Cycle-GAN

Akash

Contents

- Intro / Basics of GANs

- Architecture of the model

- Results and loss-functions

- Limitations

- Recent developments & applications

- Exploration

Intro

- Basic principles

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

- Leading upto Cycle-GAN

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

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\mathrm{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\mathrm{GAN}}(F, D_X, Y, X) \\ + \lambda \mathcal{L}_{\mathrm{cyc}}(G, F, X) + \lambda \mathcal{L}_{\mathrm{cyc}}(F, G, Y)$$
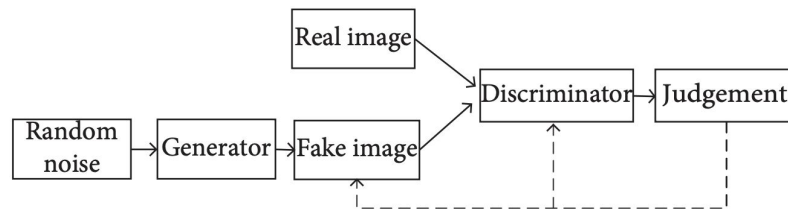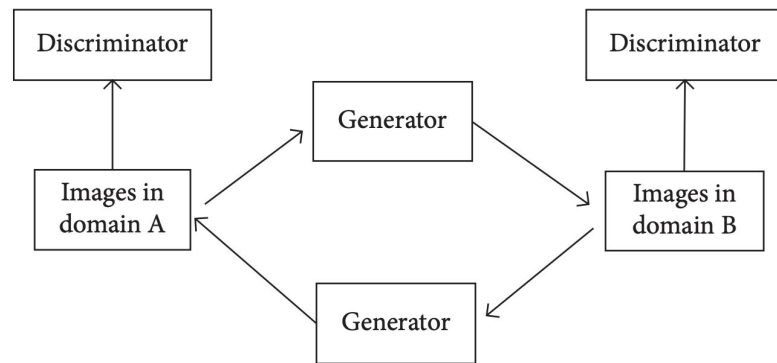
FIGURE 1: Illustration of the DCGAN model.
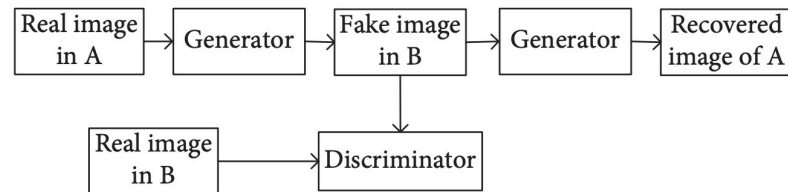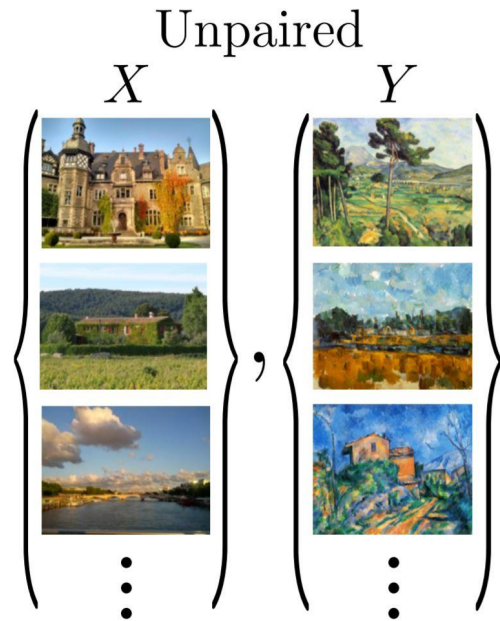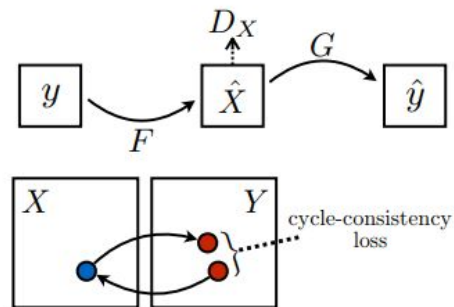
FIGURE 2: Illustration of the CycleGAN model.

