICCV 2017

Cycle Consistency (Again)



Adversarial Constraints:

- ❏ Weak
- ❏ Large mapping space

Circularity Constraints:

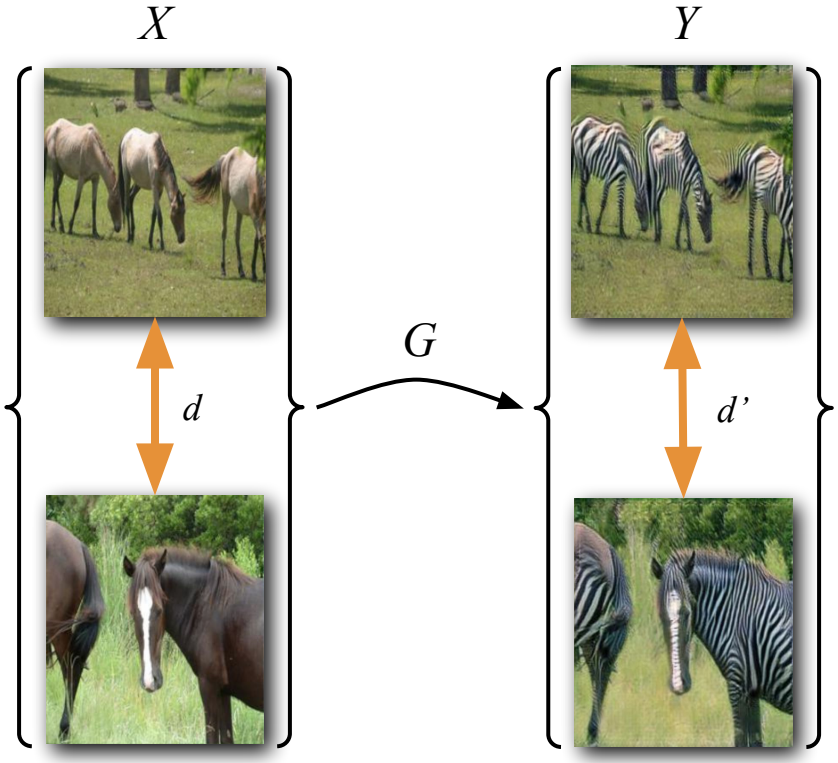
- ❏ Stronger
- ❏ Smaller mapping space

Stronger constraints?

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

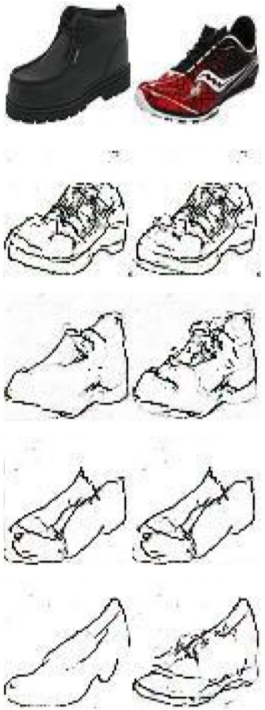
Idea: High Distance Correlation

- A pair of images of a given distance are mapped to a pair of outputs with a similar distance.
- $d = |x_i - x_j|_1$
- $d' = |G(x_i) - G(x_j)|_1$
- $d \sim d'$



Images reproduced from [One-Sided Unsupervised Domain Mapping](), Sagie Benaim and Lior Wolf. NIPS 2017

Two Assumptions



- Assumption 1:
  - Edge points are a subset of image points
  - **Similar images - similar edges**

Images from [One-Sided Unsupervised Domain Mapping](), Sagie Benaim and Lior Wolf. NIPS 2017

Two Assumptions

- Assumption 1:
  - Edge points are a subset of image points
  - **Similar images - similar edges**
- Assumption 2:
  - Similar color and spatial distributions of samples in each domain
  - Transformation can be done by pixel displacements
  - **A fixed pixel permutation may exist**
  - **Distances would be preserved**

Images from [One-Sided Unsupervised Domain Mapping](), Sagie Benaim and Lior Wolf. NIPS 2017

Justification - Nonnegative Matrix Approx.

- A **nonnegative linear transformation** T is learned to approximate CycleGAN's generator
- T ends up to be very similar to a **pixel permutation**

Images from [One-Sided Unsupervised Domain Mapping](), Sagie Benaim and Lior Wolf. NIPS 2017

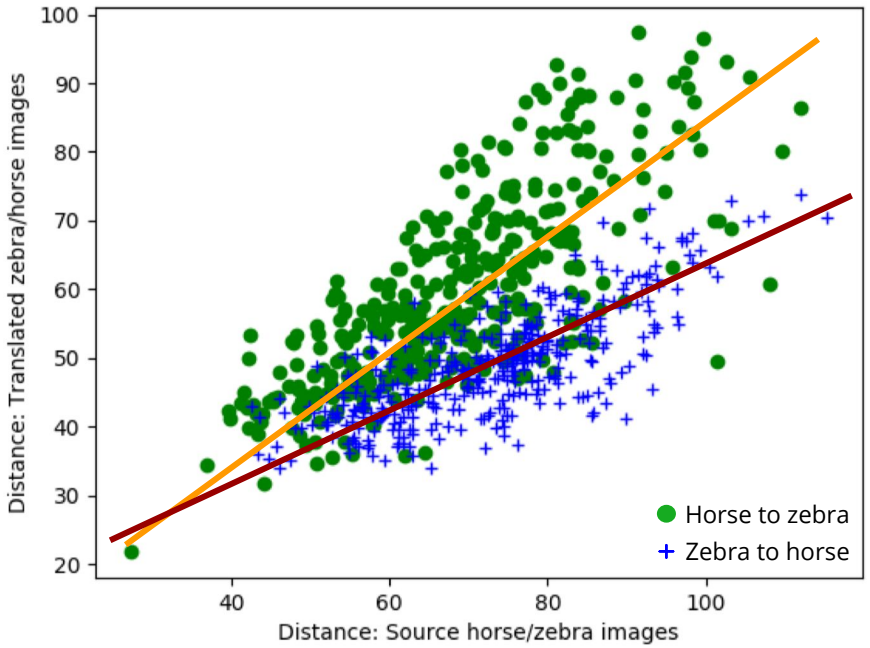
Justification - Nonnegative Matrix Approx.

