Object Transfiguration with GANs

Haojing Hu
haojing@usc.edu

Zheng Wen
zwen1423@usc.edu

Mentor: Jiali Duan
jialidua@usc.edu

June 29, 2020

Abstract

Generative Adversarial Networks (GANs) have already been widely studied and show their great performance in many image-to-image translation tasks. One of the most challenging image translation task is the object transfiguration task, which can change the object in the image from one domain to another domain. In this project, we perform the object transfiguration task with GANs. Unlike the object transfiguration task in the CyclcGAN, our network can not only incorporate color and texture transformation but also realize obvious shape deformation between two object domains. We further show our method can preserve some import information of the original images, e.g. the location and posture of the objects. In our method, unpaired data are used to make this task more applicable and multiple important loss functions are added to supervise our neural network to achieve this task, including Multi-scale Structural Similarity (SSIM) loss, Perceptual loss. We also conduct the ablation study to explore the functionality of each loss function.
Keywords: *Generative Adversarial Network; Object Transfiguration; Perceptual Loss;*

1 Introduction

Image-to-image translation is an interesting and wide topic in the computer vision area. In fact, many computer vision tasks can be viewed as the image translation task, such as the Style Transfer, Image Colorization and Super Resolution. Generative Neural Networks (GANs)[1] has shown their great power in dealing with these tasks[2][3][4][5][6]. Unsupervised and unpaired image translation is a special form of image translation. It uses unpaired images during the training, so there is no ground truth images to directly guide the GAN. However, the data collecting is much easier in this situation and thus the unpaired image translation is more applicable. CycleGAN[2] is the first GAN-based method to realize such a task and its cyclic structure is widely used as a baseline model in other works on the unpaired image translation despite some novelty improvements. However, CycleGAN shows its weakness in the object transfiguration when encountering two object domains having huge shape difference, e.g. translating between the cat and the dog. In the recent work, GANimorph[7] has solved this problem and manages to cope with shape deformation in the object transfiguration task. Two main contributions of GANimorph lie in the Multi-scale Structural Similarity (MS-SSIM) loss function[8] they added as a cycle loss term to help the GAN better reconstruct the images as well as the use of dilated convolutional discriminator to treat the discrimination as a semantic segmentation task.
In our project, we want to further explore the unpaired and unsupervised object transfiguration task. Our model is mainly based on the GANimorph but we further improve it. Our work in this project can be summarized as:

- We use the cat to dog dataset[9] to perform our object transfiguration task and implement two baseline models on it, including the CycleGAN and the GANimorph

- We add the perceptual loss function in the GAN to better preserve the spatial information and detailed shape information of the original image

- We compare our good generated results with other methods and provide some analysis

- We conduct ablation study to justify the functionality of each loss function in our GAN system

- We also show our failed results and analyze the reason of failure

2 Methodology

Our method is mainly referred to the GANimorph[7], which adopts a cycle pipeline as same as the CycleGAN[2]. Therefore, in this section, we would first briefly introduce these two models, including their pipelines and the problem setting. Next, we will introduce our modification of the loss functions and provide some intuitions about these objectives. Finally, the architecture of the generator and discriminator will be shown in the last of this section.

2.1 Problem setting

The general objective of object transfiguration is to change the object in the image from one domain to another domain. Given two object domain X and Y and sample unpaired images $\{x_i\}_{i=1}^{N}$ and $\{y_i\}_{j=1}^{M}$, our final objective is to learn a mapping function between X and Y, we denote them as $G : X \rightarrow Y$ and $F : Y \rightarrow X$. Due to the object transfiguration itself is an ill-posed problem, which means there is no unique solution to this problem, to make the mapping function meaningful, we do need to add some extra constraints on them. A common constraint proposed in [2] is to preserve the cycle-consistency of the mapping, which means $F(G(X))$ should be close to the original input X and similar for $G(F(Y))$ and Y. To make this possible, the pipeline of this GAN system should contain two generators G, F and two discriminators D_X, D_Y. The two generators and two discriminators all learn together and finally achieve the equilibrium. The overall pipeline is shown in Fig.1.

