Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Tadem Sai Pavan

SMST,Indian Institute Of Technology ,Kharagpur,India.

Abstract

This is an Implementation project from the original paper "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks".Before cycleGAN,the image to image translation is between only paired image datasets with the help of pix2pix model.In the original paper researcher came with a new mode called cycleGAN.It can do image to image translation using unpaired dataset.Even though the results of pix2pix model are good ,for many tasks the paired dataset is not available ,this is the major limitation of it.So cycleGAN can overcomes this limitation by translating image to image in the absence of paired data.cycleGAN converts images in domain-X to domain-Y and then back to domain-X.CycleGAN is tested on three different datasets and architecture,results observations are briefly discussed in this report.

1 Introduction

Image translation has become one of the most important areas to focus on.The technology behind this concept is GANs[1].GAN contains two neural networks ,both compete on each other(adversarial) to produce more accurate predictions. cycleGAN [2]consists of two GANs i.e.two two generators and two discriminators consider an example of selfie two anime translation ,one generator transforms. selfies to anime and the other transform anime image back to selfie .Discriminators check the images generated by generators are fake or real during training.This loop ensures that an image created by generator is cycle consistent, it means consecutively both generators on an image should yield a similar image.The trouble shoot with the pix2pix [3] is data set ,because it is a supervised model,It is always hard to find a paired data sets,here cycleGAN can solve that issue ,it does not require supervised training ,i.e It can perform Image to Image translation without a paired data set .

In case of image translation (supervised) there is nothing to worry about what kind of output to be generated ,but in case of unsupervised (unpaired data) it is more important to focus on the task for example orange to tomato translation ,there can be two possibilities orange is completely translated to tomato or it can be a just colour change .In unsupervised learning data sets plays a very important role to generate the mapping function .The special thing about GANs is,they can create new objects and by taking a random input.

2 Related Work

Image-to-Image Translation : The concept behind image to image translation is not only learn from the input images but also learns the loss to train the mapping .CycleGAN learns the mapping from original image to generated image, along with that it,learns a loss function to train this input and output mapping.

Generative adversarial Networks: GANs are proved to be expert at translating image to image.The same idea can be applied to video ,text translations.The reason behind the success of GANs is adversarial loss. This loss forces the created images to be indistinguishable from the input image.The adversarial loss learn mapping function such a way that the generated images can not be distinguished from the target.GANs takes random input from Gaussian noise and generates meaningful outputs.GANs are generator and discriminator which are implemented by using convolutional neural networks to perform feature extraction and mapping.Both generator and discriminator are learned through back propagation,So generator learn to produces better desired results and discriminator learn to not to become a fool.

Unpaired Image-to-Image Translation: Initially the image to image translation is supervised i.e in presence of paired data set ,but here they proposed an unsupervised model which is basically image to image translation from unpaired images.This model has wider range of applications due to being unsupervised,for example in health care,the medical data is dry and expensive for paired data.

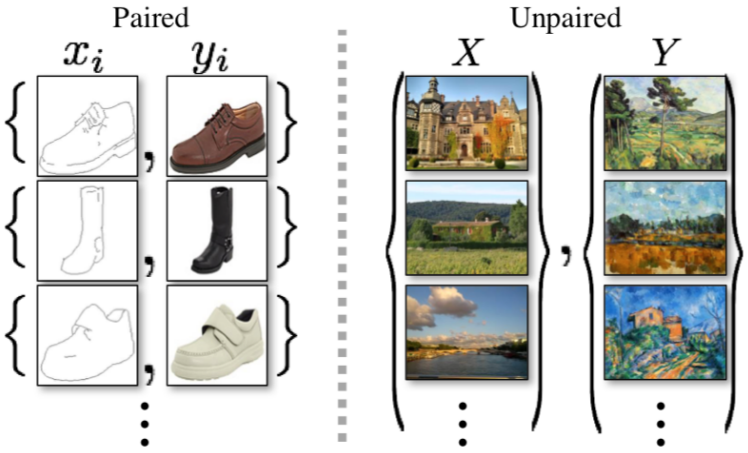


Figure 1: paired and unpaired images

Neural style transfer: It is one among the methods to convert Image to Image translation,by this method we can generate style of domain x in domain y.

Cycle consistency loss: In cycleGAN generator-G converts the original image i.e domain X to domain Y,and the generator-F tries to reconstruct that image from domain Y back to domain X,during this conversion it is important to keep follow up the loss every time to improve the reconstruction ,that responsibility is taken by cycle consistency loss.cycle consistency loss is added with the adversarial loss and back propagated,so that model become more clever at performing task.