- A **nonnegative linear transformation** T is learned to approximate CycleGAN's generator
- T ends up to be very similar to a **pixel permutation**

The distance between pair of samples is preserved after transformation

Images from One-Sided Unsupervised Domain Mapping, Sagie Benaim and Lior Wolf. NIPS 2017

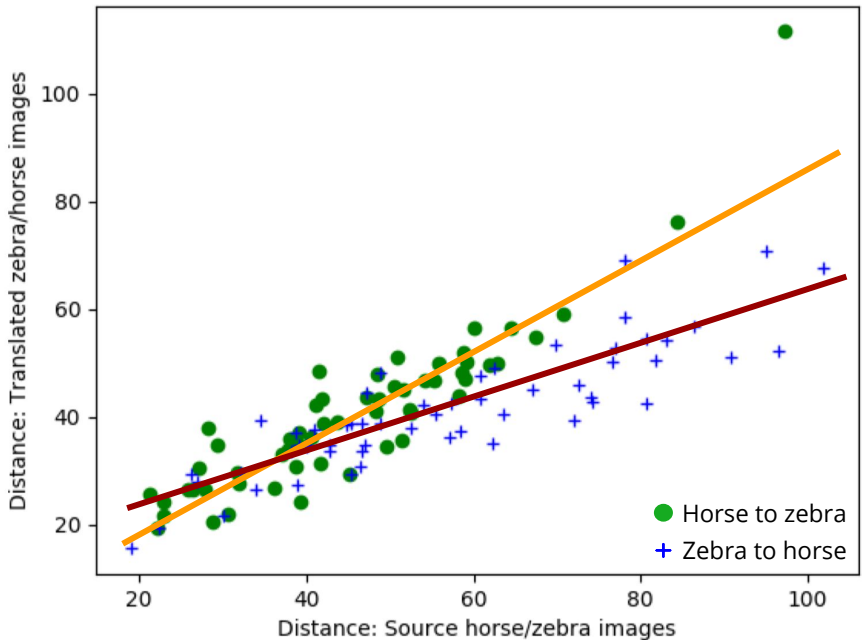
Justification - pairwise distance



- x-axis: distance of two samples in **source domain**
- y-axis: distance of the mapped samples in **target domain**
- Mapped by CycleGAN

Images from [One-Sided Unsupervised Domain Mapping](), Sagie Benaim and Lior Wolf. NIPS 2017

