Copyright Notice

deeplearning.ai

Unpaired Image-to-Image Translation

Outline

- Paired vs. unpaired image-to-image translation
- Unpaired image-to-image translation
  - Mapping between two piles of image styles
  - Finding commonalities and differences

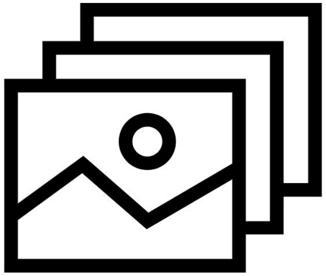


Image-to-Image Translation

Edges to photo

Paired images

Available from: https://arxiv.org/abs/1611.07004

Image-to-Image Translation

Edges to photo

Paired images

Monet to photo

Unpaired images

Available from: https://arxiv.org/abs/1611.07004

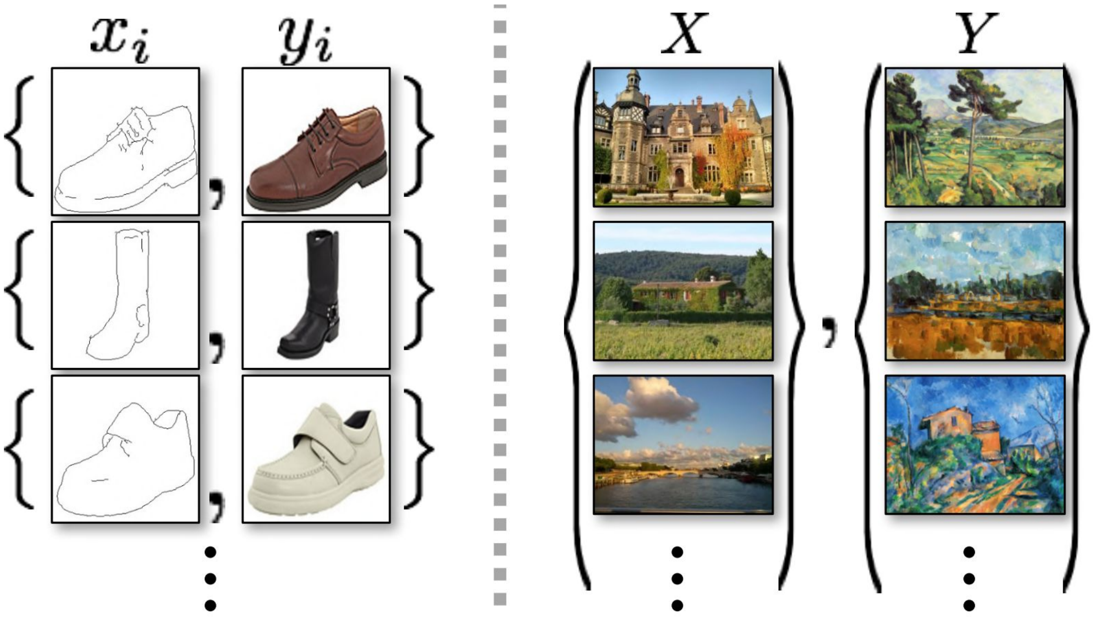
Image-to-Image Translation

Paired images

Available from: https://arxiv.org/abs/1703.10593

Image-to-Image Translation

Paired images



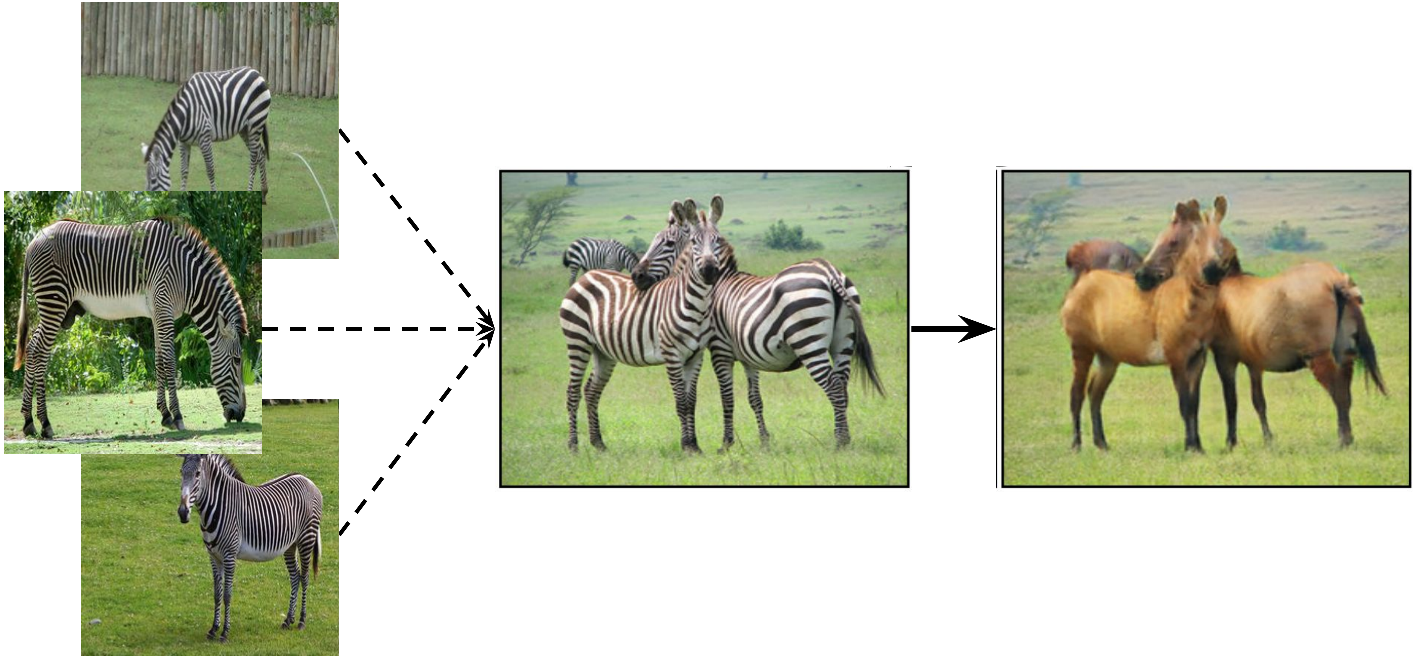
Unpaired images

Available from: https://arxiv.org/abs/1703.10593

Unpaired Image-to-Image Translation

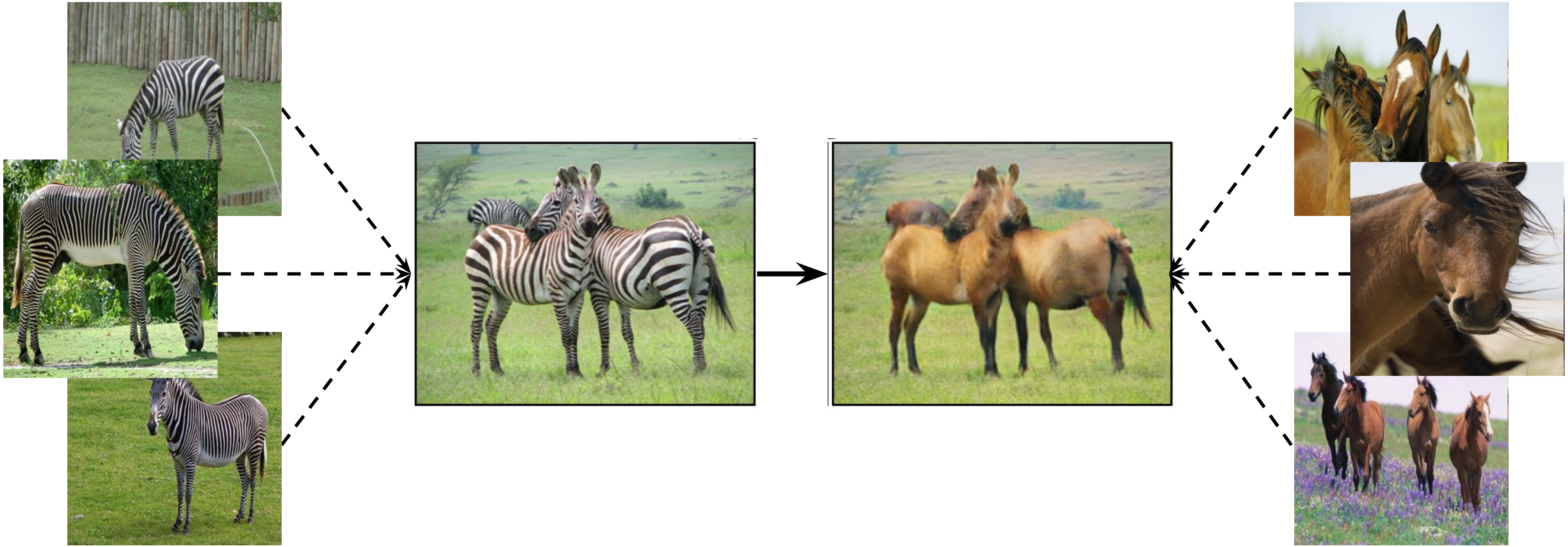
Available from: https://arxiv.org/abs/1703.10593

Mapping Between Two Piles



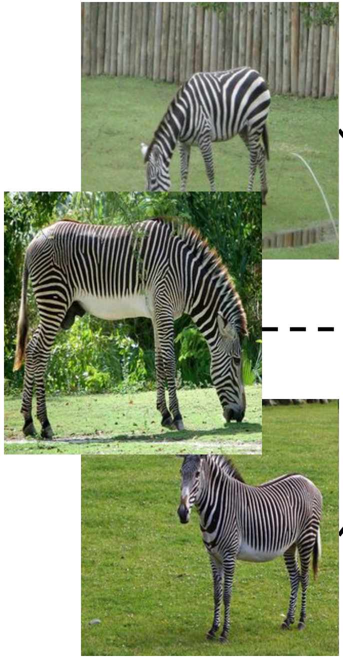
(Center) Images available from: https://arxiv.org/abs/1703.10593
(Side) Images available from: https://github.com/togheppi/CycleGAN

Mapping Between Two Piles

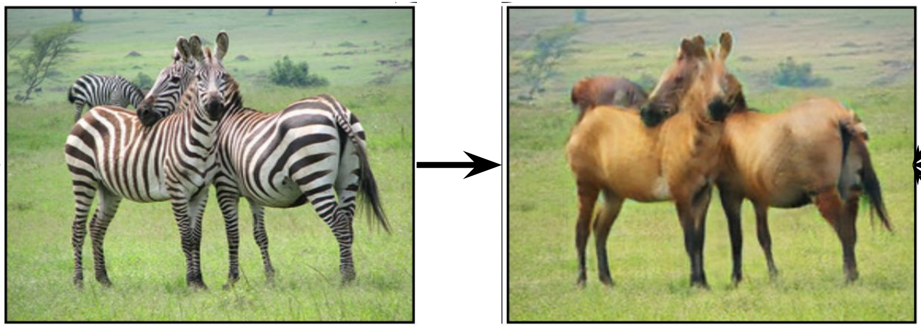


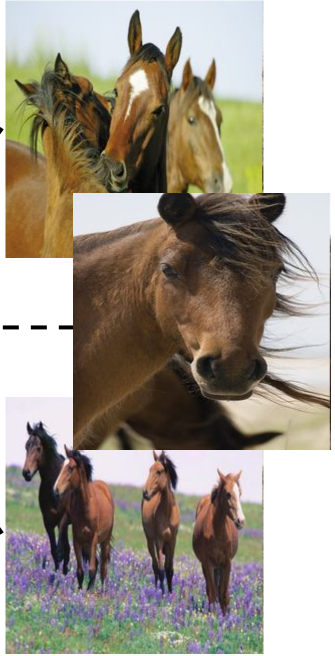
(Center) Images available from: https://arxiv.org/abs/1703.10593
(Sides) Images available from: https://github.com/togheppi/CycleGAN

Mapping Between Two Piles



Content = common elements
Style = unique elements

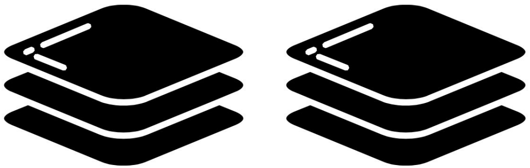




(Center) Images available from: https://arxiv.org/abs/1703.10593
(Sides) Images available from: https://github.com/togheppi/CycleGAN

Summary

- Unpaired image-to-image translation:
  - Learns a mapping between two piles of images
  - Examines common elements of the two piles (content) and unique elements of each pile (style)
- Unlike paired image-to-image translation, this method is unsupervised

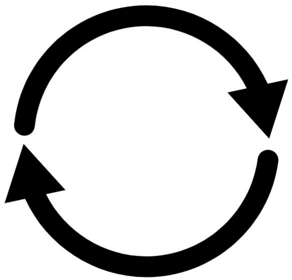


deeplearning.ai

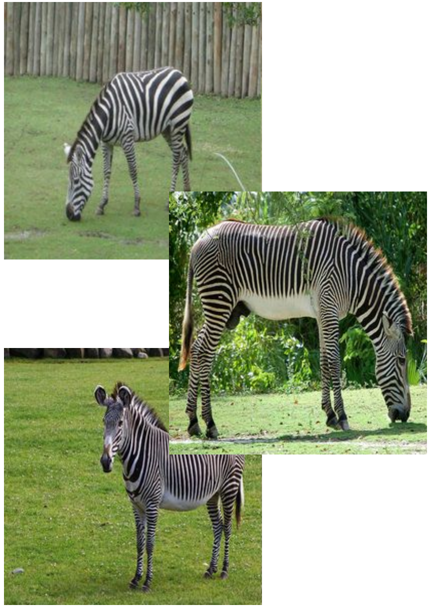
CycleGAN Overview

Outline

- Overview of CycleGAN
  - The “Cycle” in CycleGAN
  - Two GANs!



Cycle Consistency



?

