CV Final Project Report

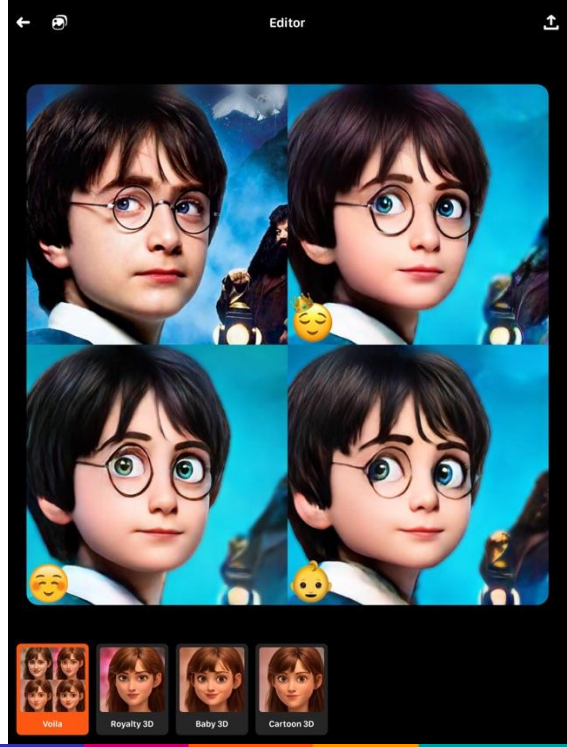
Group 5
309552012 洪立宇 309551048 陳司瑋 309551101 郭育麟

Outline

- Introduction
- Dataset
- Implementation
- Result
- Evaluation
- Discussion
- Conclusion

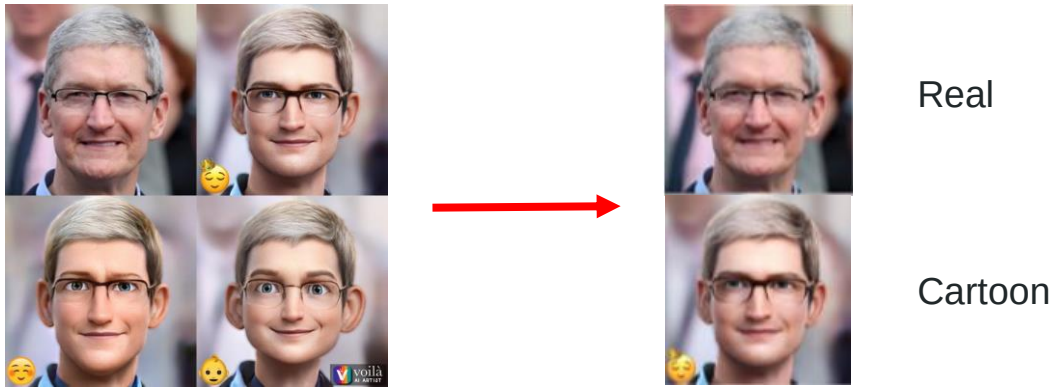
Introduction

- Voila AI Artist
- Convert human into Disney-like cartoon
- Available for animated 2D and 3D conversion
- Goal: Reconvert the Disney-like cartoon back to human



Dataset

- Colleted the dataset on our own by using the API InstaLooter
- Crawled 20000 photos with hashtag voila
- Image segmentation and manually selected the photos we want
- Finally we gethered a dataset with 10509 pairs of images



