

# Copyright Notice

These slides are distributed under the Creative Commons License.

DeepLearning.AI makes these slides available for educational purposes. You may not use or distribute these slides for commercial purposes. You may make copies of these slides and use or distribute them for educational purposes as long as you cite DeepLearning.AI as the source of the slides.

For the rest of the details of the license, see

<https://creativecommons.org/licenses/by-sa/2.0/legalcode>

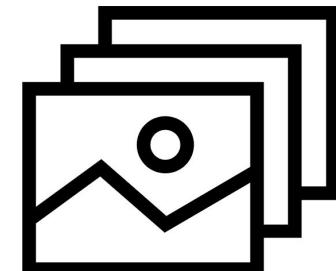


deeplearning.ai

# 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

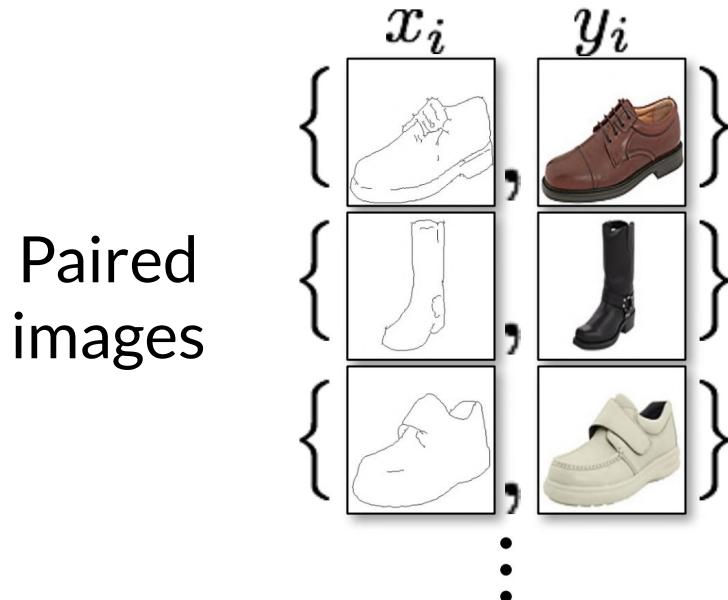
Monet to photo



Unpaired images

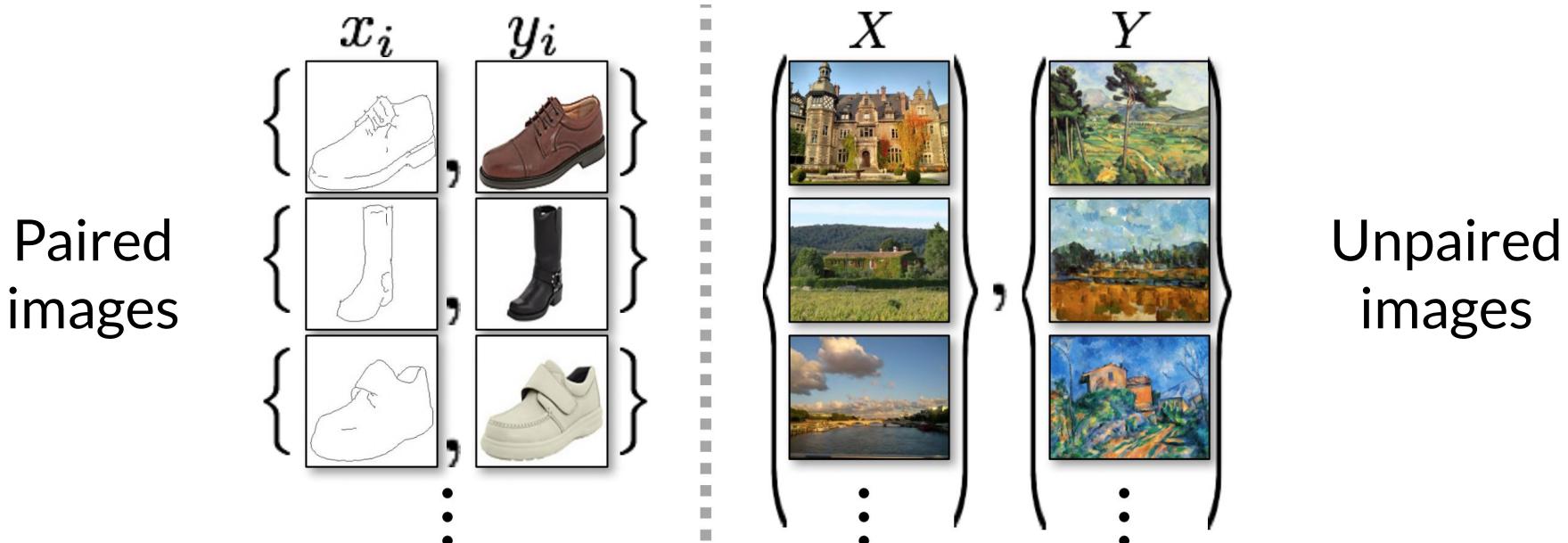
Available from: <https://arxiv.org/abs/1611.07004>

# Image-to-Image Translation



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

# Image-to-Image Translation



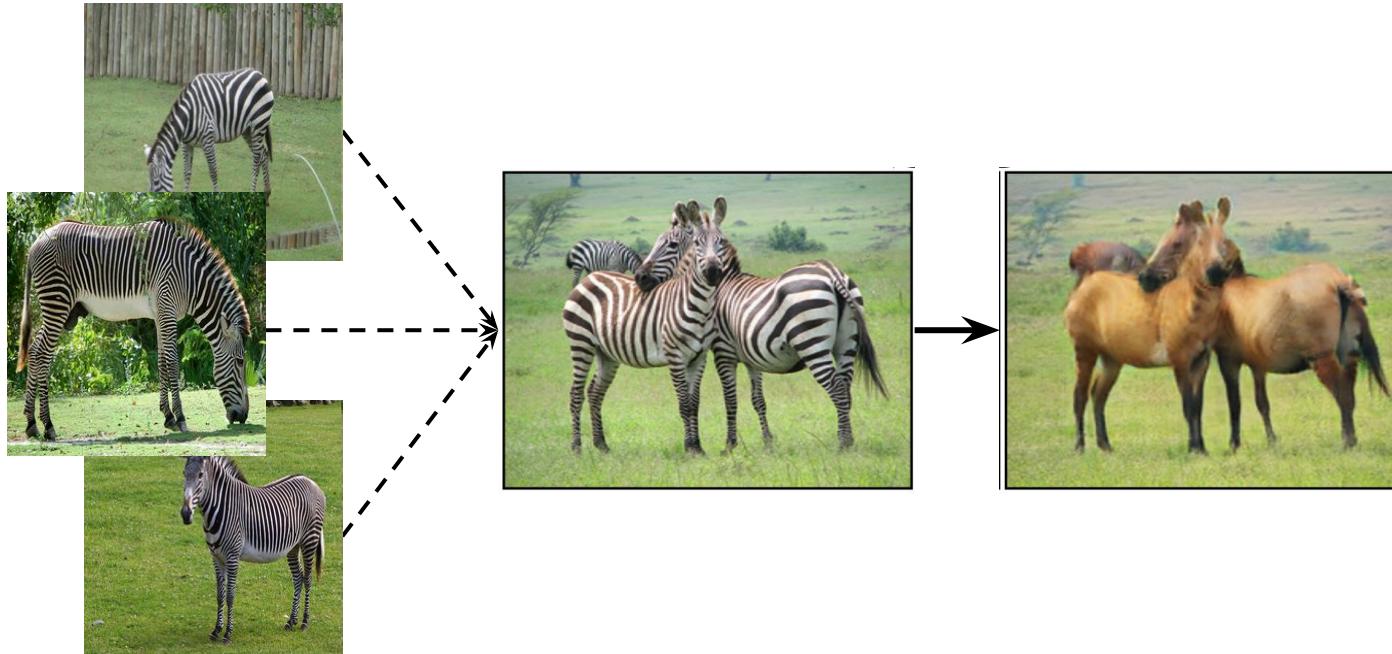
Available from: <https://arxiv.org/abs/1703.10593>

# Unpaired Image-to-Image Translation



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

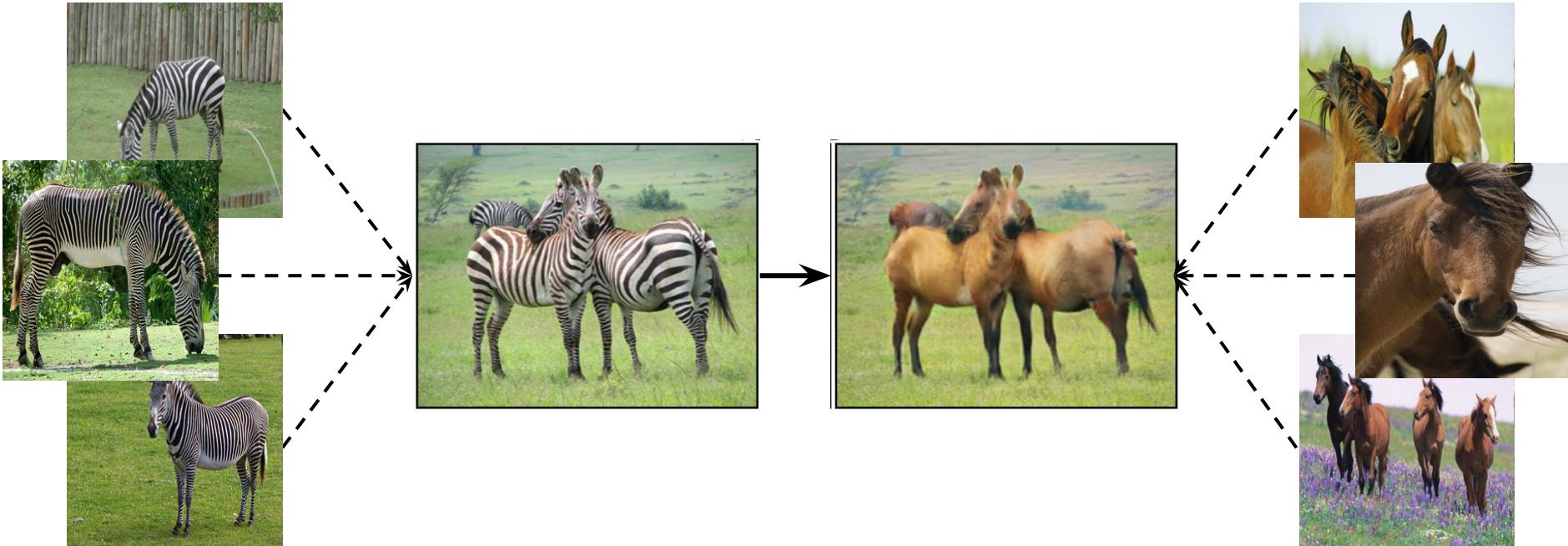
# Mapping Between Two Piles



(Center) Images available from: <https://arxiv.org/abs/1703.10593>

(Side) Images available from: <https://github.com/togheppi/CycleGAN>

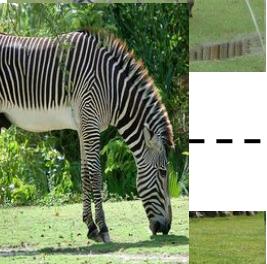
# Mapping Between Two Piles



(Center) Images available from: <https://arxiv.org/abs/1703.10593>

(Sides) Images available from: <https://github.com/togheppi/CycleGAN>

# Mapping Between Two Piles

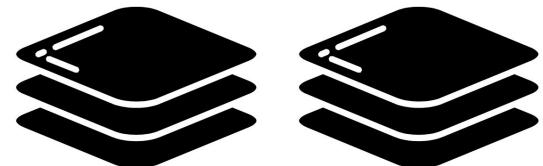


Content = common elements  
Style = unique elements

(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



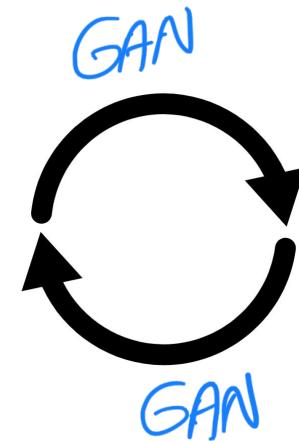


deeplearning.ai

# CycleGAN Overview

# Outline

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



# Cycle Consistency



What to  
generate?

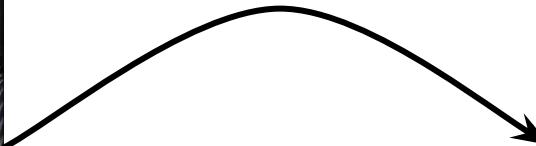
?



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

# Cycle Consistency

Real



Fake

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

# Cycle Consistency

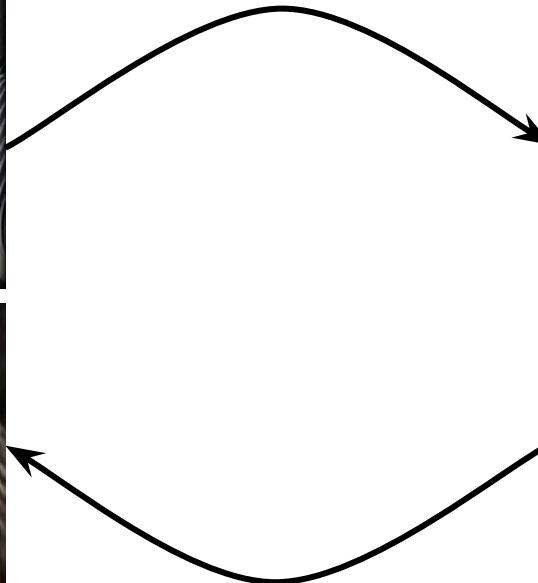
Real



Fake



Fake



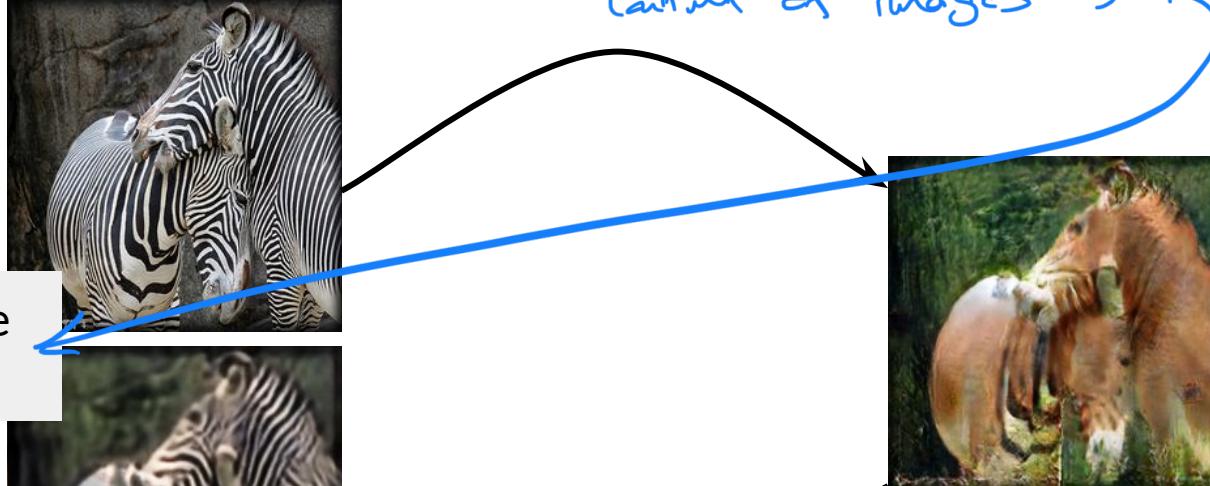
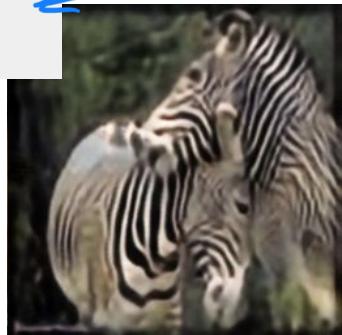
Images available from: <https://github.com/togheppi/CycleGAN>

