Ink Wash painting --- Style Transfer

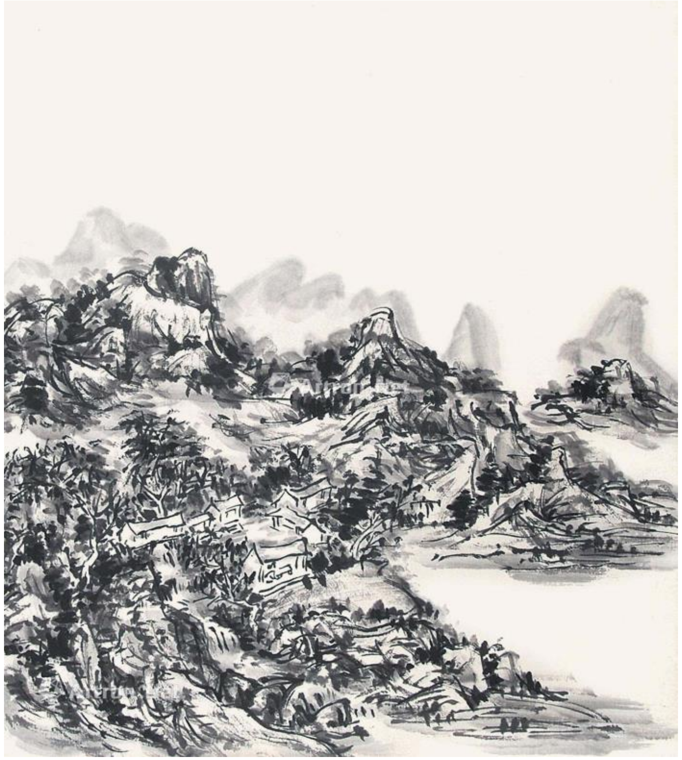
Team 4

Member1 Xingjian Qu
Member2 Zihang Zhu

Georgetown University Computer Science Department

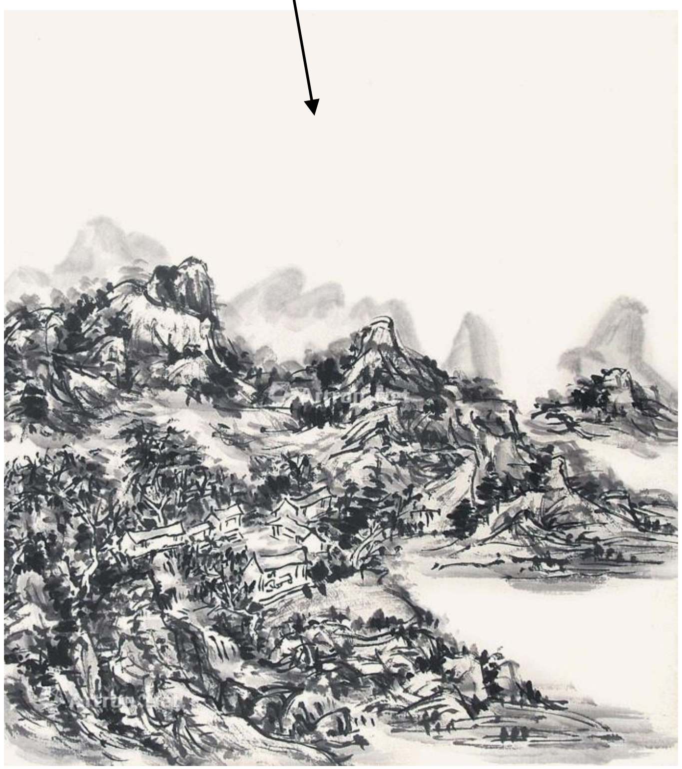
Background

What is ink wash painting?



Features

“Voids”



Color richness: ink with different gray levels.

Motivation

Generation

Inspiration

Goal

Model: CycleGan

Datasets: Horses and Landscapes

What we want to do is to transfer real-life pictures into ink wash paintings.

Horses

Landscapes

Main Reference Work(s)