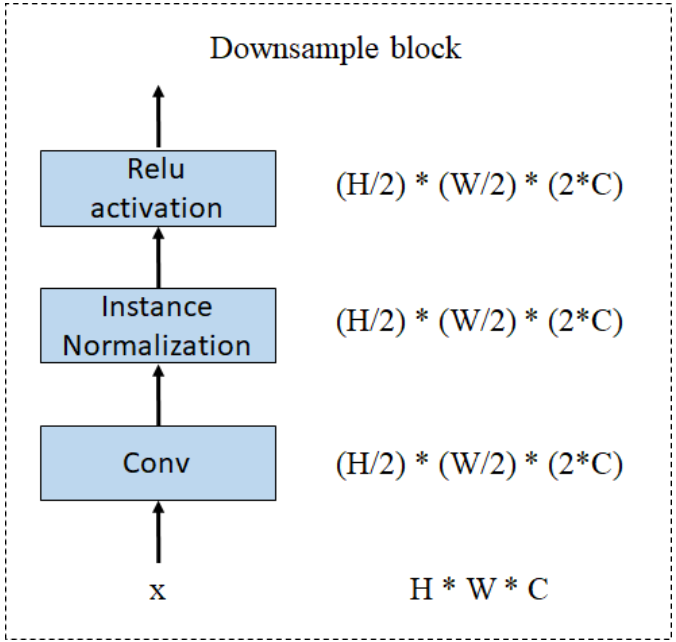
Img.1

Outline

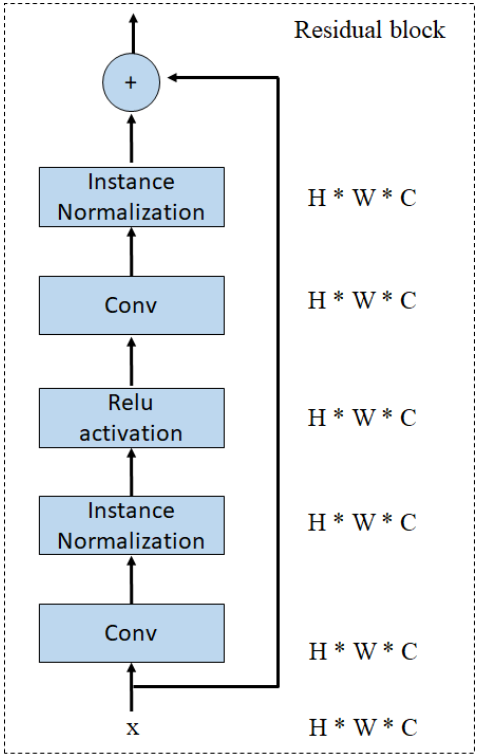
- Introduction
- Dataset
- Implementation
  - Model Architecture
  - Procedure
  - Loss function
  - Experiment settings
- Result
- Evaluation
- Conclusion

Model Architecture

- Downsample and Upsample block:

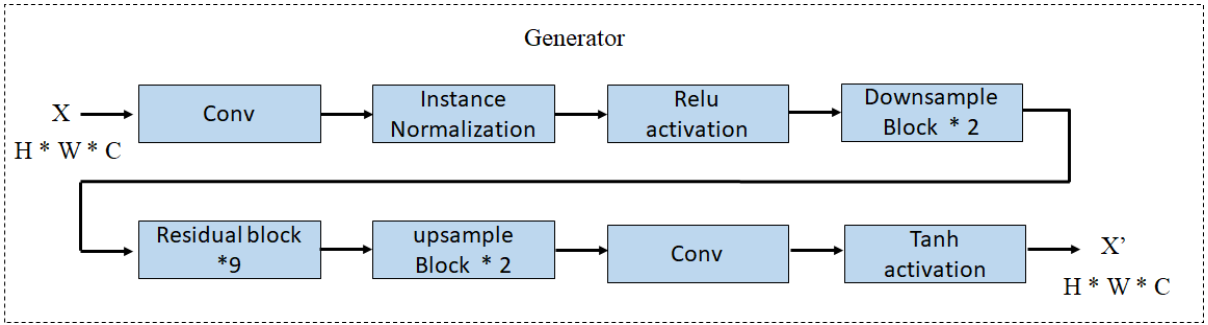


- Residual block :