Images available from: https://github.com/togheppi/CycleGAN

Cycle Consistency

Real

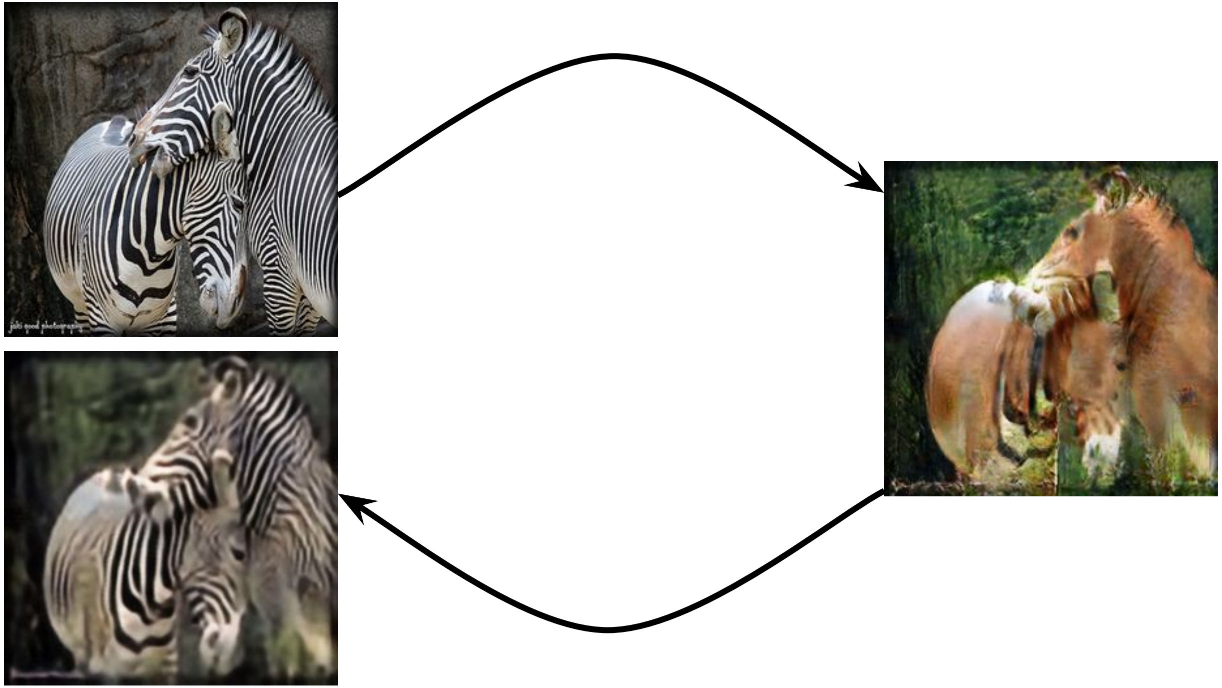
Fake

Images available from: https://github.com/togheppi/CycleGAN

Cycle Consistency

Real

Fake



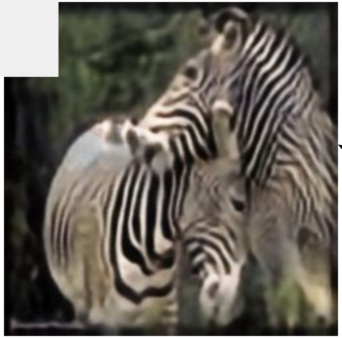
Fake

Images available from: https://github.com/togheppi/CycleGAN

Cycle Consistency

Real

Should be the same

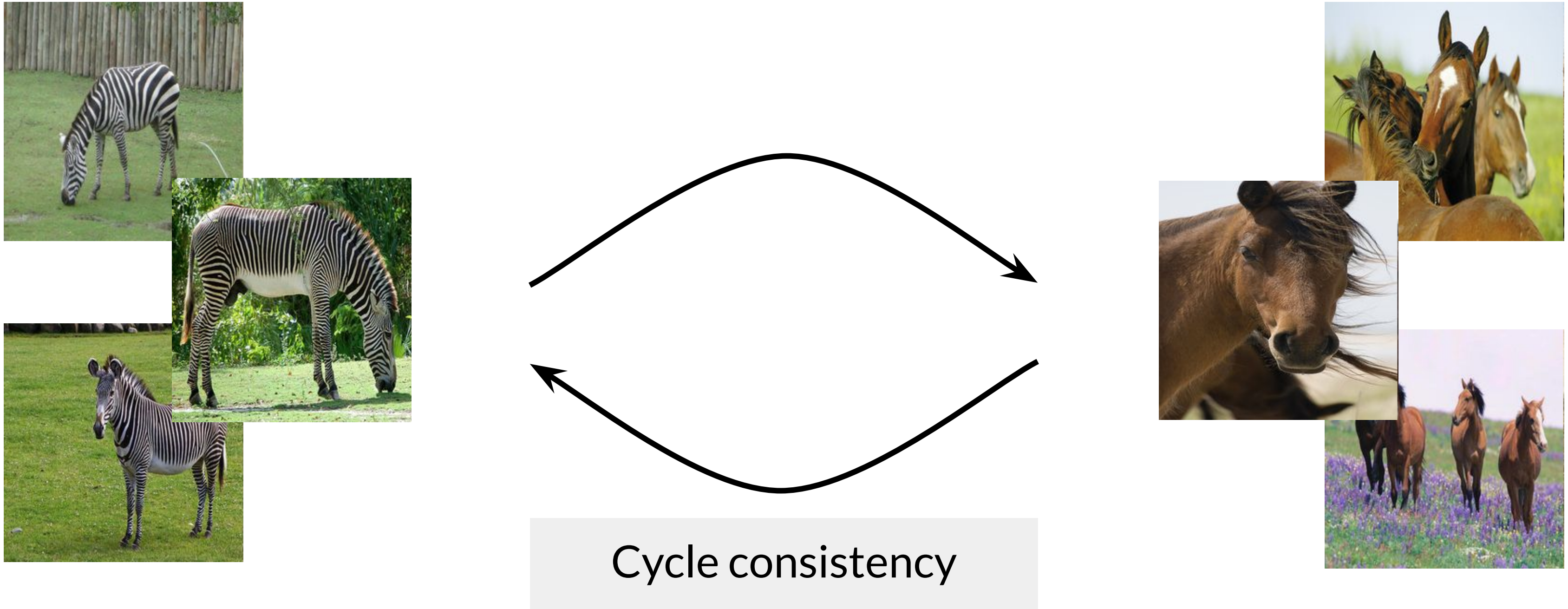


Fake

Fake

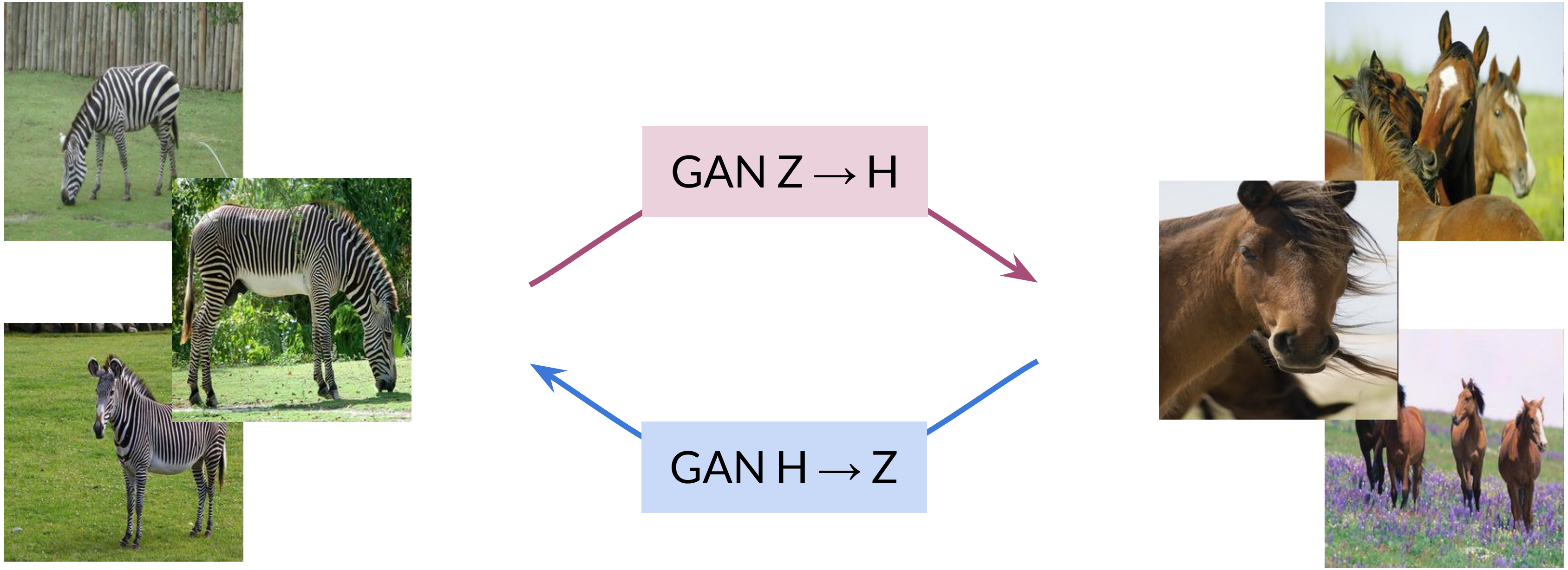
Images available from: https://github.com/togheppi/CycleGAN

Cycle Consistency



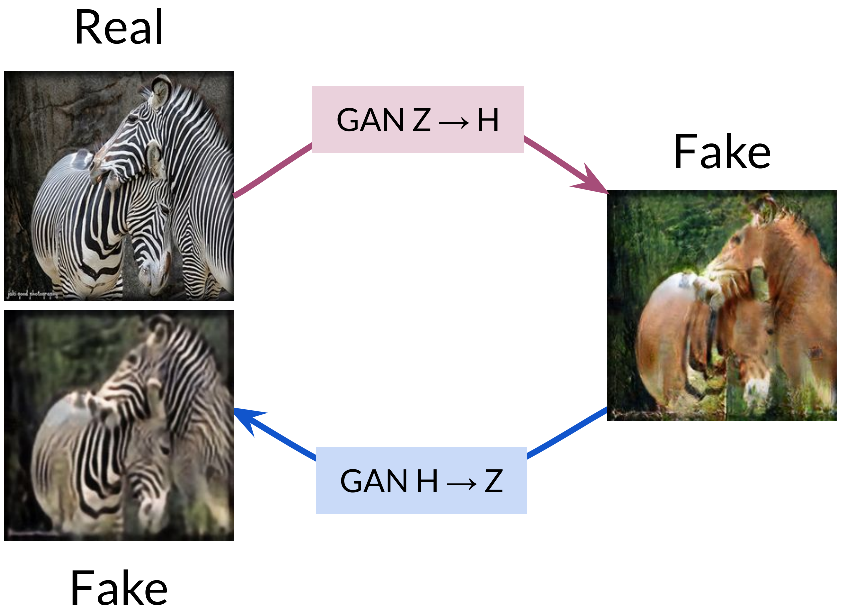
Images available from: https://github.com/togheppi/CycleGAN

Two GANs



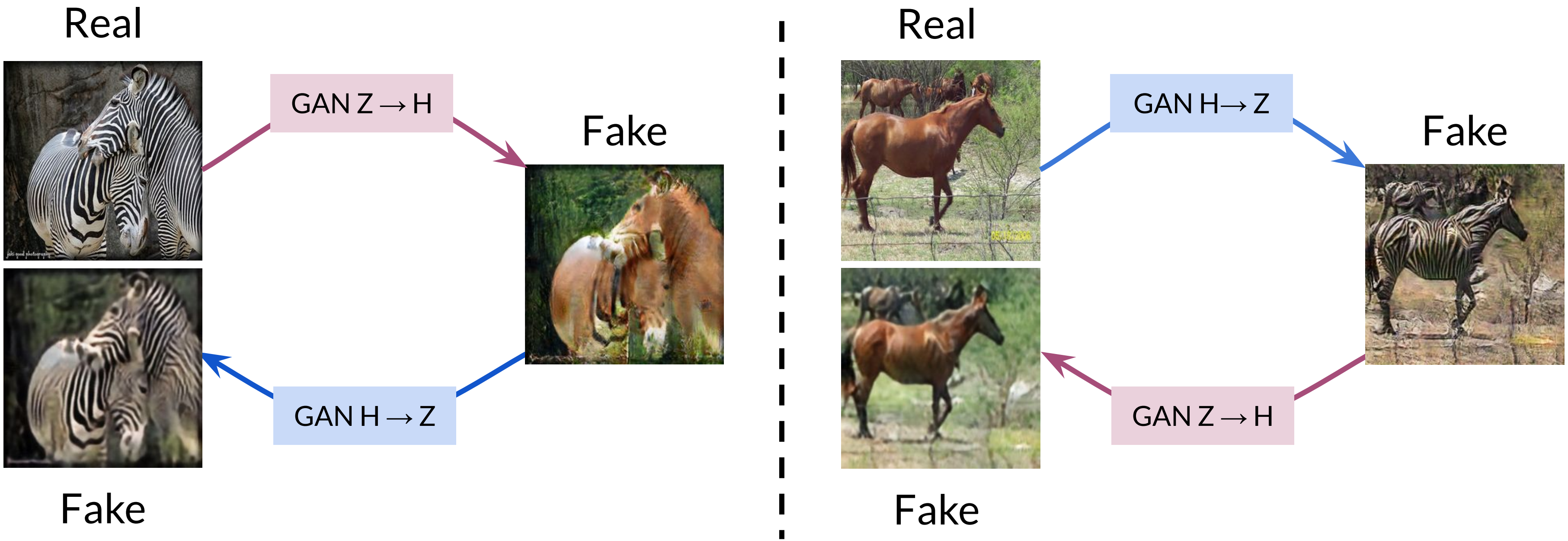
Images available from: https://github.com/togheppi/CycleGAN

Two GANs



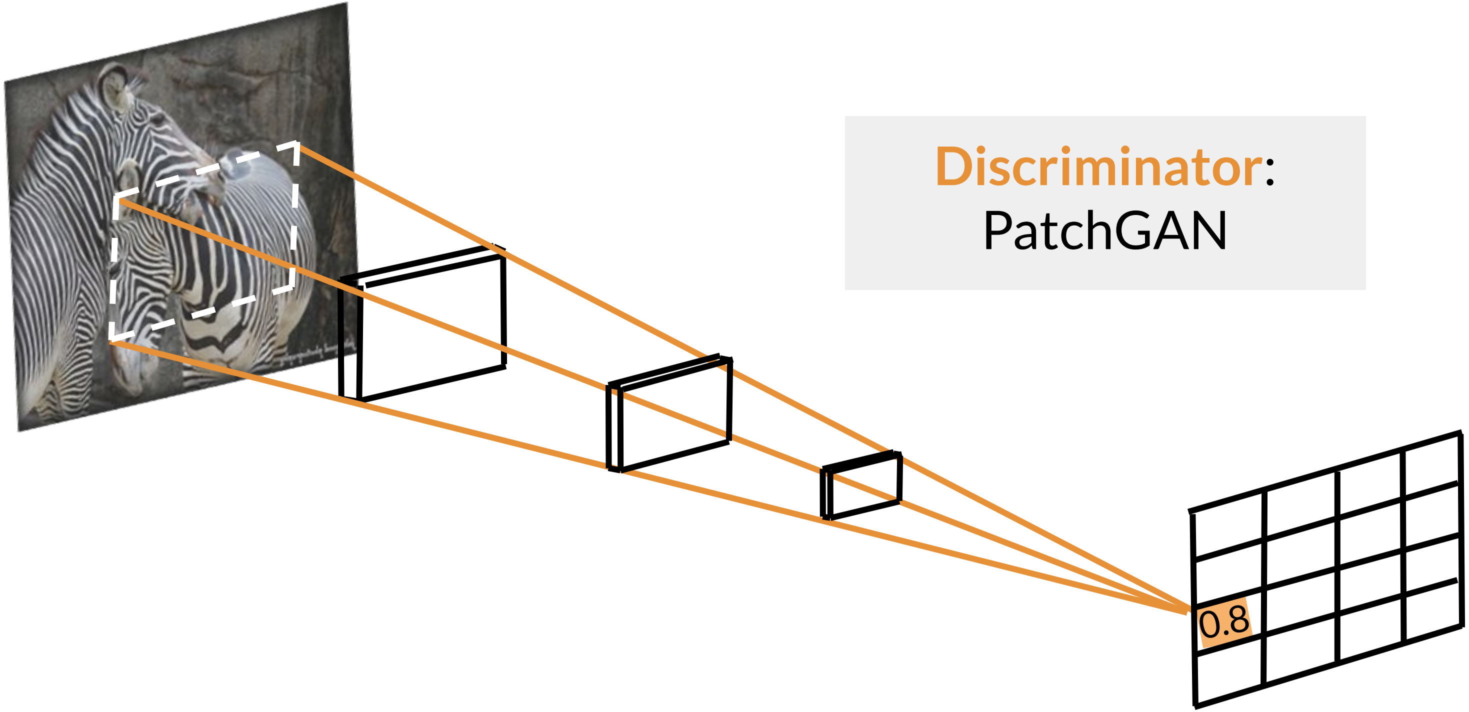
Images available from: https://github.com/togheppi/CycleGAN

Two GANs



Images available from: https://github.com/togheppi/CycleGAN

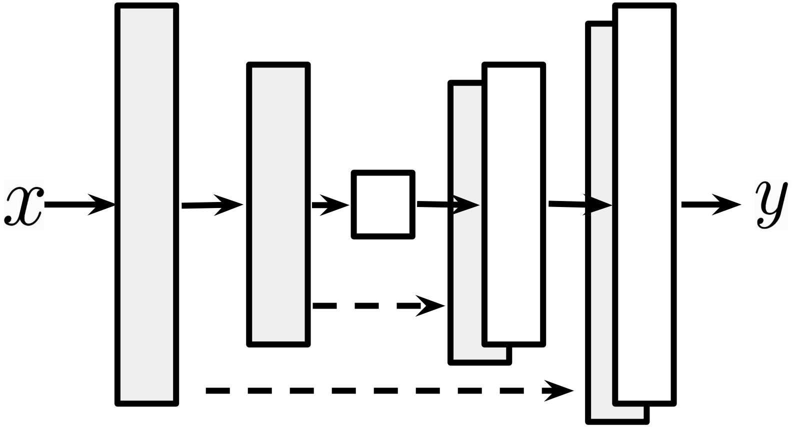
CycleGAN



Discriminator: PatchGAN

Image available from: https://github.com/togheppi/CycleGAN

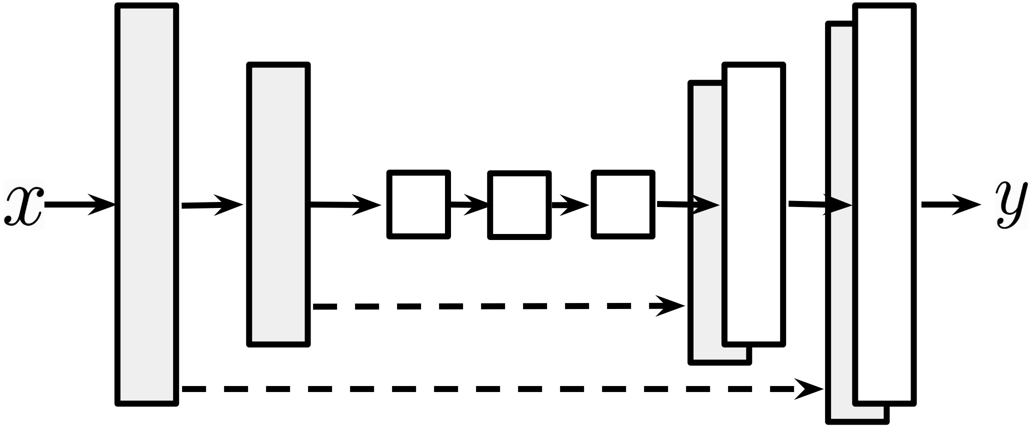
CycleGAN



Generator ≈ U-Net

Available from: https://arxiv.org/abs/1611.07004

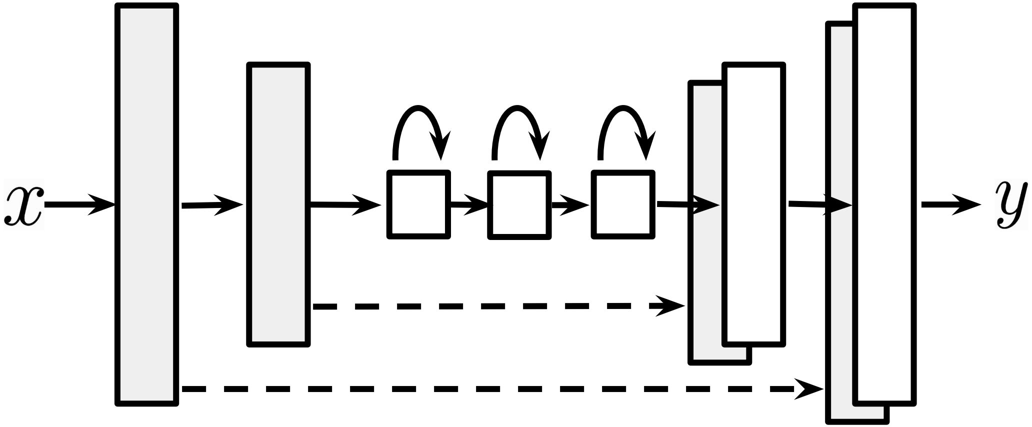
CycleGAN



Generator ≈
U-Net + DCGAN
generator

Available from: https://arxiv.org/abs/1611.07004

CycleGAN



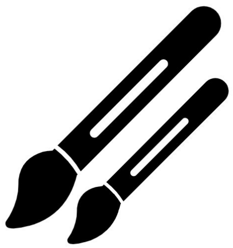
Additional skip connections

Generator ≈
U-Net + DCGAN generator

Available from: https://arxiv.org/abs/1611.07004

Summary

- CycleGAN uses two GANs for unpaired image-to-image translation
- The discriminators are PatchGAN's
- The generators are similar to a U-Net and DCGAN generator with additional skip connections