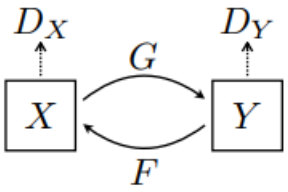

Figure 1: Cycle pipeline[2]

In the original CycleGAN model, three types of loss functions are used to supervise this system. The first one is the traditional adversarial loss function:

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

The adversarial loss term of $\mathcal{L}_{GAN}(G, D_X, X, Y)$ is similar to the Eq.1. To preserve the cycle-consistency, the cycle loss term can be formulated as:

$$\mathcal{L}_{cyc}(G, F, X, Y) = \mathbb{E}[\|F(G(x)) - x\|] + \mathbb{E}[\|G(F(y)) - y\|] \tag{2}$$

Another loss function term is the identify loss and its functionality is to preserve the information in the original image like colors. It can be formulated as:

$$\mathcal{L}_{identity}(G, F, X, Y) = \mathbb{E}[\|G(x) - x\|] + \mathbb{E}[\|F(y) - y\|] \tag{3}$$

2.2 Baseline Model

Since we want to incorporate shape deformation in our task, the CycleGAN is not appropriate to become our baseline model, because it is not successful at shape change. There are two main reasons for this failure. First, the CycleGAN uses the PatchGAN[4] as the discriminator, which assuming each image patch is independent of other patches. However, shape deformation does need the discriminator to be aware of the large-scale structure of the whole image, which relies on the relationship between image patches, so the PatchGAN is inappropriate to be used in this setting. Another reason is the loss function of the CycleGAN. The Eq.2,3 are actually based on the L1 norm difference between raw pixel values of images. Such loss functions can only represent color information but can not represent any shape information.

To overcome these drawbacks, GANimorph provides some solutions and manages to cope with shape deformation in the object transfiguration task. The first modification of the GANimorph is to use dilated convolution[10] in the discriminator and treat the discrimination as a semantic segmentation task. The dilated convolution is firstly proposed as a technique to deal with the semantic segementation task. It can greatly enlarge the receptive field of the network without the help of pooling layer, thus it makes the network better perceive the larger-scale structure of the image with less loss of detailed structures. Also, the output of the discriminator of GANimorph now becomes a map of real/fake (0/1) val-

ues. From this, it is more logical to use the LSGAN[11] loss function as the adversarial loss term:

$$\mathcal{L}_{LSGAN}(G, D_Y, X, Y) = \mathbb{E}[(D_Y(y) - 1)^2] + \mathbb{E}[D_Y(G(x))^2] \tag{4}$$

Based on this loss function, the map can tell the generator to focus more on those fields that are discriminated to be fake.

Another improvement of GANimorph is that it adds the Multi-scale structural similarity loss[8] (MS-SSIM) as the cycle loss term and also adds the feature matching loss[12] to further stabilize the training of GAN. The MS-SSIM loss function is better to represent the error if two images have the perceptual difference in the structure, so it is more conformed to the shape deformation. However, the feature matching loss is not very appropriate to use in the unpaired image translation setting. The feature matching loss can be formulated as:

$$\mathcal{L}_{FM}(G, X, Y) = \frac{1}{N-1} \sum_{i=1}^{N-1} \|f_i(y) - f_i(G(x))\| \tag{5}$$

where the f_i denotes the i-th layer activation of the discriminator D_Y. We can see that this loss function requires the generated image $G(x)$ and the real image Y from the same domain to have close activation in the discriminator. However, in the unpaired object transfiguration, the location of the object may be different in $G(x)$ and Y. Simply using L1 norm difference to force them to have close activation would be problematic in this situation. Hence, we remove the feature matching loss in our model but replace it with the perceptual loss function.