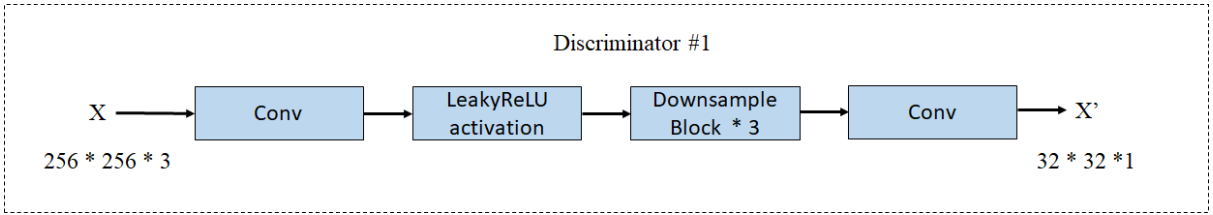


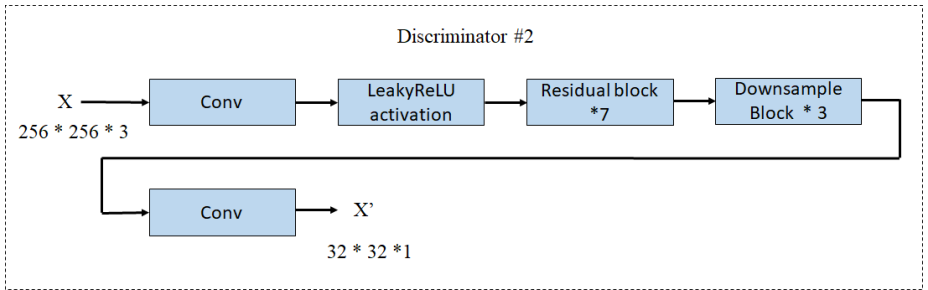
Model Architecture

- Generator



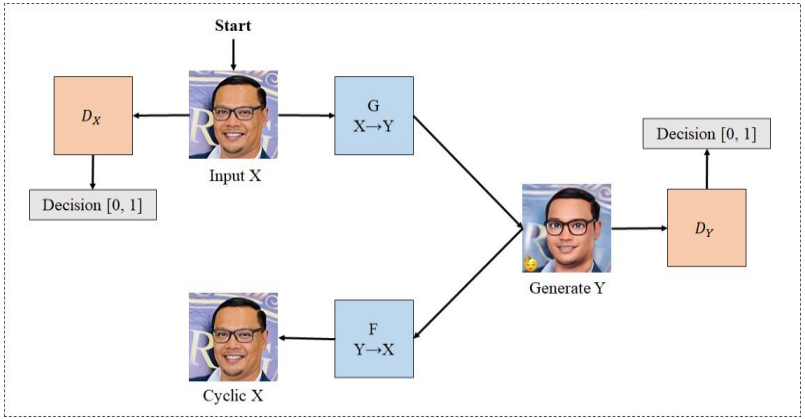
- Discriminator:

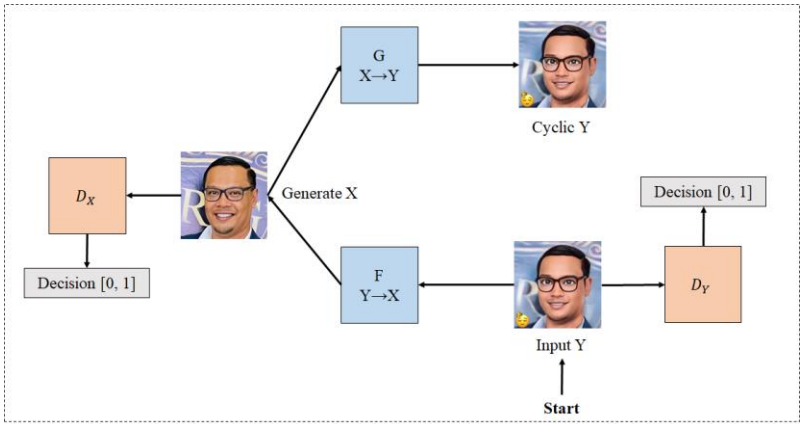




Procedure

- GAN
- Cycle - GAN

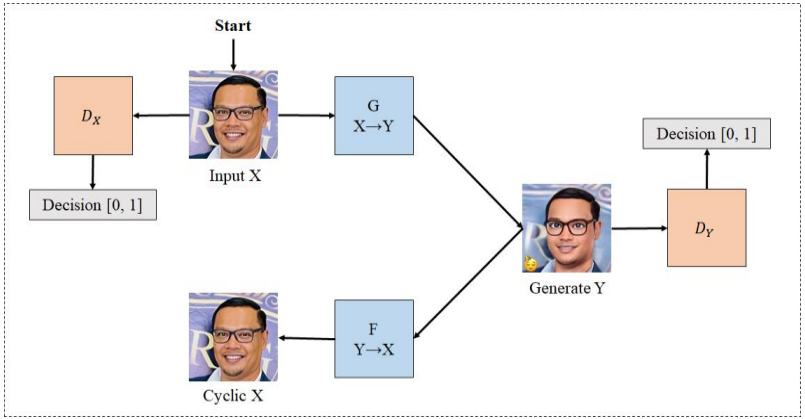


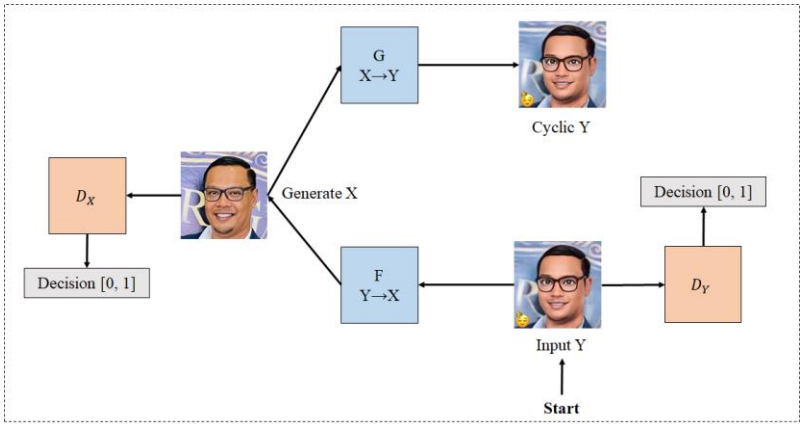


Loss function

- Base GAN loss: $L_{GAN}(G, D_Y, X, Y) = E_{y \sim p_{data}(y)}[log\ D_Y(y)] + E_{x \sim p_{data}(x)}[log(1 - D_Y(G(x))]$
- Base Cycle loss: $L_{cyc}(G, F, X) = E_{x \sim p_{data}(x)}[\|F(G(x)) - x\|_1]$
- Base identity loss: $L_{identity}(G, Y) = E_{y \sim p_{data(Y)}}[G(y) - y]_2$

$$L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda L_{cyc}(G, F, X) + \lambda L_{cyc}(F, G, Y) + L_{identity}(G, Y) + L_{identity}(F, X)$$





Loss function

- Generator loss :

$$L_{l_2}(G,\ x,\ x_g)\ =\ ||G(x)\ -\ x_g||_2$$
$$L_{adv}(G,\ x,\ x_g)\ =\ max_D E_{x \in X}|D(x_g)\ -\ D(G(x))|$$
$$L_{generator}(G,\ x,\ x_g)\ =\ \lambda L_{l_2}(G,\ x,\ x_g)\ +\ (1-\lambda)L_{adv}(G,\ x,\ x_g)$$

x

x_g

- Cycle consistency loss :

$$L_{cyc}(G,\ F,\ D_X,\ X,\ \gamma)\ =\ E_{x\ \sim p_{data}(x)}[\gamma||f_{D_X}(F(G(x))-f_{D_X}(x)||_1\ +\ (1-\gamma)||F(G(x))-x||_1]$$