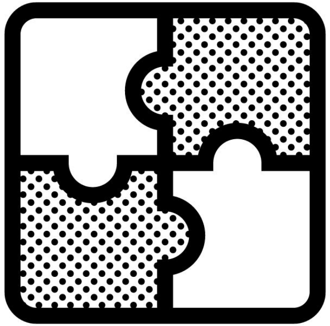


deeplearning.ai

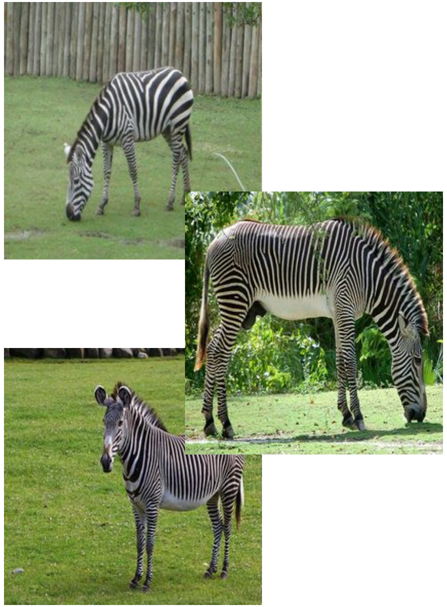
CycleGAN: Two GANs

Outline

- Two GANs, four components
  - Two generators
  - Two discriminators



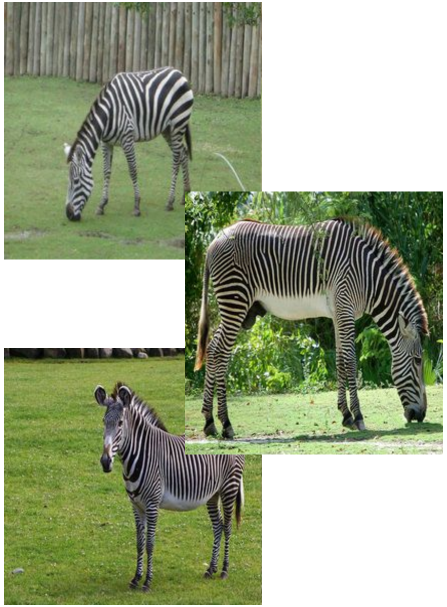
CycleGAN Components



GAN Z $\rightarrow$ H

Images available from: https://github.com/togheppi/CycleGAN

CycleGAN Components

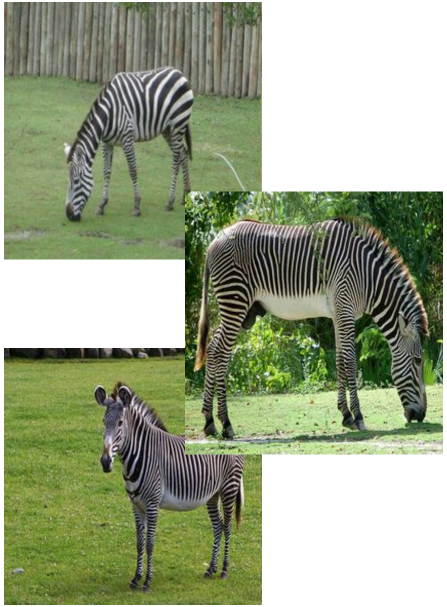


GAN Z → H

GAN H → Z

Images available from: https://github.com/togheppi/CycleGAN

CycleGAN Components



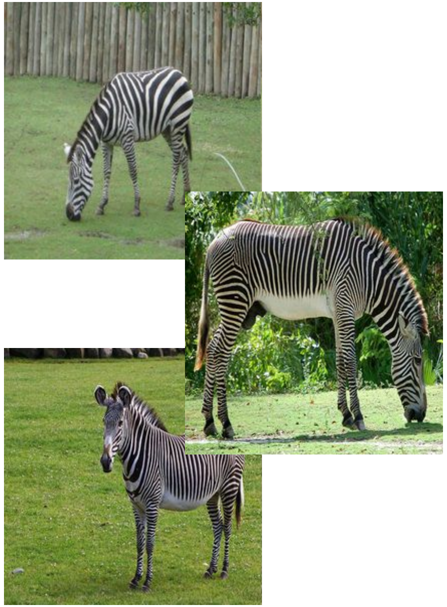
Generator Z → H
Discriminator H

GAN Z → H

GAN H → Z

Images available from: https://github.com/togheppi/CycleGAN

CycleGAN Components



Generator Z → H
Discriminator H

GAN Z → H

GAN H → Z

Generator H → Z
Discriminator Z

Images available from: https://github.com/togheppi/CycleGAN

GAN Z → H

Real

Generator
Z → H

Fake

Images available from: https://github.com/togheppi/CycleGAN

GAN $Z \rightarrow H$

Real

Generator
$Z \rightarrow H$

Fake

Discriminator
H

Real

Discriminator
H

Images available from: https://github.com/togheppi/CycleGAN

GAN Z → H

Real

→ Generator
Z → H →

Fake

→ Discriminator
H → Classification matrix

Real

→ Discriminator
H → Classification matrix

Images available from: https://github.com/togheppi/CycleGAN

GAN $H \rightarrow Z$

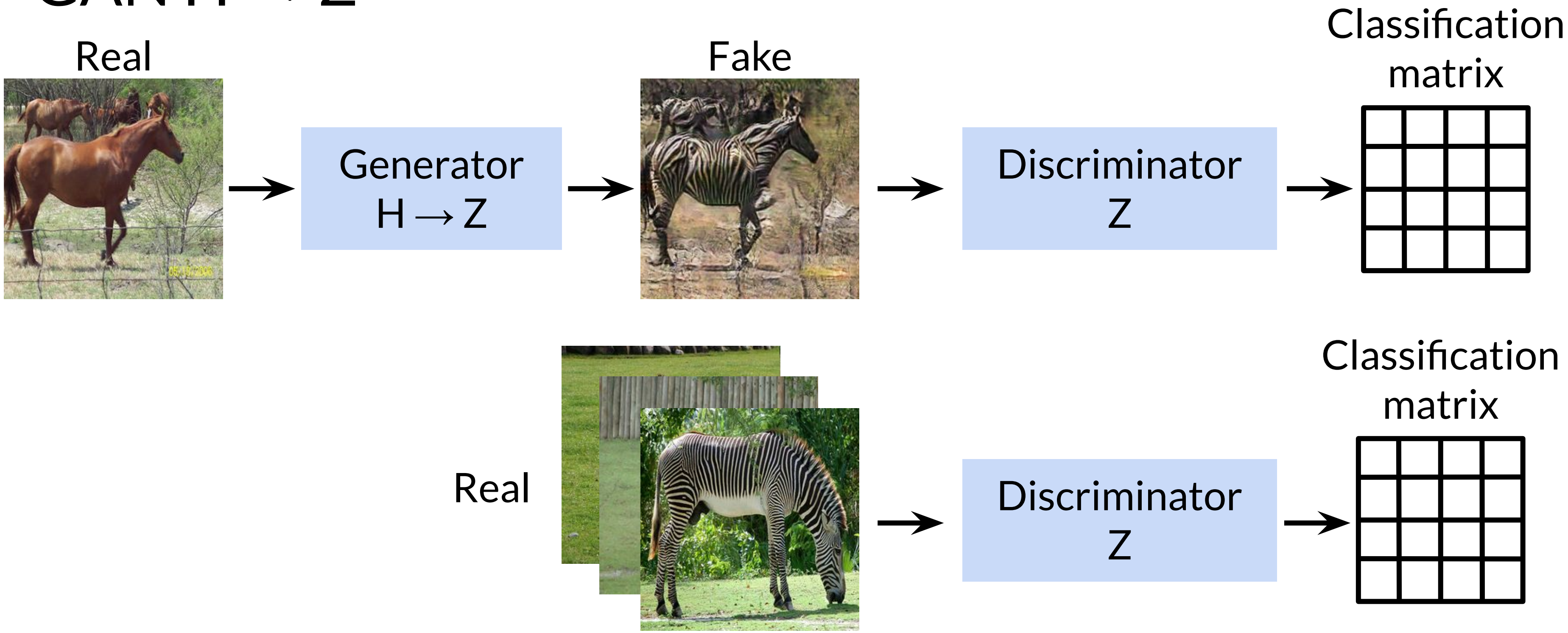
Real

Generator
$H \rightarrow Z$

Fake

Images available from: https://github.com/togheppi/CycleGAN

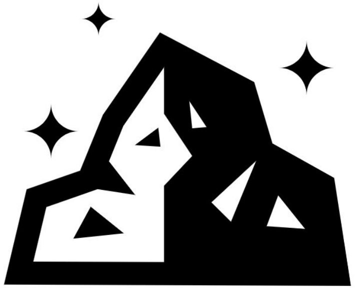
GAN H → Z



Images available from: https://github.com/togheppi/CycleGAN

Summary

- CycleGAN has four components:
  - Two generators
  - Two discriminators
- The inputs to the generators and discriminators are similar to Pix2Pix, except:
  - There are no real target outputs
  - Each discriminator is in charge of one pile of images

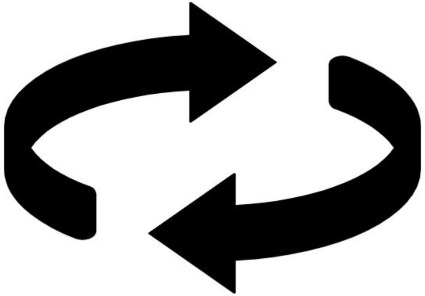


CycleGAN: Cycle Consistency

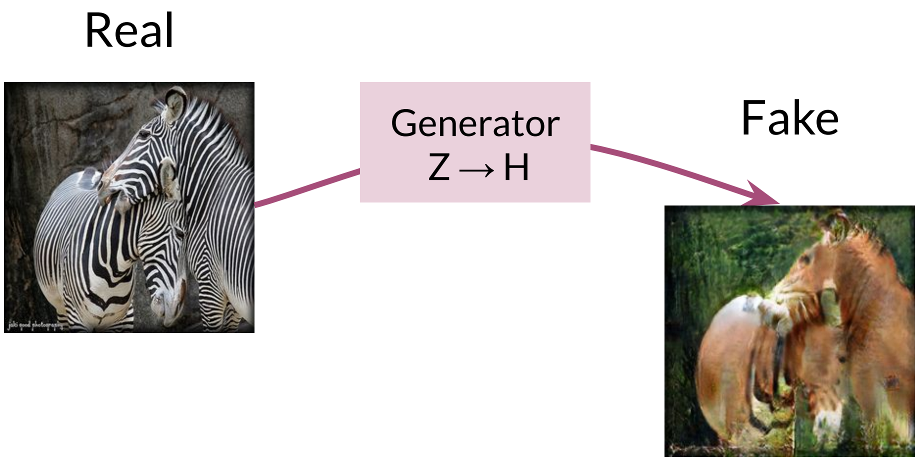
deeplearning.ai

Outline

- Encouraging cycle consistency
  - Cycle Consistency Loss term
- Loss with cycle consistency for each of two GANs
- How cycle consistency helps

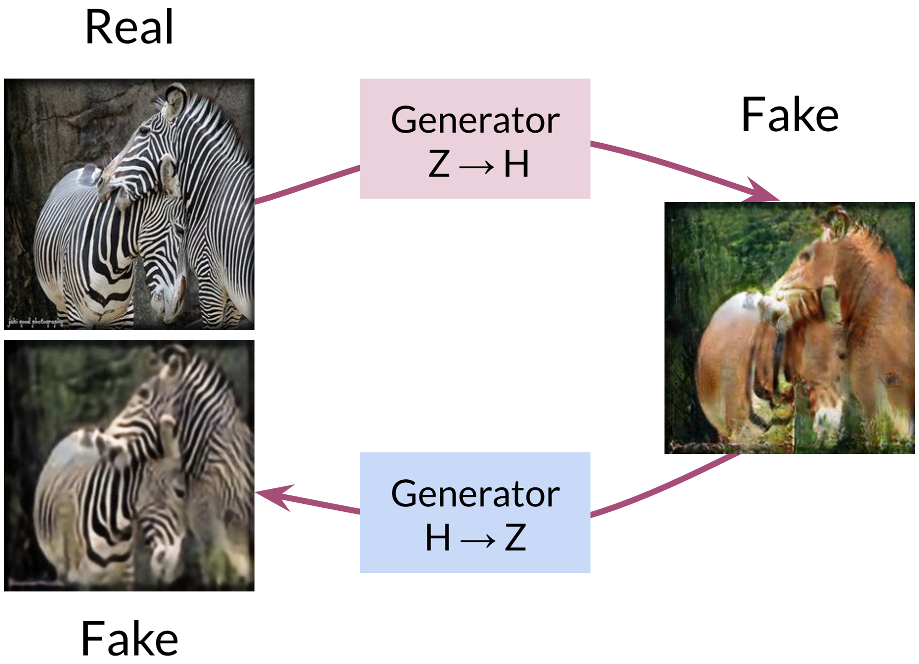


Cycle Consistency Loss



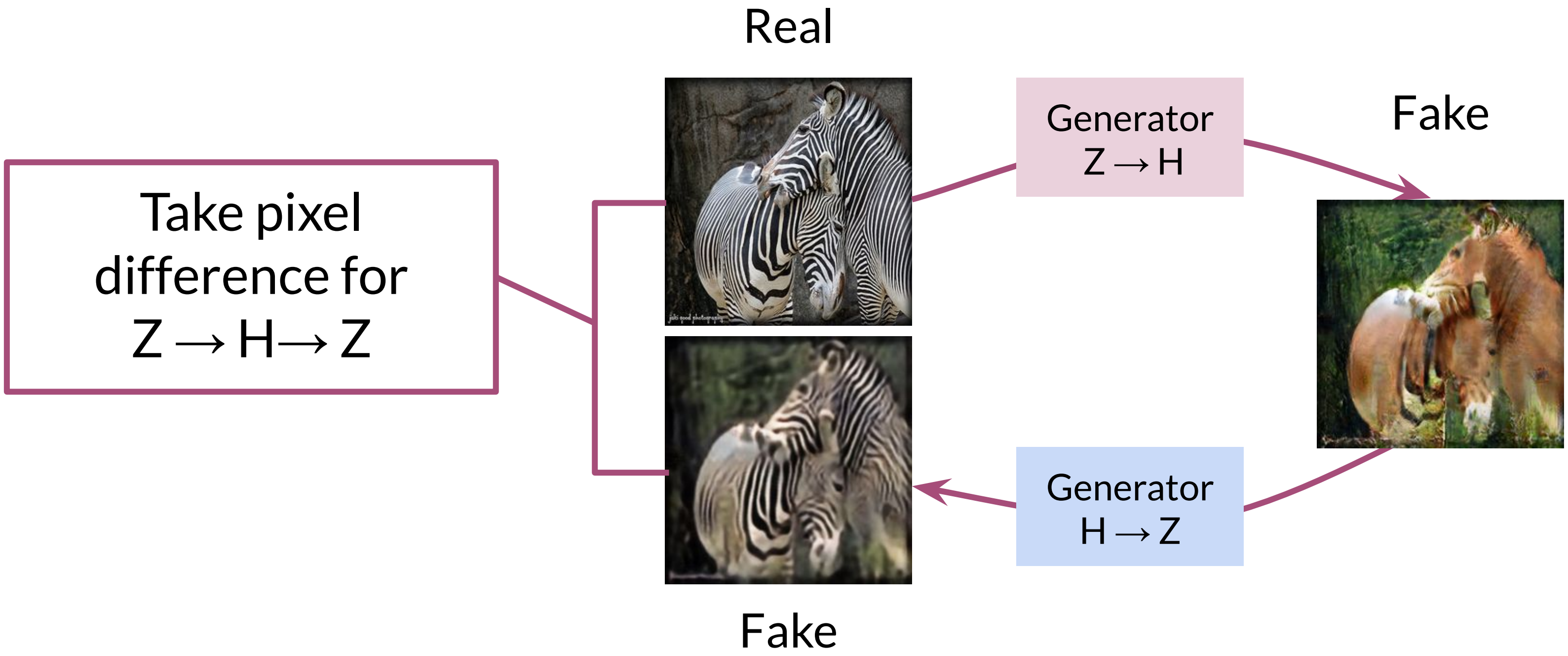
Images available from: https://github.com/togheppi/CycleGAN