3 Network Architecture

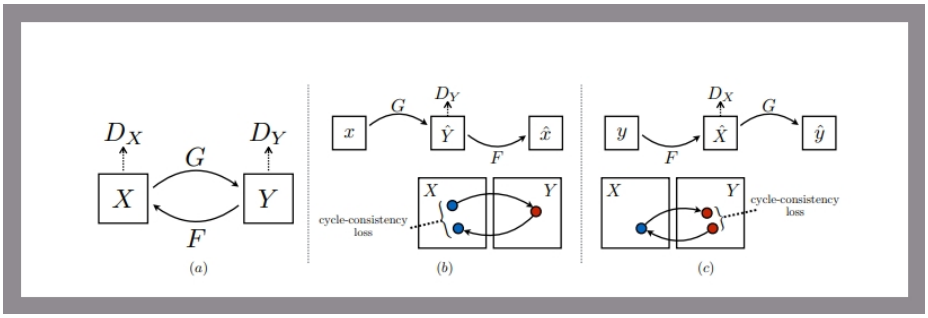


Figure 2: CycleGAN

3.1 Model

In CycleGAN there are two discriminators and two Generators.The model for the generative networks is taken from the Johnson[4].The results of that model are impressive for style transfer and super resolution.This architecture contains convolutional layers(three) , residual blocks(six), transpose convolutional layers(two) with stride of one and one convolutional layer which is going to map the features into RGB plane.The researcher's made slight modifications to the above mentioned network and designed the generators G and F,both are inverse of each other and performs one to one map operation.The aim is to creating a mapping between domain X to domain Y and then back to domain X,consider xi belongs to X and yi belongs to Y,The mathematical view of the model is $G : X \rightarrow Y$ and another translator $F : Y \rightarrow X$,So G and F are inverse of each other, and both the mappings are one to one .

3.2 Generator

Generator is a combination of encoder,decoder and residual layers. Generator is responsible of creating a fake image from the input image. Generator is sub sectioned into three parts,encoder block,residual block and decoder block..

3.2.1 Encoder

Encoder plays a key role in feature extraction from the original image. Encoder is designed with the help of convolutional layers.There are three convolutional layers in encoder,initial layer accepts input dimension of 3 and at output of third layer we have 256.Each convolutional layer is followed by instance normalization with the batch size of one.

IN-instance normalization,Conv-convolutional Layers.

Encoder **:** $Conv - 64 - IN, Conv - 128 - IN, , Conv - 3 - IN$

3.2.2 Residual Block

In generator encoder and decoder both are linked to gather by using residual blocks,deep learning models are having gradient vanishing issue,It is difficult to train them perfectly,So the results are not desired.But it can be solved by using residual blocks,which are going to learn the residual functions. There are six resnet blocks each block contains two convolutional layers.

The overall architecture is as follows for generator, here Res stands for a block Conv-IN.The output of encoder is given to residual layers.

Residualblocks **:** $Res-256, Res-256, Res-256, Res-256, Res-256, Res-256$

3.2.3 Decoder

The output of residual layers is taken by the decoders.Decoder is basically a up-sampler.It reconstructs the all the features into an image.This is constructed by two transpose convolutional layers takes 256 as input dimension and reduces to 32 as original.last layer is a convolutional layer takes 32 dimensions and converts into three dimensional image i.e RGB and this layer is followed by a tanh activation,here transpose convolutional layer is represented as TransposeConv.

Decoder **:** $TransposeConv-128-IN, TransposeConv-64-IN, Conv-3-Tanh$

3.3 Discriminator

Discriminator is full of convolutional Layers(5 layers) which are used to classify image patches of seventy by seventy sized are fake or real.The role of patchGANs is to double the number of channels and halves the size.This is repeated up to the point output converges to desired state. For discriminator the filter size is 3∗3 ,It takes 3 dimensional input image and converts into 32,64,128,256 stage wise and back to dimension 3.In discriminator ReLU are Leaky with α= 0.2.

Discriminator **:** $Conv-32, Conv-64, Conv-128, conv-256, conv-3$

4 Training Details:

Adam[5] is used as optimizer,it is one of the best performer out there,to decay the runtime average of gradients beta-1 is taken as 0.5 and to decay the square of the gradient beta-2is taken as 0.999, as initially the learning rate is set to 0.002 and the batch size is set to one i.e instance normalization which implies that batch normalization is not used .In all tasks the value of λ is taken as 10.