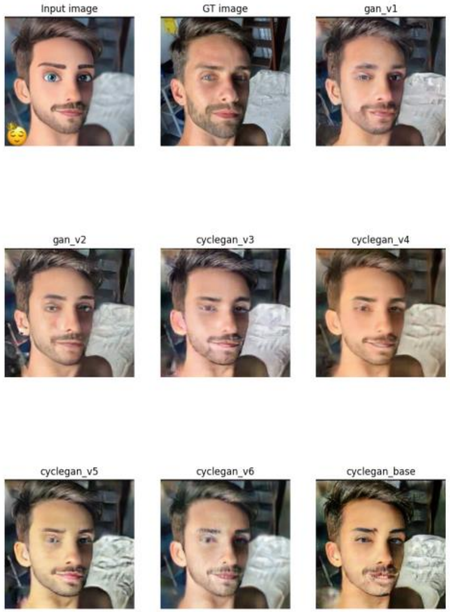
Experimental settings

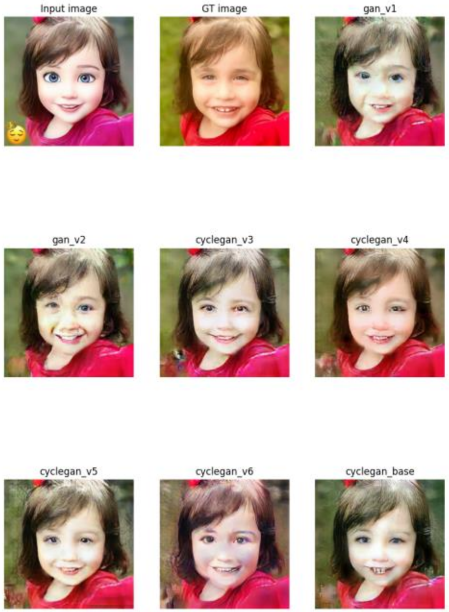
Model name	Modified
Gan v1	None
Gan v2	generator loss
Base cycle gan	None
Cycle gan v3	generator loss
Cycle gan v4	generator loss+Cycle consistency loss
Cycle gan v5	generator loss+Cycle consistency loss + Soft Label
Cycle gan v6	generator loss+Cycle consistency loss+Soft Label +discriminator network