Cycle Consistency Loss



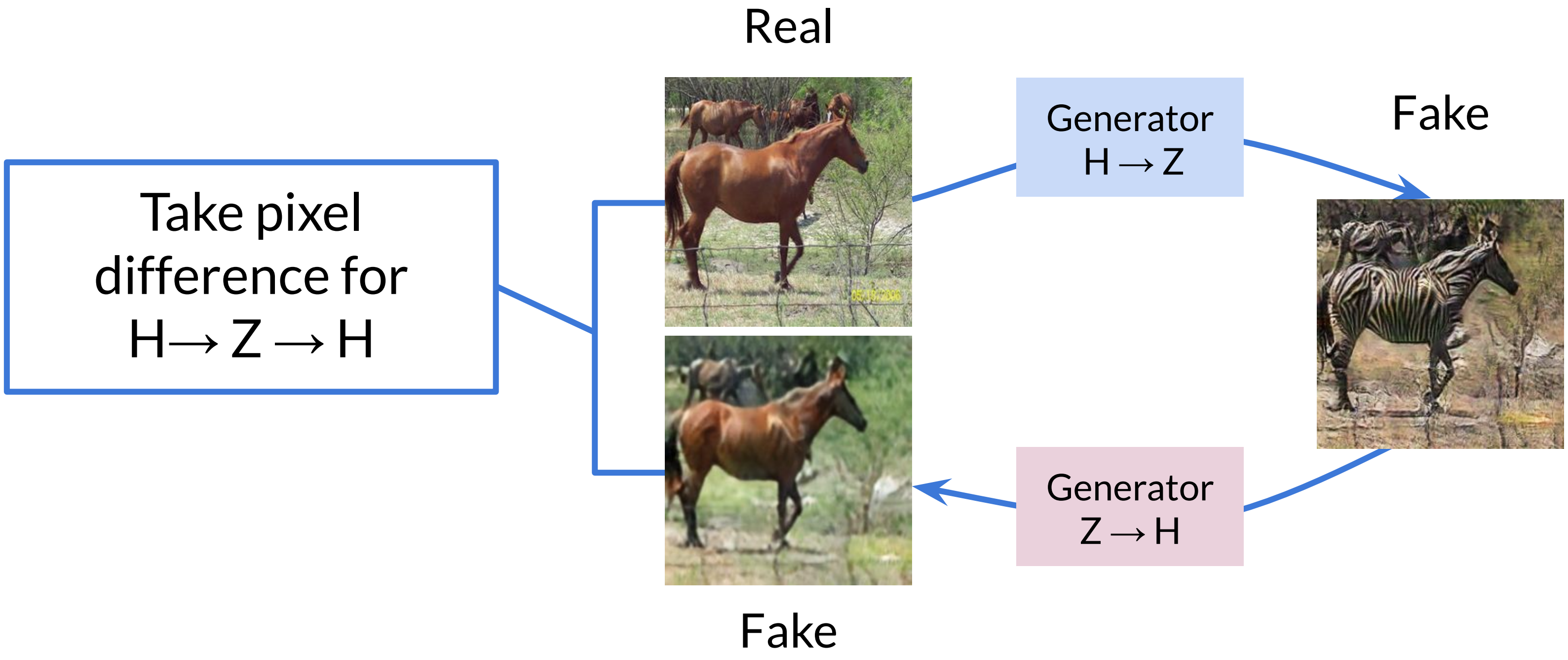
Images available from: https://github.com/togheppi/CycleGAN

Cycle Consistency Loss



Images available from: https://github.com/togheppi/CycleGAN

Cycle Consistency Loss



Images available from: https://github.com/togheppi/CycleGAN

Cycle Consistency Loss

$\sum_i |$  $-$  $|$

Z → H → Z

$+$

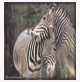
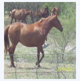
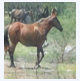
$\sum_i |$  $-$  $|$

H → Z → H

Cycle Consistency Loss is the sum of both directions

Images available from: https://github.com/togheppi/CycleGAN

Cycle Consistency Loss

Adversarial Loss + $\sum_i |$  $-$  $|$ + $\sum_i |$  $-$  $|$

Z → H → Z

H → Z → H

Images available from: https://github.com/togheppi/CycleGAN

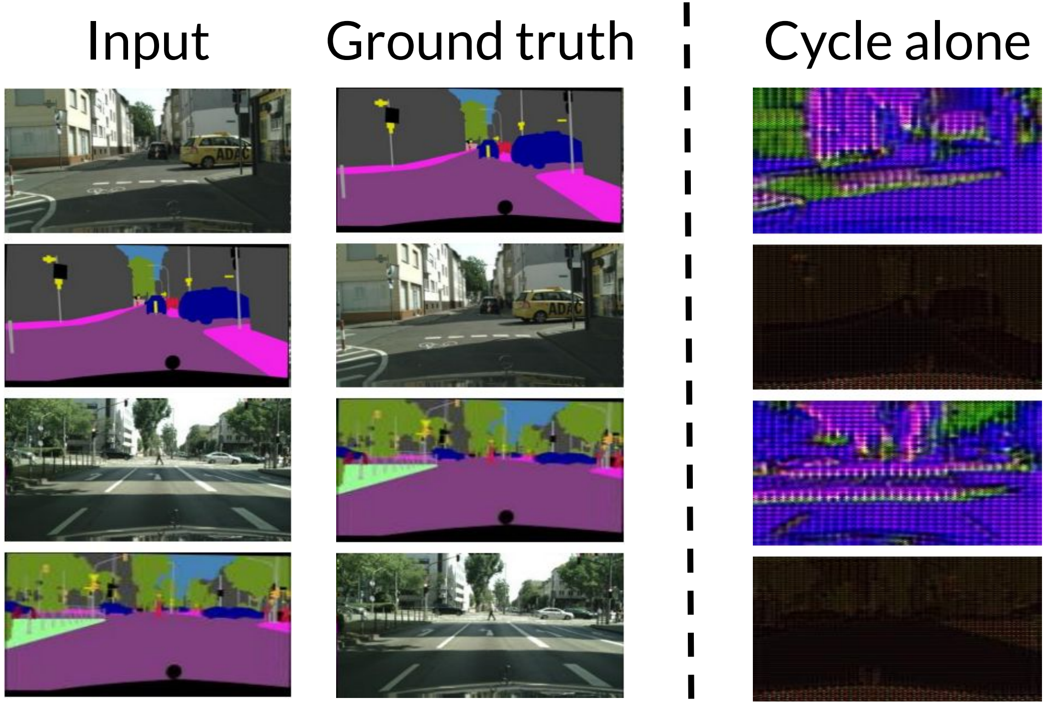
Cycle Consistency Loss

Adversarial Loss + Cycle Consistency Loss

Cycle Consistency Loss

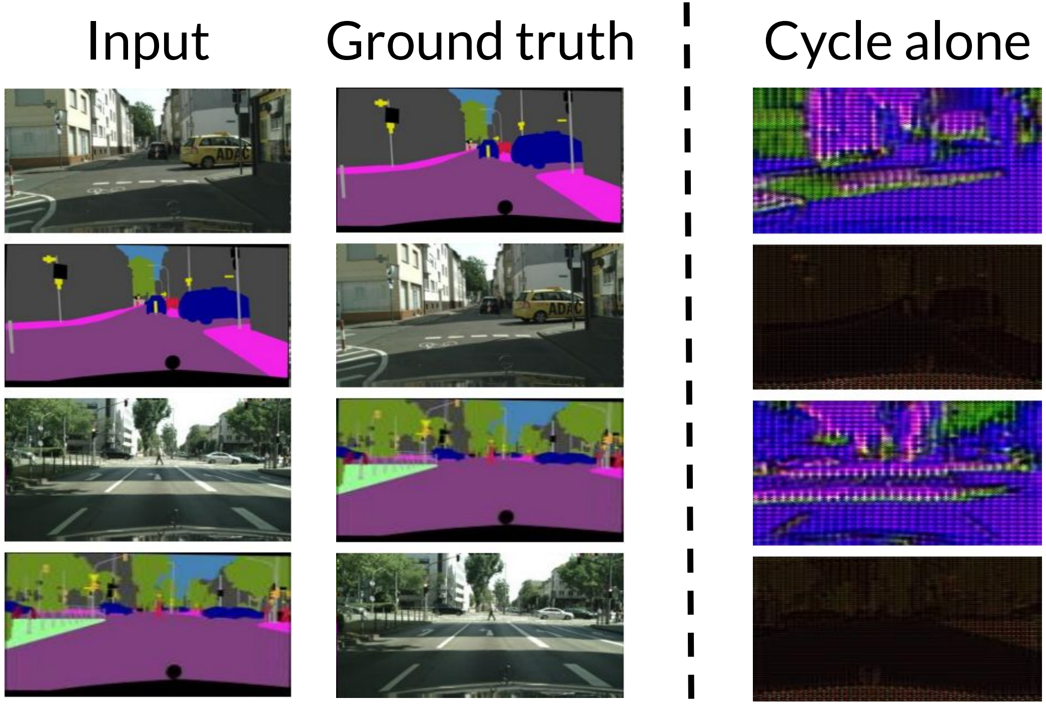
$$\text{Adversarial Loss} \quad + \ \lambda \ * \ \text{Cycle Consistency Loss}$$

Ablation Studies



Available from: https://arxiv.org/abs/1703.10593

Ablation Studies

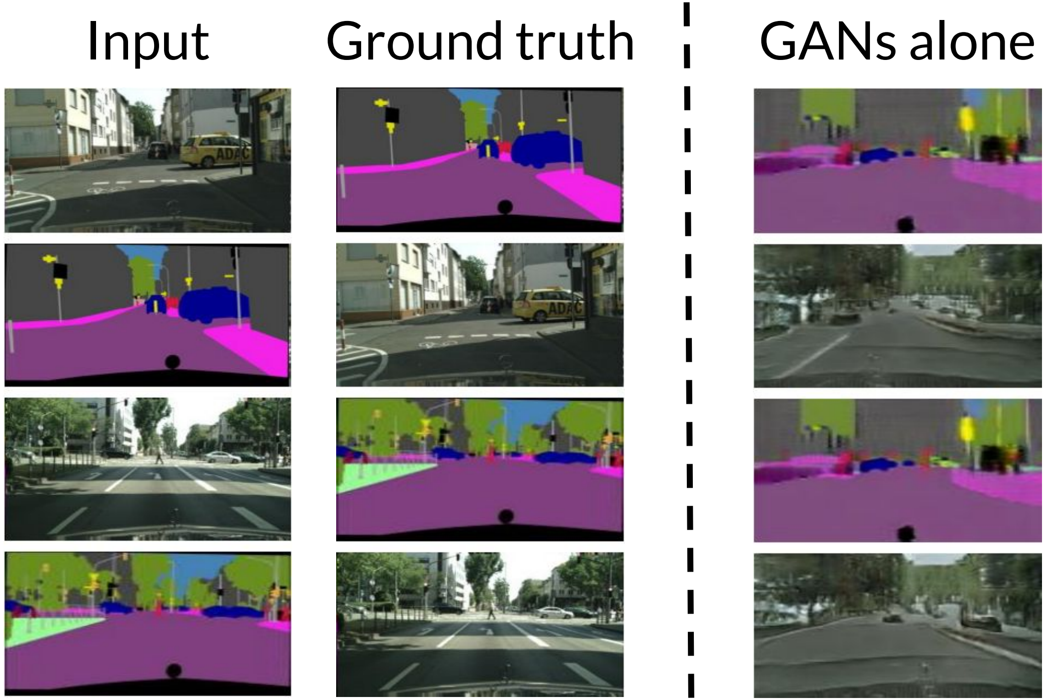


Available from: https://arxiv.org/abs/1703.10593

Without Adversarial GAN Loss, outputs are not realistic

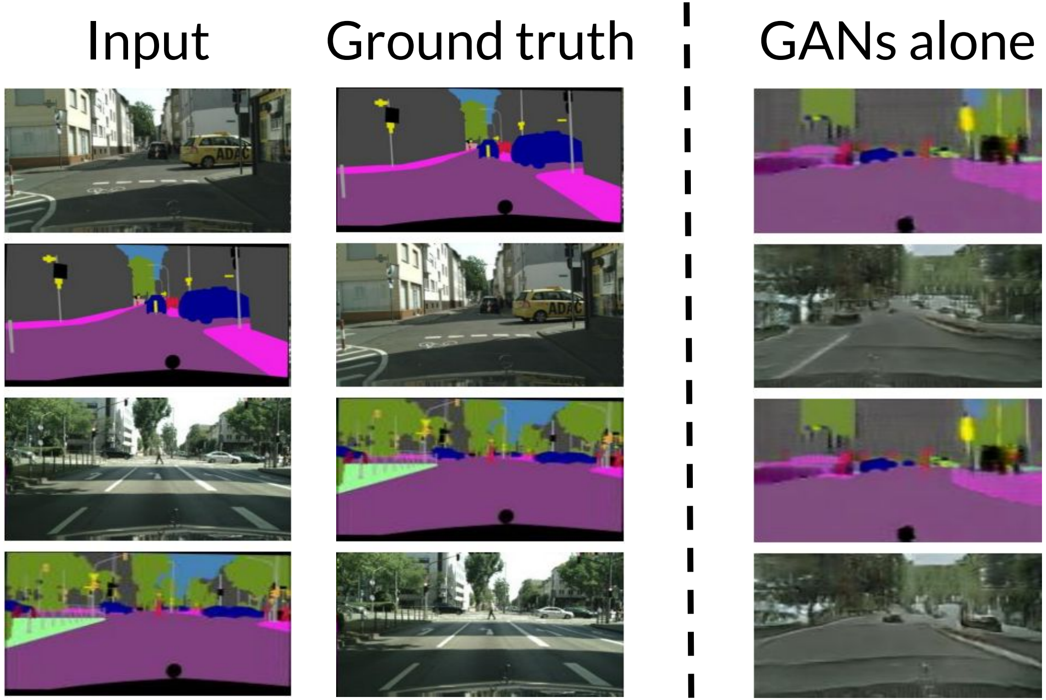
Ablation Studies

Input | Ground truth | GANs alone



Available from: https://arxiv.org/abs/1703.10593

Ablation Studies

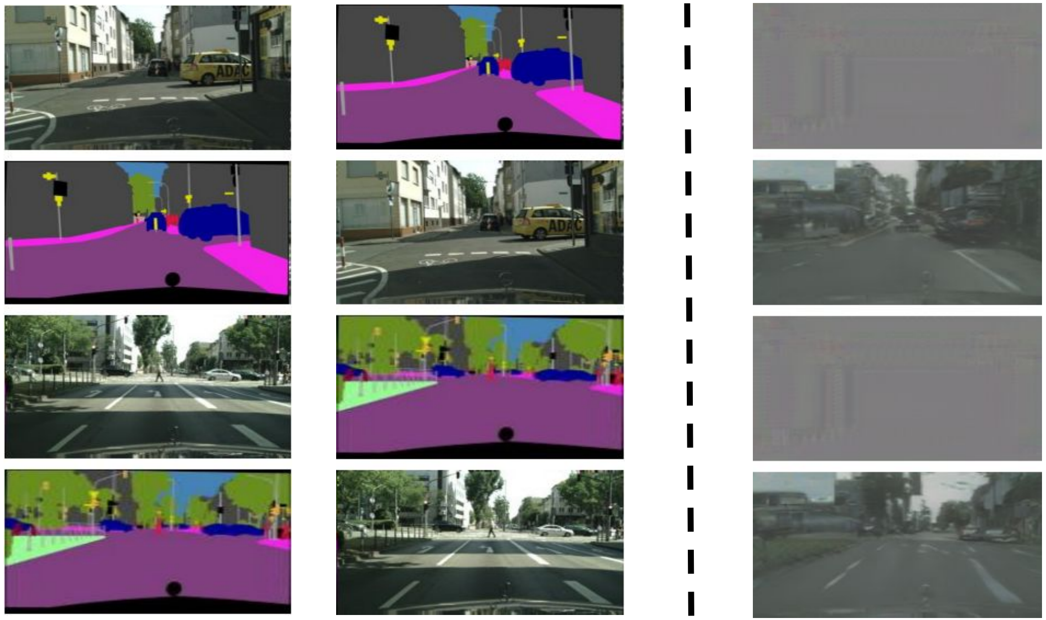


Without Cycle Consistency Loss, outputs show signs of mode collapse

Available from: https://arxiv.org/abs/1703.10593

Ablation Studies

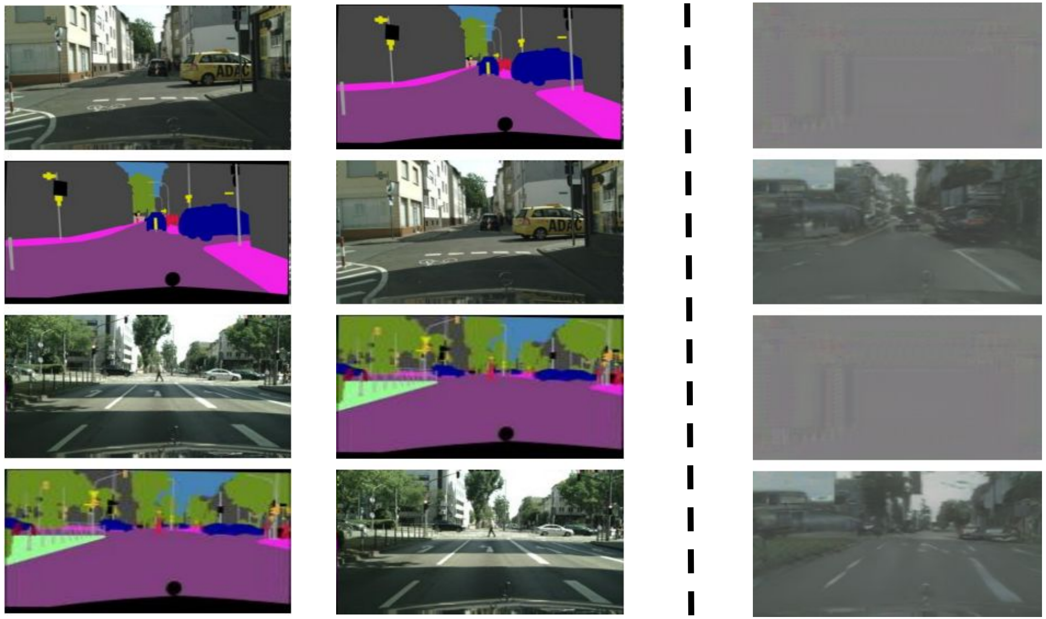
Input | Ground truth | GANs + 1-way cycle