2.3 Perceptual Loss

Perceptual loss is firstly used in neural style transfer task in [13]. It is based on the representation of high-level features extracted from a pre-trained neural network, e.g. VGG network[14]. Such loss function is superior to a per-pixel based loss function because it involves some semantic information of the image and this information is valuable and helpful for many computer vision tasks. In [13], two types of perceptual loss functions are proposed. One is the feature reconstruction loss:

$$l_{feat}^{\phi,j}(y, \hat{y}) = \frac{1}{C_j H_j W_j} \|\phi_j(y) - \phi_j(\hat{y})\| \tag{6}$$

where the $y, \hat{y}$ denote two images and ϕ denotes the pre-trained neural network. We can see this term is actually the mean L1 norm difference between the j-th layer feature maps of two images. To understand this, the example visualization of feature map in a VGG-16 network is shown in Fig.2. From this figure, we can see that only high-level spatial information is captured when we moving deeper into the network. However, the color, texture and exact shape of objects are no longer contained in the feature map of relatively deeper layers. Hence, we can conclude that this feature reconstruction loss only penalizes two

images that are different in the location information of the objects in the images.

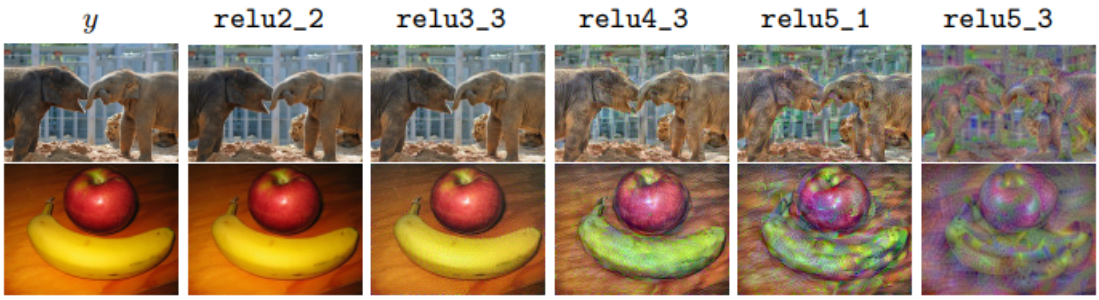

Figure 2: Example visulization of feature maps from difference layers of a pre-trained VGG-16 network. From left to right, the layer is getting deeper and left-most column is the original input images.[13]

Another perceptual loss is the style reconstruction loss, which can be formulated as:

$$l^{\phi,j}_{style}(y,\hat{y}) = \|G^{\phi}_{j}(y) - G^{\phi}_{j}(\hat{y})\| \tag{7}$$

where the $G^{\phi}_{j}(\cdot)$ denotes the Gram matrix (covariance matrix) of j-th layer feature map from pre-trained network ϕ. It can be computed by $\Psi\Psi^{T}/C_{j}H_{j}W_{j}$, where the Ψ is reshaped matrix of $\phi_{j}(x)$ of size $C_{j} \times H_{j}W_{j}$. The example visualization of the Gram matrix is shown in Fig.3. From the figure, we can see the Gram matrix only contains the information of stylistic features, like shape, structure and colors but the spatial information is lost. As we moving deeper, larger-scale structures are well preserved. Therefore, the style reconstruction loss only penalizes two images that are different in the structural information.