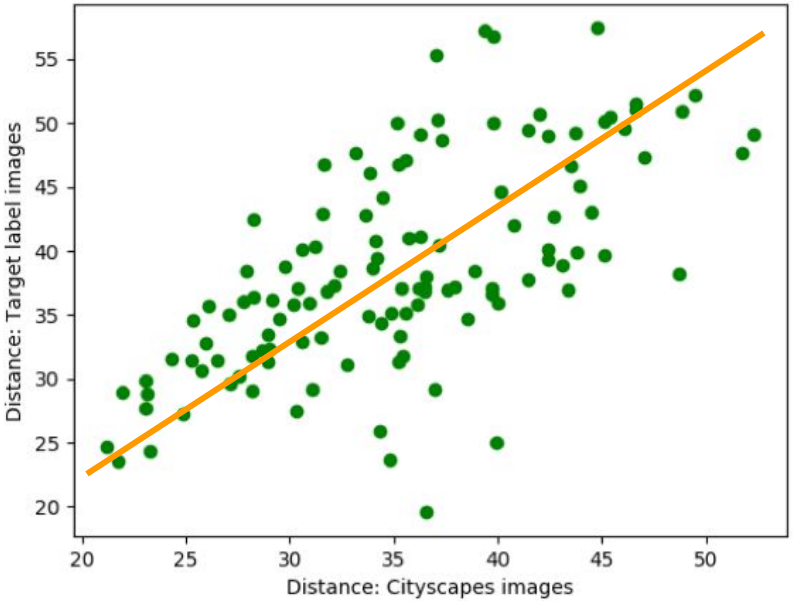
Justification - self distance



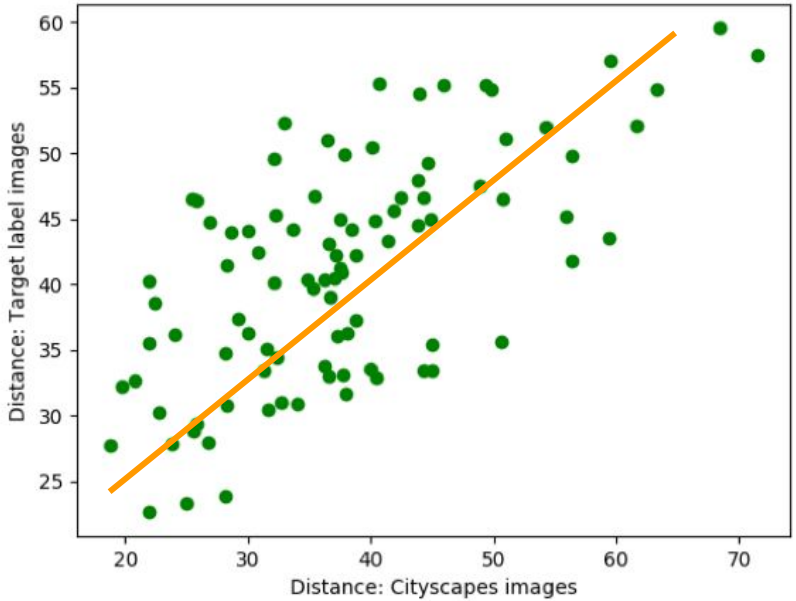
- x-axis: distance of **two halves** of one sample in **source domain**
- y-axis: distance of the **two halves** of the mapped sample in **target domain**
- Mapped by CycleGAN

Images from [One-Sided Unsupervised Domain Mapping](), Sagie Benaim and Lior Wolf. NIPS 2017

Justification - Another Evidence



pairwise distance



self-distance

Images from [One-Sided Unsupervised Domain Mapping](), Sagie Benaim and Lior Wolf. NIPS 2017

Back to the Point

★ Pairwise distances in source & target domains are **highly correlated**

★ Pairwise distances could be considered as a **new constraints**

Loss Used

Regular Adversarial Loss:

$$\mathcal{L}_{\text{GAN}}(G_{AB}, D_B, \hat{p}_A, \hat{p}_B) = \mathbb{E}_{x_B \sim \hat{p}_B}[\log D_B(x_B)] + \mathbb{E}_{x_A \sim \hat{p}_A}[\log(1 - D_B(G_{AB}(x_A))]$$

One-Sided Unsupervised Domain Mapping, Sagie Benaim and Lior Wolf. NIPS 2017

Loss Used

Regular Adversarial Loss:

$$\mathcal{L}_{\text{GAN}}(G_{AB}, D_B, \hat{p}_A, \hat{p}_B) = \mathbb{E}_{x_B \sim \hat{p}_B}[\log D_B(x_B)] + \mathbb{E}_{x_A \sim \hat{p}_A}[\log(1 - D_B(G_{AB}(x_A))]$$

Distance Loss **(main contribution)**:

$$\mathcal{L}_{\text{distance}}(G_{AB}, \hat{p}_A) = \mathbb{E}_{x_i, x_j \sim \hat{p}_A} \left| \frac{1}{\sigma_A}(\|x_i - x_j\|_1 - \mu_A) - \frac{1}{\sigma_B}(\|G_{AB}(x_i) - G_{AB}(x_j)\|_1 - \mu_B) \right|$$

Normalized L1 distance between $\mathbf{x_i}$ and $\mathbf{x_j}$

Normalized L1 distance between $\mathbf{G_{AB}(x_i)}$ and $\mathbf{G_{AB}(x_j)}$

The two distances are expected to be the same

One-Sided Unsupervised Domain Mapping, Sagie Benaim and Lior Wolf. NIPS 2017

Loss Used

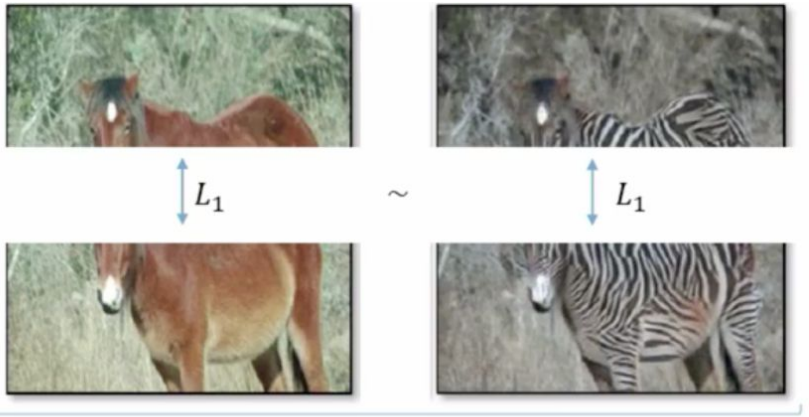


Illustration of Self Distance Loss

Self-distance Loss:

Normalized L1 distance between **two halves** of the input image x

$$\mathcal{L}_{\text{self-distance}}(G_{AB}, \hat{p}_A) = \mathbb{E}_{x \sim \hat{p}_A} \left| \frac{1}{\sigma_A}(\|L(x) - R(x)\|_1 - \mu_A) - \frac{1}{\sigma_B}(\|L(G_{AB}(x)) - R(G_{AB}(x))\|_1 - \mu_B) \right|$$

