GAN-based Chinese Painting Generation

1st Zhehao Huang
Shanghai Jiao Tong University
Department of Automation
Shanghai China
E-mail KinghtH@outlook.com

2nd Yuchao Ye
Shanghai Jiao Tong University
Department of Automation
Shanghai China
E-mail mestyeyc@sjtu.edu.cn

Abstract—Chinese painting is the representative of traditional painting of China. We use cycle-consistent adversarial network to transform pictures into Chinese painting style, which is based on new data sets collected by our own. In order to make the transformed pictures look more realistic , we utilize a more effective optimization method of the generator and the discriminator and noise data augmentation. A 221-person Vision Turing Test conducted indicates it hard to distinguish our generated Chinese paintings and the real ones. Besides, we set up modeling of the Chinese painting brush, preparing for the training of controlling robot arm to create or imitate Chinese paintings.

Index Terms—Generative adversarial networks, Chinese painting, Chinese brush modeling

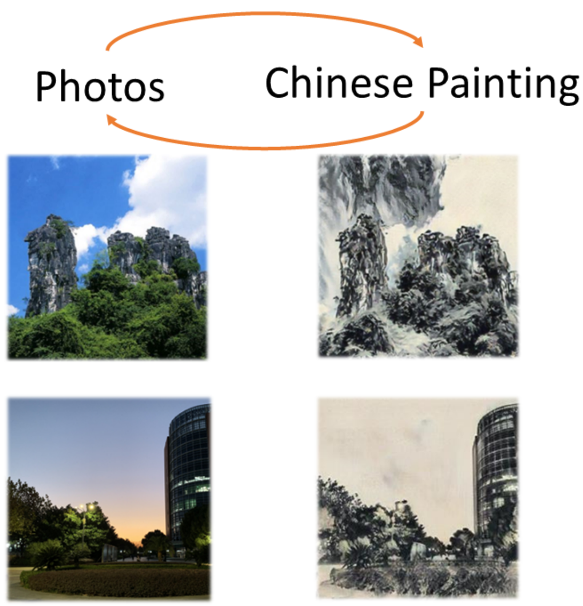


Fig. 1: Chinese paintings transfered from the original photos by our Cycle-GAN with noise data augmentation

I. Introduction

Traditional Chinese painting is one of the most brilliant Chinese culture creations, which has thousands of history. It can be classified into Chinese landscape painting, Chinese bird-and-flower painting and Chinese figure painting. But the creation of a splendid Chinese painting is quiet time-consuming. And the Chinese painting theme is mostly the same natural scenery and flora and fauna as the ancient times. Though nowadays there are still masters creating traditional Chinese paintings, the themes of modern pieces rarely change as the times developing. We try to discover how artificial intelligence can contribute to the creation of this Chinese traditional treasure.

We expect to transfer a original photo into the Chinese painting style, which actually is a neural style transfer [1] task which, recently, GAN-based models have brought new powers to [2] [3]. We have collected 2686 Chinese landscape paintings to deliberately train a cycle-consistent generative adversarial network to generate Chinese paintings not only as realistic as possible but also according to the original input photos.

The rest of this report is organized as follows. In Section II, we introduce some related works which our project is based on and the backbone network used in our training process, also some training strategies we attempt to utilize to make our transfer better. Section III generally shows our training data sets which prepares for the demonstration in the Section IV that introduce our experiments in details and some improvements contributing to the better transfer. In Section V, the Vision Turing Test designed and conducted by our own will show how realistic our generated Chinese paintings in the public aesthetic judgment. We will introduce in Section VI the modeling of the Chinese painting brush preparing for the simulation of robotic arm to create Chinese paintings. Finally we will draw a conclusion in Section VII to end this report.

II. Related Work

A. Generative Adversarial Networks [2] [4] [5]

A generative adversarial network(GAN) is a class of machine learning frameworks which learns by letting two neural networks contest with each other.