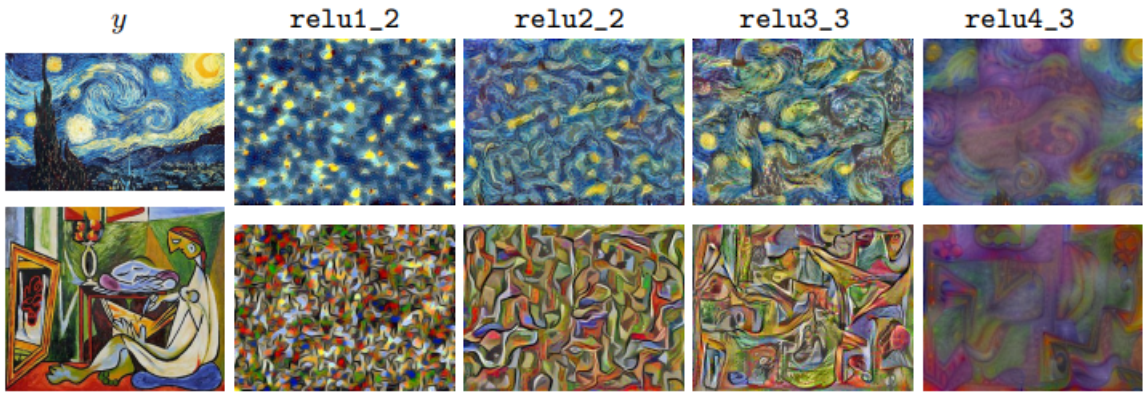

Figure 3: Example visulization of Gram matrices of feature maps from difference layers of a pre-trained VGG-16 network. From left to right, the layer is getting deeper and left-most column is the original input images.[13]

In our model, because we want the object in the generated image to have the same location and posture with the one in the original image, thus we add the feature reconstruction loss like in the Eq.6 between the X and $G(X)$ and so do the Y and $F(y)$. To better filter

out other shape information and only preserve the location information, we only use a relatively deep layer to build this loss. To avoid the ambiguity of "reconstruction", we will use the name spatial matching loss to denote this loss and it can be formulated as:

$$\mathcal{L}_{spatial}(X, G, D_Y, j) = \frac{1}{C_j H_j W_j} \| D_Y^j(x) - D_Y^j(G(x)) \| \tag{8}$$

Also, in our implementation, we find that use the discriminator network instead of the pre-trained VGG-16 is more convenient and also the features extracted by the discriminator are much task-specific, which can better represent the semantic information of the object X and Y in our transfiguration task. Besides, due to the dilated convolution is used in the discriminator, it can preserve much larger-scale semantic information. Therefore, the pre-trained network ϕ is replaced by D_X or D_Y. The spatial matching term $\mathcal{L}_{spatial}(Y, F, D_X, j)$ is similar to the Eq.8.

On the other hand, we also want the object in the generated image to have same structural information to the one in the real image, so we also add the style reconstruction loss based on Eq.7 between the $G(X)$ and Y as well as the $F(Y)$ and X. Similarly, to avoid ambiguity, we name this term as the style matching loss. To better gather the structural information from small-scale to large-scale, we sum up the loss terms obtained by different layers to build the final style matching loss:

$$\mathcal{L}_{style}(X, Y, G, D_Y, p, q) = \sum_{j=p}^{q} \| G_j^{D_Y}(Y) - G_j^{D_Y}(G(x)) \| \tag{9}$$

The style matching loss term for $\mathcal{L}_{style}(X, Y, F, D_X, p, q)$ can be obtained similarly as Eq.9.

2.4 Overall Objective Loss Function

The overall objective loss function in our model includes 4 terms: the adversarial loss term, the spatial matching loss term, the style matching loss term and the cycle loss term. It can be computed by:

$$\begin{aligned}
\mathcal{L}_{total} =& \lambda_{adv} \cdot \mathcal{L}_{LSGAN}(G, D_Y, X, Y) \\
&+ \lambda_{spatial} \cdot [\mathcal{L}_{spatial}(X, G, D_Y, j) + \mathcal{L}_{spatial}(Y, F, D_X, j)] \\
&+ \lambda_{style} \cdot [\mathcal{L}_{style}(X, Y, G, D_Y, p, q) + \mathcal{L}_{style}(X, Y, F, D_X, p, q)] \\
&+ [\lambda_{cyc-L1} \mathcal{L}_{L1}(G, F, X, Y) + \lambda_{cyc-SSIM} \mathcal{L}_{MS-SSIM}(G, F, X, Y)]
\end{aligned} \tag{10}$$

We should note that this total loss function is only used to update the generator and the objective of the generator is to make this loss as smaller as possible. The update of discriminator D_X and D_Y is only based on the adversarial loss term and the objective of discriminator is to make it as larger as possible.

To make the selection of the weight of each term easier, we still adopt the loss normalization method in [7], which normalize each loss term every several steps based on their moving average during the training process. This will eliminate the difference of the order of magnitude of each loss term. Therefore, all the weights should sum to 1. In our experiment, we find the relatively good weight values for each term are $\lambda_{adv} = 0.33, \lambda_{spatial} = 0.07, \lambda_{style} = 0.26, \lambda_{cyc-L1} = 0.1, \lambda_{cyc-SSIM} = 0.24$.