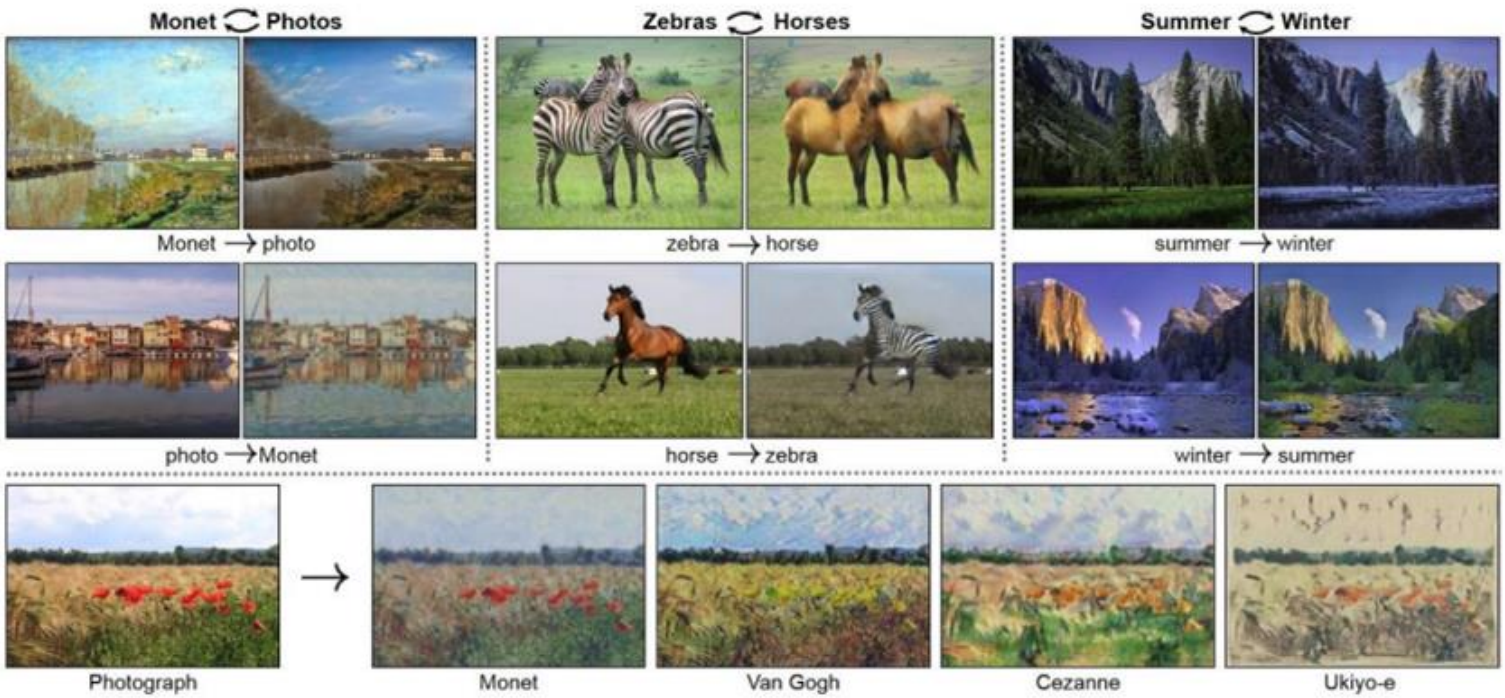
[1] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In European conference on computer vision, pages 694–711. Springer, 2016.

[2] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision, pages 2223–2232, 2017.

Method: CycleGan

CycleGAN

Style transfer problem: change the style of an image while preserving the content.



Data: Two unrelated collections of images, one for each style

CycleGAN

- If we had paired data (same content in both styles), this would be a supervised learning problem. But this is hard to find.
- The CycleGAN architecture learns to do it from unpaired data.
  - Train two different generator nets to go from style 1 to style 2, and vice versa.
  - Make sure the generated samples of style 2 are indistinguishable from real images by a discriminator net.
  - Make sure the generators are cycle-consistent: mapping from style 1 to style 2 and back again should give you almost the original image.

Architecture

CycleGAN

The discriminator tries to

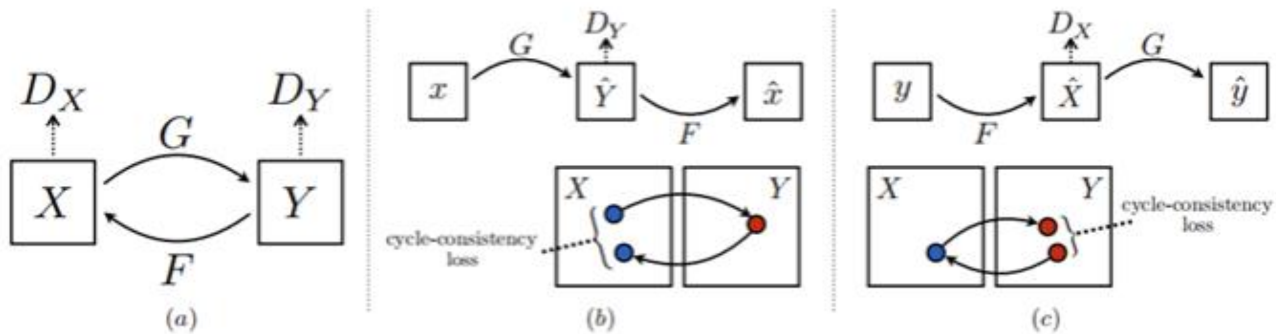


Figure 1: (a) CycleGAN contains two mapping functions G:X → Y and F:Y → X, and associated adversarial discriminators D_Y,D_X. D_Y encourages G to translate X into outputs in distinguishable from domain Y, and vice versa for D_Y and F. To further regularize the mappings, two-cycle consistency losses are introduced: (b) forward cycle-consistency loss: x → G(x) → F(G(x)) ≈ x, and (c) backward cycle-consistency loss: y →F(y) → G(F(y)) ≈ y.