FIGURE 3: Illustration of a one-way GAN model in CycleGAN.

Paired

x_i y_i

Unpaired

X Y

Model & architecture

$x \xrightarrow{G} \hat{Y} \xrightarrow{F} \hat{x}$

D_Y

$y \xrightarrow{F} \hat{X} \xrightarrow{G} \hat{y}$

D_X

cycle-consistency loss

X Y

cycle-consistency loss

X Y
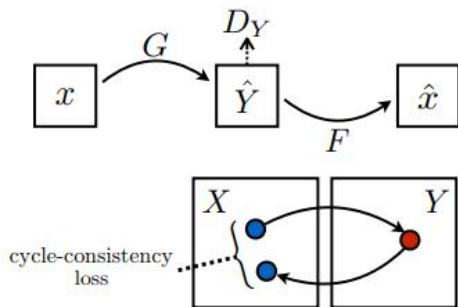
Discriminator

Generator

Network

x   G(x)   F(G(x))   y   F(y)   G(F(x))

x $\hat{Y}$ $\hat{x}$ y $\hat{X}$ $\hat{y}$

G F F G

$D_Y(G(x))$ $D_G(F(x))$
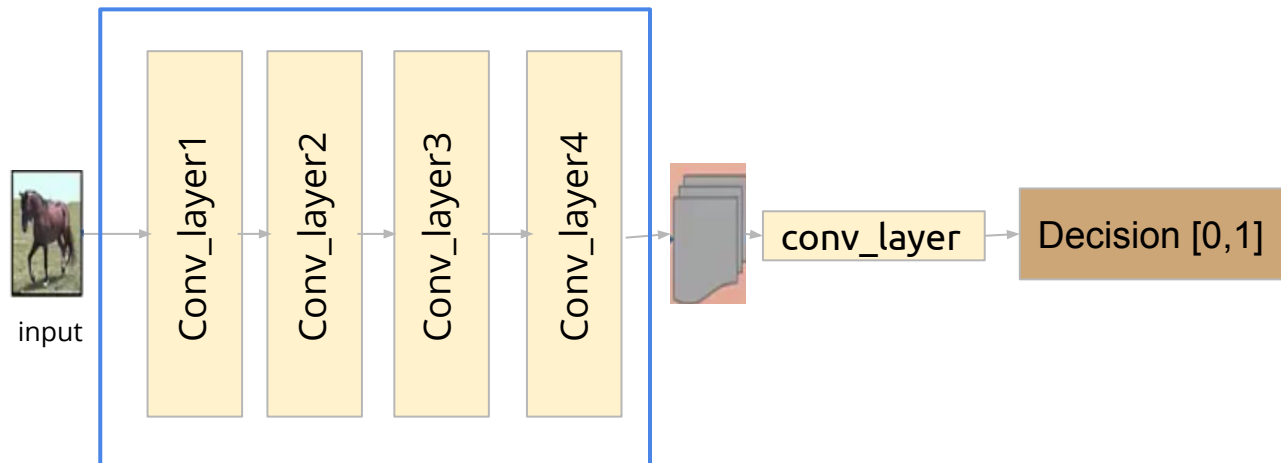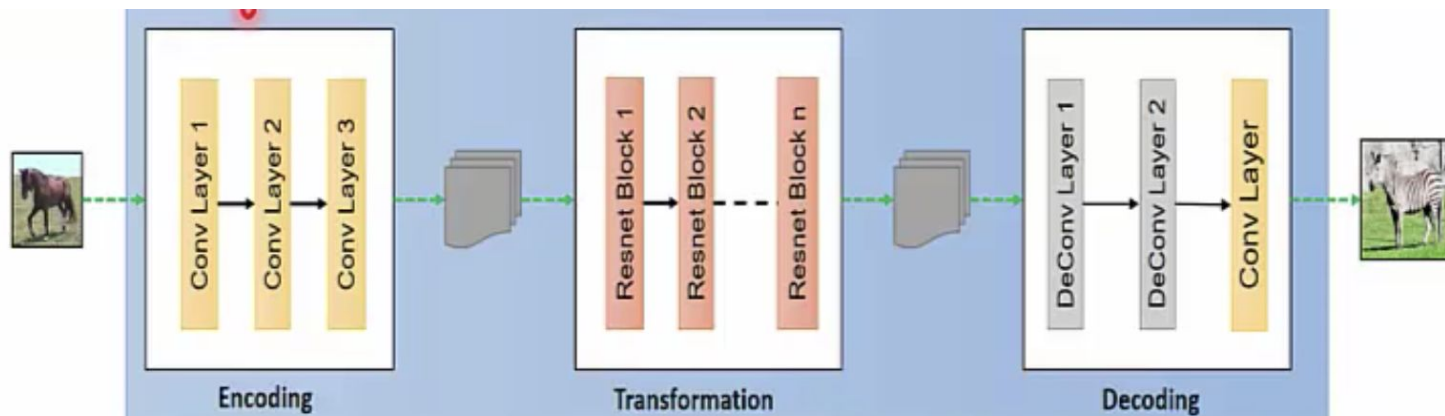
Generator A2B