# Cycle Consistency

Real

Should be the same

Fake

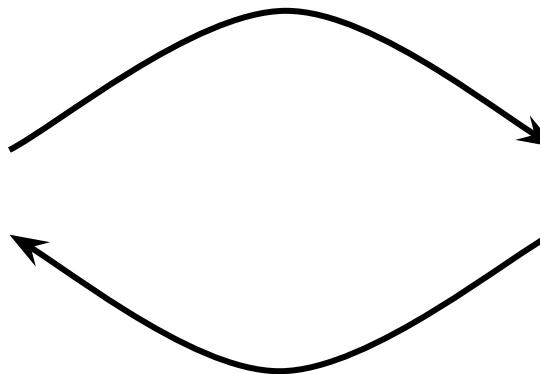


Fake

Content preservation  
Cycle Consistency

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

# Cycle Consistency

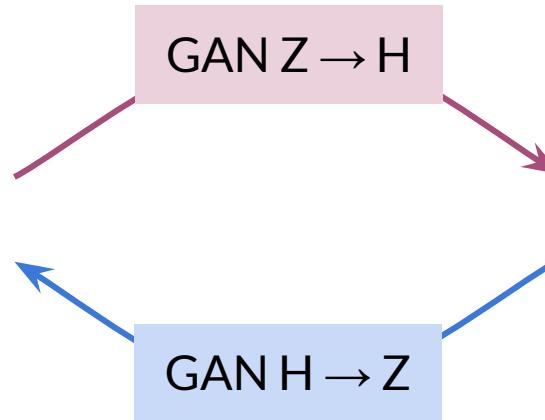
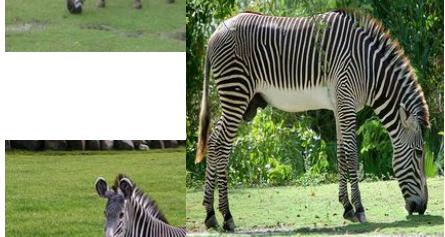


Cycle consistency



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

# Two GANs



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

# Two GANs

Real



GAN Z → H

Fake



GAN H → Z

Fake

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

# Two GANs

Real



GAN  $Z \rightarrow H$

Fake



GAN  $H \rightarrow Z$

Fake

Real



GAN  $H \rightarrow Z$

Fake



GAN  $Z \rightarrow H$

Fake

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

# CycleGAN

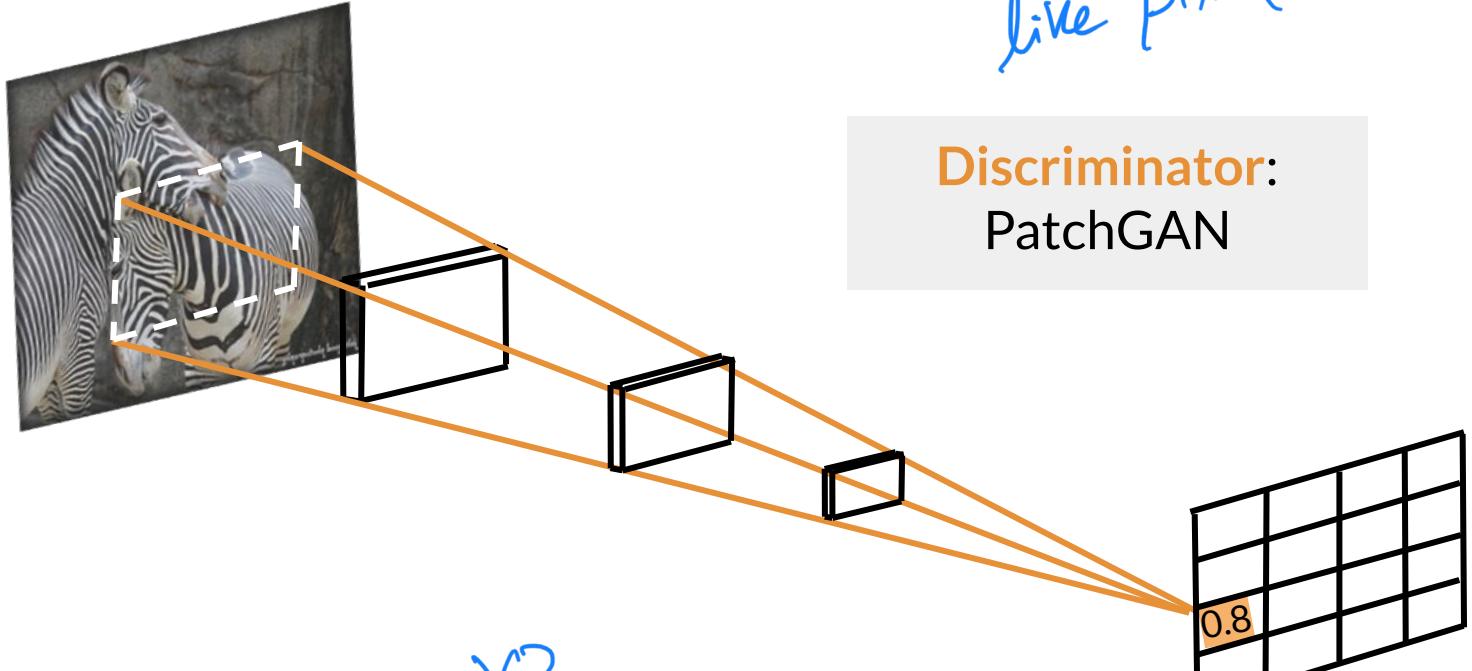
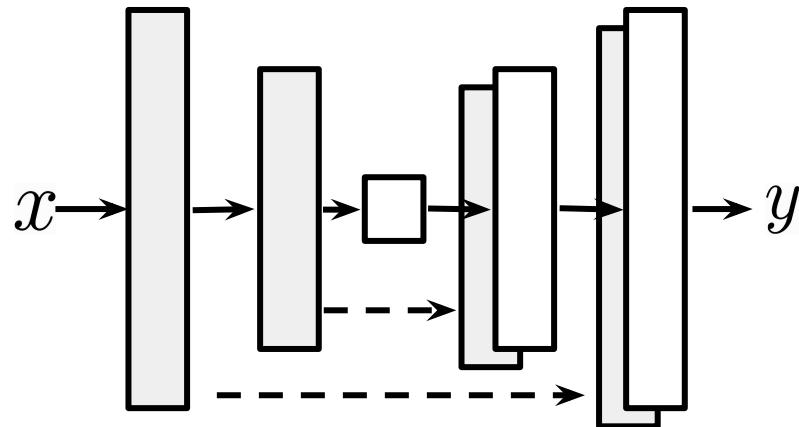


Image available from: <https://github.com/togheppi/CycleGAN>

# CycleGAN

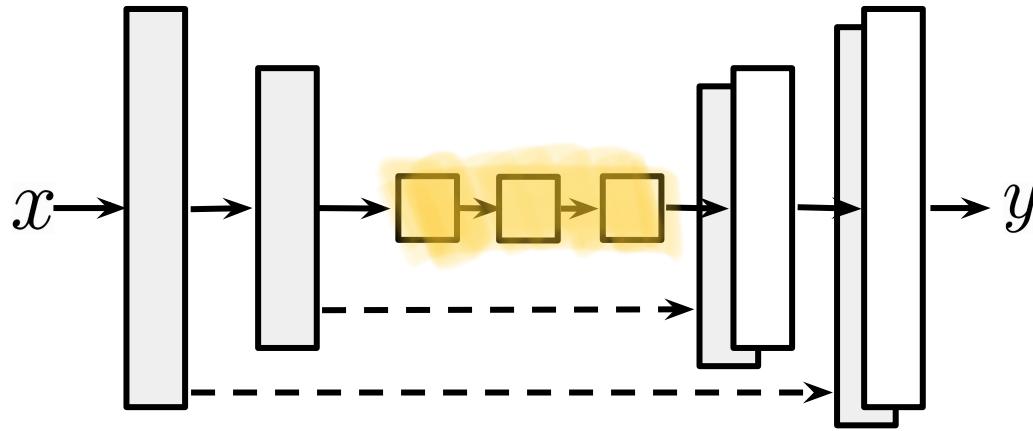


Generator ≈  
U-Net

$\times 2$   
For 2GANs

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

# CycleGAN



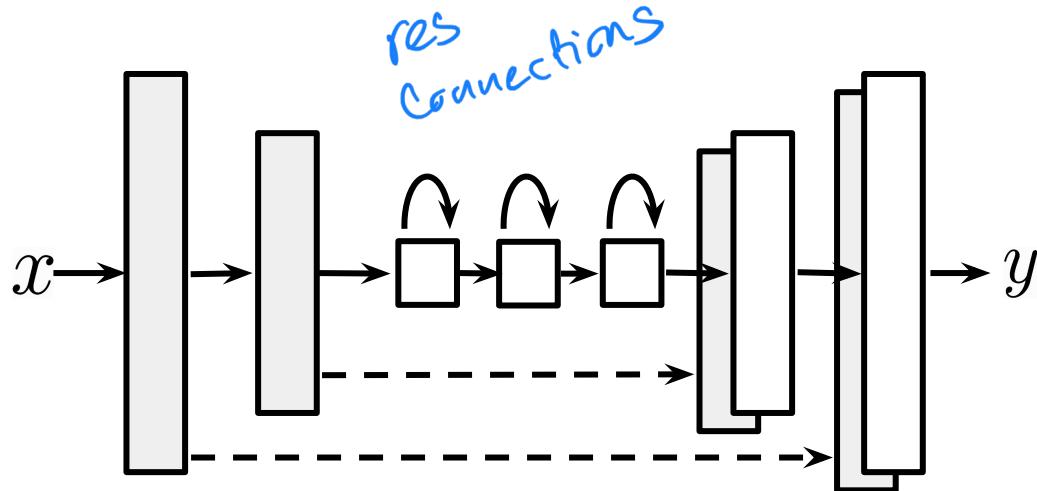
Just an improvement  
nothing essential

Generator ≈  
U-Net + DCGAN  
generator

expand bottleneck  
section with more  
Conv layers

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

# CycleGAN



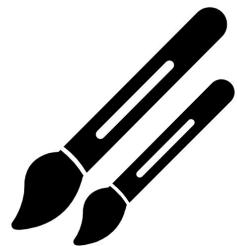
Additional skip connections

**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



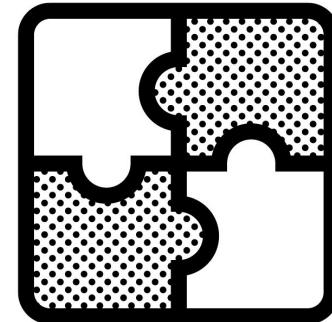


deeplearning.ai

# CycleGAN: Two GANs

# Outline

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

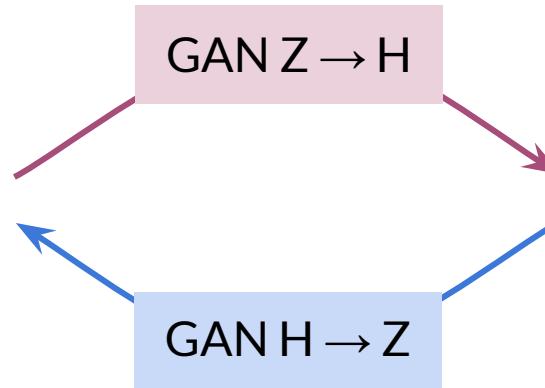


# CycleGAN Components



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

# CycleGAN Components

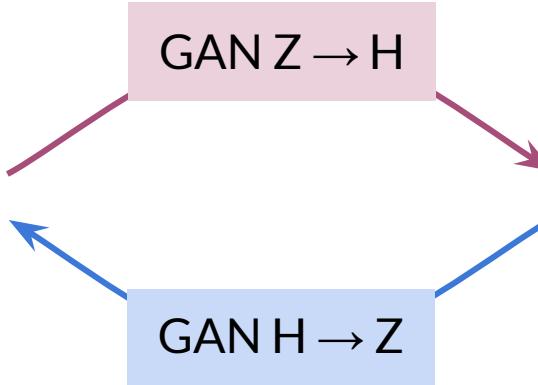


Images available from: <https://github.com/togheppi/CycleGAN>

# CycleGAN Components

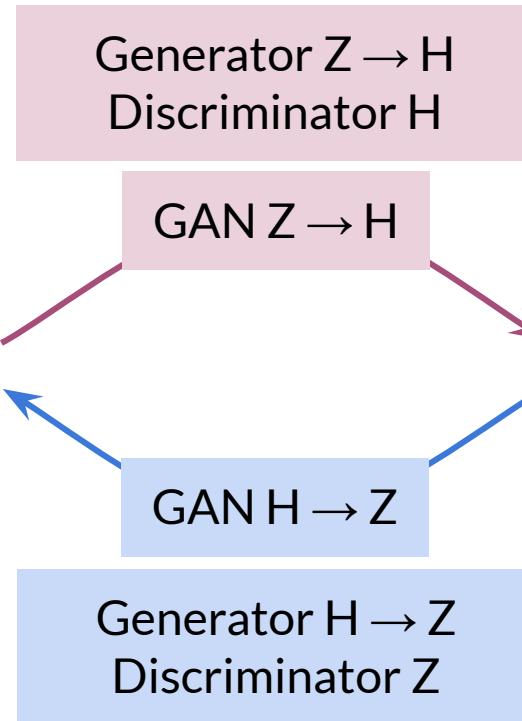
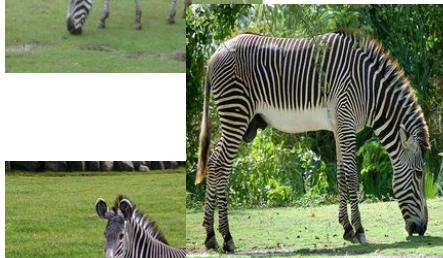


Generator  $Z \rightarrow H$   
Discriminator  $H$



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

# CycleGAN Components



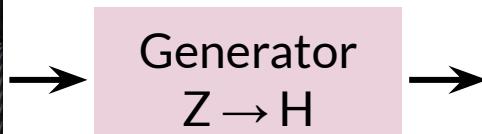
Images available from: <https://github.com/togheppi/CycleGAN>