Common generative adversarial network often consists of a generative network(generator) and a discriminant network(discriminator), as shown in Fig. 2. The work of generative network is to randomly sample from the latent space as input, and the output result of it needs imitate the real sample in the training set as much as possible [2]. In other words, generator is trained to generate new data with the same statistics as the training set from noise input. The function of discriminant network is to distinguish the output of the generative network from the real samples as much as possible, which means that the generator is not trained to minimize the distance to a specific sample, but rather to fool the discriminator.

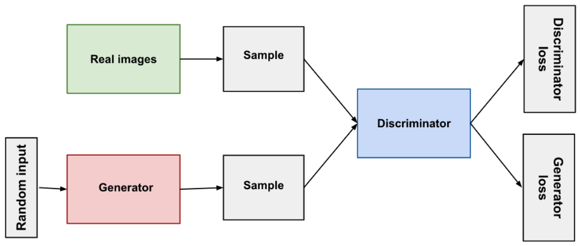


Fig. 2: Common GAN structure diagram

The two networks confront each other and constantly adjust their parameters. The ultimate goal is to make the discriminator unable to judge whether the output of the generator is true. This is why the machine learning method called the generative adversarial network.

GAN have achieved impressive results in image editing, image generation and representation learning [6]. A kind of GAN called the condition GAN(cGAN) is able to generate samples with specified characteristcs. [4] [7]

B. Cycle-Consistent Adversarial Networks [3] [8]

Cycle-consistent adversarial networks(cycleGAN) is an algorithm based on generative adversarial network that can learn to translate between domains without paired input-output examples.

There are two domain of diffenrent style samples (Domain X and Domain Y). To transfer the samples from one domain to another, cycleGAN uses two generators($G_{X\to Y}$, $G_{Y\to X}$), two discriminators(D_X, D_Y) and cycle-consistency loss to train the whole networks, the diagram of basic transfer structure is shown in Fig. 3.

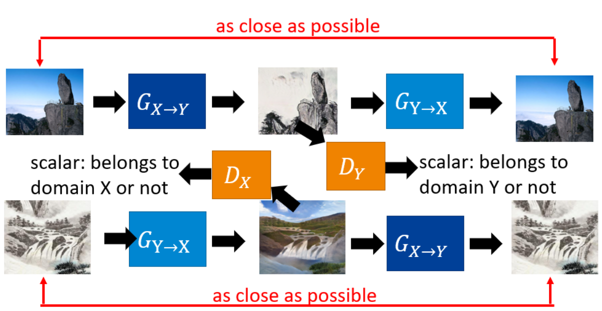


Fig. 3: CycleGAN structure diagram

Take transfer from Domain X to Domain Y as example, the $G_{X\to Y}$ is trained such that the output $\hat{y} = G_{X\to Y}(x), x \in X$, is indistinguishable from images $y \in Y$ as generative adversarial network training process. The $G_{Y\to X}$ is also trained in the same methods. Thus, both generators and discriminators are trained to be enough prominent. Meanwhile, to ensure that the content of the images will not change in the style transfer process, cycleGAN add one more structure called cycle consistency loss which can force the transfer to be ”cycle consistent”. To minimize the cycle consistency loss, the algorithm encourages $G_{Y\to X}(G_{X\to Y}(x)) \approx x$ and $G_{X\to Y}(G_{Y\to X}(y)) \approx y$.

Combining this loss and the front generators, discriminators, cycleGAN yields their full objective for unpaired image-to-image transfer.

C. Learning Rate Scheduler

[9] For networks training process there are several learning rate schedulers with different features which have influence on the training results. Learning rate schedules seek to adjust the learning rate during training by reducing the learning rate according to a pre-defined schedule, including time-based decay, step decay, cosine decay, etc.

1) Linear Decay
There are no changes in the first n epochs, which means remaining a constant learning rate in the front rounds. From n epochs on, learning rate decays linearly to a constant small enough value.

$$lr = \begin{cases} lr_{max} & E < E_0 \\ lr_{max} - DR \times (E - E_0) & E_0 < E < E_1 \\ lr_{min} & E_1 < E \end{cases}$$

where lr_{max} is the start learning rate, DR is the decay rate, E is the current epoch number, E_0 is the starting decay epoch, E_1 is the ending decay epoch.