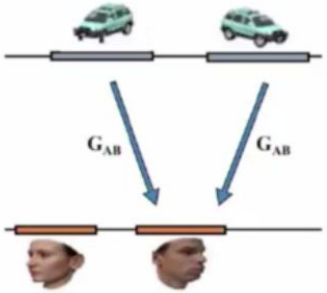
Normalized L1 distance between **two halves** of the generated image $G_{AB}(x)$

Where $L, R \in \mathbb{R}^{h \times w/2}$ (or $\mathbb{R}^{h/2 \times w}$) are operators that give a **half** of the input image x

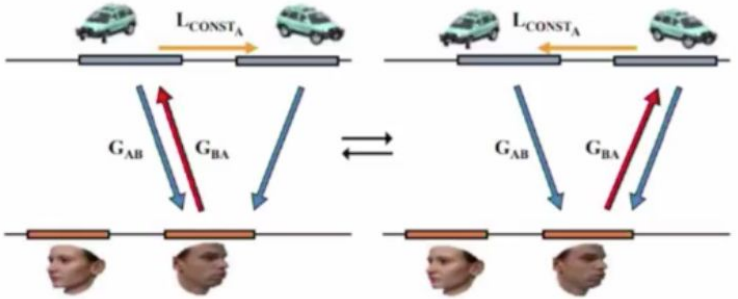
Images from the talk of DistanceGAN given at NIPS 2017, by Sagie Benaim

Solves Asymmetry Problem

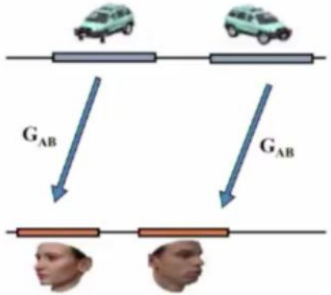
- GAN:



Cycle:



- Distance:



Images from the [talk of DistanceGAN given at NIPS 2017](), by Sagie Benaim

Full Loss Function

$$\alpha_{1A}\mathcal{L}_{\text{GAN}}(G_{AB}, D_B, \hat{p}_A, \hat{p}_B) + \alpha_{1B}\mathcal{L}_{\text{GAN}}(G_{BA}, D_A, \hat{p}_B, \hat{p}_A) + \alpha_{2A}\mathcal{L}_{\text{cycle}}(G_{AB}, G_{BA}, \hat{p}_A) +$$
$$\alpha_{2B}\mathcal{L}_{\text{cycle}}(G_{BA}, G_{AB}, \hat{p}_B) + \alpha_{3A}\mathcal{L}_{\text{distance}}(G_{AB}, \hat{p}_A) + \alpha_{3B}\mathcal{L}_{\text{distance}}(G_{BA}, \hat{p}_B) +$$
$$\alpha_{4A}\mathcal{L}_{\text{self-distance}}(G_{AB}, \hat{p}_A) + \alpha_{4B}\mathcal{L}_{\text{self-distance}}(G_{BA}, \hat{p}_B)$$

Goal: minimize over G and maximize over D

α_{iA}, α_{iB} are trade-off parameters

When performing **one-sided** mapping from A to B:

→All α set to 0 except for α_{1A} and either α_{3A} or α_{4A}

One-Sided Unsupervised Domain Mapping, Sagie Benaim and Lior Wolf. NIPS 2017

Variants Methods

One-sided distance:

$$\alpha_{1A}\mathcal{L}_{\text{GAN}}(G_{AB}, D_B, \hat{p}_A, \hat{p}_B) + \alpha_{1B}\mathcal{L}_{\text{GAN}}(G_{BA}, D_A, \hat{p}_B, \hat{p}_A) + \alpha_{2A}\mathcal{L}_{\text{cycle}}(G_{AB}, G_{BA}, \hat{p}_A) + \alpha_{2B}\mathcal{L}_{\text{cycle}}(G_{BA}, G_{AB}, \hat{p}_B) + \alpha_{3A}\mathcal{L}_{\text{distance}}(G_{AB}, \hat{p}_A) + \alpha_{3B}\mathcal{L}_{\text{distance}}(G_{BA}, \hat{p}_B) + \alpha_{4A}\mathcal{L}_{\text{self-distance}}(G_{AB}, \hat{p}_A) + \alpha_{4B}\mathcal{L}_{\text{self-distance}}(G_{BA}, \hat{p}_B)$$

One-sided self-distance:

$$\alpha_{1A}\mathcal{L}_{\text{GAN}}(G_{AB}, D_B, \hat{p}_A, \hat{p}_B) + \alpha_{1B}\mathcal{L}_{\text{GAN}}(G_{BA}, D_A, \hat{p}_B, \hat{p}_A) + \alpha_{2A}\mathcal{L}_{\text{cycle}}(G_{AB}, G_{BA}, \hat{p}_A) + \alpha_{2B}\mathcal{L}_{\text{cycle}}(G_{BA}, G_{AB}, \hat{p}_B) + \alpha_{3A}\mathcal{L}_{\text{distance}}(G_{AB}, \hat{p}_A) + \alpha_{3B}\mathcal{L}_{\text{distance}}(G_{BA}, \hat{p}_B) + \alpha_{4A}\mathcal{L}_{\text{self-distance}}(G_{AB}, \hat{p}_A) + \alpha_{4B}\mathcal{L}_{\text{self-distance}}(G_{BA}, \hat{p}_B)$$

*In order to perform one-sided learning, either the loss of A → B or B → A will be used