4.1 Generator Training

The real image(x) is given to Generator-Gx and it creates fake image fake−y.and that fake image given to the discriminator-Dy and it predicts the fake decision(Dyfakedecision) and by using that fake decision mean square loss of Gx is estimated.The generated fake−y is given to the Generator-Gy,and it tries to regenerate the original image(Rx) with some loss.Now training Generator-Gy it takes an original image(y) from domain Y and generates fake image i.e fake-x.This fake-x is passed to discriminator Dx to and predicts fake decision(Dxfakedecision).Mean square loss of generator Gy

is calculated by using Dxfake decision.

forward cyclic loss is calculated from the reconstructed image(Rx) and real image x.**Backward cyclic loss** is calculated from the reconstructed image (Ry) and original image y.And total generator loss is sum of Gx loss, Gy loss,forward cyclic loss and backward cyclic loss.This final loss is back propagated.In all cases Adam optimizer is used.

4.2 Discriminator Training

The discriminator Dx is trained with real image x and fake−x generates Dxrealdecision,Dxfakedecision.Decision loss of Dx is calculated by real and fake decisions and result is back propagated.Similarly discriminators Dyis trained with real image y , fake−y generates Dyrealdecision,Dyfakedecision.By the help of real and fake decision,Decision loss of Dy is calculated by real and fake decisions and result is back propagated.

Both discriminators Dx and Dy are trained by using real image x,y and generates decision-x,decision-y.and both discriminators Dx and Dy are trained by using fake images x,y and generates fake-decision-x,fake-decision-y,sum of the real and fake losses of Dx and Dy are back propagated.

4.3 Image buffer

Generator and discriminator both are trained at a time,it is most important to take care of model not to change drastically for every simultaneous epochs.To avoid that[6] discriminator is fed with the previously generated images,instead of just one image generated by the generator.In image pool we should store 50 recently generated images,If we train like this both generator and discriminator overfits and then mode collapse will going to occur,by doing this model oscillations[7] and overfitting both can be reduced.

5 Experiments:

Maps dataset: Total number of samples in the data set are 1100 and they partitioned into trainA,trainB,testA,testB to train and test the model .This data set is collected from Kaggle[8]. By this data set CycleGAN is trained up to 150 epochs with 0.02 learning rate and then applied linearly decay of learning rate up to 315 epochs.

Vangogh2photo dataset: It is a small data set also partitioned into four sets for training and testing purpose,trained up to 150 with the 0.002 learning rate and from 150 to 230 epochs with decayed learning rate,i.e learning rate becomes zero gradually.This data set is collected from kaggle [9].

Summer2winter dataset: It is used to show the season transfer.The entire data is partitioned into four parts and used for training and testing. This is trained for 120 epochs with 0.002 learning rate and then from 120 to 230 epochs with linearly decay learning to zero.This data set is taken from the kaggle[10] this images are normalized ro 256 * 256 pixels .Total summer training images are 1273 and winter images are 854.

6 Objective functions

There are two loss functions in CycleGAN an adversarial loss and cycle consistency ,both are important and essential to bring good outputs. Two Components to the CycleGAN objective function, an adversarial loss, and Cycle consistency loss.Both the generators tries to fool their respective discriminator.

The loss of mapping function $G : X \rightarrow Y$is as follows.

$$L_{GAN}(G, DY, X, Y) = E_{y,pdata(y)}[logDY(y)] + E_{x,pdata(x)}[log(1 - DY(G(x))] \tag{1}$$

The discriminator tries to maximise the above expression and generator tries to minimize the adversary Discriminator.The mathematical formulation is $min_G max_{DY} L_{GAN}(G, DY, X, Y)$ .similarly $F : YtoX$ mapping loss is as below

$$L_{GAN}(F, DX, Y, X) = E_{x,pdata(x)}[logDX(x)] + E_{y,pdata(y)}log(1 - DX(G(y))] \tag{2}$$

similarly from Y $\rightarrow Xtheadversarialactslikeis$ $\min_F max_{DX} L_{GAN}(F, DX, Y, X)$. Only Adversarial loss is not enough to produce good output(images),because we are training both Generators at a time we need to form a cyclic loss .

$$L_{cyc}(G, F) = E_{x,pdata(x)}[||F(G(x)) - x||_1] + Ey, pdata(y)[||G(F(y)) - y||_1]. \tag{3}$$

The full objective function is formed by using above loss function together, and measuring the Cycle-consistency loss by using a hyper parameter λ .

$$L(G, F, DX, DY) = L_{GAN}(G, DY, X, Y) + L_{GAN}(F, DX, Y, X) + \lambda L_{cyc}(G, F), \tag{4}$$

So the final aim is to solve $G,F = argmin_G max_{Dx,DY} L(G, F, DX, DY)$.