# GAN $Z \rightarrow H$

Real



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 $Z \rightarrow H$

Real



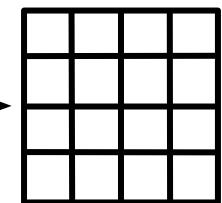
Generator  
 $Z \rightarrow H$

Fake



Discriminator  
 $H$

Classification matrix

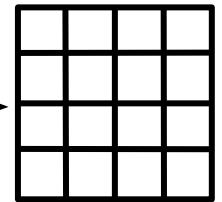


Real



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 $H \rightarrow Z$

Real



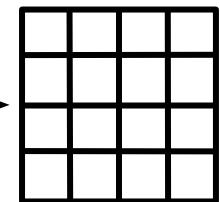
Generator  
 $H \rightarrow Z$

Fake

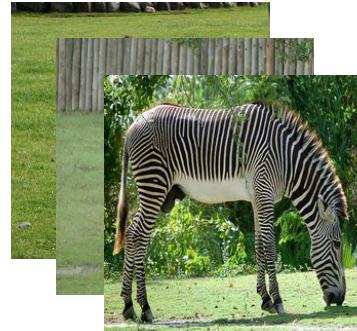


Discriminator  
 $Z$

Classification matrix

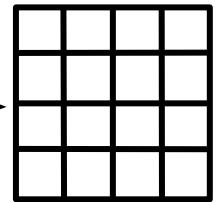


Real



Discriminator  
 $Z$

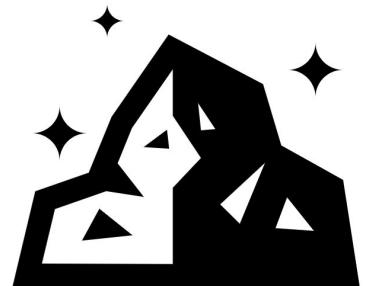
Classification matrix



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



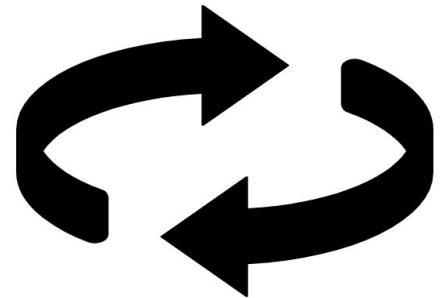


deeplearning.ai

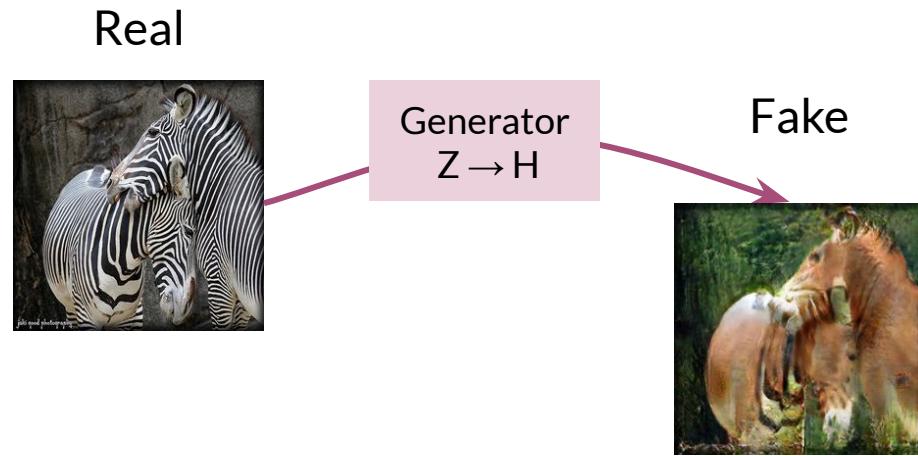
# CycleGAN: Cycle Consistency

# 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

Real



Generator  
 $Z \rightarrow H$

Fake



Generator  
 $H \rightarrow Z$

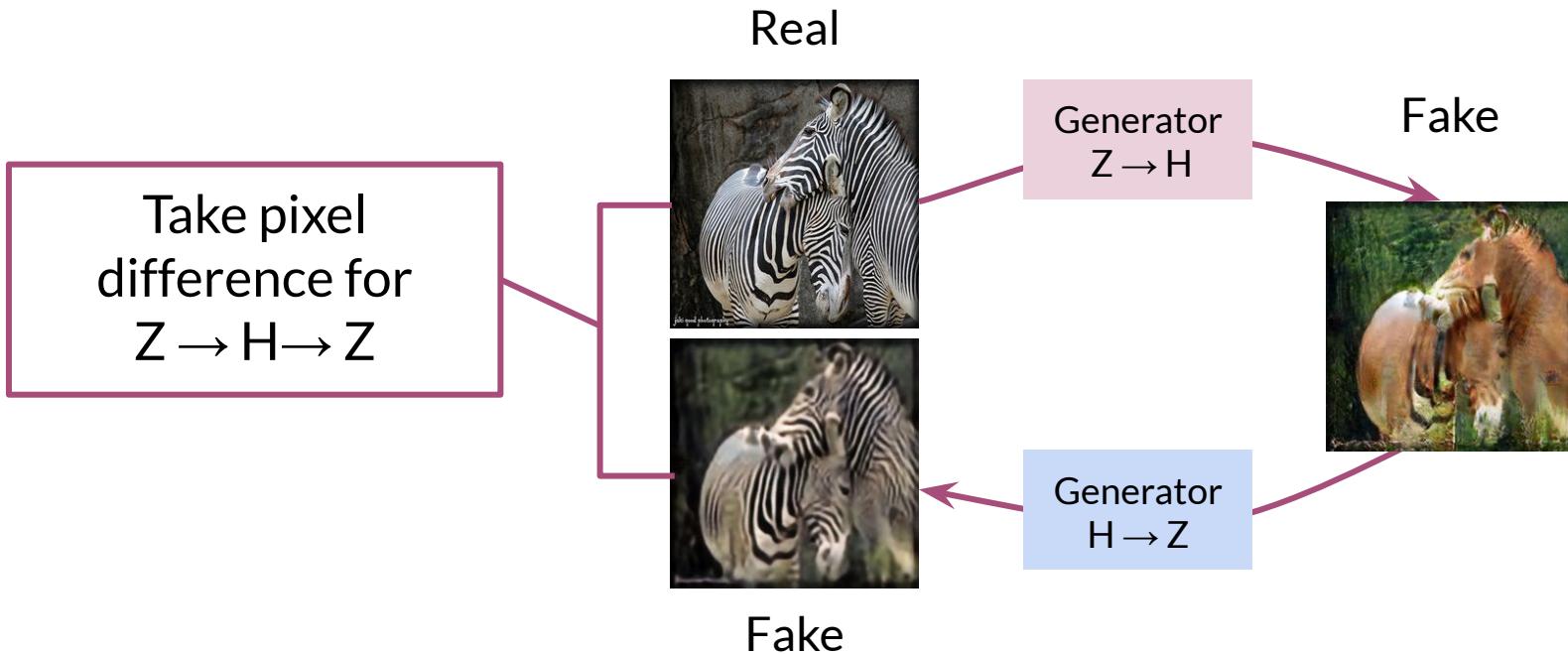
Fake



1

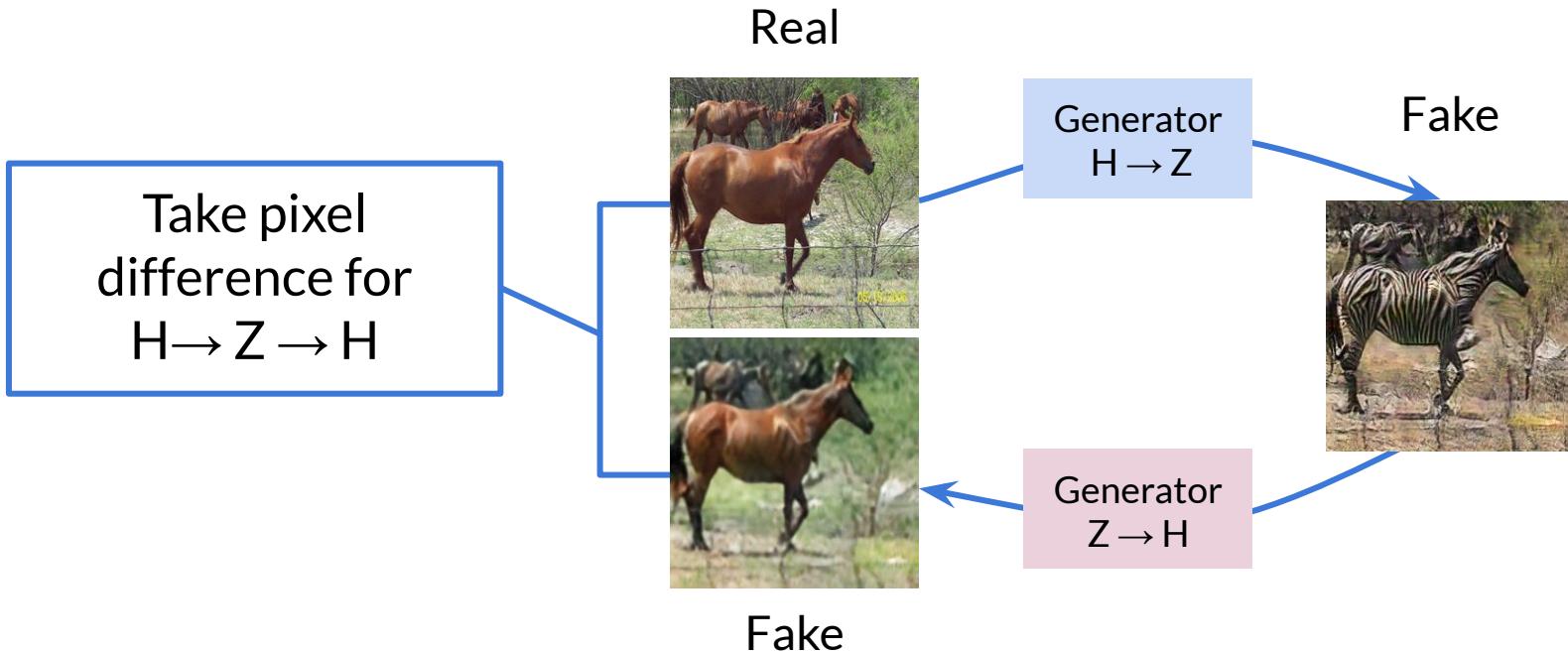
Images available from: <https://github.com/togheppi/CycleGAN>

# 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

$$\sum_i |$$



—



+

$$\sum_i |$$



—



$Z \rightarrow H \rightarrow Z$

$H \rightarrow Z \rightarrow H$

Cycle Consistency Loss is the  
sum of both directions

# Cycle Consistency Loss

Adversarial Loss +

$$\sum_i | \begin{array}{c} \text{zebra} \\ \text{---} \\ \text{horse} \end{array} - \begin{array}{c} \text{zebra} \\ \text{---} \\ \text{horse} \end{array} | + \sum_i | \begin{array}{c} \text{horse} \\ \text{---} \\ \text{zebra} \end{array} - \begin{array}{c} \text{horse} \\ \text{---} \\ \text{zebra} \end{array} |$$

$Z \rightarrow H \rightarrow Z$        $H \rightarrow Z \rightarrow H$

# Cycle Consistency Loss

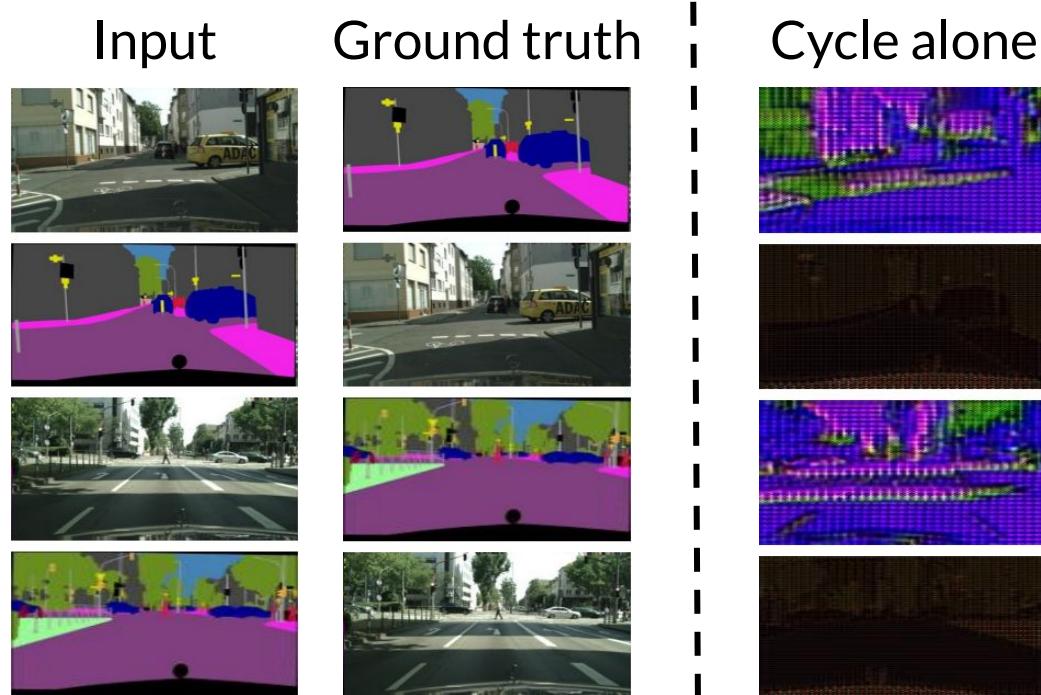
Adversarial Loss +

Cycle Consistency Loss

# Cycle Consistency Loss

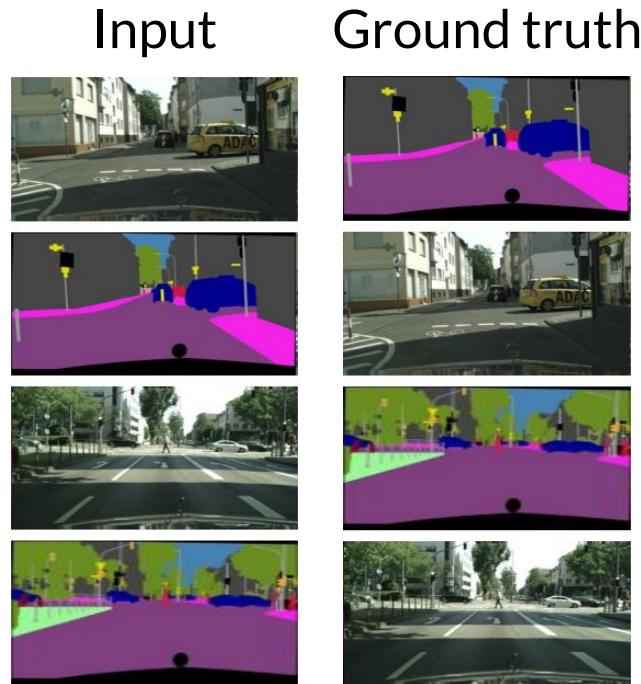
Adversarial Loss +  $\lambda$  \* Cycle Consistency Loss