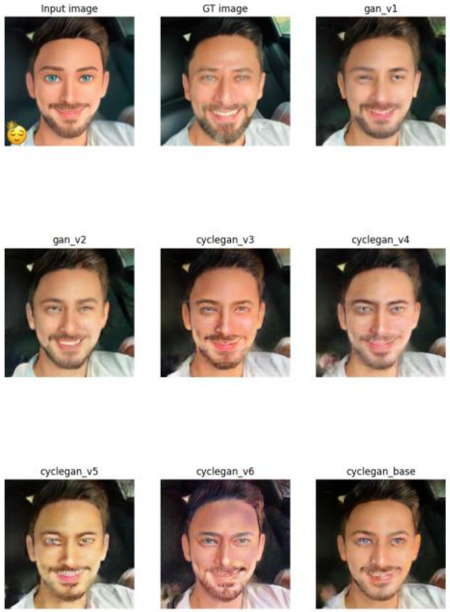
Outline

- Introduction
- Dataset
- Implementation
- Result
- Evaluation
- Conclusion

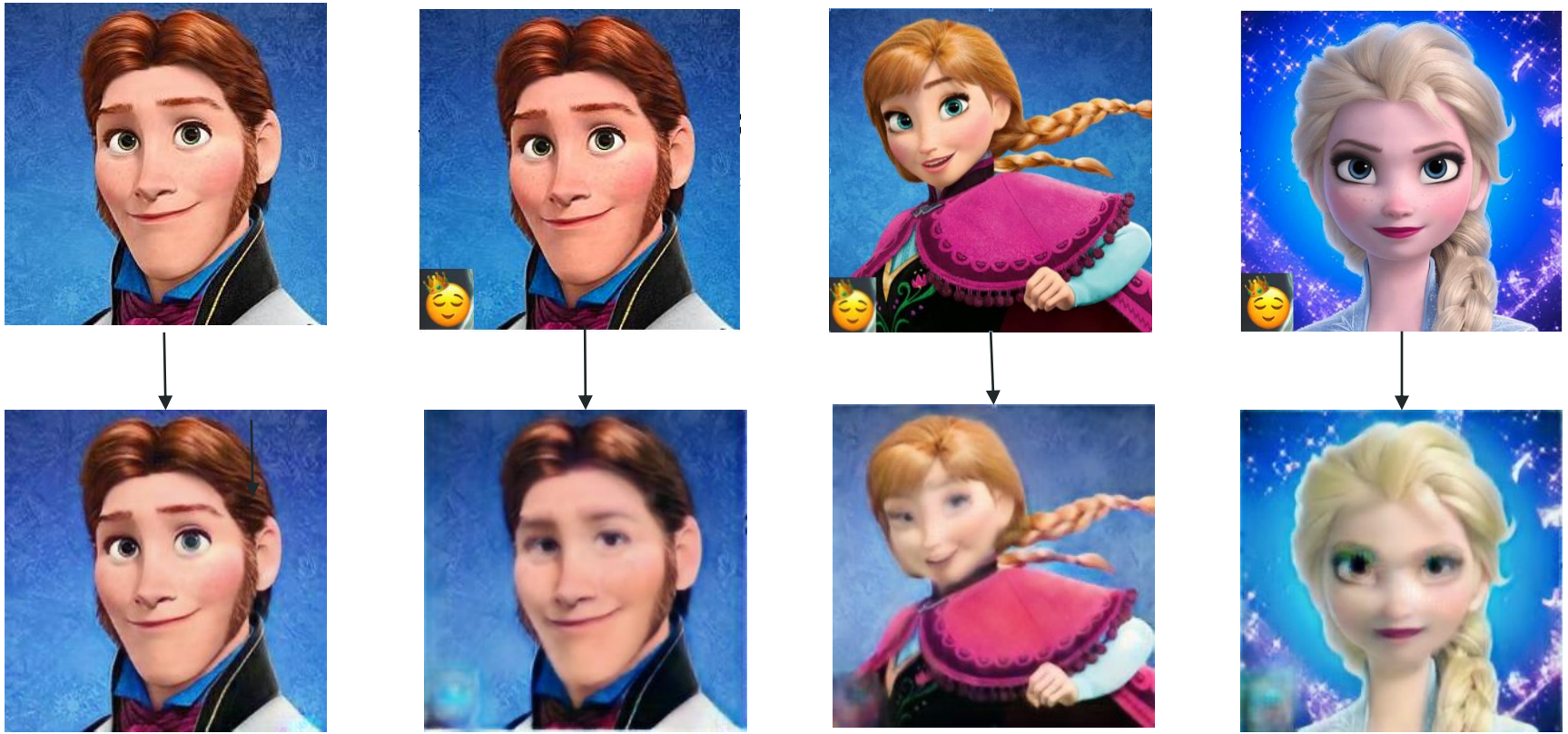
Result


Input image
GT image
gan_v1
gan_v2
cyclegan_v3
cyclegan_v4
cyclegan_v5
cyclegan_v6
cyclegan_base


Input image
GT image
gan_v1
gan_v2
cyclegan_v3
cyclegan_v4
cyclegan_v5
cyclegan_v6
cyclegan_base


Input image
GT image
gan_v1
gan_v2
cyclegan_v3
cyclegan_v4
cyclegan_v5
cyclegan_v6
cyclegan_base

Result



Outline

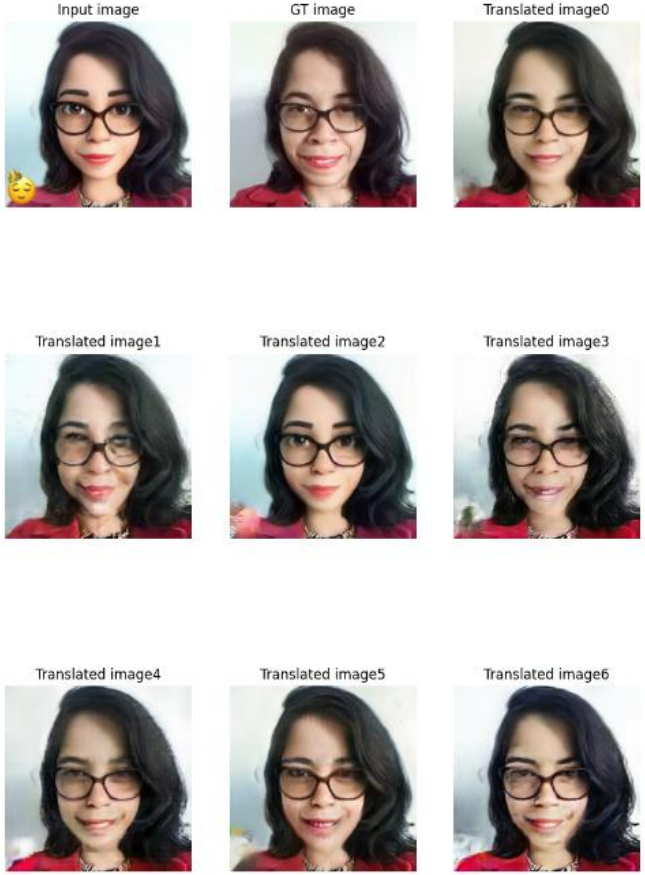
- Introduction
- Dataset
- Implementation
- Result
- Evaluation
- Conclusion