2.5 Architecture of Generator and Discriminator

We adopt the architecture of generator and discriminator in [7] and they are shown in the Figure 4 and 5.

Figure 4: The architecture of the generator[7]. Blue block: Convolutional Layer; Green Block: Residue Block; Orange Block: Deconvolutional Layer; Orange Arrow: Skip Connection

The generator in the Fig.4 is based on the standard encoder-decoder architecture. However, there are several skip connections to connect the encoder and decoder, which do not exist in the traditional encoder-decoder network. These skip connections allow the network to learn the transformation between features in a multi-scale way and thus improve the capacity of the network. Besides, the role of the residual blocks is to help the network preserve the low-frequency information of images when getting deeper into the network.

Figure 5: The architecture of the discriminator[7]. Blue block: Convolutional Layer; Yellow Block: Dilated Convolutional Layer; Orange Arrow: Skip Connection

In the discriminator, we can see there are 3 dilated convolution layers. As stated before,

these layers greatly enlarge the receptive field of the network and help it to perceive larger-scale structures in the image. The skip connection is used to aggregate the information from both low level and high level. This also increases the information flow between the generator and the discriminator and help the generator to focus more on those regions are unrealistic.

In our experiment, because we use a relatively smaller and simpler dataset compared to [7], we reduce the number of filters N in the generator and discriminator to 32. To construct the spatial matching loss term, the feature extracted from the last dilated convolutional layer is computed. For the style matching loss term, the features from three dilated convolutional layers are all computed.

3 Experiment

In this part, the experiment process is exploited. In our experiment, we adopt some image preprocessing techniques to make full use of the training set, giving more information to the network, which is described in 3.1. The hyperparameters of our network are described in 3.2. To show the validity of the shape deformation of our model and the difference between baseline, we compared the result of our model with cycleGAN and GANimorph as is shown in 3.3. Moreover, to verify the function of each loss function mentioned in 2.5, some ablation study is carried out and shown in 3.4. Some failure results and reasons are shown in 3.5.

3.1 Dataset and Preprocessing

In our implementation, the dataset contains less samples to shorten the training time and test our model more conveniently. Our experimental dataset only contains images about face of dog and cat. The number of images in training set and test set of each category is shown in Table 1.

Category	Training set	Test set
Cat	771	1264
Dog	100	100

Table 1: Number of Samples in dataset

Before the images are fed into the network, they are first resized to 128*128 and we also apply some data augmentation method on each image to get more information from the finite training set. In order to give more posture and orientation information to the network, each image is randomly rotated by -10 degree to 10 degree and horizontal flipping is randomly carried out on each training image. To utilize more information about the location, each training image is randomly translated.

3.2 Hyperparameter

In our experiment, the network is trained for 40 epochs, and in each epoch, the network is updated for 1000 steps, the learning rate is set to 2×10^{-4}.

3.3 Results Comparison

In this section, we compare our result with CycleGAN and GANimorph to show the effect of our model.

Figure 6: CycleGAN
Left: True Right: Fake

Fig.6 shows the deformation effect of CycleGAN, there are three groups of images are given, the left image of each group is the original input image, and the right image is the corresponding fake one. From Fig.6, we could see the CycleGAN completely fails on the shape deformation task. We believe this is because of the L1 norm loss in CycleGAN between the real image and the fake image. This constraint does not allow huge difference like deformation. The only change allowable in CycleGAN is the texture change like contrast and color shift. Besides, since the reconstruction loss only contains L1 norm loss, if the error measured between the reconstructed image and the real image contains shape difference, the generator will learn how to eliminate it to mimic the pattern of the true image.