# Ablation Studies

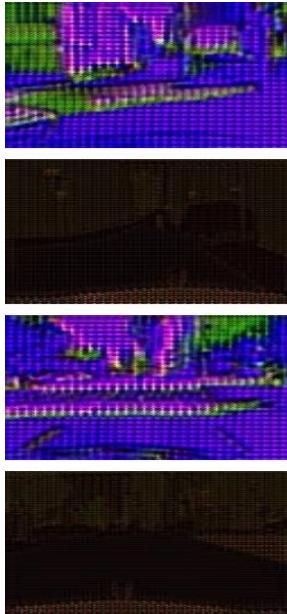


Available from: <https://arxiv.org/abs/1703.10593>

# Ablation Studies



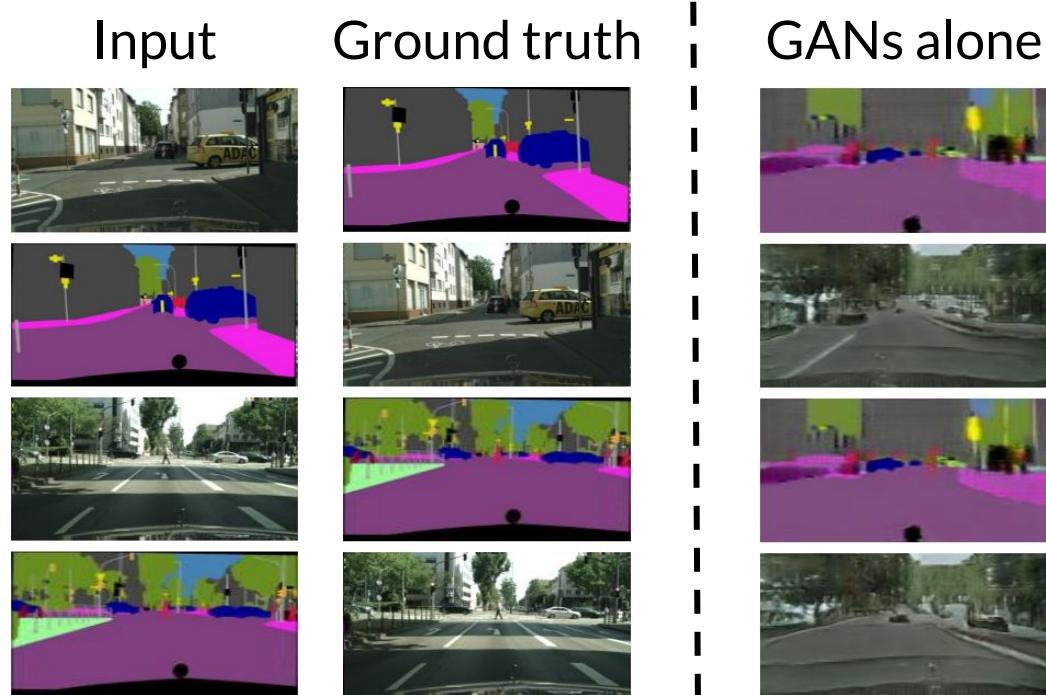
Cycle alone



Without Adversarial  
GAN Loss, outputs are  
not realistic

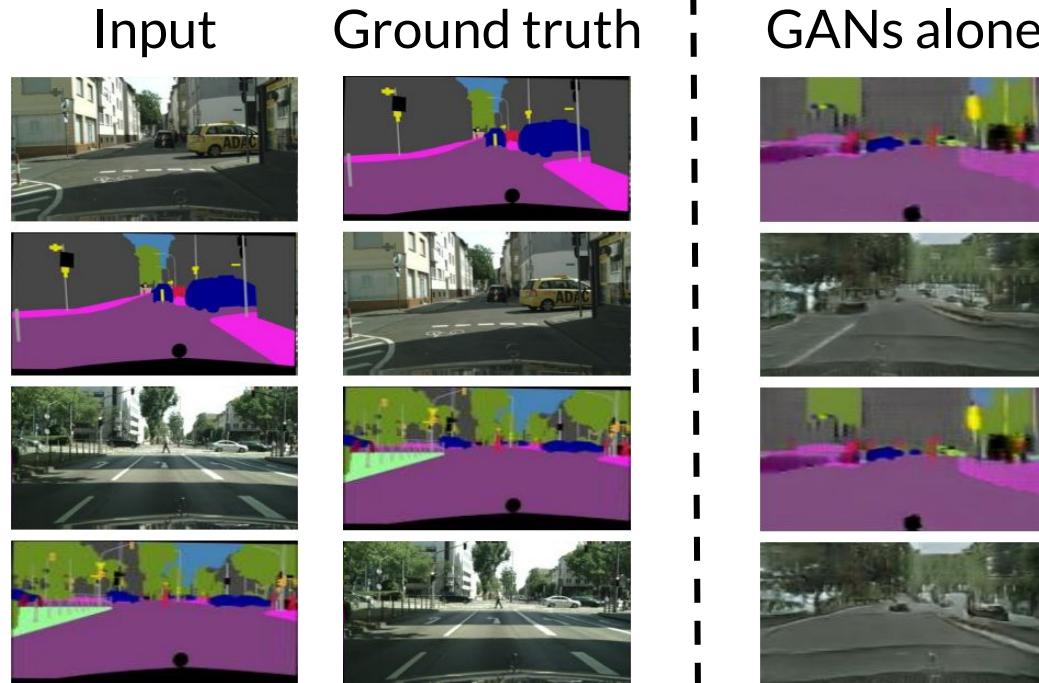
Available from: <https://arxiv.org/abs/1703.10593>

# Ablation Studies



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

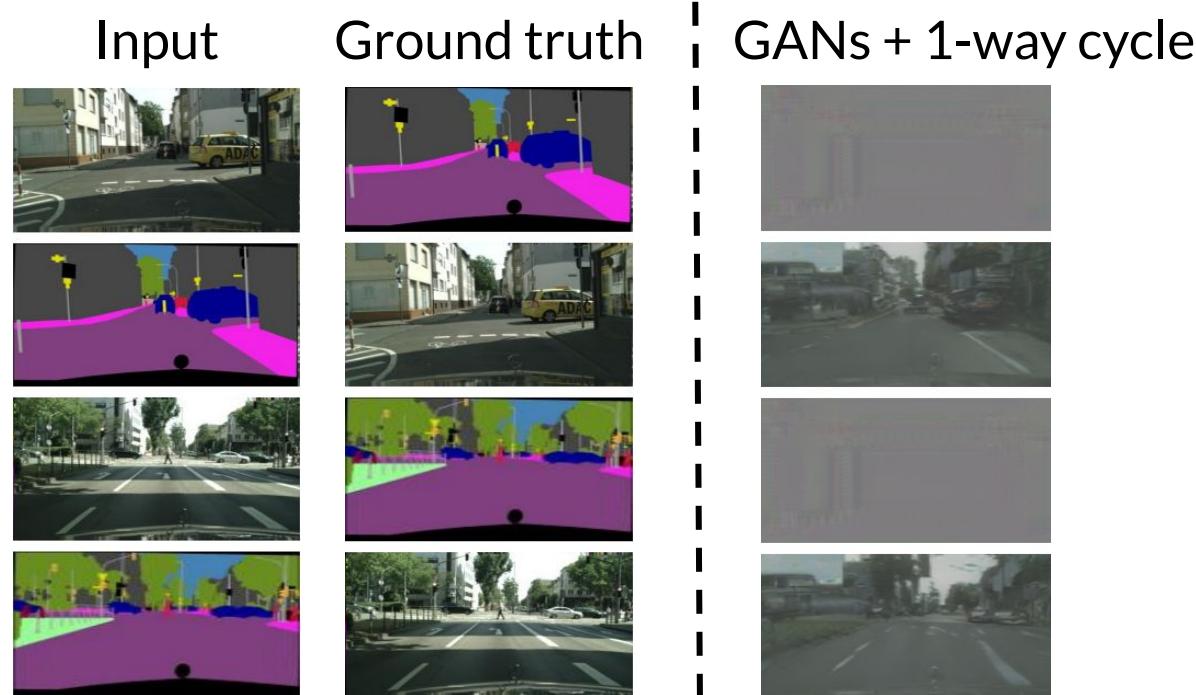
# Ablation Studies



Without Cycle Consistency Loss, outputs show signs of mode collapse

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

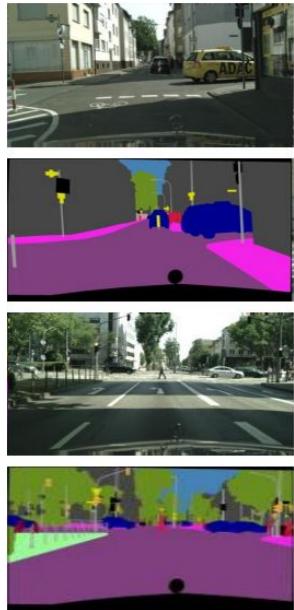
# Ablation Studies



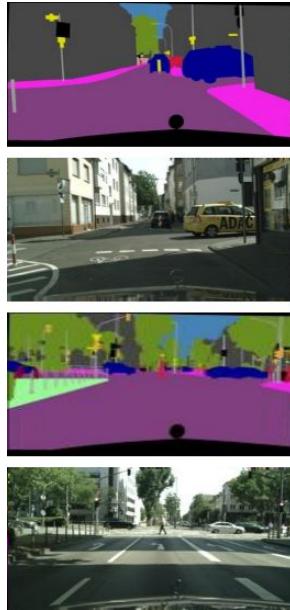
Available from: <https://arxiv.org/abs/1703.10593>

# Ablation Studies

Input



Ground truth



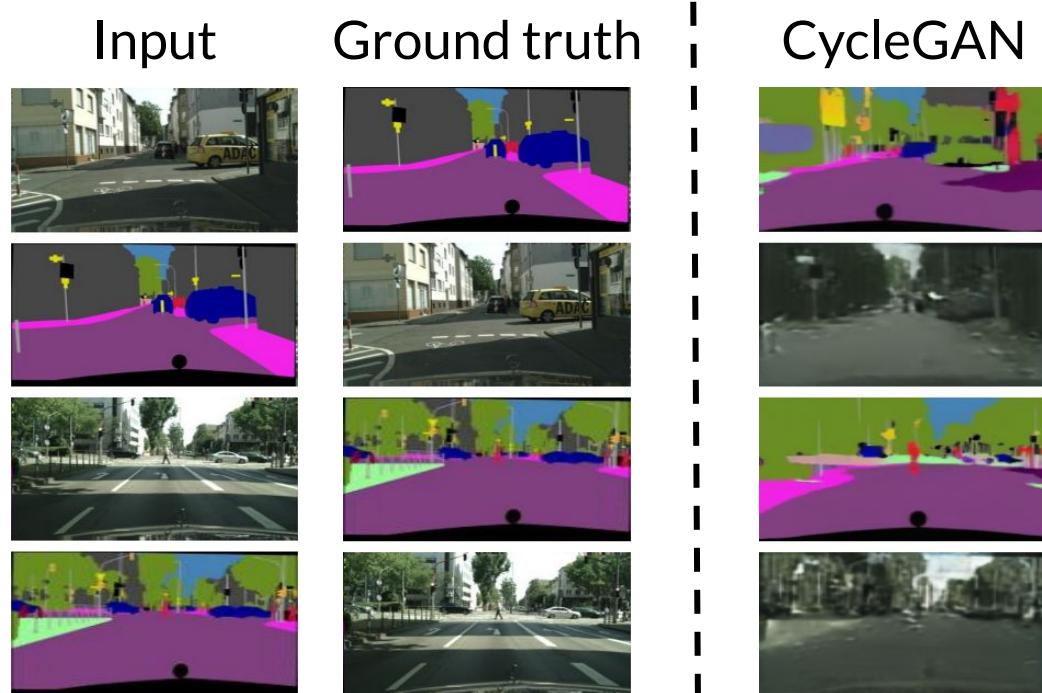
GANs + 1-way cycle



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

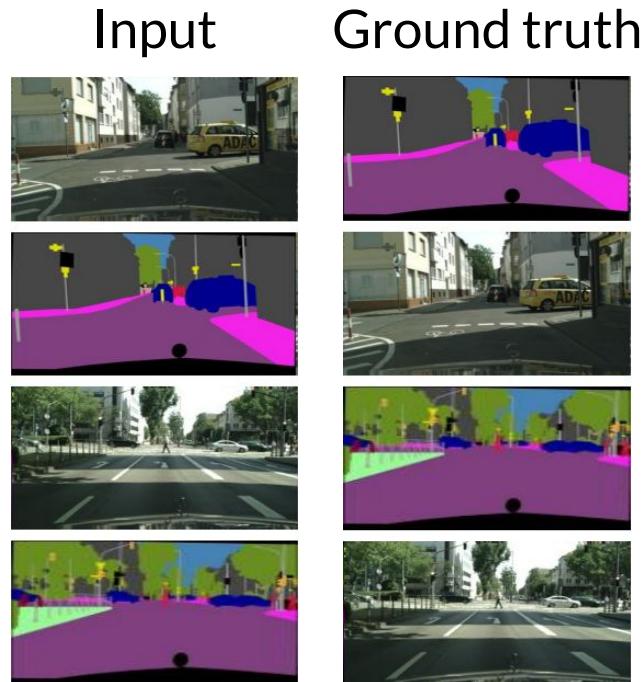
Available from: <https://arxiv.org/abs/1703.10593>

# Ablation Studies



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

# Ablation Studies



CycleGAN

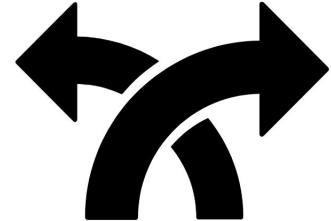


**CycleGAN uses both  
Adversarial Loss and  
Cycle Consistency Loss**

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

# 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



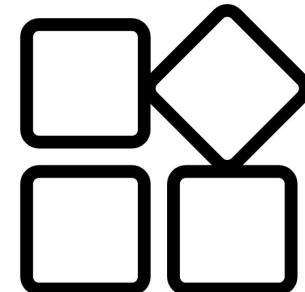


deeplearning.ai

# 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



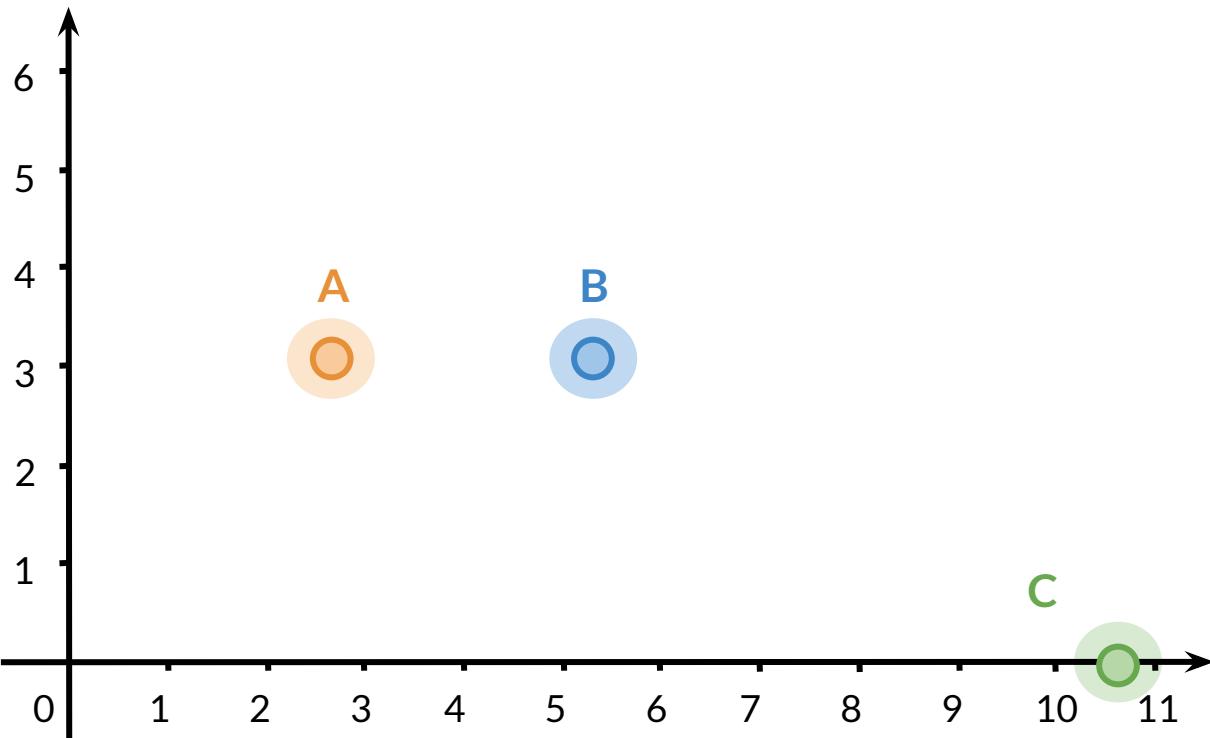
# Least Squares Loss: Another GAN Loss Function