Generator A2B

Photo Van Gogh Monet Ukiyo-e

Paintings to pictures / vice-versa (so far ..)

Reconstruction

Identity loss

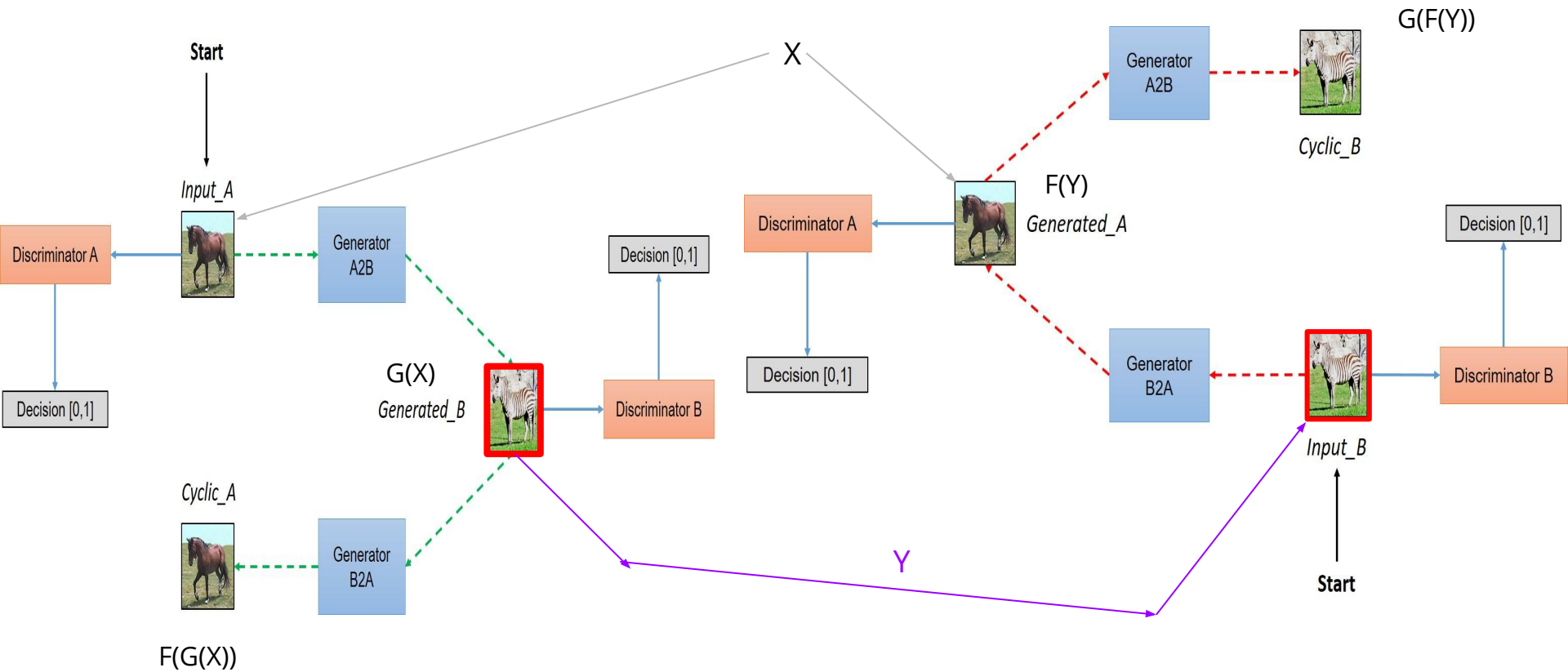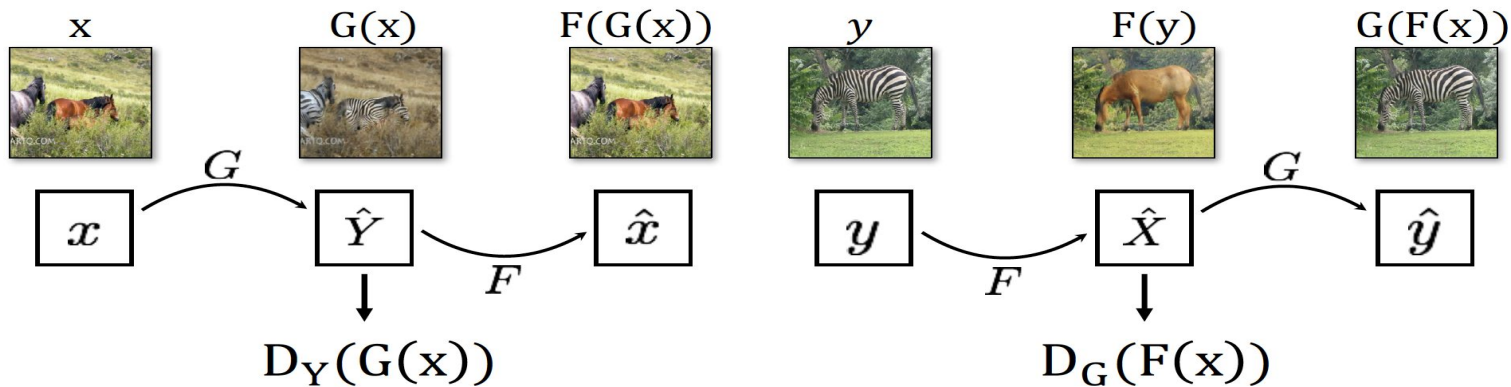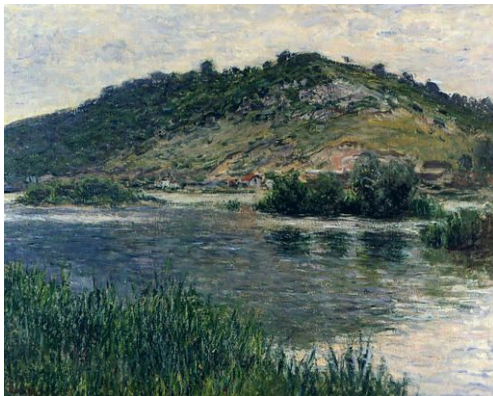$$\mathcal{L}_{\text{identity}}(G, F) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(y) - y\|_1] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(x) - x\|_1]$$
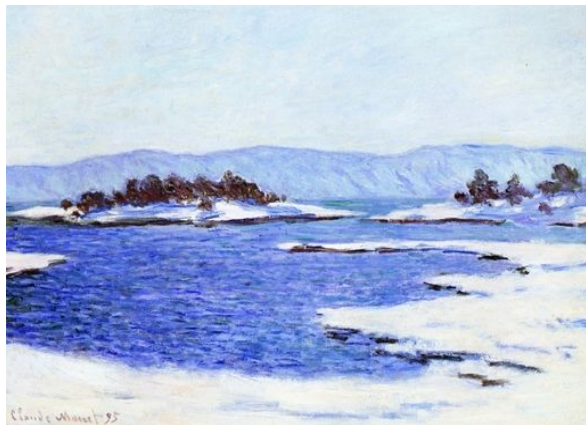