2) Step Decay
Step decay schedule drops the learning rate by a factor every few epochs. The mathematical form of step decay is :

$$lr = lr_0 \times dropval^{\lfloor epoch/epochs_{drop} \rfloor}$$

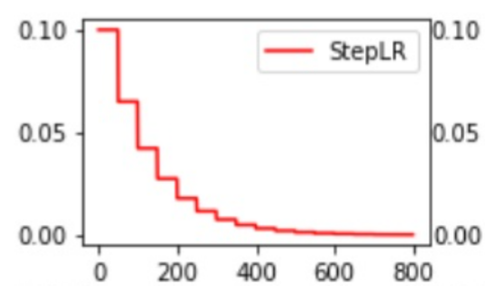


Fig. 4: Step decay

3) Plateau Decay
Plateau decay method is a kind of adaptive learning rate schedule. It changes the learning rate when some parameters have little changes which indicate the networks maybe convergence.

4) Cosine Annealing Decay [10]
This is a type of learning rate schedule that has the effect of starting with a large learning rate and decreasing to a minimun value before increasing again. The feature of cosine annealing decay is like resetting of the learning rate to simulated restart of the learning process so that it has more possibility to lead to the better networks weights.

$$lr = lr_0 \times (1 + \cos(\frac{epoch_{current}}{epoch_{set}}\pi))$$

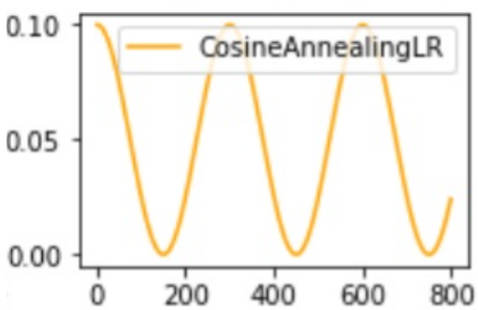


Fig. 5: Cosine annealing decay

III. Datasets

We use cycleGAN to achieve the task of transferring style from photo to traditional Chinese painting which can be classified into Chinese landscape painting, Chinese bird-and-flower painting and Chinese figure painting. To train the unsupervised cycleGAN network, we need to collect enough samples of the two domain. We evaluate our data sets during the whole algorithm implementation process. Thus, we create three versions of data sets, each of which is updated based on the previous version. All the images are collected from the Internet.

A. cp2photo_v1

The first version of data set is used to validate whether the algorithm code can be run. It contains 5382 photos including landscape photos, plant photos and animal photos, 1070 Chinese paintings including Chinese landscape paintings, Chinese bird-and-flower paintings and Chinese figure paintings.

Though the number of images is large enough and the images in one domain also have different styles like Chinese landscape painting and Chinese bird-and-flower painting, the train results on this data set is quiet unacceptable.

Fig. 6: Dateset version 1

B. cp2photo_v2

To improve the training results, we filtered the images into the same domain to ensure their style is close enough. The version two data set contains 826 photos including almost all landscape photos and 522 traditional Chinese paintings almost all of which are landscape paintings.

After filtering the version one data set, the image numbers of data set became smaller but the data set also becomes more pure. The results show that the training on this version of data set gains great improvement.

Fig. 7: Dateset version 2

C. cp2photo_v3

The third version of data set contains 2194 landscape photos and 2686 Chinese landscape paintings. This is expanded from the previous version by adding more required images collected from the Internet.

Compared with the previous version, the training on this larger version gain a certain improvement.

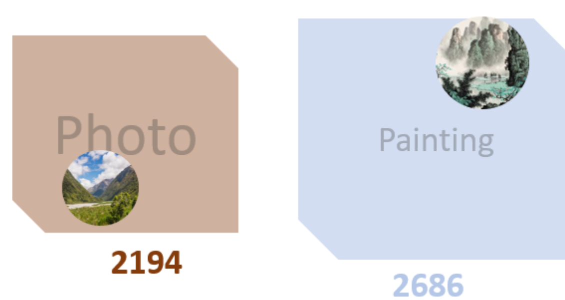


Fig. 8: Dateset version 3

IV. Experiments

In the preliminary attempt we use the data sets given by the original Cycle-GAN and combine different hyperparameters to first grasp a general training setting which can make the model stably achieve the same transfer results as in the original works. Then we gradually collect and change our own data sets to train the generator from the original photos to the Chinese paintings, during the training experiments we also propose a more effective optimization method of the generator and the discriminator and noise data augmentation to solve some problems we met with.