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

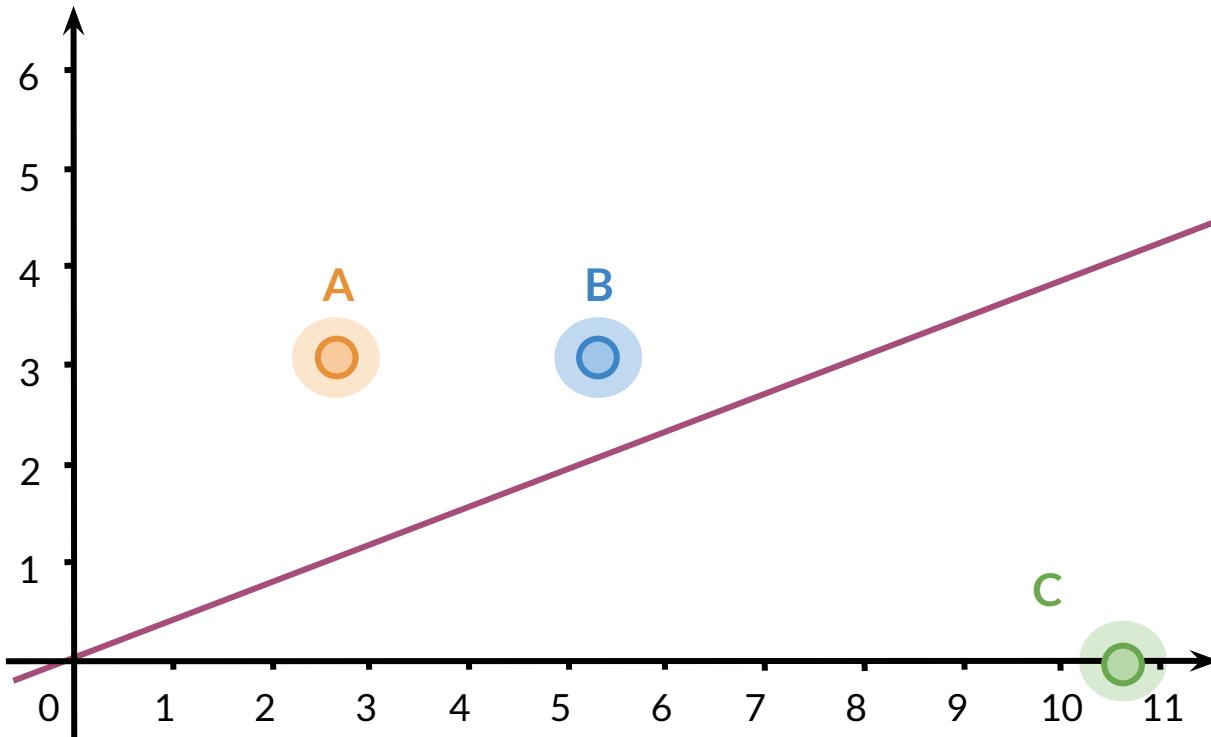


- GAN loss functions are chosen empirically

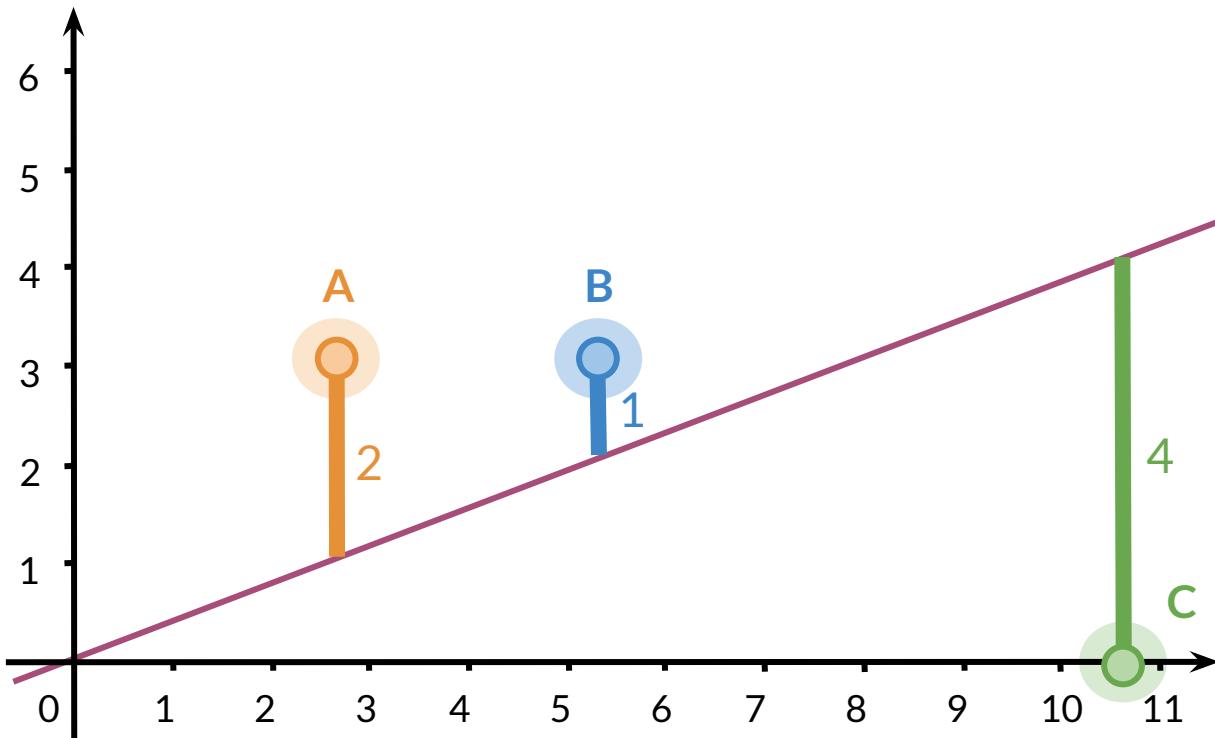
# Least Squares



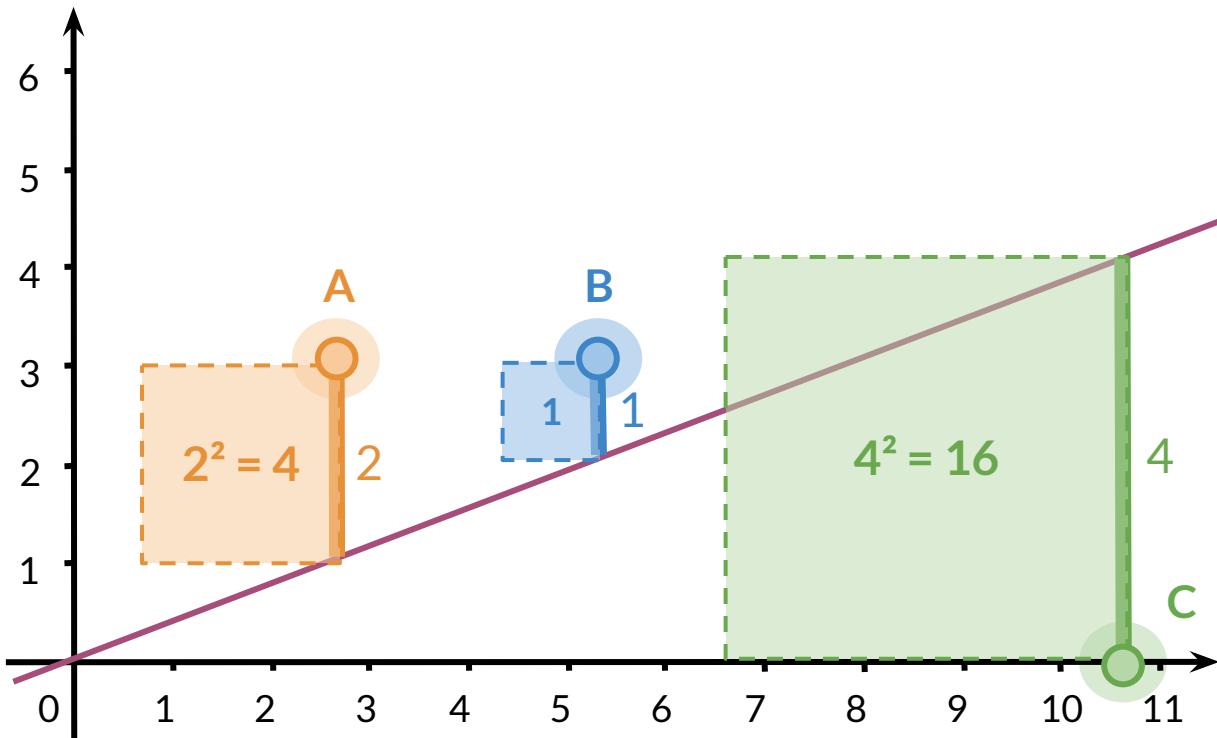
# Least Squares



# Least Squares

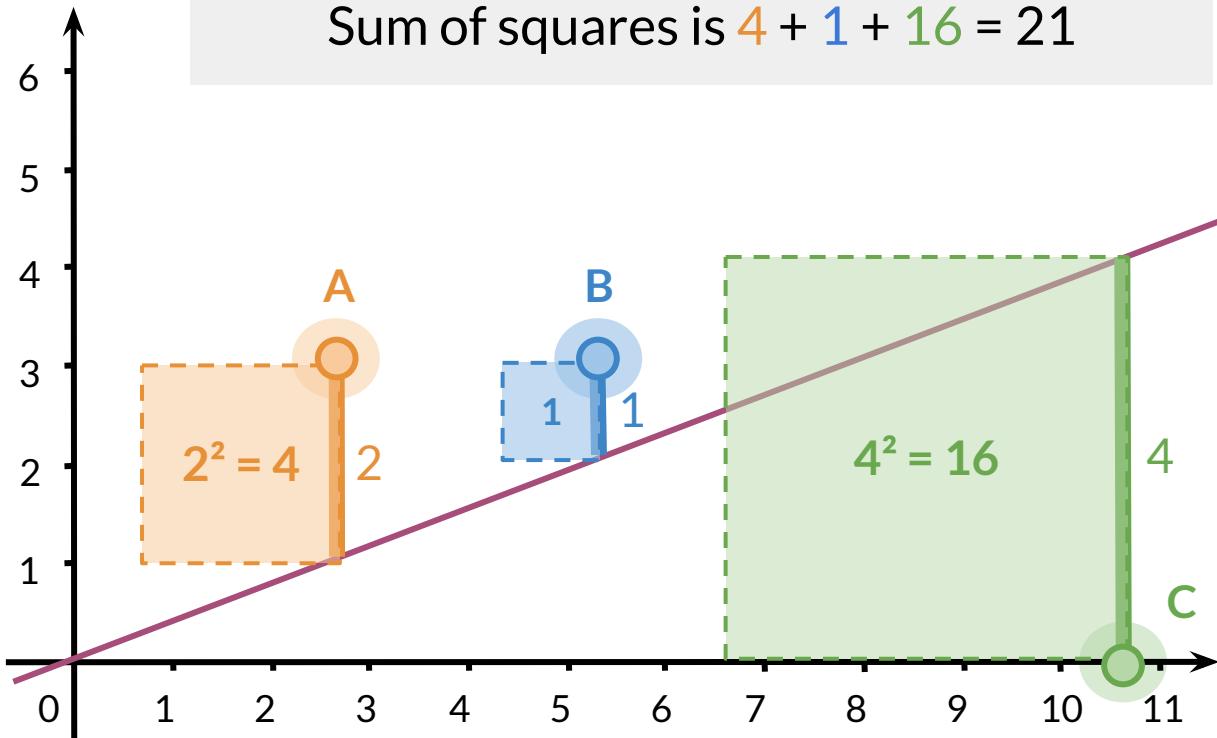


# Least Squares



# Least Squares

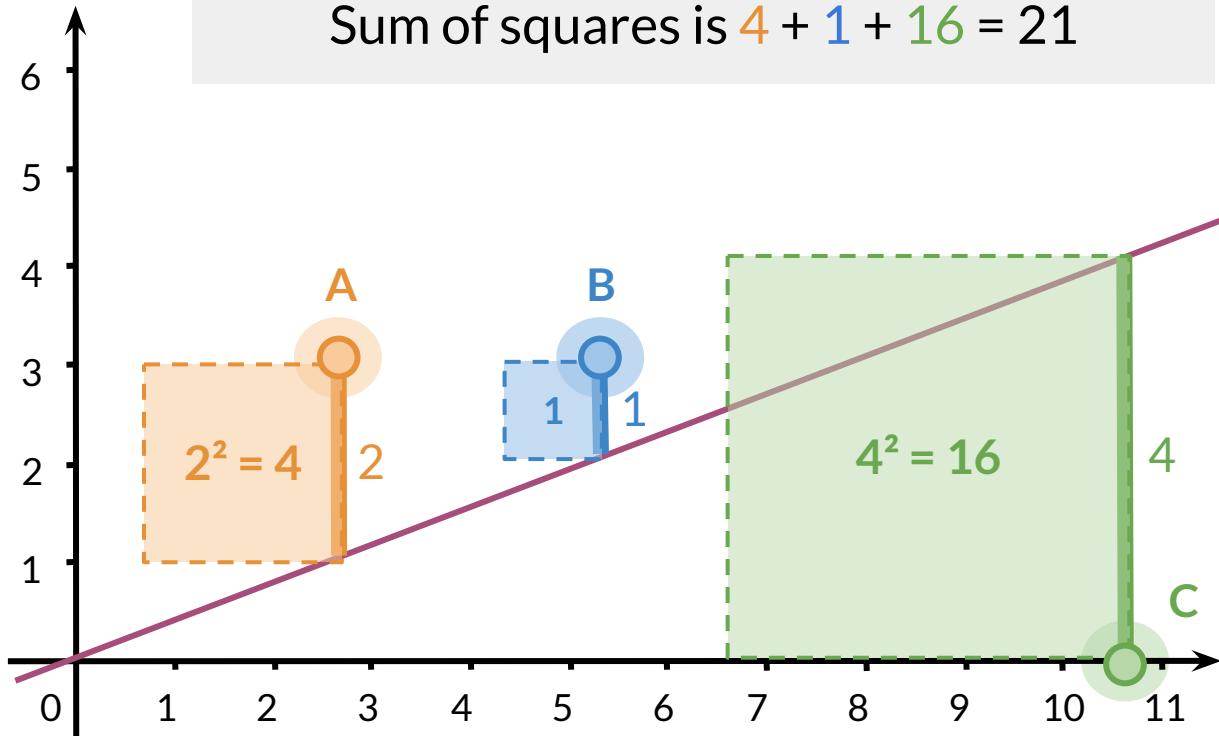
Sum of squares is  $4 + 1 + 16 = 21$



# Least Squares

Minimize  
sum of squares

Sum of squares is  $4 + 1 + 16 = 21$



# Least Squares Loss: Discriminator

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



Discriminator classification  
of real image  $\mathbf{x}$

# Least Squares Loss: Discriminator

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

# Least Squares Loss: Discriminator

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

Discriminator classification  
of fake image  $G(\mathbf{z})$

# Least Squares Loss: Discriminator

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

# Least Squares Loss: Discriminator

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

# Least Squares Loss: Generator

$$\mathbb{E}_z [(D(G(z)) - 1)^2]$$

# Least Squares Loss

Discriminator  
Loss

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

# Least Squares Loss

Discriminator  
Loss

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

Generator  
Loss

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

# Least Squares Loss

Discriminator  
Loss

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

Generator  
Loss

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

Reduces vanishing gradient problem

# Least Squares Loss

Discriminator  
Loss

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

Generator  
Loss

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

Also known as Mean Squared Error!