Available from: https://arxiv.org/abs/1703.10593

Ablation Studies

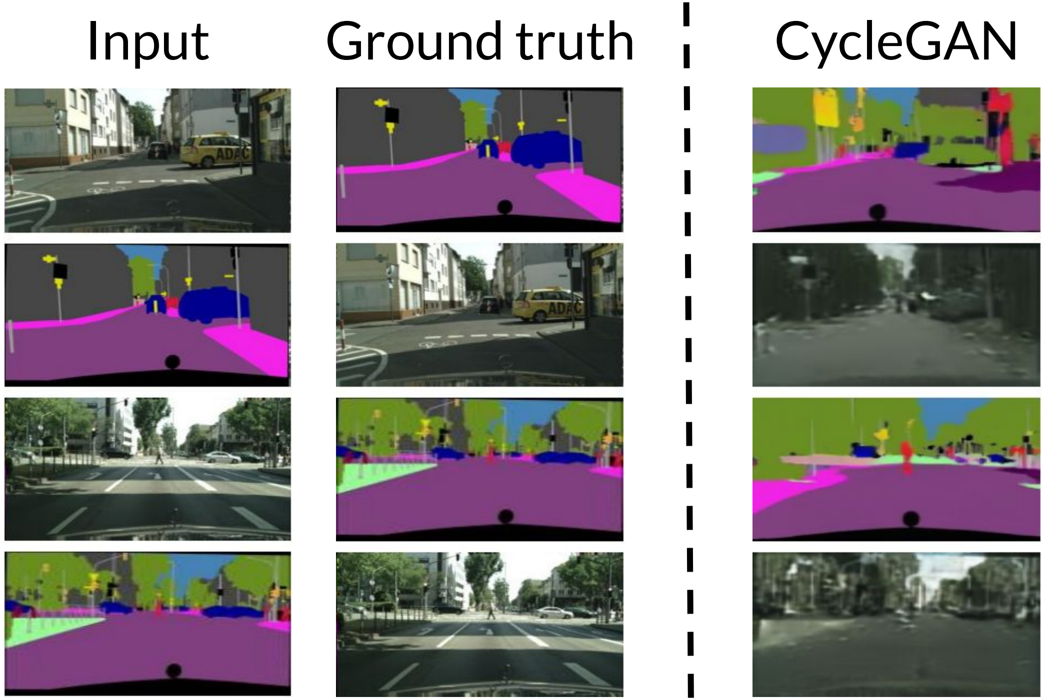
Input | Ground truth | GANs + 1-way cycle



Without **full** Cycle Consistency Loss, outputs see mode collapse too

Available from: https://arxiv.org/abs/1703.10593

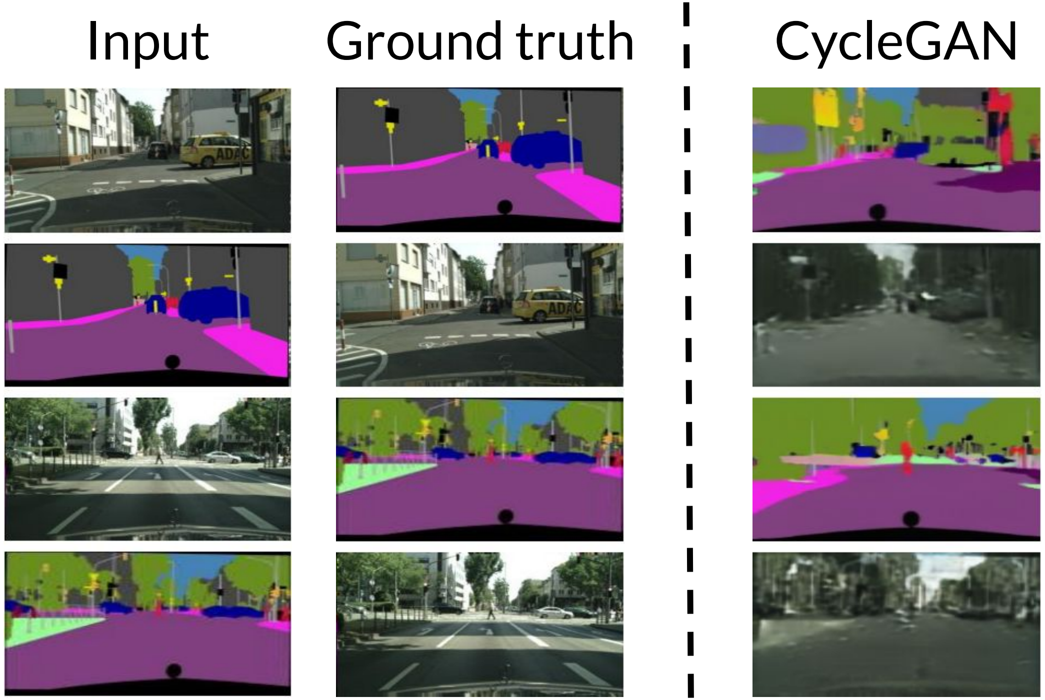
Ablation Studies



Available from: https://arxiv.org/abs/1703.10593

Ablation Studies

Input | Ground truth | CycleGAN

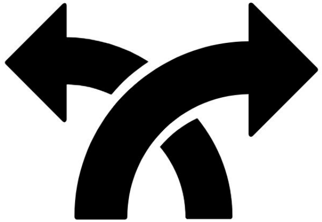


Available from: https://arxiv.org/abs/1703.10593

CycleGAN uses both **Adversarial Loss** and **Cycle Consistency Loss**

Summary

- Cycle consistency helps transfer uncommon style elements between the two GANs, while maintaining common content
- Add an extra loss term to each generator to softly encourage cycle consistency
- Cycle consistency is used in both directions



deeplearning.ai

CycleGAN: Least Squares Loss

Outline

- Least squares in statistics
- Least Squares Loss in GANs
  - Discriminator
  - Generator

Least Squares Loss: Another GAN Loss Function

- Came out when training stability was a big problem in GANS
  - Similar time to WGAN-GP

Least Squares Loss: Another GAN Loss Function

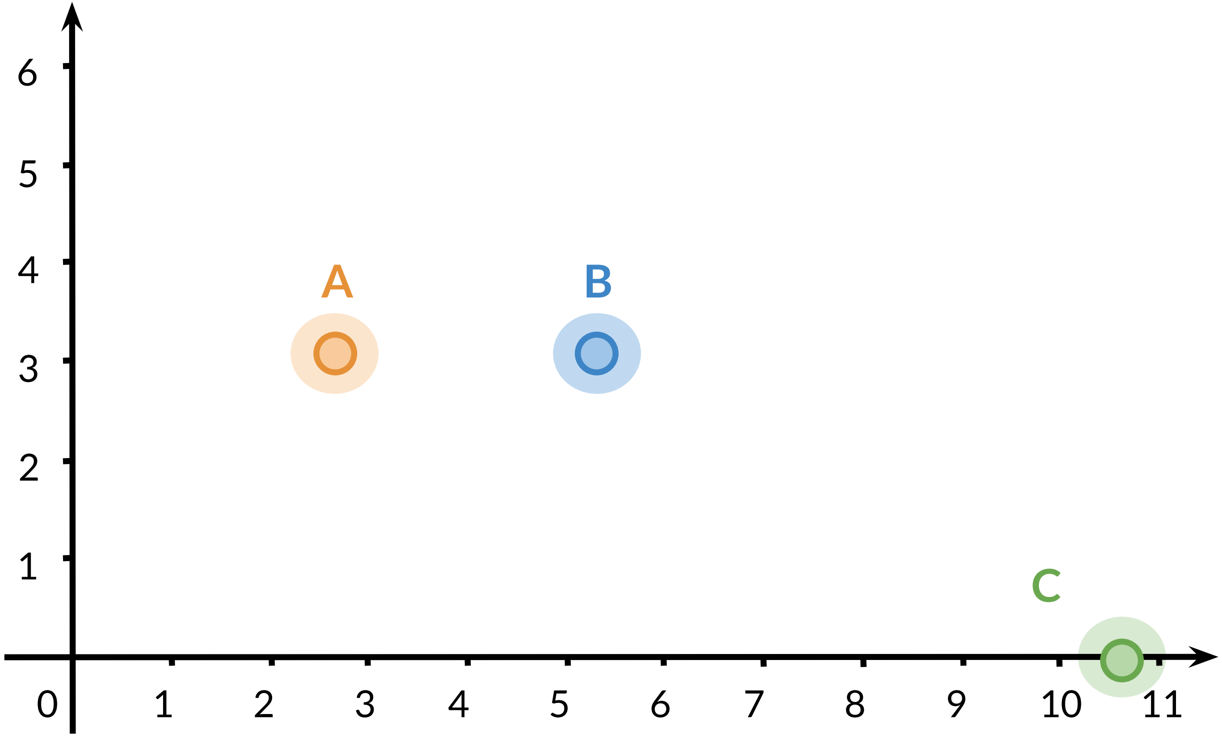
- Came out when training stability was a big problem in GANS
    - Similar time to WGAN-GP
- Helps with vanishing gradients and mode collapse

Least Squares Loss: Another GAN Loss Function

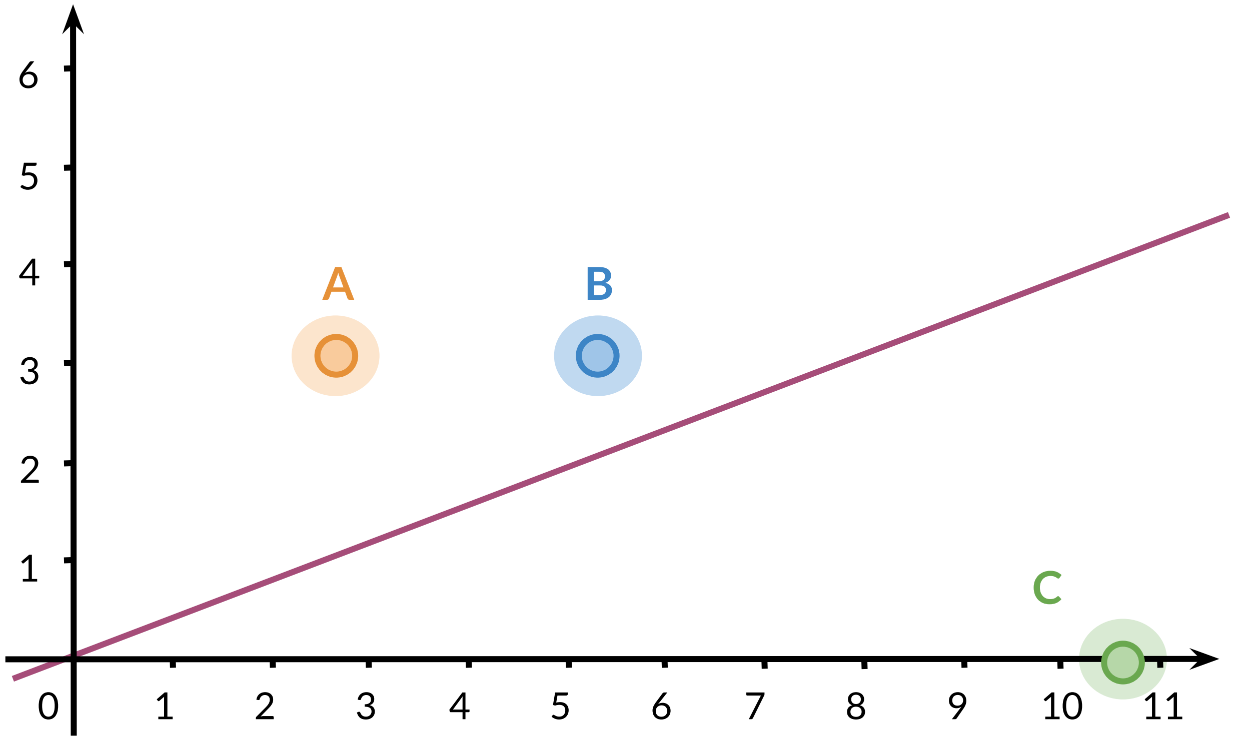
- Came out when training stability was a big problem in GANS
  - Similar time to WGAN-GP
- Helps with vanishing gradients and mode collapse