A. Stage 0

At first, we utilize the data sets used in the original Cycle-GAN and try different combinations of the hyperparameters to discover how they influence the transfer effect of the model. Most of the training settings are the same as the original defaults, such as the mode of the GAN we stick to the LSGAN [11] which changes the output of the classification from sigmoid function to the linear output to achieve the effect of optimizing gradient transfer. The other training settings are as follows in TABLE I.

TABLE I: Training settings in stage 0

Name	Data set	netG	epochs	lr policy
maps cyclegan	maps	resnet	200	linear
s2w cyclegan	summer2winter	resnet	200	linear
monet0	monet2photo	resnet	300	linear
monet1	monet2photo	unet	300	linear
monet2	monet2photo	resnet	300	plateau
monet3	monet2photo	unet	300	cosine

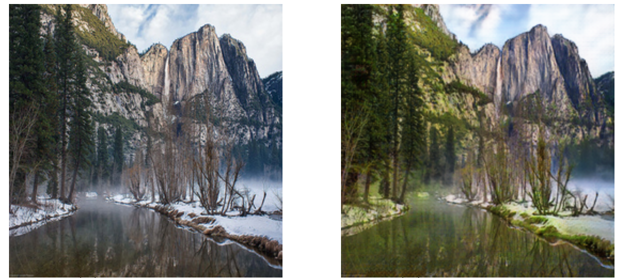
Fig. 9: Data sets in stage 0

In the experiments of stage 0, the centers of the most photos transfer better than the margin parts(Fig. 10). We discover that most models converge in 200 epochs, and the more epochs they train the more realistic the targete domain pictures generated are. The difference of the backbone networks of generator between ResNet [12] and Unet [13] isn't so obvious, and we will continue to use both networks to compare their effect on the transfer.

B. Stage 1

After discovering the default training settings and the original data sets, we implement the cycle-GAN model on our data set version 1 which including 5382 photos and 1070 Chinese paintings to try to achieve some transfer results. The training setting in detail is shown in TABLE II.

In stage 1 we actually haven't trained to produce some effective results that the loss of the discriminator is almost zero which means it can perfectly distinguish the fake Chinese paintings generated by the generator and the real



(a) winter real (b) summer fake

Fig. 10: Transfer done better in the center of the photos

TABLE II: Training settings in stage 1

Name	Data set	netG	epochs	lr policy
v1	cp2photo v1	unet	300	cosine

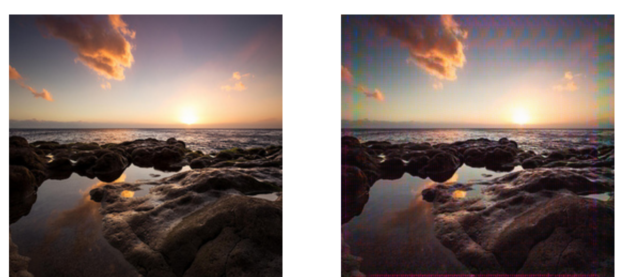
ones, and in this situation, the generator doesn't work as our expectation but generates paintings that are quiet the same as the original input photos(Fig. 11).

We find that it's because the discriminator is so powerful that it makes the gradients of the generator vanish and the generator degenerates to keep the photos the same as the original ones to just minimize the identity loss, which can be solved by reducing the learning rate or implementing gradient clip to the discriminator.

C. Stage 2

After failing training an effective model in stage 1 of the experiment, we decide to concentrate our data sets to make the general structure of the photos in original domain the same as in target domain, which brings about our data set version 2 that contains 826 photos of natural scenery and 522 traditional Chinese paintings of natural landscapes, expecting to obtain the baseline transfer effect.

What's more, we utilize a different optimization method from the original cycle-GAN algorithm that we train discriminator more often than the generator, which means we will update the parameters of discriminator n_D times per



(a) real photo (b) fake Chinese painting

Fig. 11: Chinese painting generated the same as the input one

update of the generator, where n_D is the hyperparameter decided before training.