# Context of Least Squares Loss

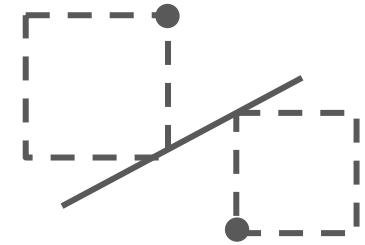
Adversarial Loss +  $\lambda$  \* Cycle Consistency Loss



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



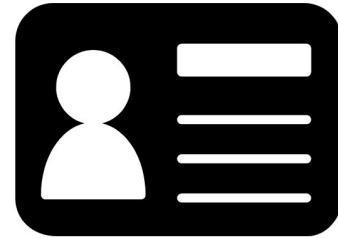


deeplearning.ai

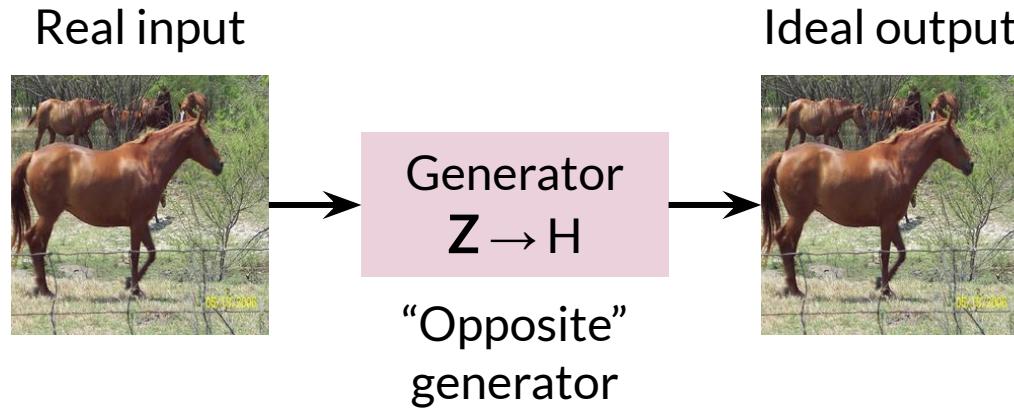
# CycleGAN: Identity Loss

# Outline

- Identity Loss
  - How it works
  - Impact on outputs

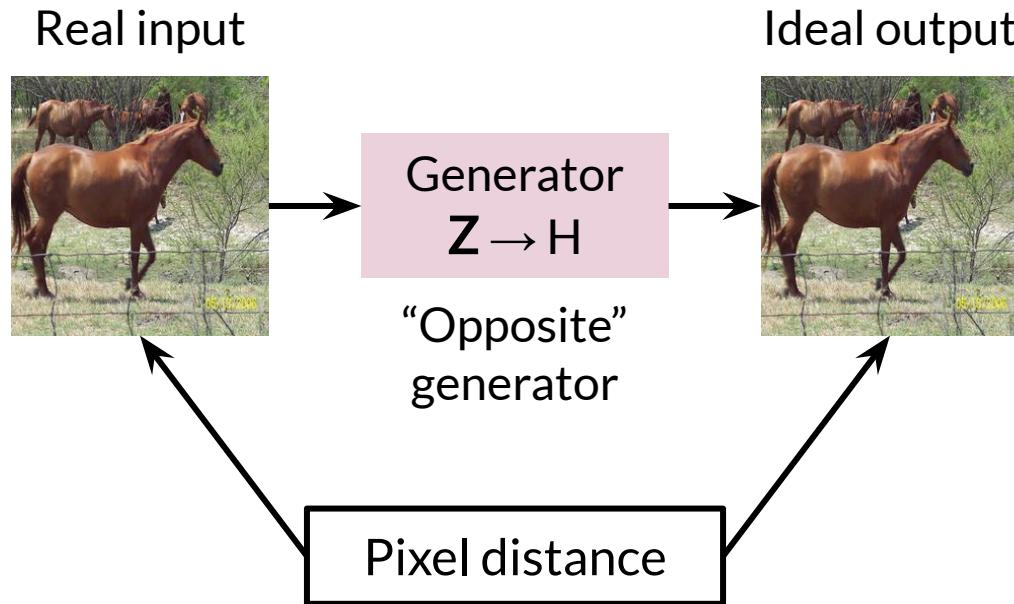


# Identity Loss



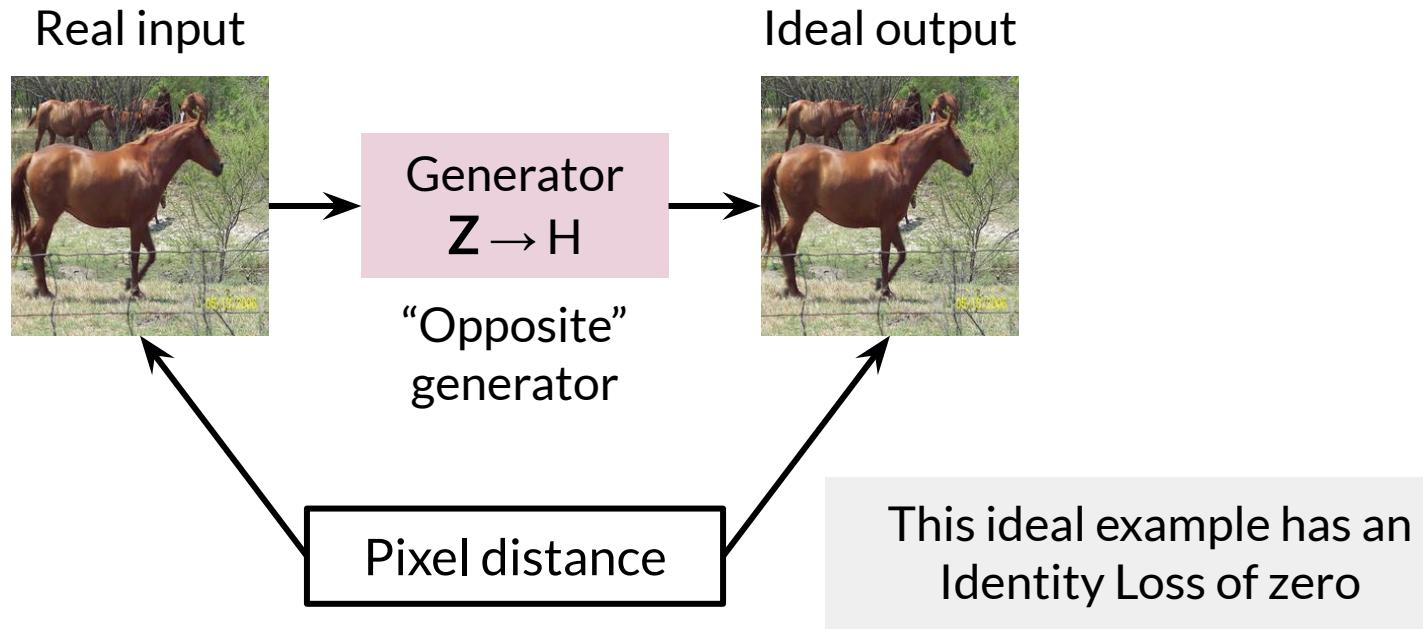
Images available from: <https://github.com/togheppi/CycleGAN>

# Identity Loss



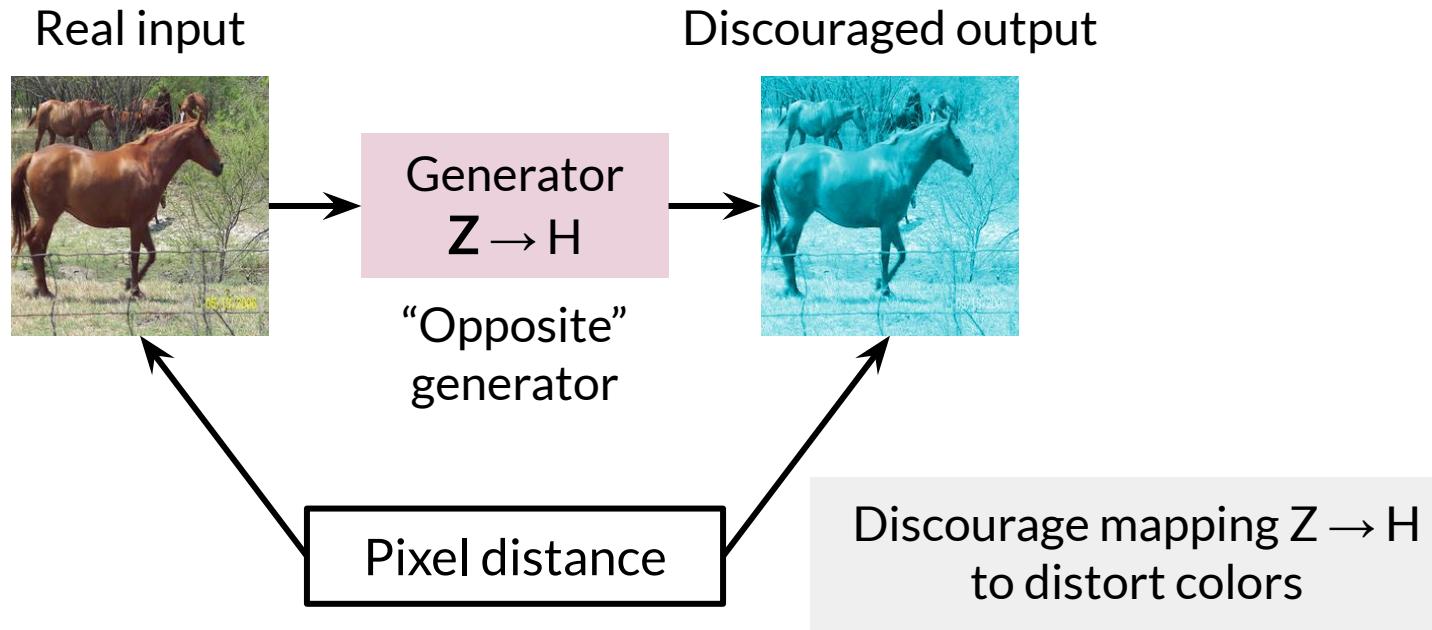
Images available from: <https://github.com/togheppi/CycleGAN>

# Identity Loss



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

# Identity Loss



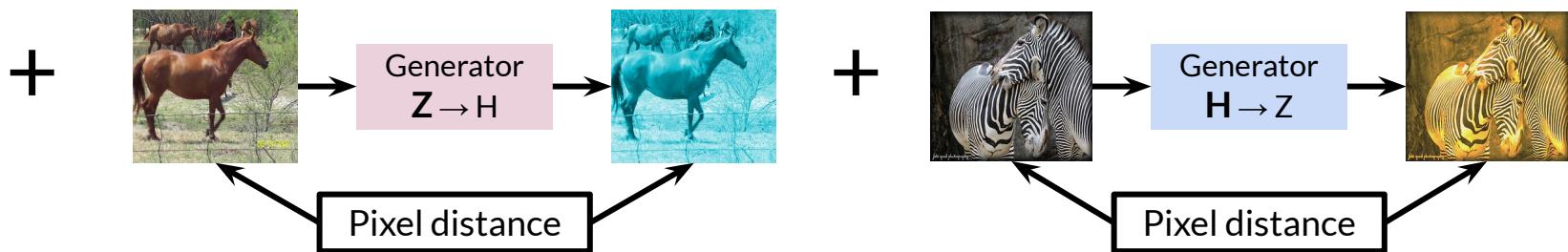
Images available from: <https://github.com/togheppi/CycleGAN>

# Context of Identity Loss

Adversarial Loss +  $\lambda$  \* Cycle Consistency Loss

# Context of Identity Loss

Adversarial Loss +  $\lambda$  \* Cycle Consistency Loss



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

# Context of Identity Loss

Adversarial Loss +  $\lambda$  \* 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:  $A \rightarrow B$ , 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