One-Sided Unsupervised Domain Mapping, Sagie Benaim and Lior Wolf. NIPS 2017

Variants Methods

Cycle + distance:

$$\alpha_{1A}\mathcal{L}_{\text{GAN}}(G_{AB}, D_B, \hat{p}_A, \hat{p}_B) + \alpha_{1B}\mathcal{L}_{\text{GAN}}(G_{BA}, D_A, \hat{p}_B, \hat{p}_A) + \alpha_{2A}\mathcal{L}_{\text{cycle}}(G_{AB}, G_{BA}, \hat{p}_A) +$$
$$\alpha_{2B}\mathcal{L}_{\text{cycle}}(G_{BA}, G_{AB}, \hat{p}_B) + \alpha_{3A}\mathcal{L}_{\text{distance}}(G_{AB}, \hat{p}_A) + \alpha_{3B}\mathcal{L}_{\text{distance}}(G_{BA}, \hat{p}_B) +$$
$$\alpha_{4A}\mathcal{L}_{\text{self-distance}}(G_{AB}, \hat{p}_A) + \alpha_{4B}\mathcal{L}_{\text{self-distance}}(G_{BA}, \hat{p}_B)$$

*pairwise distance and self-distance constraints are not tested jointly

Network Architecture:

Discriminators and generators are implemented based on both **DiscoGAN** and **CycleGAN** for comparison

One-Sided Unsupervised Domain Mapping, Sagie Benaim and Lior Wolf. NIPS 2017

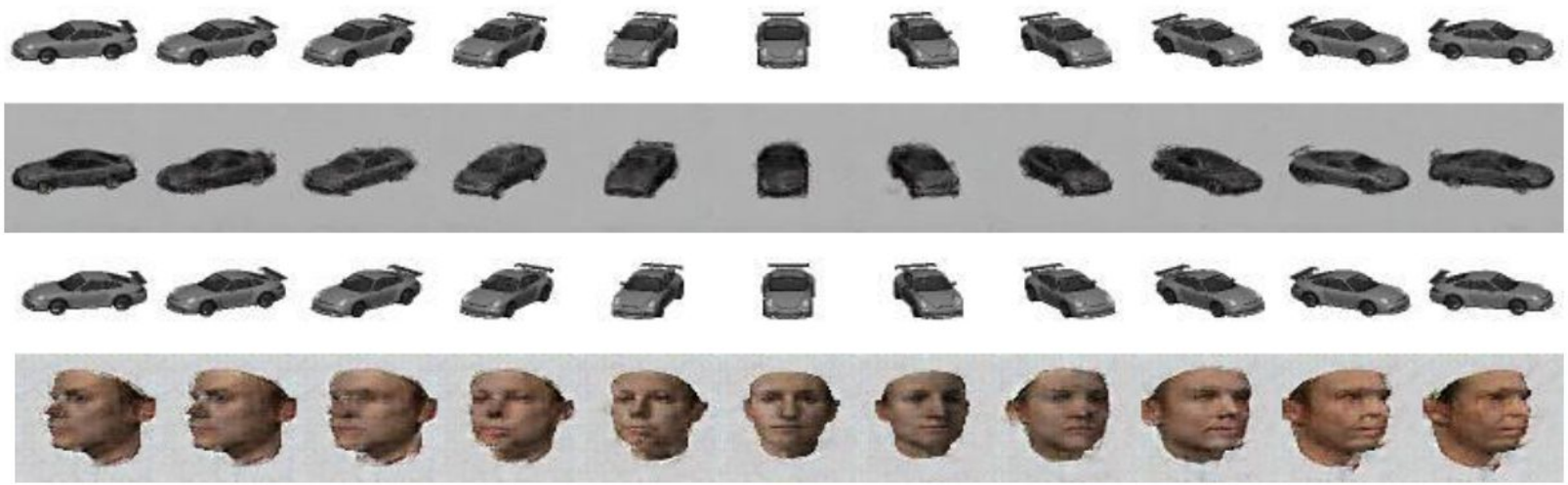
Results

Table 4: CelebA mapping results using the VGG face descriptor.

Method	Male → Female		Blond → Black		Glasses → Without	
	Cosine Similarity	Separation Accuracy	Cosine Similarity	Separation Accuracy	Cosine Similarity	Separation Accuracy
DiscoGAN	0.23	0.87	0.15	0.89	0.13	**0.84**
Distance	0.32	**0.88**	**0.24**	**0.92**	**0.42**	0.79
Distance+Cycle	**0.35**	0.87	**0.24**	0.91	0.41	0.82
Self Distance	0.24	0.86	**0.24**	0.91	0.34	0.80
Other direction						
DiscoGAN	0.22	0.86	0.14	0.91	0.10	**0.90**
Distance	0.26	0.87	**0.22**	**0.96**	**0.30**	0.89
Distance+Cycle	**0.31**	0.89	**0.22**	0.95	**0.30**	0.85
Self Distance	0.24	**0.91**	0.19	0.94	**0.30**	0.81

Images from [One-Sided Unsupervised Domain Mapping](), Sagie Benaim and Lior Wolf. NIPS 2017

Results



Method	car2car	car2head
DiscoGAN	0.306	0.137
Distance	0.135	**0.097**
Dist.+Cycle	**0.098**	0.273
Self Dist.	0.117	0.197

Images from [One-Sided Unsupervised Domain Mapping](), Sagie Benaim and Lior Wolf. NIPS 2017