The detail of the train settings are as follows in TABLE III.

TABLE III: Training settings in stage 2

Name	Data set	netG	epochs	lr policy	n_D
v2	cp2photo v2	resnet	300	cosine	3
v3	cp2photo v2	resnet	550	cosine	3
v4	cp2photo v2	unet	350	cosine	1
v5	cp2photo v2	unet	900	cosine	3
v6	cp2photo v2	resnet	700	step	2
v7	cp2photo v2	resnet	950	cosine	2
v8	cp2photo v2	resnet	1000	step	2

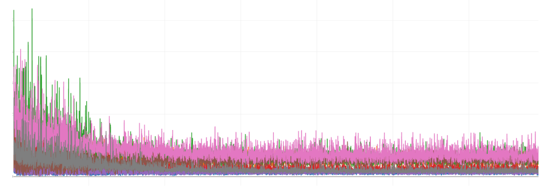
The changed optimization method of discriminator and generator makes a difference on the train effect of the data set version 2. And comparing the training effect between the generator network backbone of ResNet and Unet, we find ResNet is less likely to overfit than Unet because Unet extracts all the features [13] of the landscapes and tends to generate more figures like mountains to make the paintings hard to distinguish but also generates them in the area supposed to be empty, which actually makes the effect of the transfer worse than ResNet.

As to the learning rate schedulers, we try cosine annealing learning rate scheduler and the step decay learning rate scheduler, which both come to quiet satisfactory transfer results. But as the goal of the generative adversarial network is to compete against each other to find the Nash balance between the generator and the discriminator, we don't want it trapped in a local optimal solution too early, and the cosine annealing learning rate scheduler makes it possible to escape from the local optimal solution to find a better balance(Fig. 12). So in the following experiments we almost use cosine annealing learning rate scheduler as our learning rate adjustment policy.

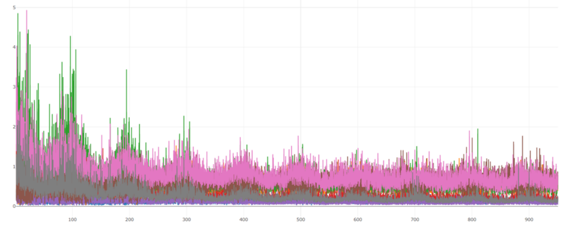
Though we achieve quiet better transfer effect on data set version 2(Fig. 13). There still is a problem of the transfer that the sky and the clouds in the original photos will become some strange noisy points appearing in the generated Chinese paintings, which is easy to be judged fake by human(Fig. 14). This problem can hardly be solved by just increasing the training epochs, thus we consider enlarging our data sets to improve the training effect. In the next stage we actually use noise data augmentation to eliminate these noisy points generated from the sky and the clouds.

D. Stage 3

Beside using the data set version 3 as the training data set, a milestone of our project is that we add noise data augmentation to the original input photos in order to reduce the model's overfitting of feature learning and make the generator better adapt to some unseen or relatively rare features in the photos.

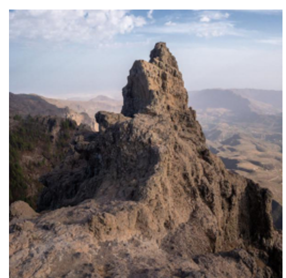


(a) loss curve of model cp2photo v6(step)

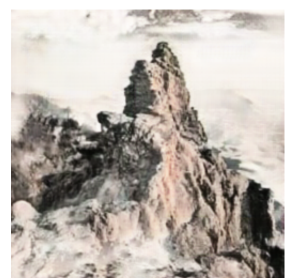


(b) loss curve of model cp2photo v7(cosine)

Fig. 12: different shape of loss curve between learning rate scheduler step and cosine

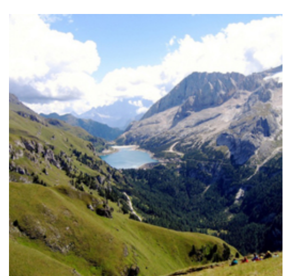


(a) real photo



(b) fake Chinese painting

Fig. 13: well done transfer by model cp2photo version 7 on data set verison 2



(a) real photo

(b) fake Chinese painting

Fig. 14: bad transfer of the sky and the clouds

We use two types of noise sources of salt and pepper noise and gaussian noise.