Input

with

without

Implementation

- Proposal / improvements made:
1. Weight the cycle consistency by quality of image formed
2. Cycle consistency on discriminator CNN feature level, can be done based on recent arxiv paper.

Task	dataset	metric	value
Img-to-img translation	*Cityscape*	Per-pixel accuracy	0.568
Img-to-img translation	*Cityscape*	Class IoU score	0.11
Multi-modal unsuperv. Img-to-img translation	*Cats-to-dogs*	CIS	0.075
Img-to-img translation	*monet-photo*	Quality	0.408
Img-to-img translation	*horse-zebra*	Per-pixel accuracy /ablation study score	0.52
Img-to-img translation	*horse-zebra*	Per-class accuracy	0.17

Current Status: (AMI)
- Runs of horse2zeb, monet and seasonal data have been done.
- Issues with latency in the nvidia-docker image setup, resolving the same with prebuilt image training, but improvements have been done in AWS.

Improvement results:
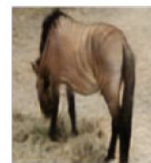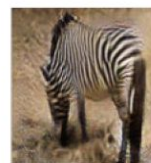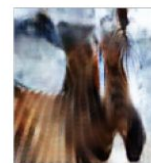
The proposed architecture is currently being trained, with the above image sheet illustrating the changes achieved in results.