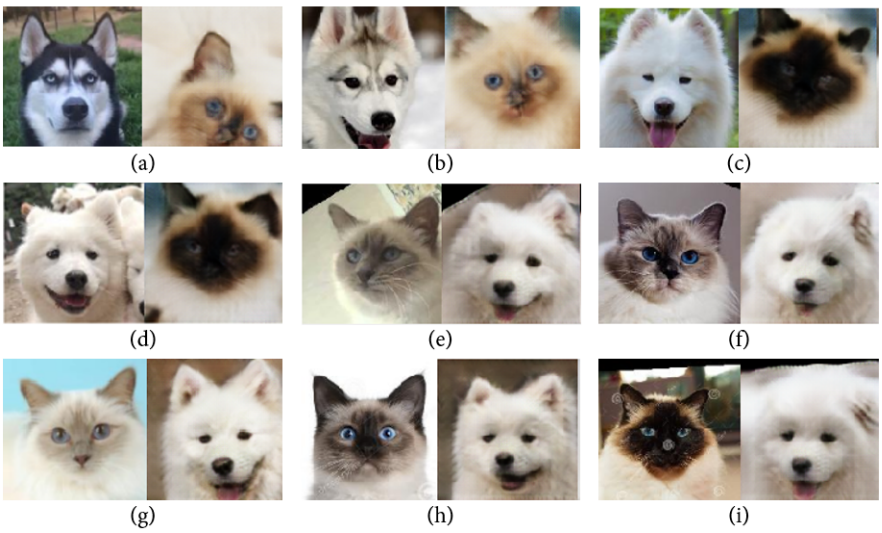

Figure 7: GANimorph
Left: True Right: Fake

Fig.7 shows the result of GANimorph, which is set as our baseline. There are nine groups of result images. From the Fig.7(a)(b)(c), we could see that the GANimorph could not preserve the location information well. In the first image, the face of the Husky in (a) lies in the middle of the image, but the face of the generated cat lies in the bottom of the image. In (b) and (c), the eyes of the true image are horizontal, but the generated image is somehow with inclined eyes. Fig.7(d)(e)(f) show the posture variance of GANimorph. In (d), the true dog is looking to the right, while the generated cat seems to look to the left. In (e)(f), the true cat is with different orientations while the generated dog looks with the same orientation. Fig.7(g)(h)(i) show that the generated images by GANimorph have relatively low diversity, even the true images of (g)(h)(i) are quite different, the generated dog are all Samoyed with almost the same expression, which could also be proved in comparison with (e)(f).

The main reason for these drawbacks of GANimorph lies in the feature matching loss. In the GANimorph model, the feature matching loss compares the outputs of all the convolutional layers with no specific selection. As is shown in 2.3, applying feature matching loss onto the outputs of different convolutional layers, the effect is different. The shallow layers tend to protect the detail information like color and texture, and deeper layers tend to protect the location information. So, if these two kinds of information are combined together to form a single loss, the effect will be ambiguous. In this situation, we could not predict which part takes a larger part of the loss, and the algorithm will find an output with the smallest loss, the location and posture information are not guaranteed to be preserved. Besides, as stated above, because the features extracted from shallow layers are also compared, the output image will contain similar color and texture for sure. These drawbacks is extremely obvious when the dataset is small for the information in the image is less diverse.