- GAN loss functions are chosen empirically

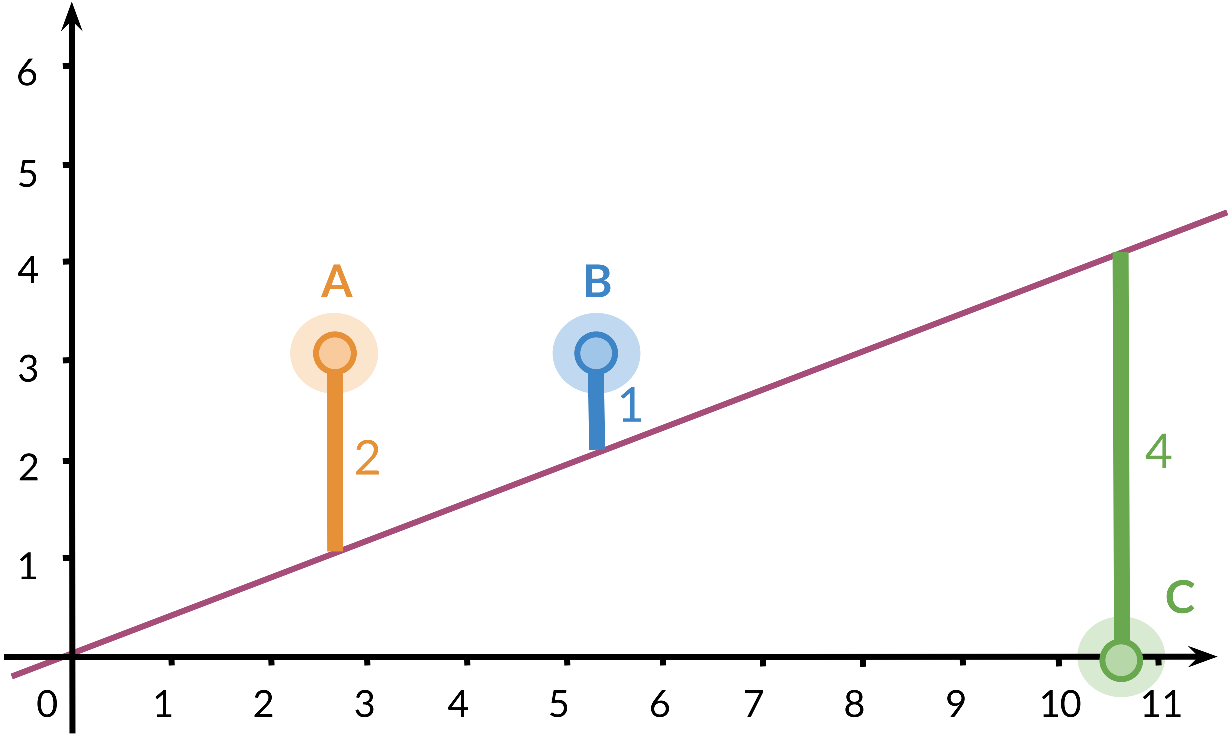
Least Squares



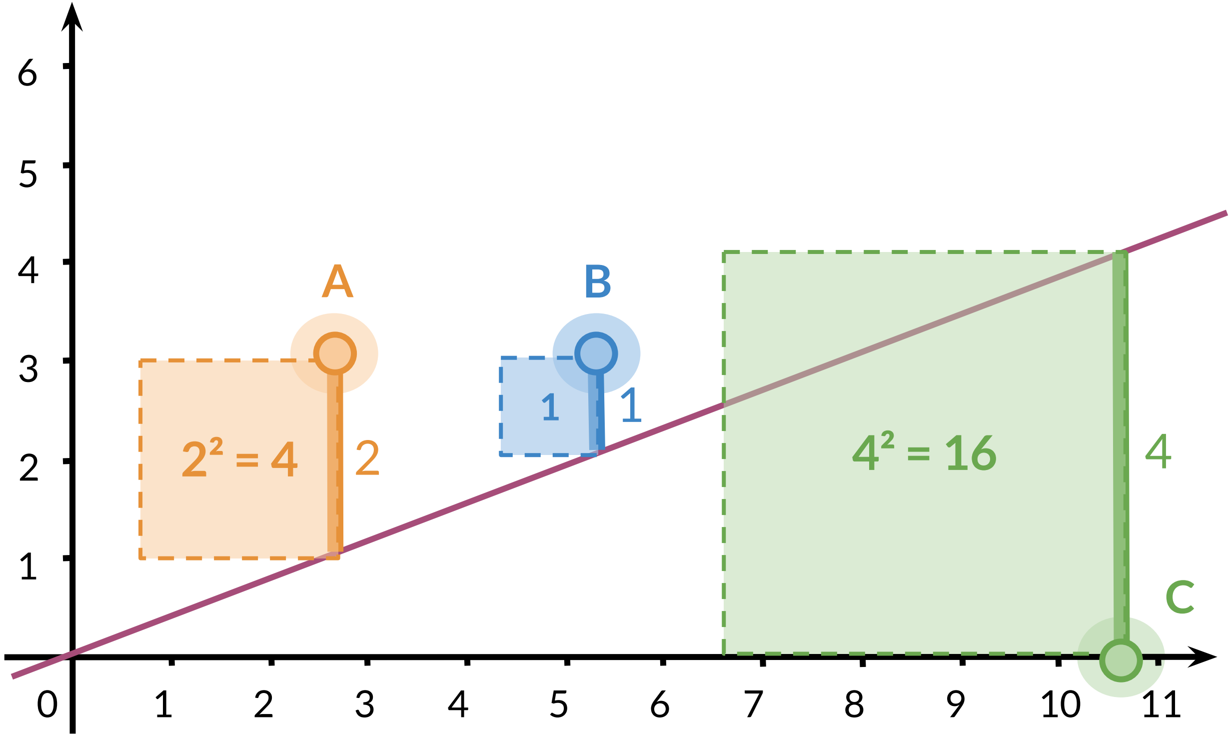
Least Squares



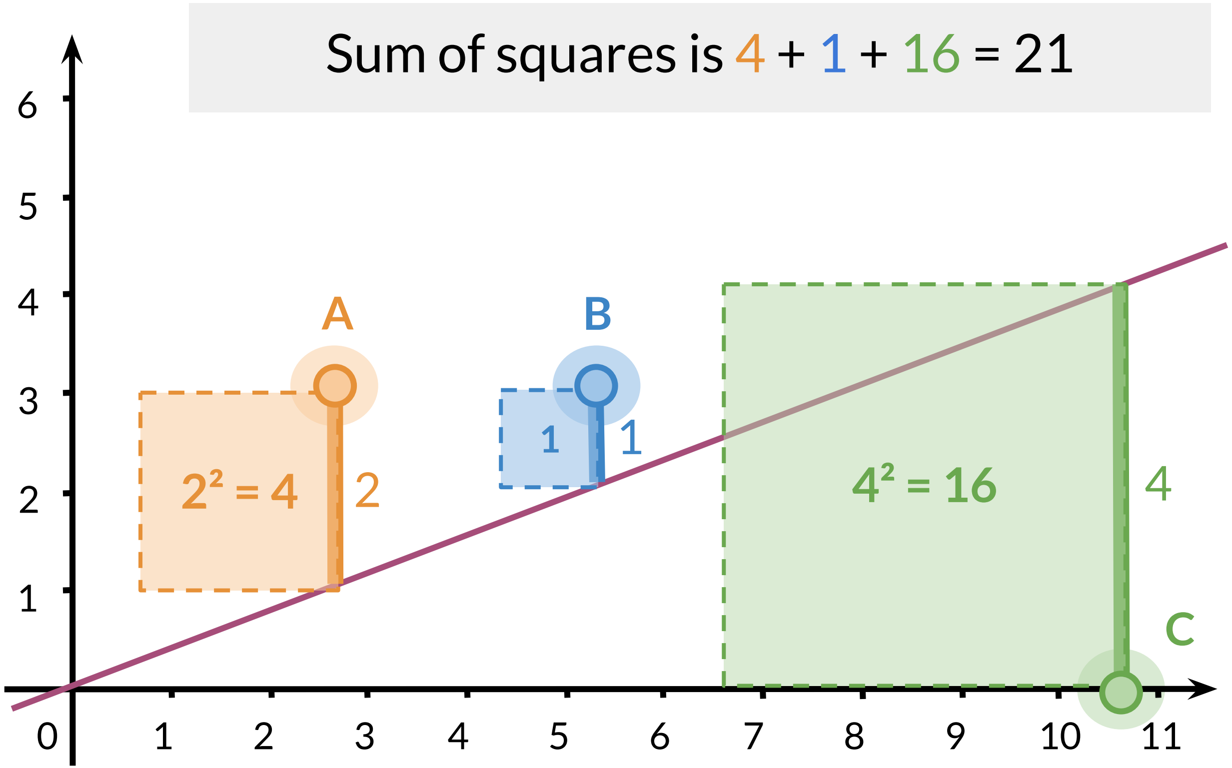
Least Squares



Least Squares

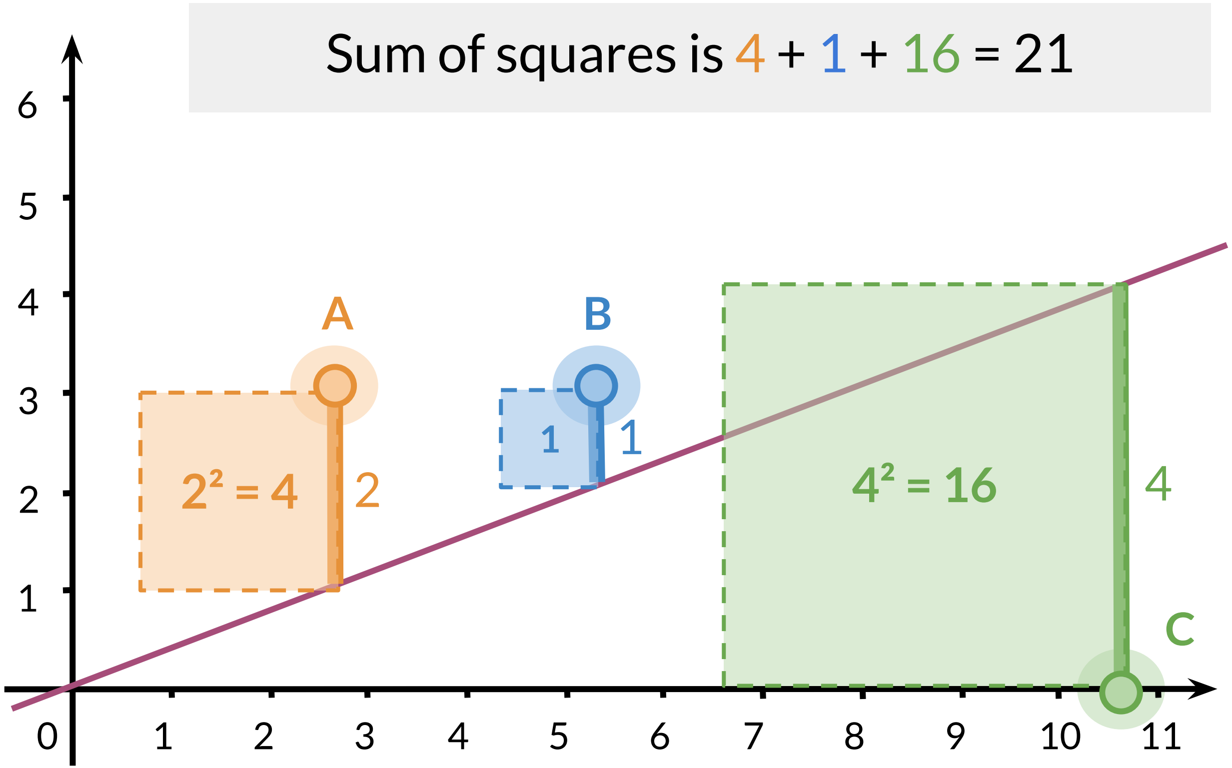


Least Squares



Least Squares

Minimize
sum of squares



Least Squares Loss: Discriminator

$$(D(\boldsymbol{x}) - 1)^2$$

Discriminator classification of real image ***x***

Least Squares Loss: Discriminator

$$\mathbb{E}_{\boldsymbol{x}}\left[(D(\boldsymbol{x})-1)^2\right]$$

Least Squares Loss: Discriminator

$$\mathbb{E}_{\boldsymbol{x}}\left[(D(\boldsymbol{x})-1)^2\right] + \qquad (D(G(\boldsymbol{z}))-\mathbf{0})^2$$

Discriminator classification
of fake image G(***z***)

Least Squares Loss: Discriminator

$$\mathbb{E}_{\boldsymbol{x}}\left[(D(\boldsymbol{x})-1)^2\right]+\mathbb{E}_{\boldsymbol{z}}\left[(D(G(\boldsymbol{z}))-\mathbf{0})^2\right]$$

Least Squares Loss: Discriminator

$$\mathbb{E}_{\boldsymbol{x}}\left[(D(\boldsymbol{x})-1)^2\right]+\mathbb{E}_{\boldsymbol{z}}\left[(D(G(\boldsymbol{z})))^2\right]$$

Least Squares Loss: Generator

$$\mathbb{E}_{\boldsymbol{z}}\left[(D(G(\boldsymbol{z}))-1)^2\right]$$

Least Squares Loss

Discriminator Loss

$$\mathbb{E}_{\boldsymbol{x}}\left[(D(\boldsymbol{x})-1)^2\right]+\mathbb{E}_{\boldsymbol{z}}\left[(D(G(\boldsymbol{z})))^2\right]$$

Least Squares Loss

Discriminator Loss

$$\mathbb{E}_{\boldsymbol{x}}\left[(D(\boldsymbol{x})-1)^2\right]+\mathbb{E}_{\boldsymbol{z}}\left[(D(G(\boldsymbol{z})))^2\right]$$

Generator Loss

$$\mathbb{E}_{\boldsymbol{z}}\left[(D(G(\boldsymbol{z}))-1)^2\right]$$

Least Squares Loss

Discriminator Loss

$$\mathbb{E}_{\boldsymbol{x}}\left[(D(\boldsymbol{x})-1)^2\right]+\mathbb{E}_{\boldsymbol{z}}\left[(D(G(\boldsymbol{z})))^2\right]$$

Generator Loss

$$\mathbb{E}_{\boldsymbol{z}}\left[(D(G(\boldsymbol{z}))-1)^2\right]$$

Reduces vanishing gradient problem

Least Squares Loss

Discriminator Loss	$\mathbb{E}_{\boldsymbol{x}}\left[(D(\boldsymbol{x})-1)^2\right]+\mathbb{E}_{\boldsymbol{z}}\left[(D(G(\boldsymbol{z})))^2\right]$
Generator Loss	$\mathbb{E}_{\boldsymbol{z}}\left[(D(G(\boldsymbol{z}))-1)^2\right]$

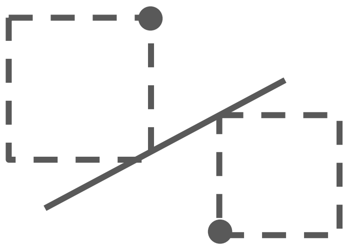
Also known as Mean Squared Error!

Context of Least Squares Loss

Adversarial Loss + λ * Cycle Consistency Loss

Least Squares Loss

Summary



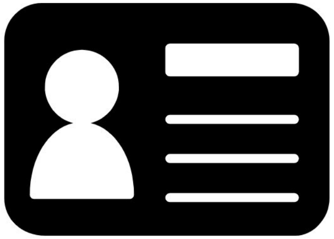
- Least squares fits a line from several points
- Least Squares Loss is used as the Adversarial Loss function in CycleGAN
- More stable than BCELoss, since the gradient is only flat when prediction is exactly correct

deeplearning.ai

CycleGAN: Identity Loss

Outline

- Identity Loss
  - How it works
  - Impact on outputs



Identity Loss

Real input

Generator

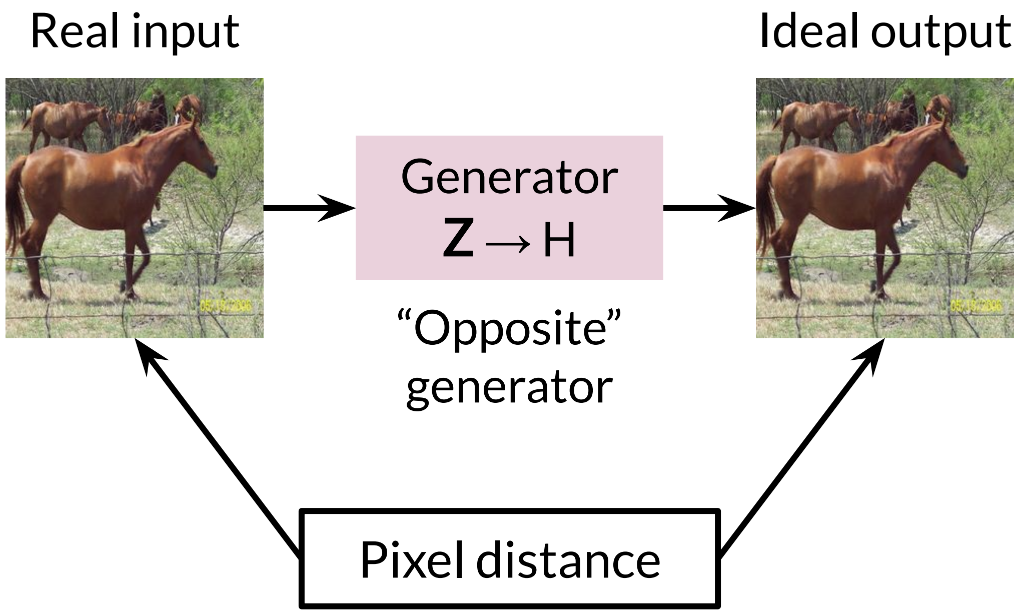
Z → H

"Opposite" generator

Ideal output

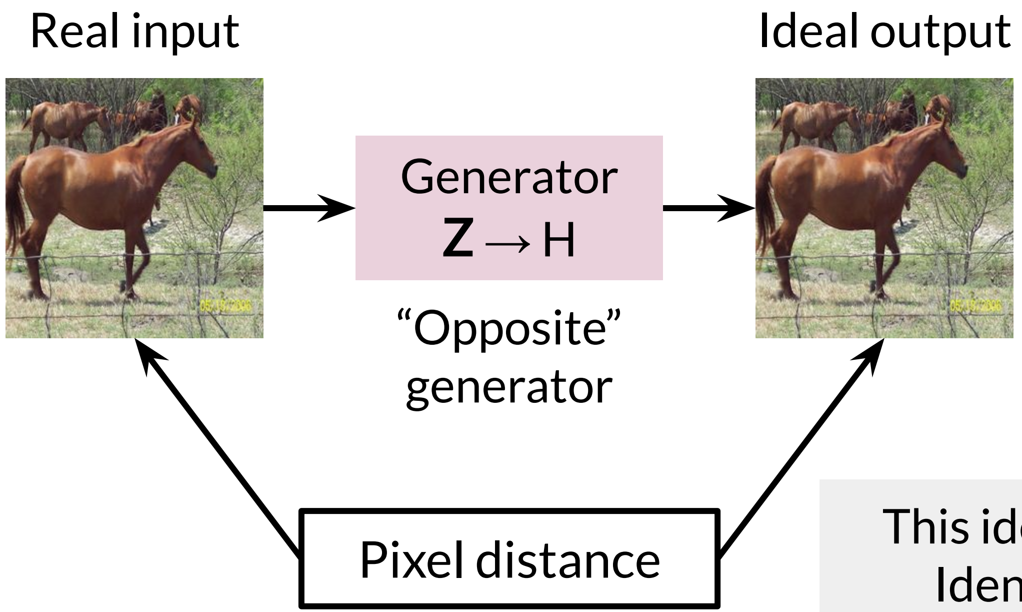
Images available from: https://github.com/togheppi/CycleGAN