7 Applications of CycleGANs

we can apply cycleGAN to many areas of applications for example converting Selfie of person to anime,winter season to summer season ,Animal to animal,sketch to photo ,to convert low resolution images to high resolution,object transfer ,and photo enhancement.There are wide range of application in computer vision ,graphics,video games etc

8 Results

The results of the experiments are shown below.CycleGAN is converting well from domain X to domain Y and then back to domain x as mentioned in original paper[2].we can observe that here Aerial image is converted to satellite and then back to aerial and vice versa.

NOTE-1 all Images are in the format of domain-X (Original Image)− > domain-Y (Generated)− > Re-generated (back to original Image).

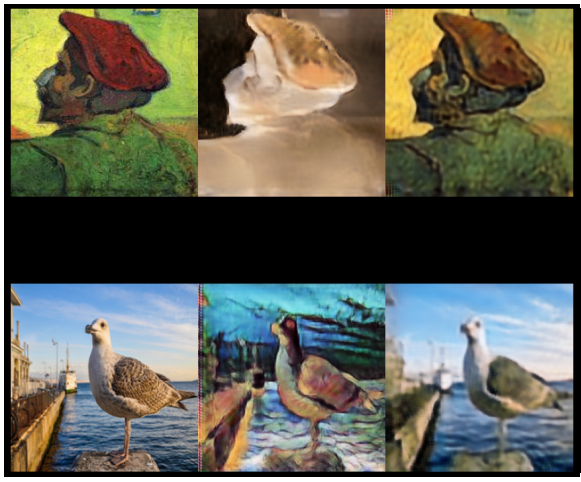


Figure 3: **Row-1:**Vangogh>Pictures>Vangogh and **Row-2:**Picture > Vangogh>Picture

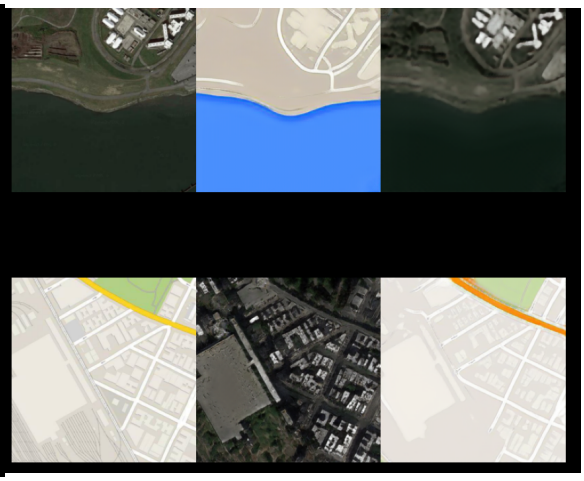


Figure 4: **Row-1:**Satellite>Aerial>Satellite and **Row-2:**Aerial> Satellite>Aerial

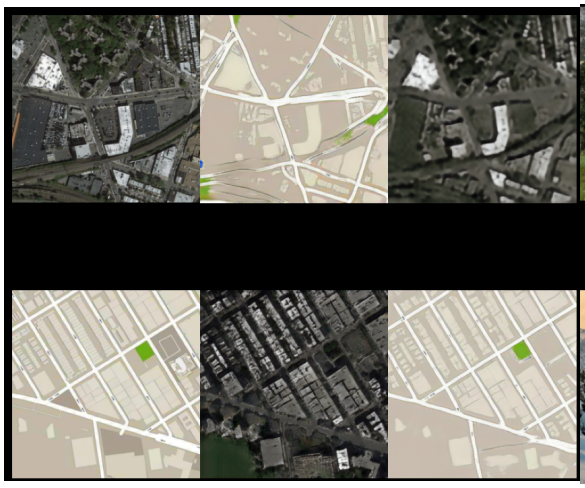
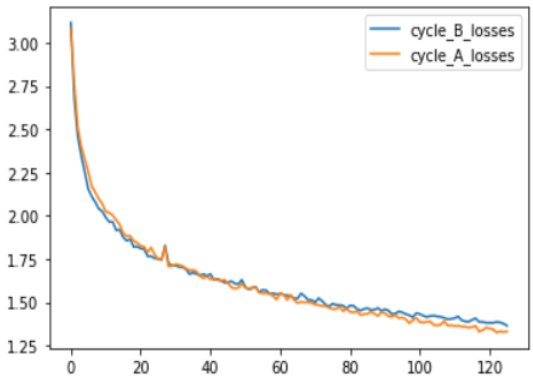