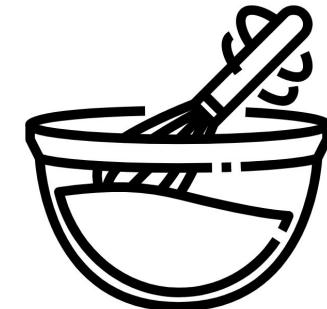


deeplearning.ai

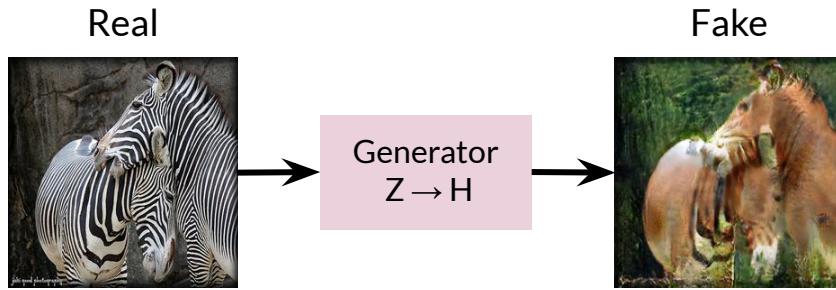
# CycleGAN: Putting It All Together

# Outline

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

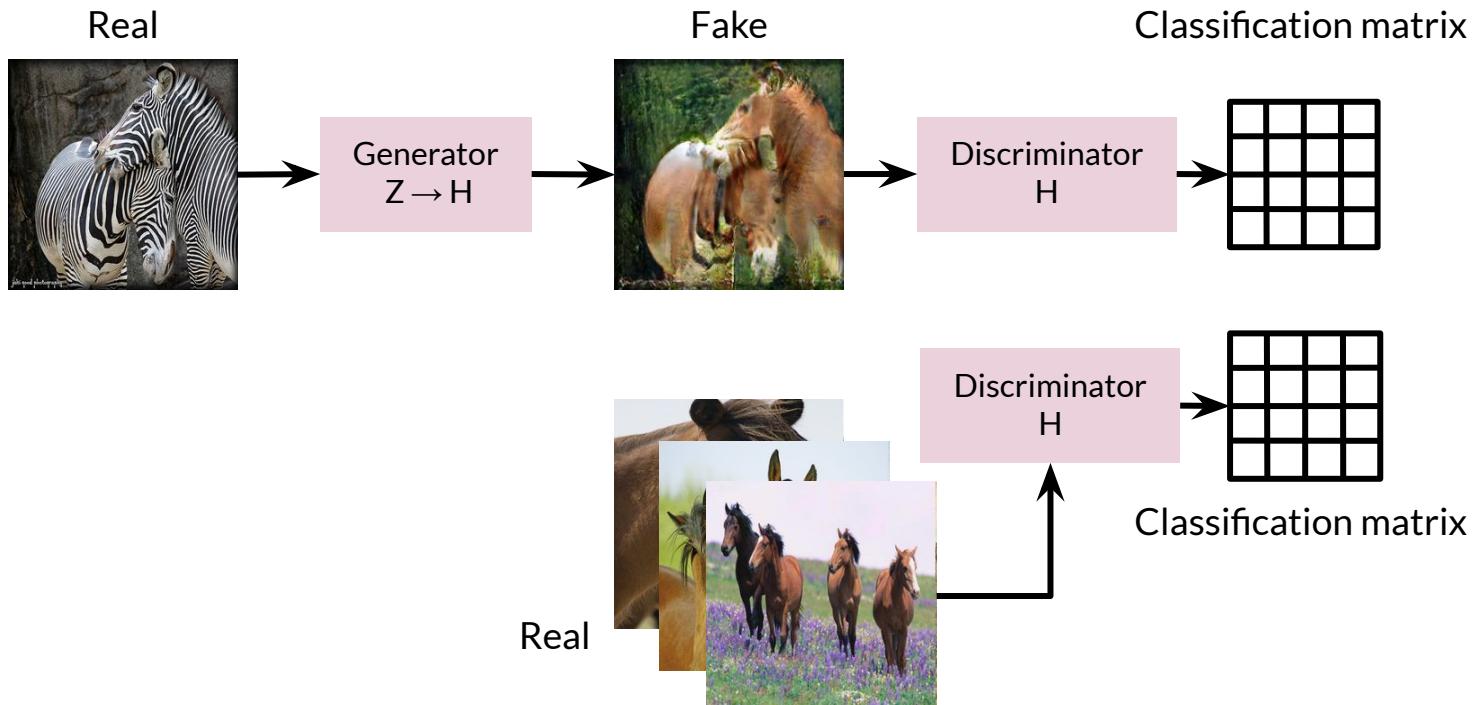


# CycleGAN



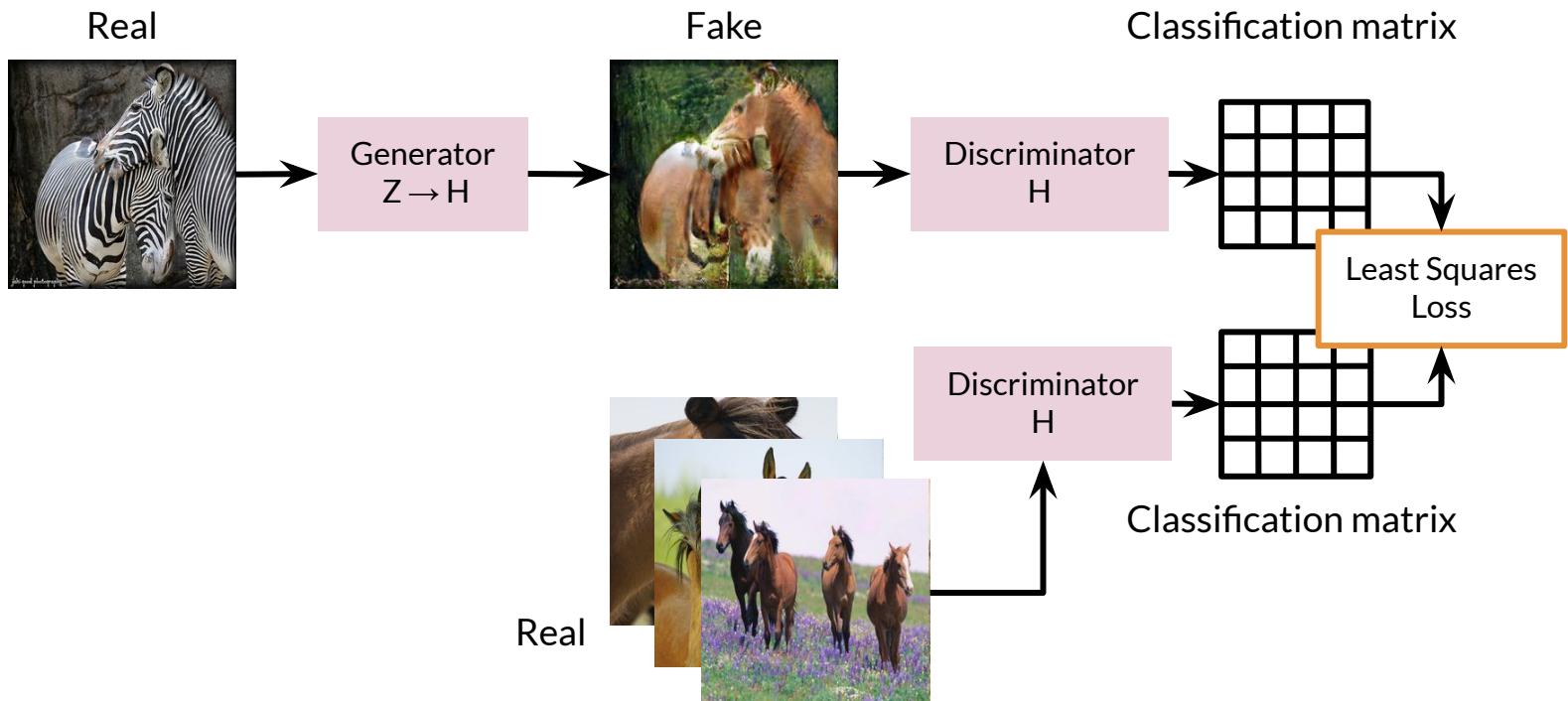
Images available from: <https://github.com/togheppi/CycleGAN>

# CycleGAN



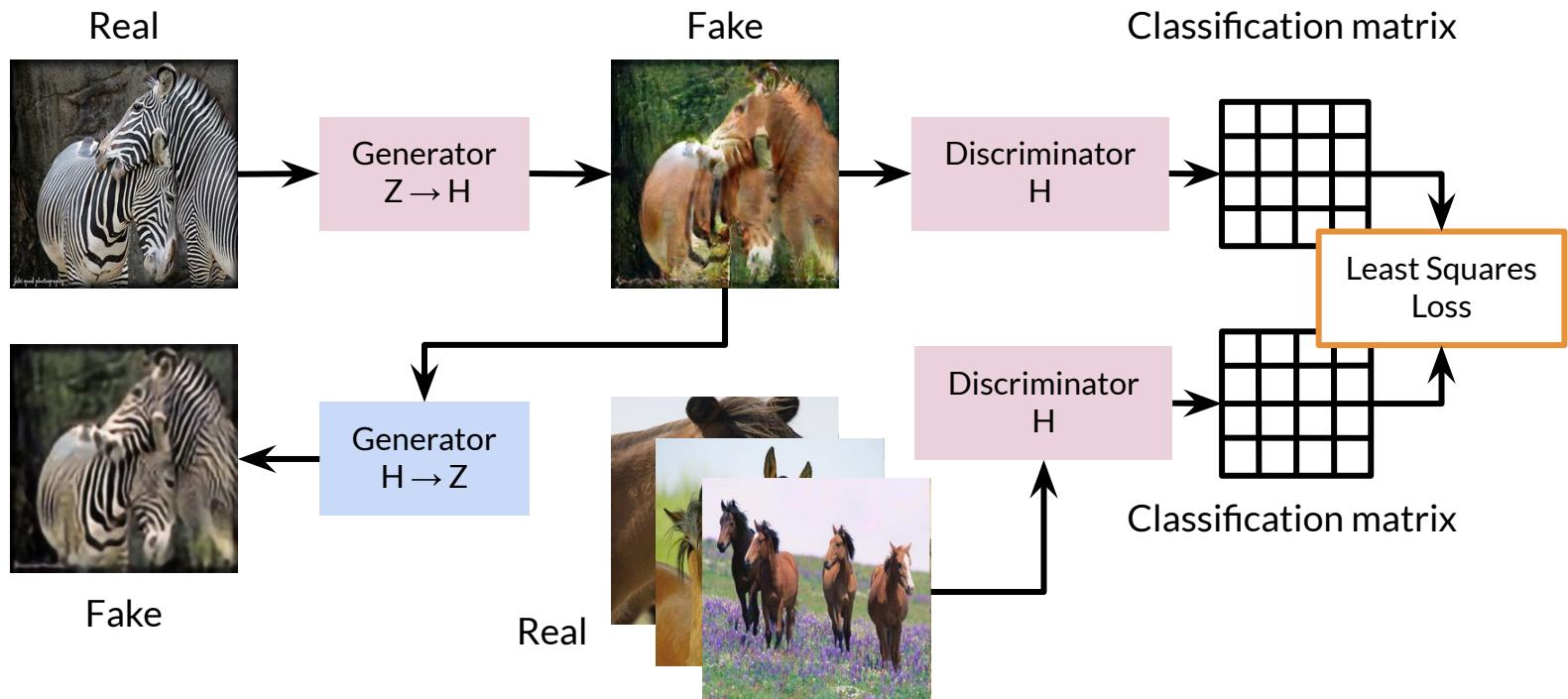
Images available from: <https://github.com/togheppi/CycleGAN>

# CycleGAN



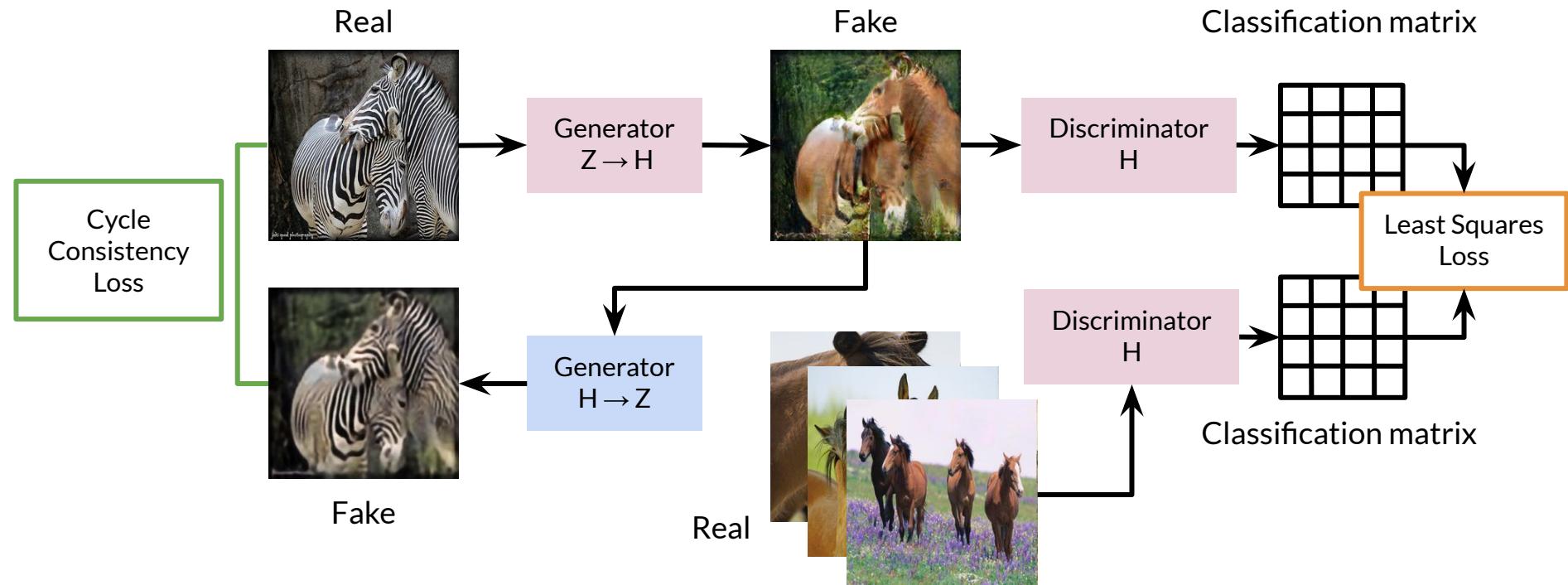
Images available from: <https://github.com/togheppi/CycleGAN>

# CycleGAN



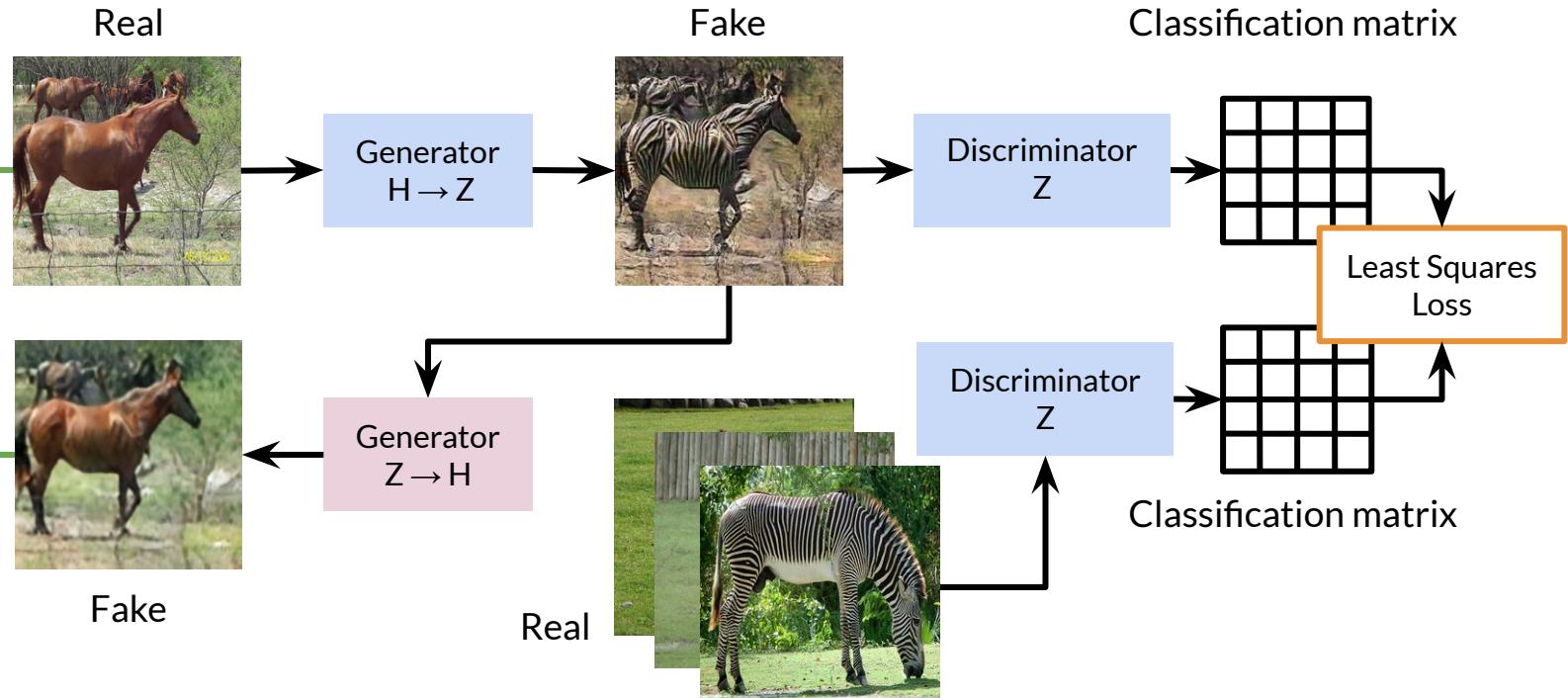
Images available from: <https://github.com/togheppi/CycleGAN>

# CycleGAN



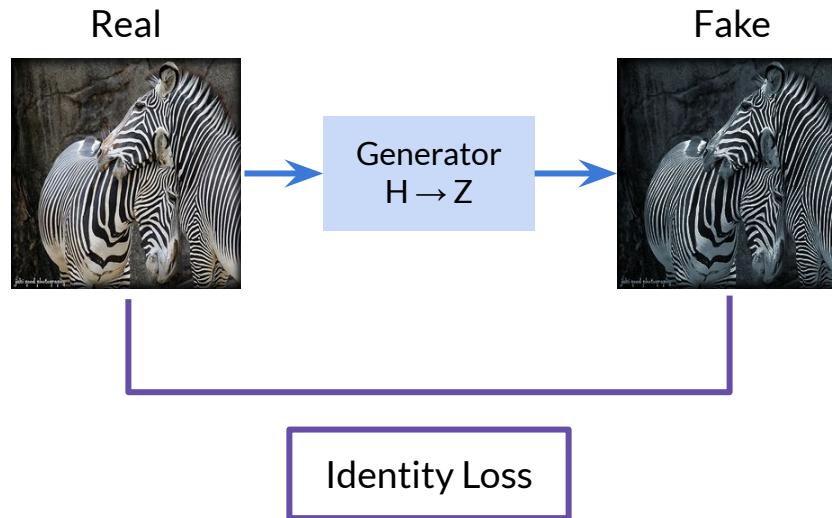
Images available from: <https://github.com/togheppi/CycleGAN>