Identity Loss



Images available from: https://github.com/togheppi/CycleGAN

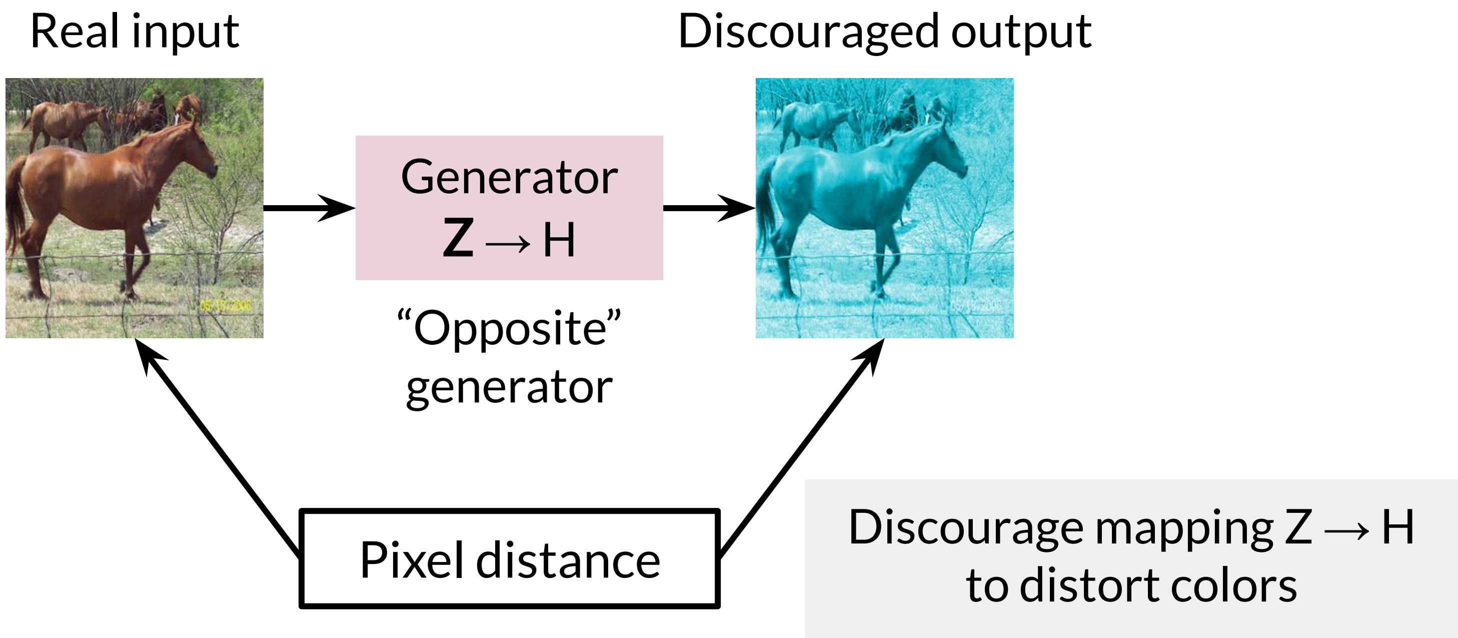
Identity Loss



This ideal example has an Identity Loss of zero

Images available from: https://github.com/togheppi/CycleGAN

Identity Loss



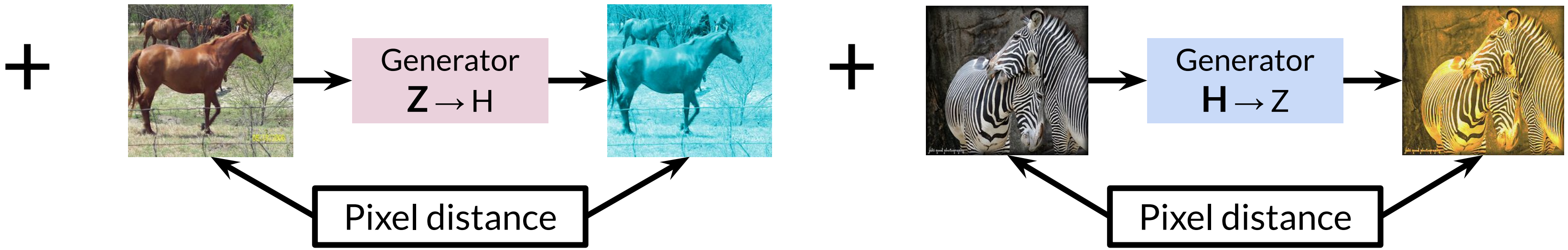
Images available from: https://github.com/togheppi/CycleGAN

Context of Identity Loss

Adversarial Loss + λ * Cycle Consistency Loss

Context of Identity Loss

Adversarial Loss + λ * Cycle Consistency Loss



Images available from: https://github.com/togheppi/CycleGAN

Context of Identity Loss

Adversarial Loss + λ * Cycle Consistency Loss

\+ Identity Loss

Context of Identity Loss

Adversarial Loss $+ \lambda_1 *$ Cycle Consistency Loss

$+ \lambda_2 *$ Identity Loss

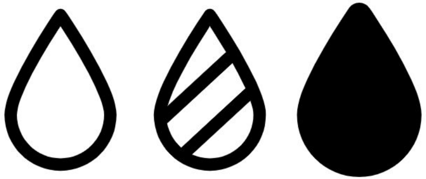
Identity Loss Example: Photo → Monet

Identity Loss helps preserve original photo color

Available from: https://arxiv.org/abs/1703.10593

Summary

- Identity Loss takes a real image in domain B and inputs it into Generator: A → B, expecting an identity mapping
  - An identity mapping means the output is the same as the input
- Pixel distance is used
  - Ideally, no difference between input and output!
- Identity Loss is optionally added to help with color preservation

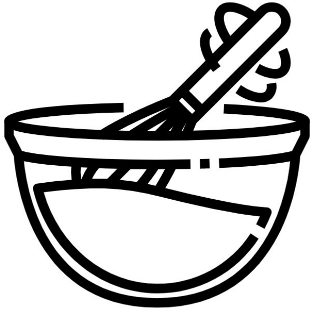


CycleGAN: Putting It All Together

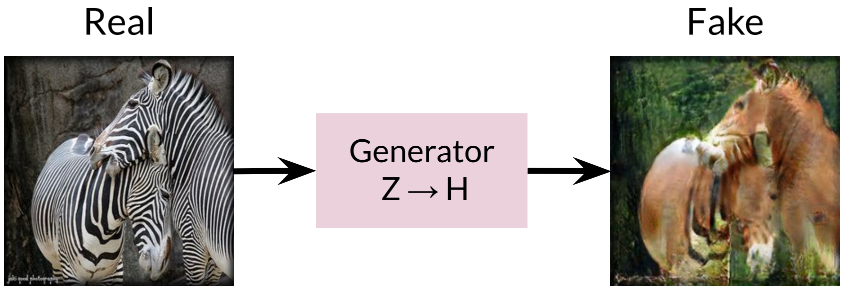
deeplearning.ai

Outline

- Putting CycleGAN together!
    - Two GANs
    - Cycle Consistency Loss
    - Least Squares Adversarial Loss
    - Identity Loss (optional)

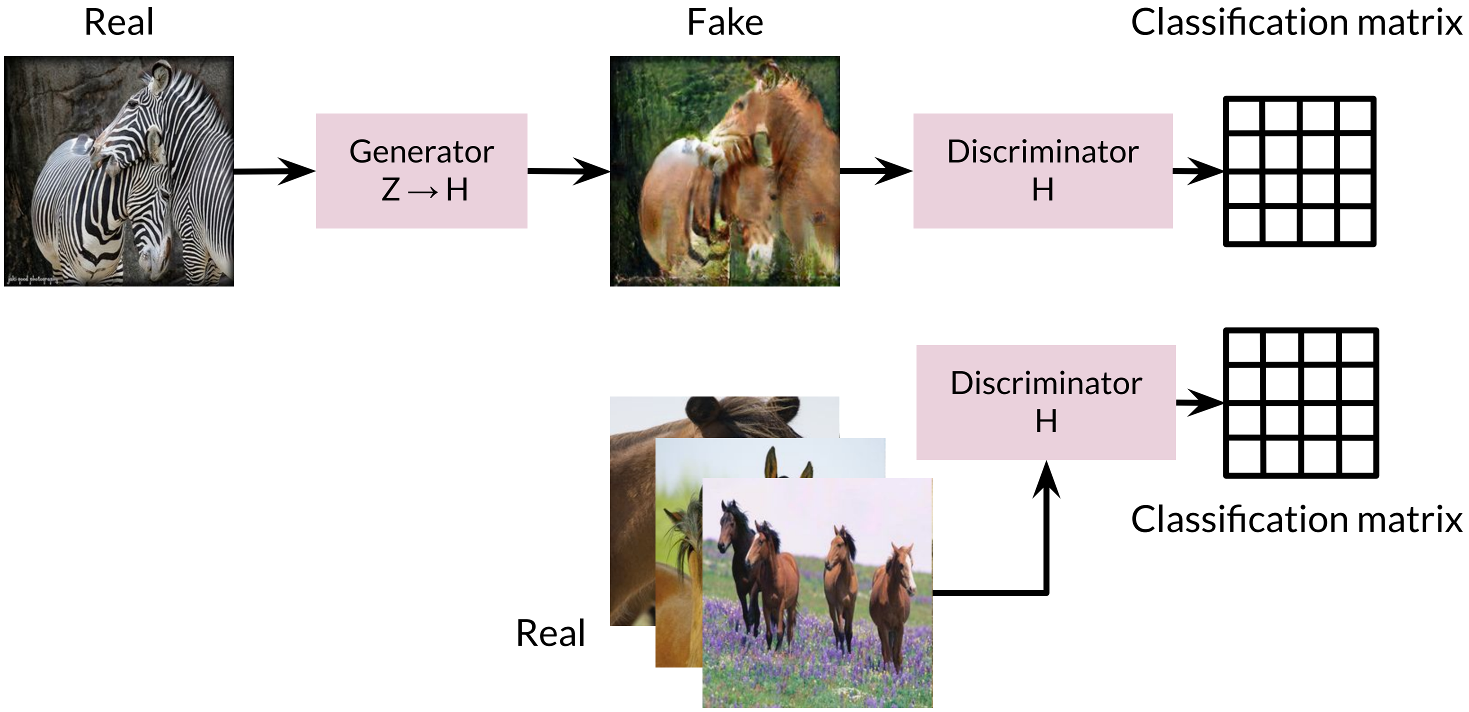


CycleGAN



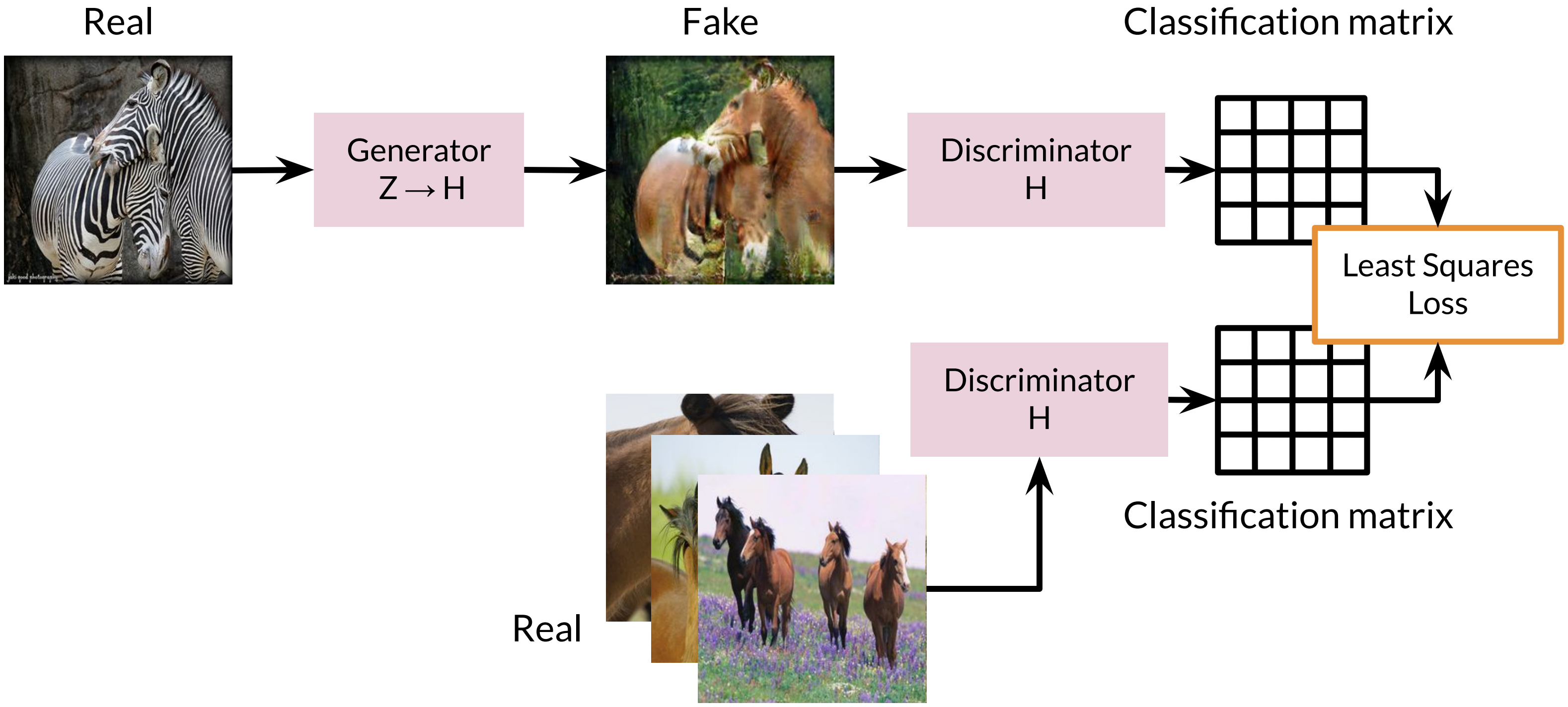
Images available from: https://github.com/togheppi/CycleGAN

CycleGAN



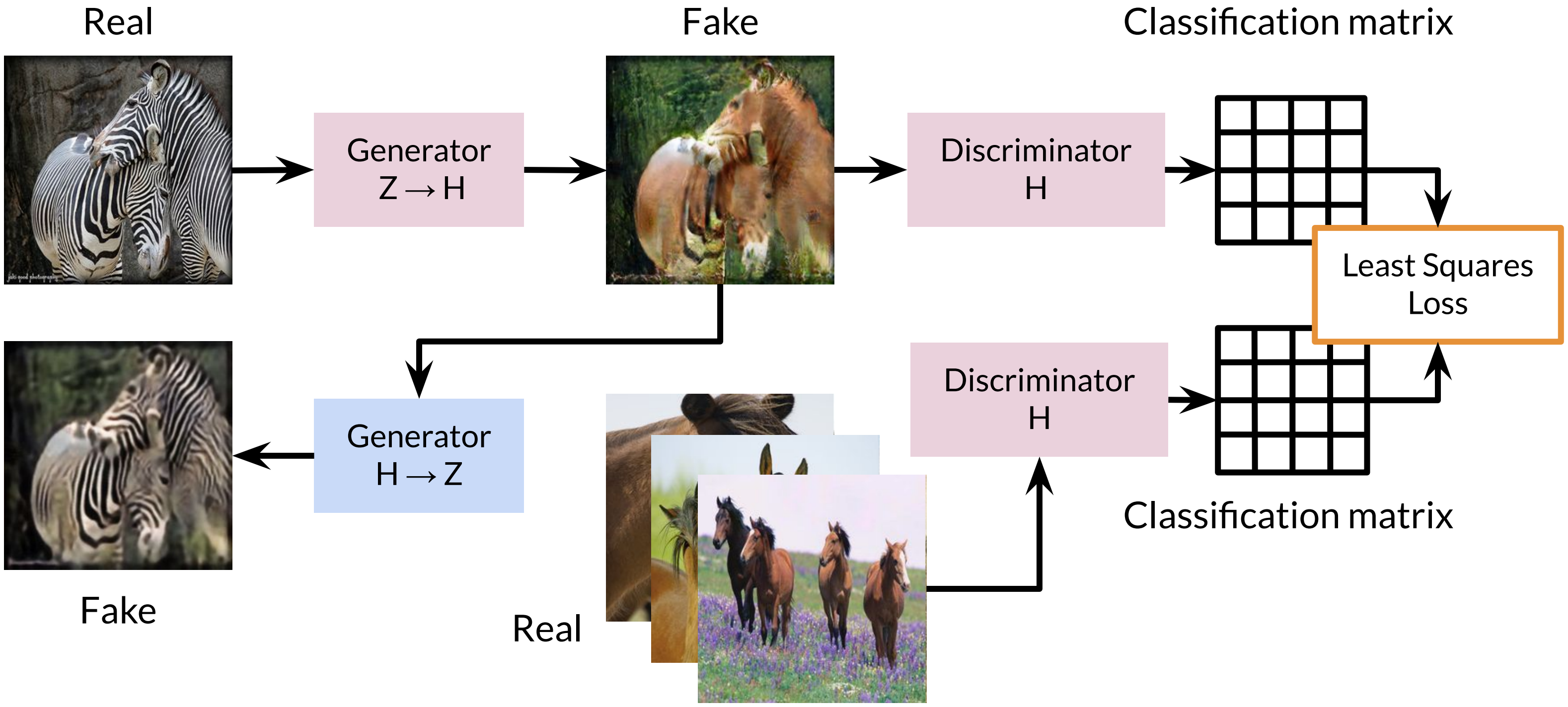
Images available from: https://github.com/togheppi/CycleGAN