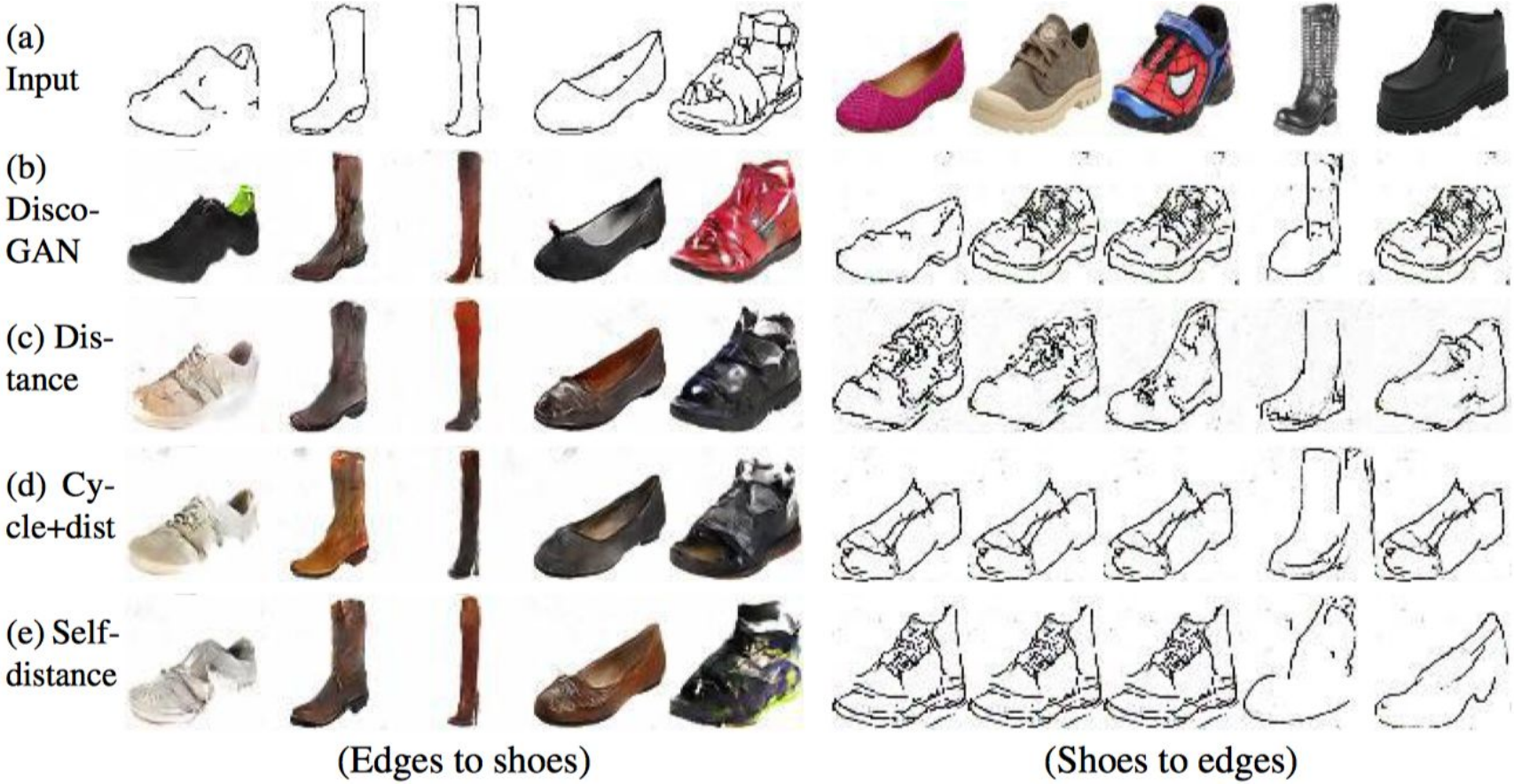
Results

Images from [One-Sided Unsupervised Domain Mapping](), Sagie Benaim and Lior Wolf. NIPS 2017