# CycleGAN



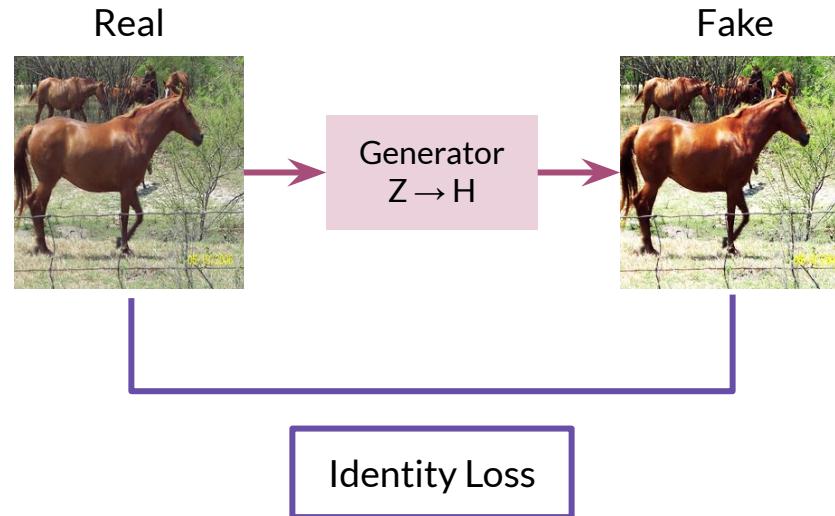
Images available from: <https://github.com/togheppi/CycleGAN>

# CycleGAN



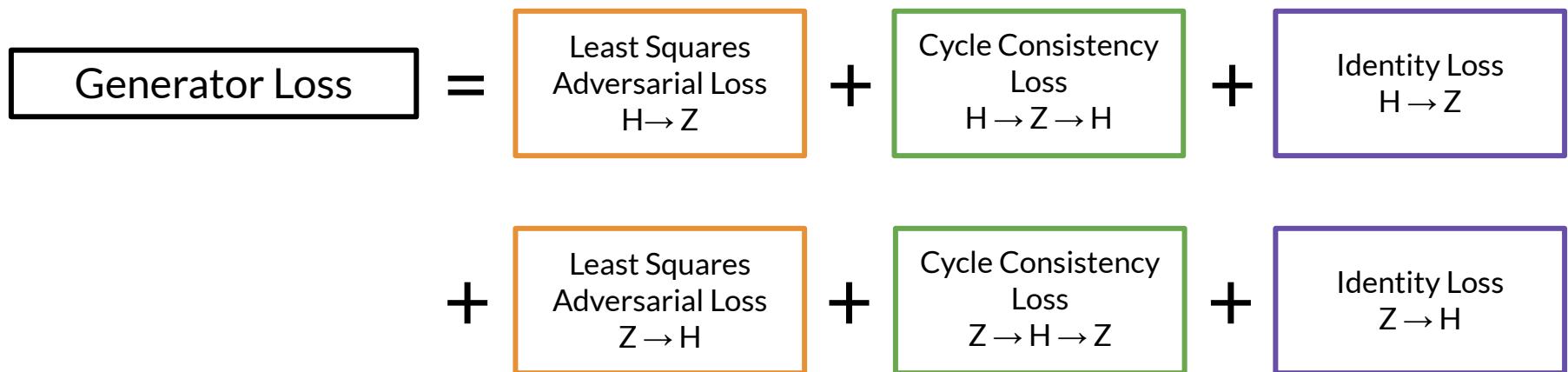
Images available from: <https://github.com/togheppi/CycleGAN>

# CycleGAN



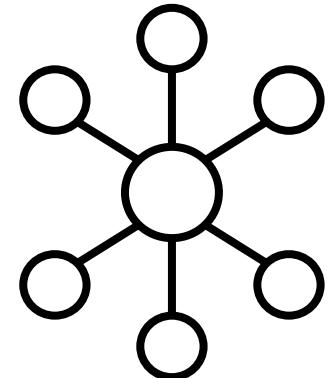
Images available from: <https://github.com/togheppi/CycleGAN>

# CycleGAN Loss



# 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

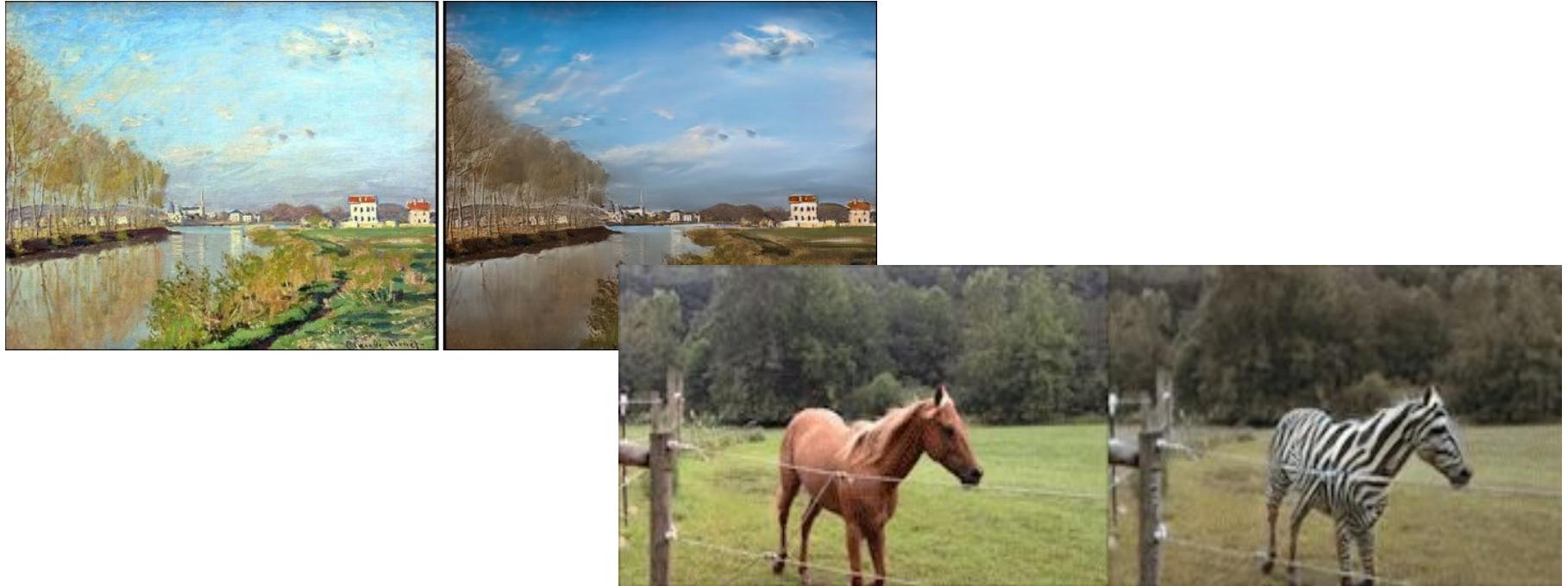
# CycleGAN Applications & Variants

# Outline

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

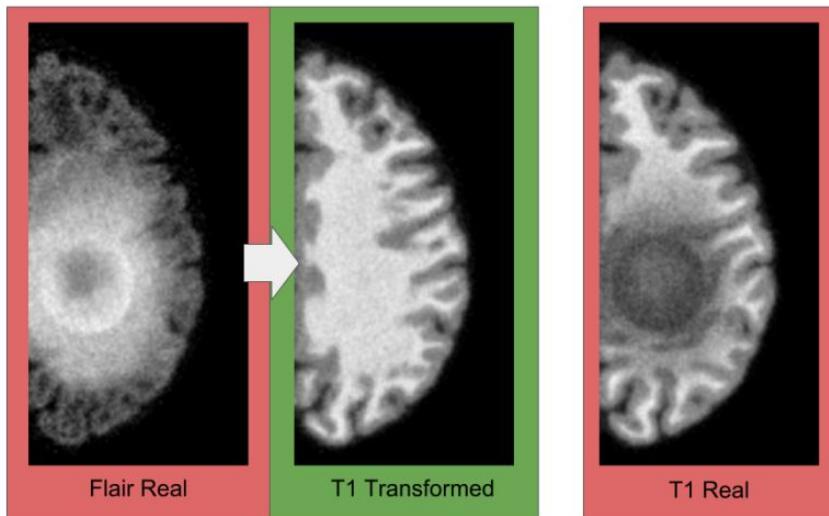


# Applications

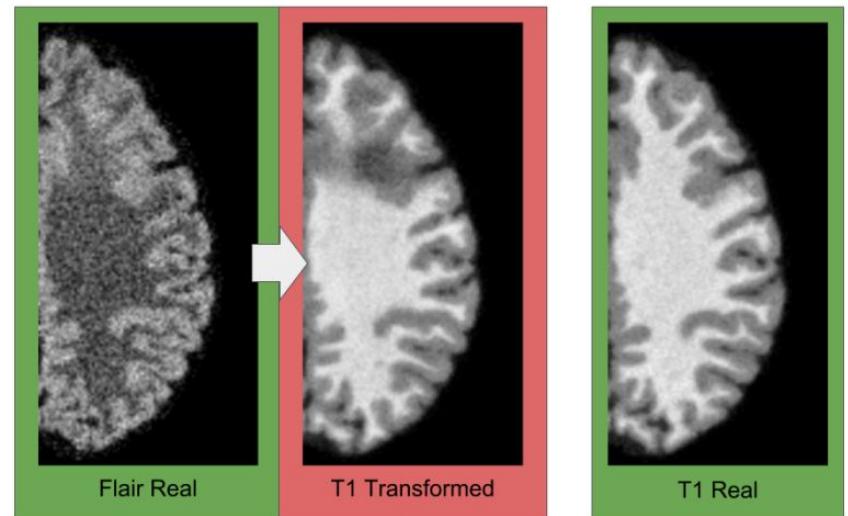


Available from: <https://arxiv.org/abs/1611.07004>

# Applications



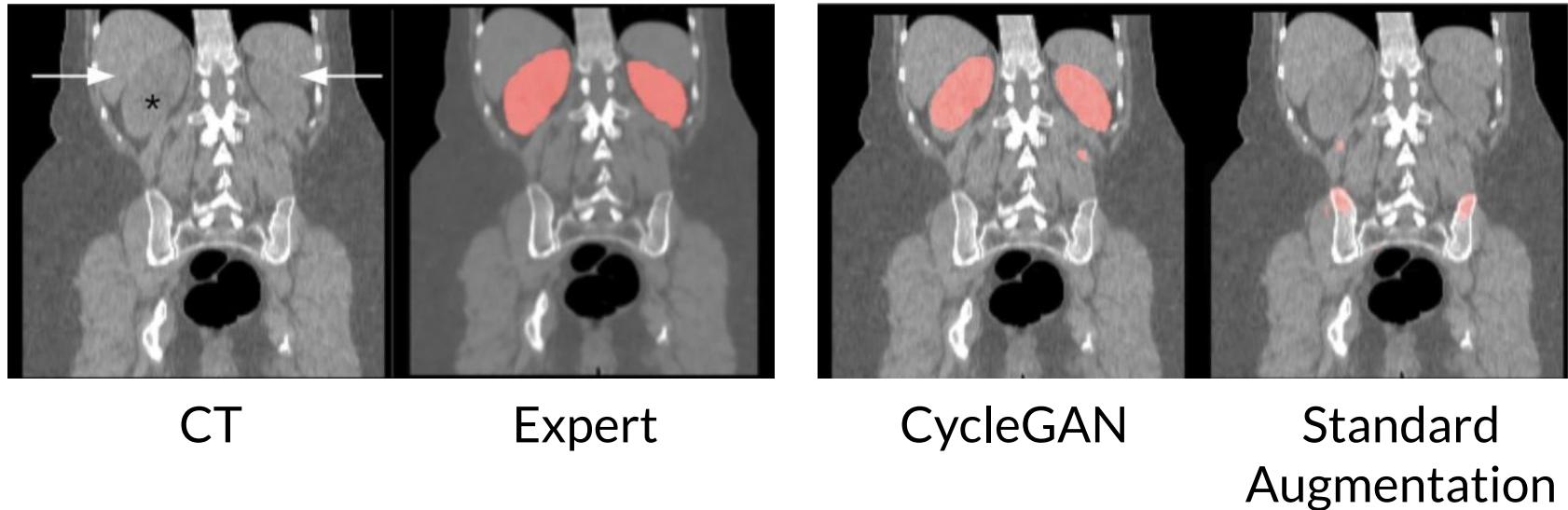
(a) A translation removing tumors



(b) A translation adding tumors

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

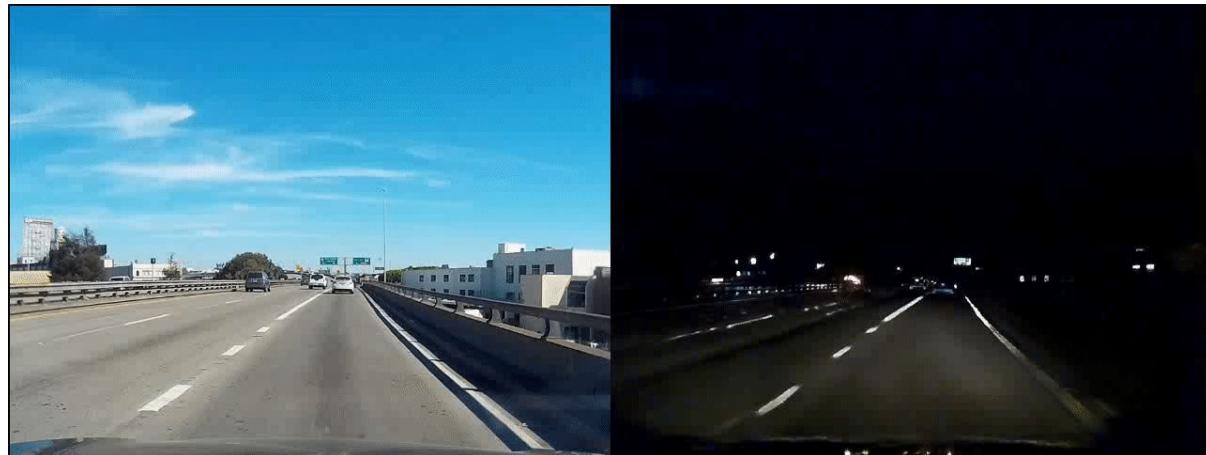
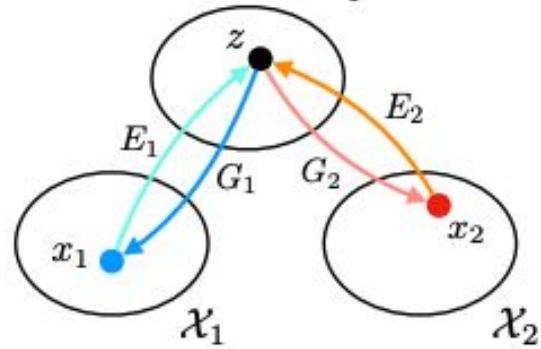
# Applications



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

# Variant: UNIT

$\mathcal{Z}$  : shared latent space



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