Manual evaluation

Number of reviewer: 3 people

Photo : 50 images

Number of different Model : 7



GAN v1	GAN v2	Base Cycle GAN	Cycle GAN v3	Cycle Gan v4	Cycle Gan v5	Cycle Gan v6
18	25	11	41	26	6	23

Kernel MMD

- A measure of dissimilarity between P_r and P_g for some fixed kernel function k. Given two sets of samples from P_r and P_g
- We use the euclidean distance calculating the images differences
- Kernel function: y= (x / (x_r,x_r').mean * 2 * beta * beta)

$$MMD^2(P_r, P_g) = E_{x_r,\, x'_r \sim P_r,\, x_g,\, x'_g \sim P_g}[k(x_r, x_r') - 2k(x_r, x_g) + k(x_g, x'_g)]$$

1-NN

- Given two sets of samples $S_r \sim P_r^n$ and $S_g \sim P_g^m$, with $|S_r| = |S_g|$, one can compute the leave-one-out accuracy of a 1-NN classifier trained on S_r and S_g with positive labels for S_r and negative labels for S_g
- 50% leave-one-out accuracy when $|S_r| = |S_g|$ is large, and 1NN classifier can't seperate 2 set well which is a good result!!
- ~0% leave-one-out accuracy means the two sets of images are close but in opposite labels
- ~100% leave-one-out accuracy, it means GAN generates a complete different set of pictures and is able to separate 2 sets easily

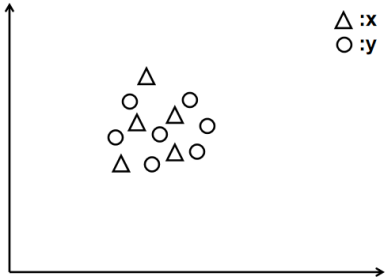


Figure 1

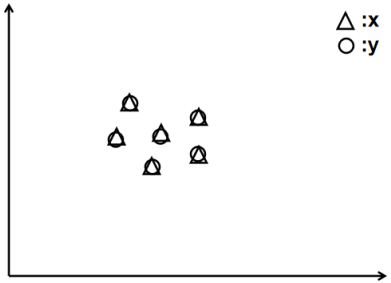


Figure 2

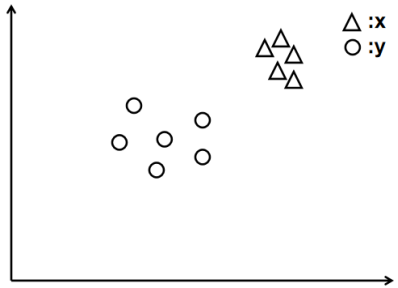


Figure 3

Table

Model name	mmd ↓ (epoch)	knn (epoch)
Gan v1	0.05003(161)	0.86329 (162)
Gan v2	0.05139 (134)	0.87341 (199)
Base cycle gan	0.05865 (124)	**0.83544** (126)
Cycle gan v3	**0.04959** (141)	0.87088 (149)
Cycle gan v4	0.05114 (159)	0.87848 (89)
Cycle gan v5	0.05170 (85)	0.88354 (102)
Cycle gan v6	0.05144 (54)	0.87594 (39)

Discussion

- We modify the loss function and architecture. However, we do not see a lot of difference from the results. The reason why v5 and v6 do not get better results might be due to the fact that we don’t have enough time to train more epochs.
- We found that a man with a moustache can be converted well often while children and infants are usually converted awful.
- It is easy to see that the losses of the discriminators decrease continuously. Both losses of the discriminators and generators change a lot during training, and the generator seems hard to converge.

Conclusion

- MMD value is more reliable since it becomes lower when we train the model longer, which can be confirmed by the table of the different epochs of v3 model and v3 perform also the best for manual evaluation.
- The 1NN index is not that reliable in our case. From the table of the different epochs of v3 model, the 1NN results are always high and hardly become lower.
- Emoji does matter!! Maybe we should get a new dataset or it will be an feature learned in the GAN.

Thanks for your Attention