- Salt and pepper noise:
  Salt and pepper noise is a special case of impulse noise. It refers to random point noise with very low gray value or very high gray value, which appears as a random distribution of very dark points(peppers) and very bright points(salt) on the image(Fig. 15). A model trained on a data set with salt and pepper noise data augmentation should be more robust.

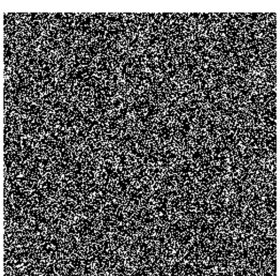
Fig. 15: the image after adding salt and pepper noise

- Gaussian noise:
  Gaussian noise, noise obeying a normal distribution(Equation 1), is a very common noise that is superimposed on every point of the image(Fig. 16). The main parameters are variance , mean μ and amplitude A.

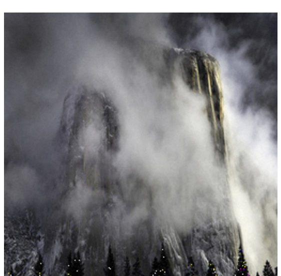
$$p(x) = \frac{A}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (1)$$

Training settings in the stage 3 of the experiment are shown in TABLE IV.

We test settings with the noise data augmentation under three cases which are just using SaltPepper noise,



(a) original Gaussian noise



(b) the image after adding Gaussian noise

Fig. 16: Gaussian noise

TABLE IV: Training settings in stage 3(netG=resnet, n_D=2)

Name	Data set	noise	epochs	lr policy
v7 Gaussian	cp2photo v3	Gaussian	500	cosine
v7 SaltPepper	cp2photo v3	SaltPepper	500	cosine
v7 Both	cp2photo v3	Both	500	cosine
v9	cp2photo v3	None	500	cosine
v10	cp2photo v3	None	500	step
v11	cp2photo v3	Both	500	cosine
v12	cp2photo v3	None	900	cosine
v13	cp2photo v3	Both	900	cosine

just using Gaussian noise and using both noise on the training of model version 7 which performs best on the data set version 2. We find that using both noise randomly performs better than other two settings on the elimination of the noisy points of the transfer of the sky and the clouds in the original input photos. We use the improved optimization method and both noise data augmentation to obtain the best model version 11 on the data set version 3(Fig. 17).

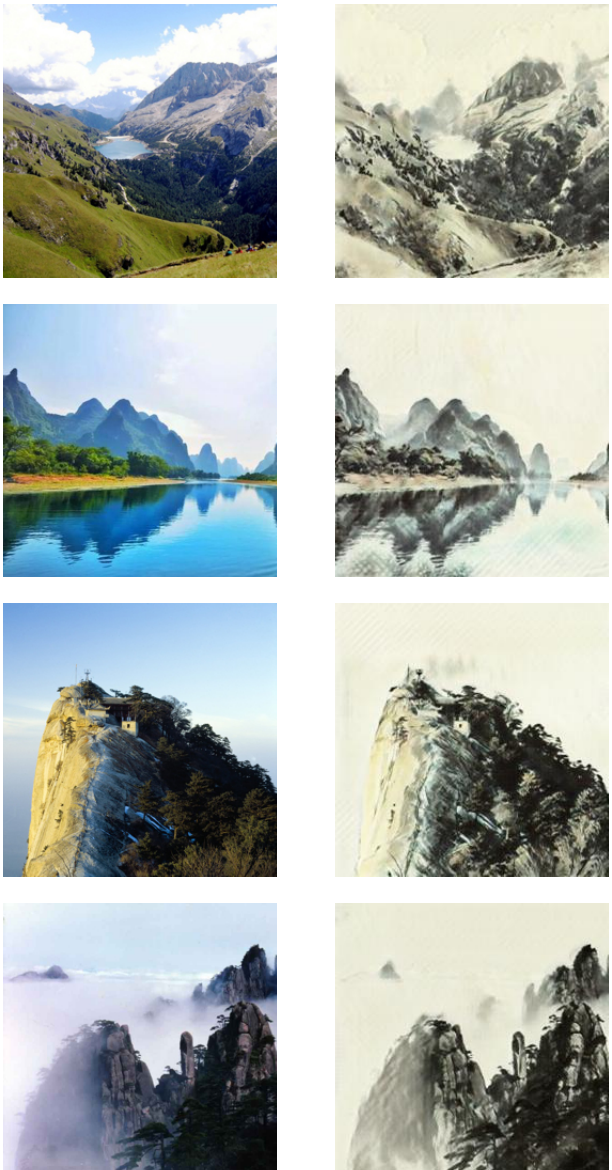
V. Human Study: Visual Turing Test [14]