Results

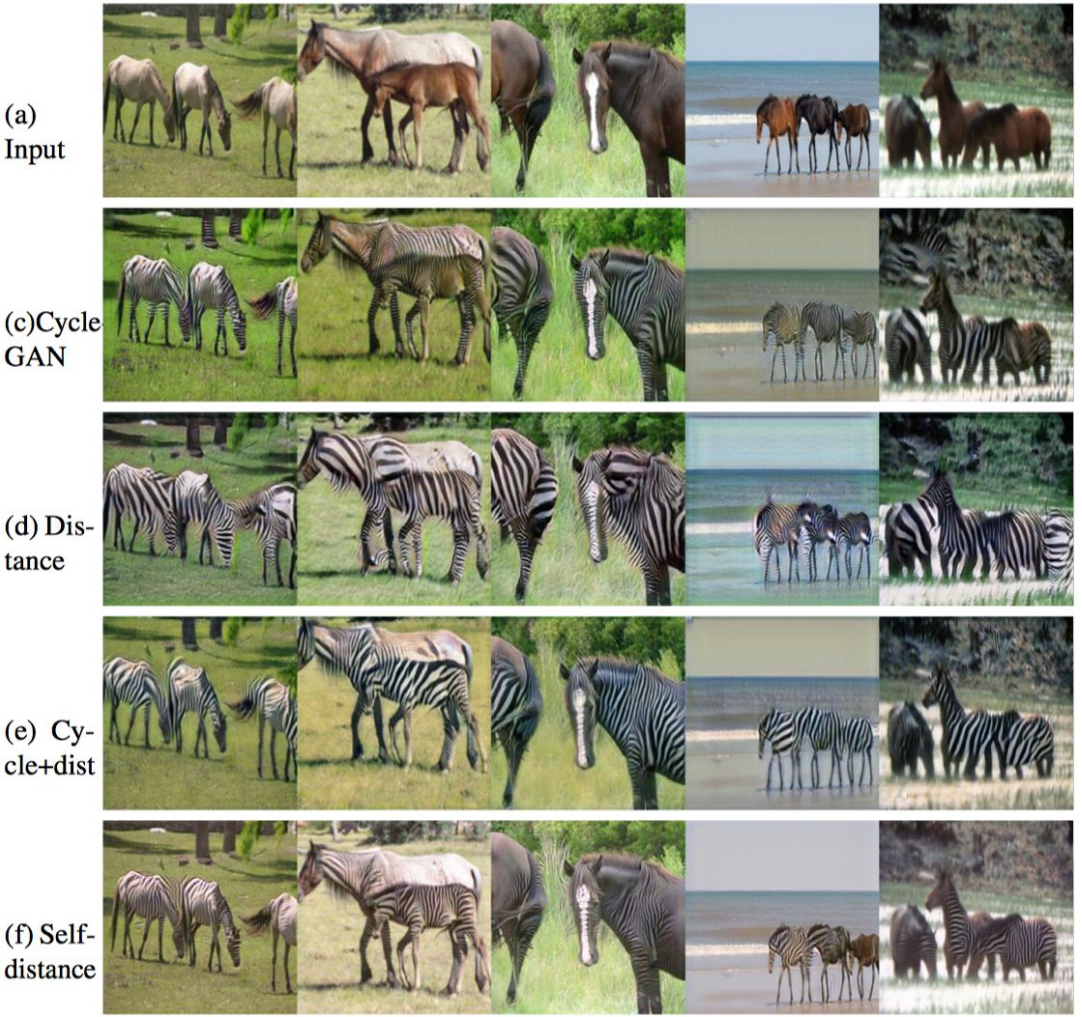
Images from [One-Sided Unsupervised Domain Mapping](), Sagie Benaim and Lior Wolf. NIPS 2017

Results



Images from [One-Sided Unsupervised Domain Mapping](), Sagie Benaim and Lior Wolf. NIPS 2017

Results



Images from [One-Sided Unsupervised Domain Mapping](), Sagie Benaim and Lior Wolf. NIPS 2017

Results

Images from [One-Sided Unsupervised Domain Mapping](), Sagie Benaim and Lior Wolf. NIPS 2017