Input image (real horse image)

Generator 1 learns to map from horse images to zebra images while preserving the structure

Generated sample

Generator 2 learns to map from zebra images to horse images while preserving the structure

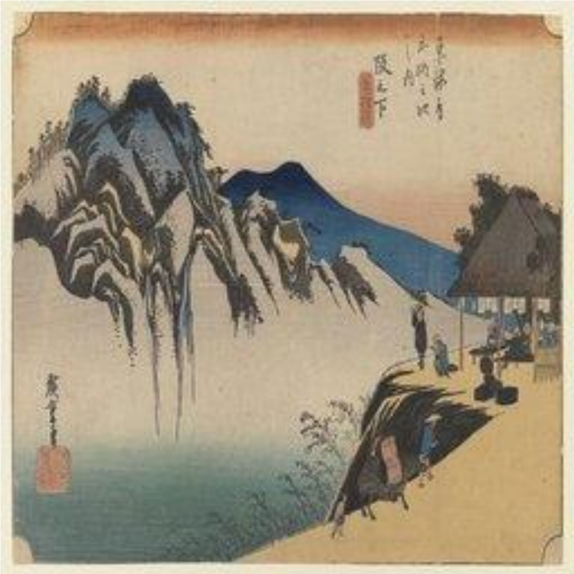
Reconstruction

Total loss = discriminator loss + reconstruction loss

Training Details

1. Use Pretrained Model: Style Ukiyoe

apple2orange.pth	2018-07-24 16:47	43M
cityscapes_label2pho..>	2018-07-24 16:48	43M
cityscapes_photo2lab..>	2018-07-24 16:47	43M
facades_label2photo.pth	2018-07-24 16:48	43M
facades_photo2label.pth	2018-07-24 16:48	43M
horse2zebra.pth	2018-07-24 16:47	43M
iphone2dslr_flower.pth	2018-07-24 16:48	43M
map2sat.pth	2018-07-24 16:47	43M
monet2photo.pth	2018-07-24 16:47	43M
orange2apple.pth	2018-07-24 16:47	43M
sat2map.pth	2018-07-24 16:47	43M
style_cezanne.pth	2018-07-24 16:47	43M
style_monet.pth	2018-07-24 16:47	43M
style_ukiyoe.pth	2018-07-24 16:47	43M
style_vangogh.pth	2018-07-24 16:47	43M
summer2winter_yosemi..>	2018-07-24 16:47	43M
winter2summer_yosemi..>	2018-07-24 16:47	43M
zebra2horse.pth	2018-07-24 16:47	43M



2. Fine-Tuning and Hyper-parameter tuning

```
# training parameters
parser.add_argument('--n_epochs', type=int, default=100, help='number of epochs with the initial learning rate')
parser.add_argument('--n_epochs_decay', type=int, default=100, help='number of epochs to linearly decay learning rate to zero')
parser.add_argument('--beta1', type=float, default=0.5, help='momentum term of adam')
parser.add_argument('--lr', type=float, default=0.0002, help='initial learning rate for adam')
parser.add_argument('--gan_mode', type=str, default='lsgan', help='the type of GAN objective. [vanilla| lsgan | wgangp]. vanilla GAN loss is the cross-entrop
parser.add_argument('--pool_size', type=int, default=50, help='the size of image buffer that stores previously generated images')
parser.add_argument('--lr_policy', type=str, default='linear', help='learning rate policy. [linear | step | plateau | cosine]')
parser.add_argument('--lr_decay_iters', type=int, default=50, help='multiply by a gamma every lr_decay_iters iterations')
```

↓

```
parser.add_argument('--n_epochs', type=int, default=130, help='number of epochs with the initial learning rate')
parser.add_argument('--n_epochs_decay', type=int, default=100, help='number of epochs to linearly decay learning rate to zero')
parser.add_argument('--beta1', type=float, default=0.5, help='momentum term of adam')
parser.add_argument('--lr', type=float, default=0.0004, help='initial learning rate for adam')
parser.add_argument('--gan_mode', type=str, default='lsgan', help='the type of GAN objective. [vanilla| lsgan | wgangp]. vanilla GAN
parser.add_argument('--pool_size', type=int, default=50, help='the size of image buffer that stores previously generated images')
parser.add_argument('--lr_policy', type=str, default='linear', help='learning rate policy. [linear | step | plateau | cosine]')
parser.add_argument('--lr_decay_iters', type=int, default=50, help='multiply by a gamma every lr_decay_iters iterations')
```

Experimental Results(What we want you to see)

Experimental Results(What we don’t want you to see)

Github Link

https://github.com/StarFarming99/COSC_5470-CV

Thank you!

QUESTIONS?