We conduct a survey to ask public to judge the paintings presented are whether created by real people or generated by our Cycle-GAN model. The detailed settings of our questionnaire are as following:

- There are respectively 25 real and fake Chinese paintings and in total 50 paintings in our data base. They are randomly selected from our data set version 3 that just remove meaningless samples containing obvious modern elements, which can reveal the total performance of our model not just the most realistic ones.
- For the convenience of the participants, every questionnaire will present 10 paintings randomly, which means the ratio of the real and fake Chinese paintings in a questionnaire is not necessarily 1:1.
- We also ask whether the participants have learned traditional Chinese painting skill or not.

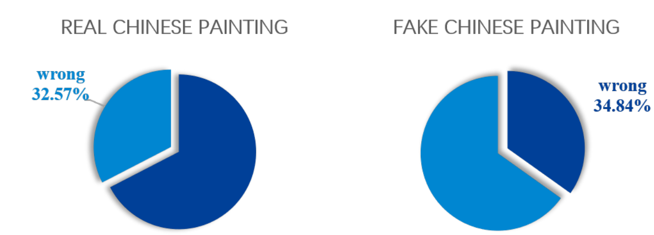
We collect the results of the questionnaires for 221 people(Fig. 18). The pie charts clearly demonstrate that they recognize 32.57% of real Chinese paintings as fake while consider 34.84% fake Chinese paintings as real, which meanings that the Chinese paintings generated by our model somehow confuse them and make their judgment of the true Chinese paintings wrong.

Further take out those who have learned traditional Chinese painting skill(38 people)(Fig. 19), it turns out that they have a stronger ability to distinguish real paintings that they only miss 27.78% true Chinese paintings created by real people, but they also have a higher error rate of 35.53% in distinguishing fake Chinese paintings. It is obvious that some fake Chinese paintings generated by our Cycle-GAN model is so realistic to distinguish(Fig. 20).



(g) real photo (h) fake Chinese painting

Fig. 17: well done transfer by model cp2photo version 11 on data set veriosn 3



(a) pie chart of real Chinese paintings (b) pie chart of fake Chinese paintings

Fig. 18: questionnaire result of all participants

(a) pie chart of real Chinese paintings (b) pie chart of fake Chinese paintings

Fig. 19: questionnaire result of participants who have learned Chinese painting

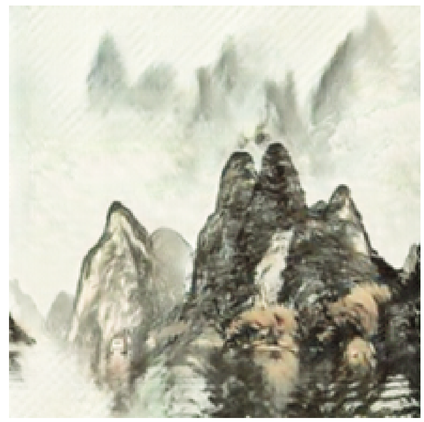


Fig. 20: The generated Chinese painting of highest error rate of 62.79%

VI. Chinese Painting Brush Modeling

After completing the task of transferring photos to Chinese paintings, we consider how to training a controlling robot arm to create or imitate the Chinese painting we transferred.