CycleGAN



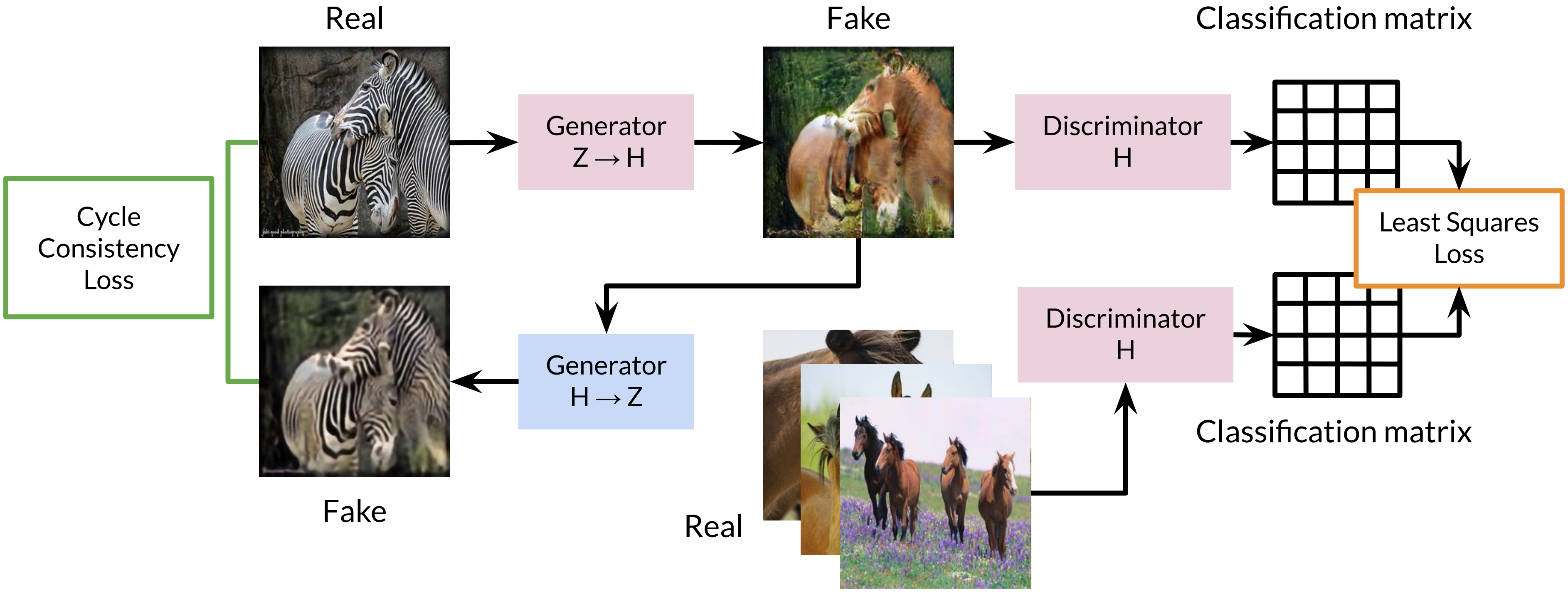
Images available from: https://github.com/togheppi/CycleGAN

CycleGAN



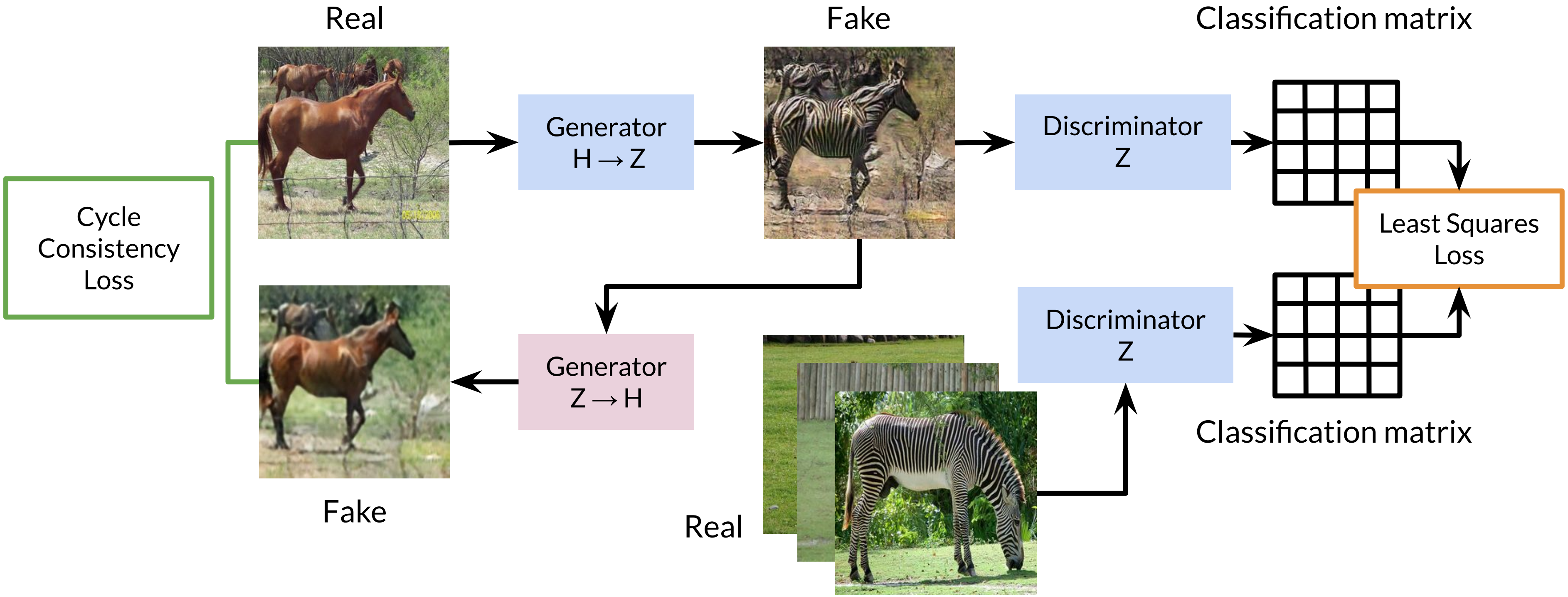
Images available from: https://github.com/togheppi/CycleGAN

CycleGAN



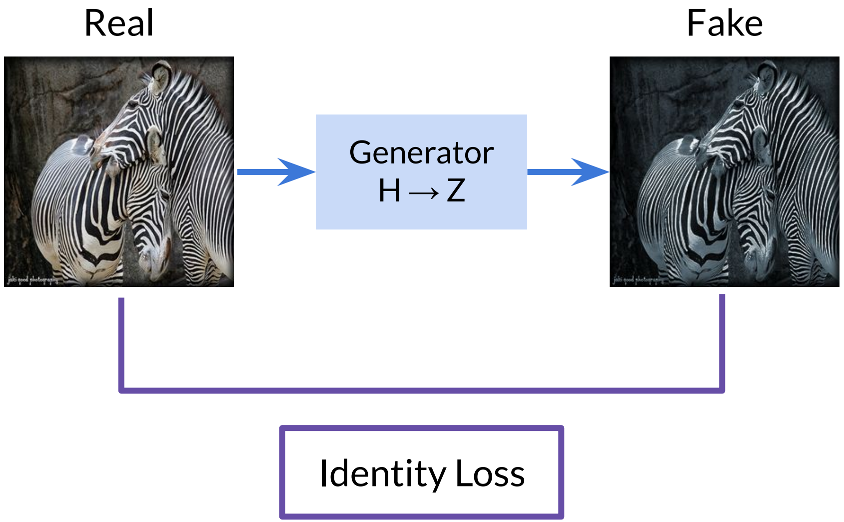
Images available from: https://github.com/togheppi/CycleGAN

CycleGAN



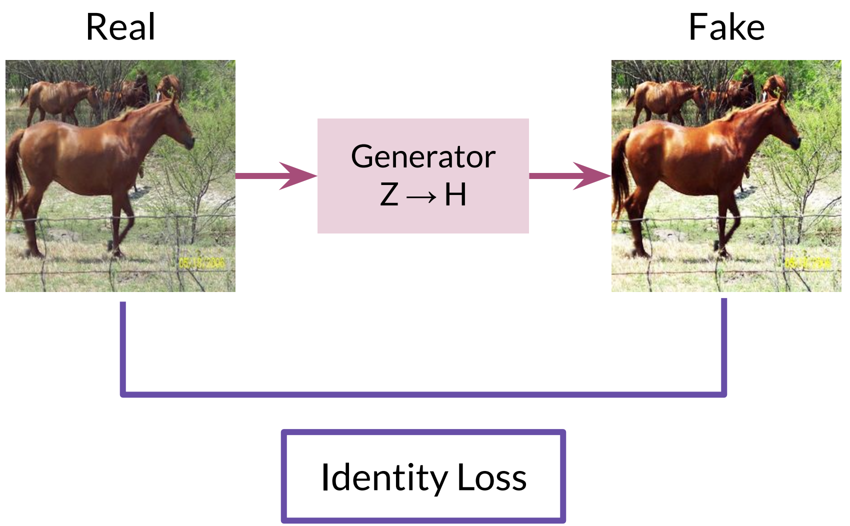
Images available from: https://github.com/togheppi/CycleGAN

CycleGAN



Images available from: https://github.com/togheppi/CycleGAN

CycleGAN



Images available from: https://github.com/togheppi/CycleGAN

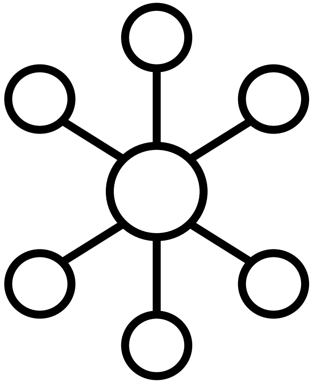
CycleGAN Loss

Generator Loss = Least Squares Adversarial Loss $H \rightarrow Z$ + Cycle Consistency Loss $H \rightarrow Z \rightarrow H$ + Identity Loss $H \rightarrow Z$

+ Least Squares Adversarial Loss $Z \rightarrow H$ + Cycle Consistency Loss $Z \rightarrow H \rightarrow Z$ + Identity Loss $Z \rightarrow H$

Summary

- CycleGAN is composed of two GANs
- Generators have 6 loss terms in total, 3 each:
  - Least Squares Adversarial Loss
  - Cycle Consistency Loss
  - Identity Loss
- Discriminator is simpler, with BCELoss using PatchGAN



deeplearning.ai

CycleGAN Applications & Variants

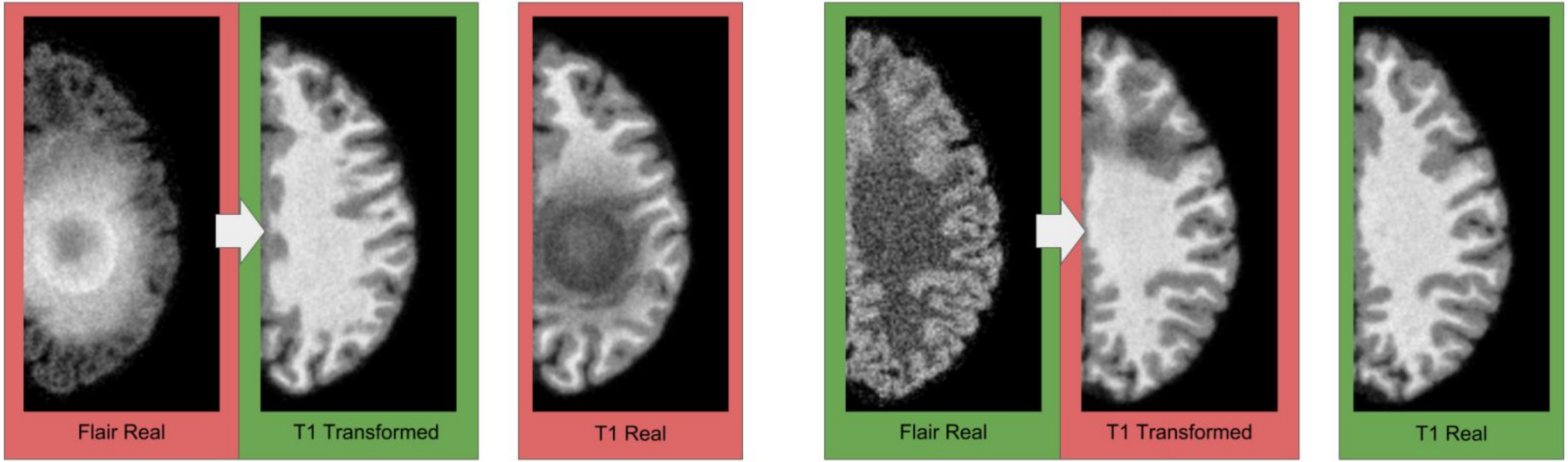
Outline

- Overview of some CycleGAN applications
- Some variants of unpaired image-to-image translation

Applications

Available from: https://arxiv.org/abs/1611.07004

Applications

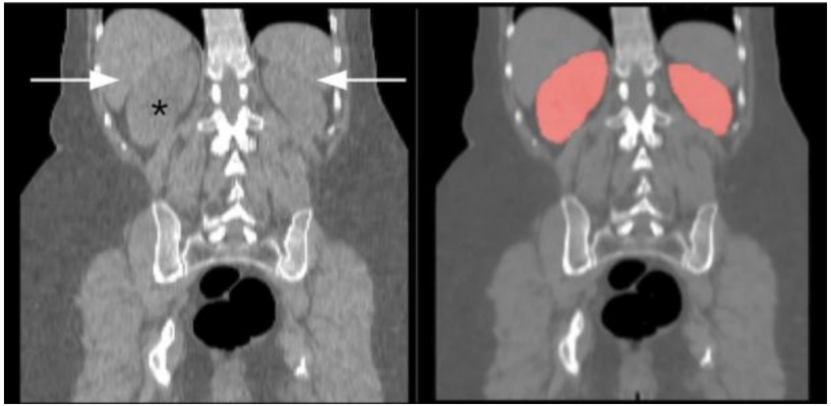


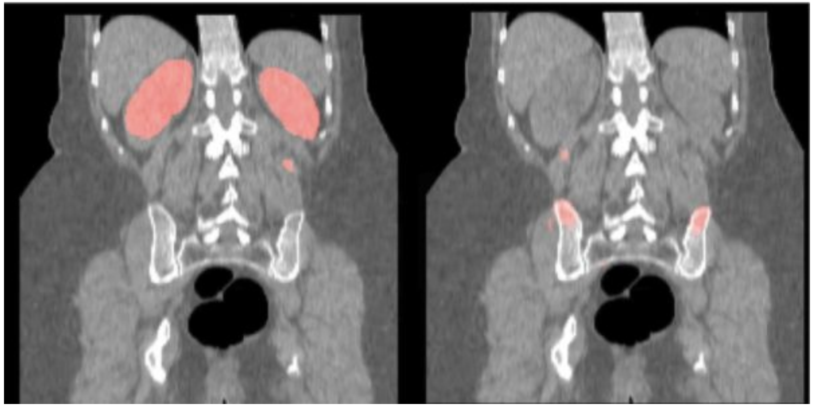
(a) A translation removing tumors

(b) A translation adding tumors

Available from: https://arxiv.org/abs/1805.08841

Applications

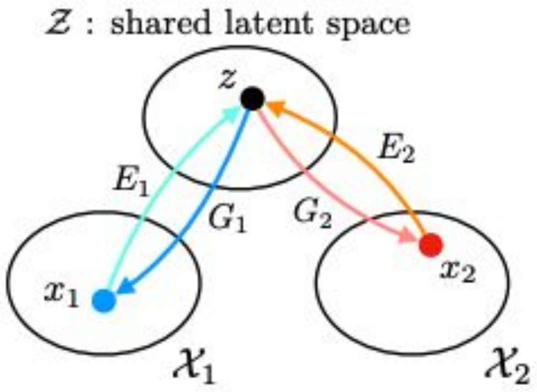




CT Expert CycleGAN Standard Augmentation

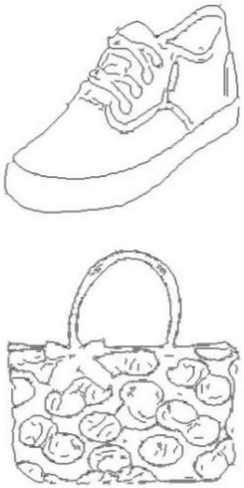
Available from: https://www.nature.com/articles/s41598-019-52737-x.pdf

Variant: UNIT



Available from: https://github.com/mingyuliutw/UNIT

Variant: Multimodal UNIT (MUNIT)



Available from: https://github.com/NVlabs/MUNIT

Variant: Multimodal UNIT (MUNIT)

Available from: https://github.com/NVlabs/MUNIT

Summary

- Various applications of CycleGAN including:
  - Democratized art and style transfer
  - Medical data augmentation
  - Creating paired data
- UNIT and MUNIT are other models for unpaired (unsupervised) image-to-image translation