Limitations & failures

Input   Output   Input   Output   Input   Output

apple → orange   zebra → horse   winter → summer

dog → cat   cat → dog   Monet → photo

photo → Ukiyo-e   photo → Van Gogh   iPhone photo → DSLR photo

I_{MR}

Dis_{CT}

$\|Syn_{MR}(Syn_{CT}(I_{MR}))-I_{MR}\|_1$

Syn_{CT}

$Syn_{CT}(I_{MR})$

$Syn_{MR}(Syn_{CT}(I_{MR}))$

Syn_{MR}

Recent developments / literature

Cycle consistency loss

Unlabeled image
$\mathcal{X}_\mathcal{U}$

Image to labels generator

G_{IS}

Generated labels
$G_{IS}(\mathcal{X}_\mathcal{U})$

Labels to image generator

G_{SI}

Reconstructed image
$G_{SI}(G_{IS}(\mathcal{X}_\mathcal{U}))$

Ground truth labels
$\mathcal{Y}_\mathcal{L}$

Labels to image generator

G_{SI}

Generated image
$G_{SI}(\mathcal{Y}_\mathcal{L})$

Image to labels generator

G_{IS}

Reconstructed labels
$G_{IS}(G_{SI}(\mathcal{Y}_\mathcal{L}))$
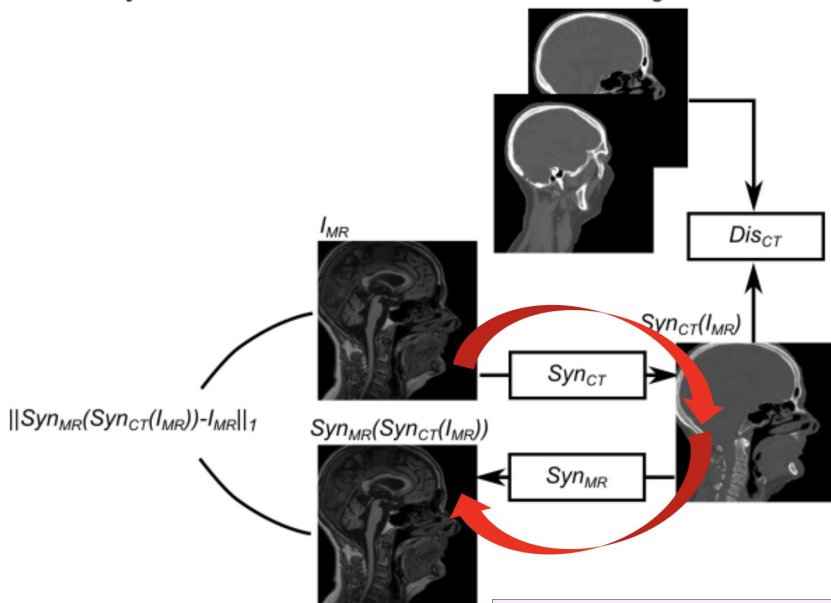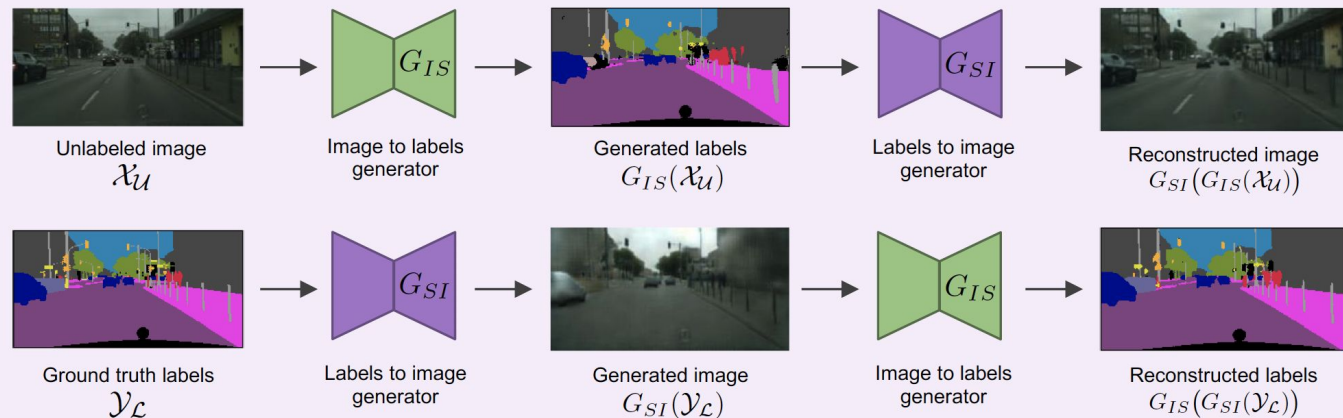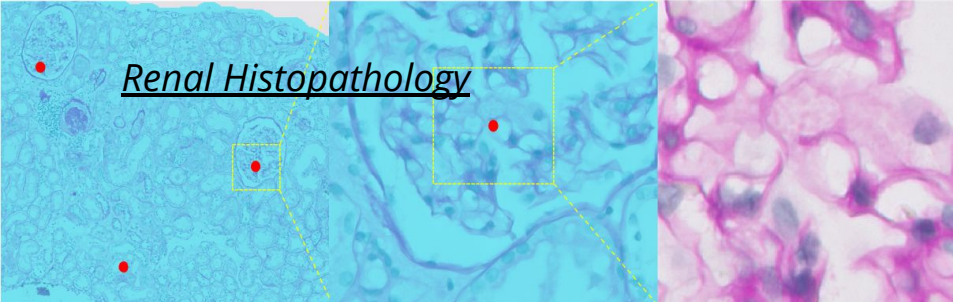
Renal Histopathology

_VC2_

(a) 1D CNN

T ... Q C C T Q
Downsample ($1/r$)   ResBlocks   Upsample ($\times r$)

(b) 2D CNN

Downsample ($1/r$)   ResBlocks   Upsample ($\times r$)

(c) 2-1-2D CNN (proposed)

Reshape   1x1Conv   ResBlocks   1x1Conv   Reshape
Downsample ($1/r$)   Upsample ($\times r$)

- Driver gaze estimations
- Natural speech perturbations (Automatic Speech Recognition)
- etc.