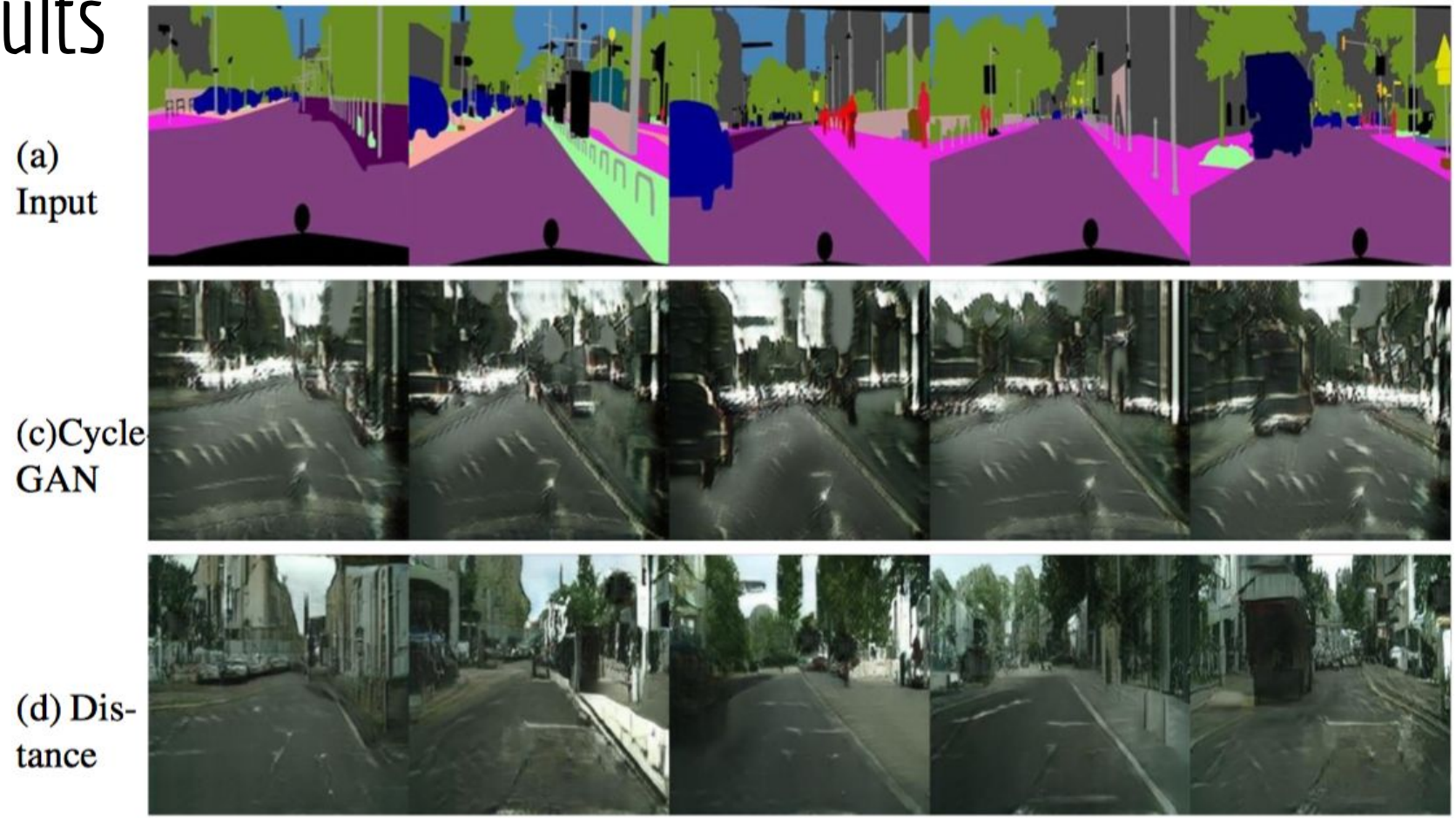
Results

Table 3: MNIST classification on mapped SHVN images

Method	Accuracy
CycleGAN	26.1%
Distance	**26.8%**
Dist.+Cycle	18.0%
Self Dist.	25.2%

Images from [One-Sided Unsupervised Domain Mapping](), Sagie Benaim and Lior Wolf. NIPS 2017

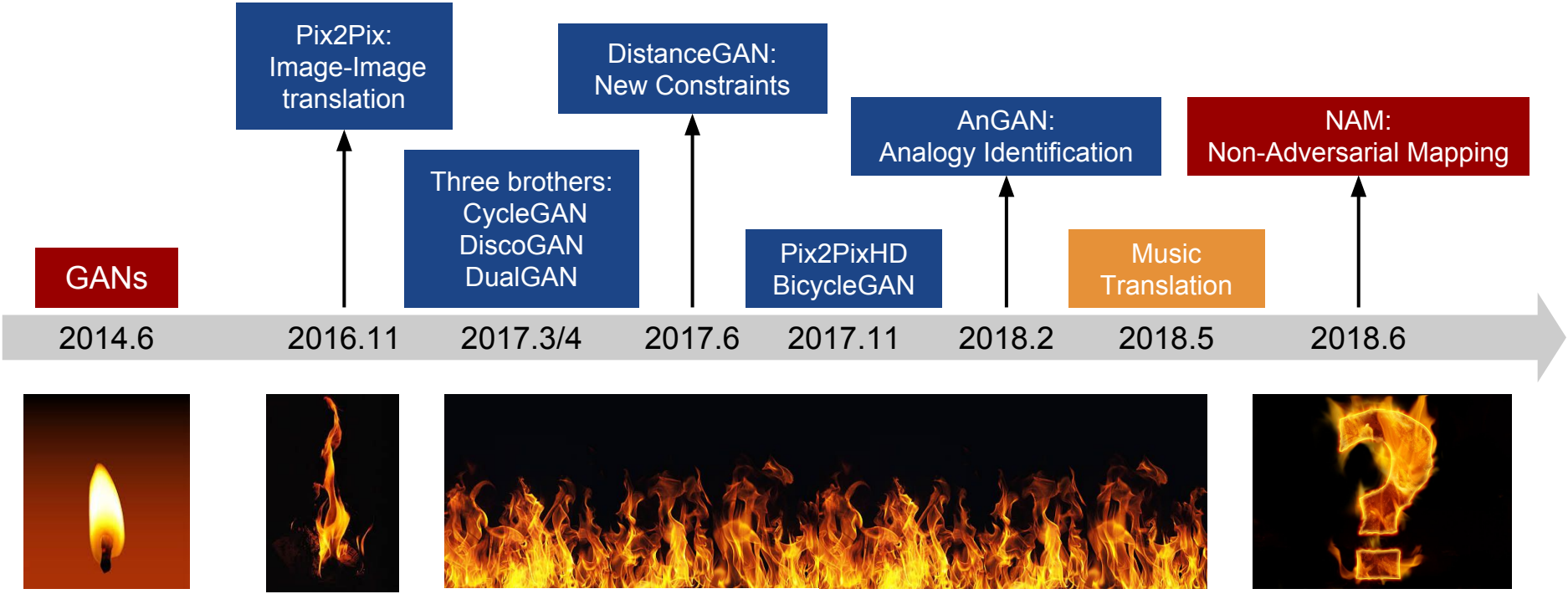
Results



Images from [One-Sided Unsupervised Domain Mapping](), Sagie Benaim and Lior Wolf. NIPS 2017

To Conclude Today's Talk

Timeline of Domain Transfer



Take Home Messages

- GANs can be used to learn a mapping from one distribution to another
- Two main directions - Paired and Unpaired datasets
- Typical architectures: DCGANs, U-Nets, PatchGAN + encoders (multimodal)
- Current approaches are to find useful constraints to improve performance
- Domain Transfer is still a fast evolving field covering many domains (not just images)

THE END

Have a Good Summer!