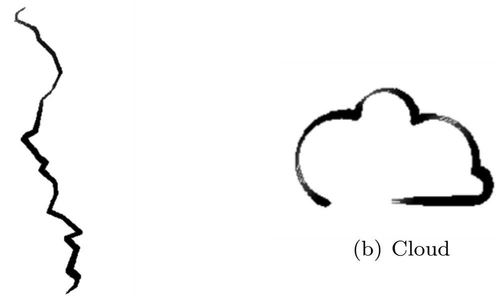
Rather than using brush stack on the real robot arm to do experiment, we first modeling a virtual Chinese painting brush so that we can create Chinese paintings in computer virtual environment.

The *turtle* library in Python is used to draw the digital images and provide the drawing canvas. But different from the common drawing brush, the strokes of Chinese painting brush have more details like thickness changing and shade changing through the drawing process [15] [16], as shown in Fig. 21

Thus, based on the *turtle* library we create a *brush* class to imitate the stroke of Chinese painting brush. The main

(a) Common brush (b) Chinese painting brush

Fig. 21: Difference of brushes



(b) Cloud

(a) Random drawing lines

Fig. 22: Images drew by our brush model

used function in *brush* class is *drawline* which accepts three input parameters including direction, drawing speed and move distance and drawing the corresponding line.

To obtain the effect of line thickness changing and shade changing, we set up a function relationship of thickness and speed to change the thickness, also shade and thickness and speed to change the shade. The coefficient in the equtions is set by experience. To achieve a more realistic effect, we also add some offset noise line to the main drawing line.

$$currentThickness = maxThickness - 0.5 \times speed$$

$$shade = 0.4 \times (1 - \frac{curThick}{maxThick - 7} + \frac{speed}{40} - 0.9)$$

From the above, the Chinese painting brush model can be controlled by three parameter, direction, speed and move distance. The agent can draw lines in different direction and different length by adjusting the direction and distance parameters, also change the thickness and color shade of a line by adjusting the speed parameters for which more fast the speed is, thinner and lighter the line is.

In the future work, deep reinforcement learning can be apply to the Chinese paintings drawing imitation works. By adjusting the three parameters, the agent is able to draw various lines with features of lines in Chinese painting. Giving the suitable reward and punishment policy, we are able to perfect complete our algorithm and train the agent. Finally, transplant the agent in virtual environment to the real robot arms.

VII. Conclusion

To summarize this report, we try to use Cycle-GAN model to transfer original input photos into Chinese painting style. And we collect three data sets to help train a powerful neural style transfer model. In order to solve the gradient vanishing problem and the noisy points created by the sky and the clouds, we improve our algorithm with a more effective optimization method of the generator and the discriminator and the noise data augmentation respectively, which indeed brings about the satisfactory transfer effect. To test our model with the public aesthetic judgment, we conduct a Vision Turing Test for 221 participants that demonstrates some of the Chinese paintings generated by us is so realistic to distinguish. We also work out a method to simulate the robotic arm creating Chinese Paintings, which provides an interface for future use of robotic arms for painting.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2414–2423.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," Advances in neural information processing systems, vol. 27, pp. 2672–2680, 2014.

[3] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.

[4] B. Dai, S. Fidler, R. Urtasun, and D. Lin, "Towards diverse and natural image descriptions via a conditional gan," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2970–2979.

[5] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," arXiv preprint arXiv:1609.03126, 2016.

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1125–1134.

[7] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," arXiv preprint arXiv:1605.05396, 2016.

[8] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2849–2857.

[9] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv preprint arXiv:1609.04747, 2016.

[10] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017.

[11] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2794–2802.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.

[14] A. Xue, "End-to-end chinese landscape painting creation using generative adversarial networks," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 3863–3871.

[15] H. T. Wong and H. H. Ip, "Virtual brush: a model-based synthesis of chinese calligraphy," Computers & Graphics, vol. 24, no. 1, pp. 99–113, 2000.

[16] N. S. Chu and C.-L. Tai, "Real-time painting with an expressive virtual chinese brush," IEEE Computer Graphics and applications, vol. 24, no. 5, pp. 76–85, 2004.