Cryptography

Phase 1: CycleGAN training

Generator G_1 D_1 Restorer F D_2
X → Y → X
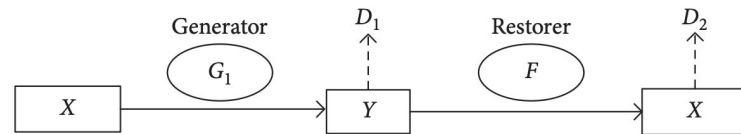
Phase 2: generator training

Generator G_2 D_3
Random noise Z → X
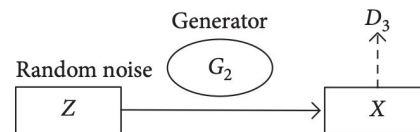
Phase 3: extractor training

Extractor E
Input Z_1 → G_2 → X → G_1 → Y → Output Z_2

Phase 4: send and receive

M Secret data
Map ↓
Z → G_2, G_1 → Marked image $G_1(G_2(Z))$ → Send

E → Z → Map → M
F → Recovered image

Sender   Receiver

Preparation process

Implementation process

Applications

Real Human Demos   Generated Robot Translations

-   Object transfiguration
-   Seasonal transfer
-   Photo generation from paintings
-   MR images to CT scans
-   Attribute editing (photos / facial features)
-   CG2real - video game to live and vice-versa
-   Day-to-night driving video rendering (domain adaptation)
-   Image dehazing
-   Maps to satellite imagery
-   Cryptography, Robotics, NLP - text style transfer etc.
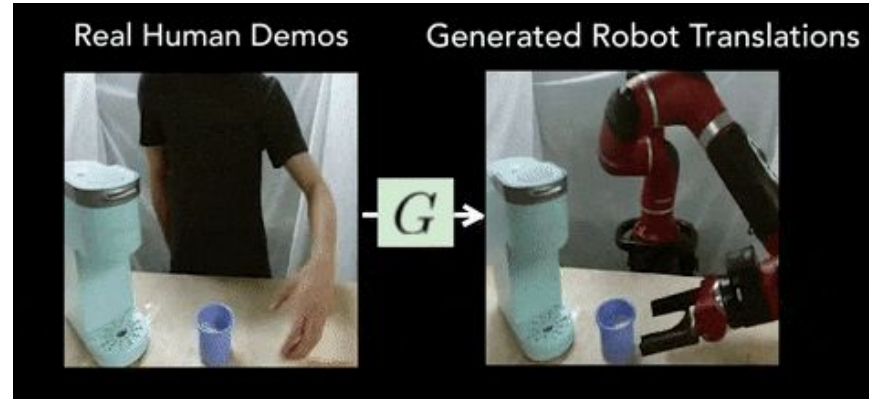
Explore

- taesung.me/cyclegan/2017/03/25/maps-comparison.html
- BicycleGAN: cVAE-GAN + cLR-GAN
- github.com/yunjey/mnist-svhn-transfer
- Tensorflow & colab notebooks.
- hardikbansal.github.io/CycleGANBlog
- Another approach: github.com/PramuPerera/In2I
- web.stanford.edu/~jaustinb/papers/CS236.pdf
- downloads.hindawi.com/journals/scn/2019/4932782.pdf
- arxiv.org/pdf/1712.02950.pdf (cycle-gan : master of stenography)

References / links

- GANs (Ian Goodfellow): youtube.com/watch?v=RvgYvHyT15E
- MILA blog: ishmaelbelghazi.github.io/ALI
- paperswithcode.com/paper/unpaired-image-to-image-translation-using
- Pen & paper explanation of cycle: youtu.be/T-lBMrjZ3_0
- blog.aylien.com/introduction-generative-adversarial-networks-code-tensorflow/
- BAIR blogs (bair.berkeley.edu/blog/2019/12/13/humans-cyclegan)
- TF tutorial: machinelearningmastery.com/cyclegan-tutorial-with-keras/