Figure 5: **Row-1:**Satellite>Aerial>Satellite and **Row-2:**Aerial > Satellite >Aerial

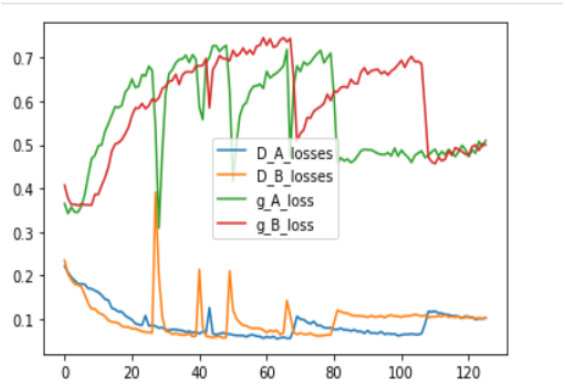
Figure 6: **Row-1:**Summer>Winter>Summer and **Row-2:**Winter> Summer>Winter

8.1 Loss Curves

The below loss curves are for summer to winter and winter to summer conversion task.There is no perfect measurement to show the performance of cycleGAN.But we can observe the model accuracy by the loss figures.



(a)



(b)

Figure 7: Loss plots for summer to winter vice versa.(a) Cyclic losses. (b) Generator and discriminator losses.

9 Limitations

(a)

(b)

Figure 8: These are some figures which shows cycleGAN difficulties. (a) Summer to Winter conversion(Women is not there in generated Images). (b) Statue is covered with the grass in generated image.

Even though cycleGAN creating good results,still there are some instances where it is struggling,above figures shows the some failure cases of summer to winter and winter to summer translations,here in fig-a the women in front of tree is not there in generated and re-constructed images,in fig-b the original image is a big statue,but in generated image the floor grass is also appeared on the statue.This observation is explained in horse to zebra translation by researchers in original paper.Not only that we can observe there is a resolution change from original image to generated and regenerated images,the clarity of the image is slightly reduced.

10 Conclusion

Out of three data sets that are used to test the cycleGAN,In all cases the results are much accurate as shown in paper.In few cases the model is recreating with low resolution images,some times back ground changes or colour changes are there,all above results and limitation clearly shows that CycleGAN is still needs a good amount of research.

References

[1] I. J. Goodfellow **andothers**, *Generative adversarial networks*, 2014. arXiv: 1406.2661 [stat.ML].

[2] J.-Y. Zhu, T. Park, P. Isola **and** A. A. Efros, *Unpaired image-to-image translation using cycle-consistent adversarial networks*, 2020. arXiv: 1703.10593 [cs.CV].

[3] P. Isola, J.-Y. Zhu, T. Zhou **and** A. A. Efros, *Image-to-image translation with conditional adversarial networks*, 2018. arXiv: 1611.07004 [cs.CV].

[4] J. Johnson, A. Alahi **and** L. Fei-Fei, *Perceptual losses for real-time style transfer and super-resolution*, 2016. arXiv: 1603.08155 [cs.CV].

[5] D. P. Kingma **and** J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: 1412.6980 [cs.LG].

[6] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang **and** R. Webb, *Learning from simulated and unsupervised images through adversarial training*, 2017. arXiv: 1612.07828 [cs.CV].

[7] I. Goodfellow, *Nips 2016 tutorial: Generative adversarial networks*, 2017. arXiv: 1701.00160 [cs.LG].

[8] suyashdamle, *Cyclegan-maps2satellite data set*, 2019. **url**: https://www.kaggle.com/suyashdamle/cyclegan?select=maps.

[9] ——, *Cyclegan-vangogh2photo data set*, 2019. **url**: https://www.kaggle.com/suyashdamle/cyclegan?select=vangogh2photo.

[10] ——, *cycleGAN dataset*, 2019. **url**: https://www.kaggle.com/suyashdamle/cyclegan?select=summer2winter_yosemite.