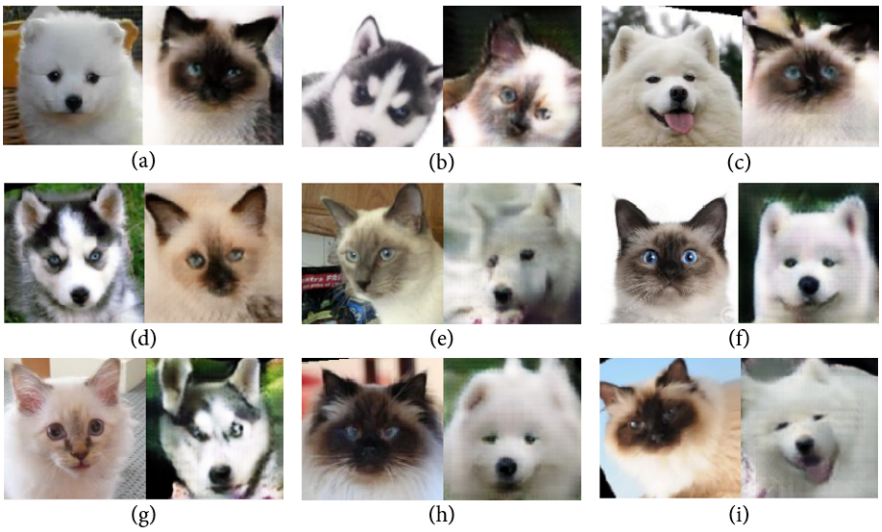

Figure 8: Our model
Left: True Right: Fake

Fig.8 shows the effect of our model. To compare with GANimorph, there are 9 groups of images correspondingly. In Fig.7(a)(b)(c), we could see our model could better preserve the location of the object face better than GANimorph. The representative result is the group(b), where the face of the Husky even does not lie in the center, but our model could generate a cat face in a corresponding region. In Fig.8 (d)(e)(f), the posture and orientation of the real image could be well preserved. The diversity of our model is shown in Fig.8(g)(h)(i). In our experiment, the portion of generated Husky and Samoyed is well balanced and there is also high diversity in the expressions of generated images.

Because of the function of designed spatial matching loss and style matching loss, some important information of the original image could be preserved largely. The source of the diversity is that the functionalities of different loss terms are well disentangled, so that shape deformation can be achieved alone without any posture adjustment. After the location and some detail information be preserved by spatial matching loss and style matching loss, the MS-SSIM loss term could help to reconstruct a complete face. There is no overlap in these three loss functions.

3.4 Ablation Study

In this session, ablation study is carried out to further exploit and justify the functionality of the key loss functions used in our model, including style matching loss, spatial matching loss and MS-SSIM loss.

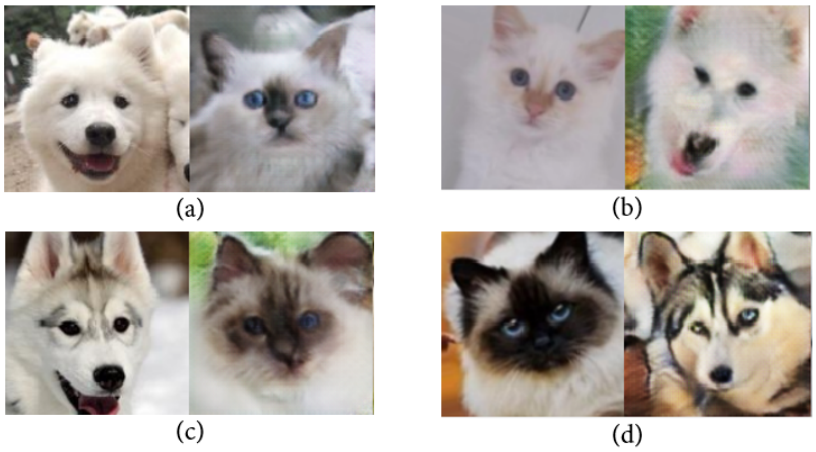

Figure 9: MS-SSIM + Spatial Matching
Left: True Right: Fake

The functionality style matching loss term is tested in the first setting. In this setting, we drop out the style matching loss introduced in 2.3 and allocate its weight to the spatial loss term in order to ensure the sum of weights is still 1. Therefore, in this setting, the weights are set as $\lambda_{adv} = 0.33, \lambda_{spatial} = 0.33, \lambda_{cyc-L1} = 0.1, \lambda_{cyc-SSIM} = 0.24$. The other hyperparameters leave unchanged.

Fig.9 shows the result of our model without style matching loss. The relative positions among different part of a face is distorted in this setting. This kind of detailed shape information or so-called stylistic information is not natural compared to a real image, even if we could still recognize face from the generated images. In Fig.10(a), the nose of the generated cat does not lie in the middle of the cat face, and the mouth of generated image in (b)(c)(d) is kind of twisted. Because the weight of spatial matching loss is larger, so the location of the generated face is still the same with the true image.

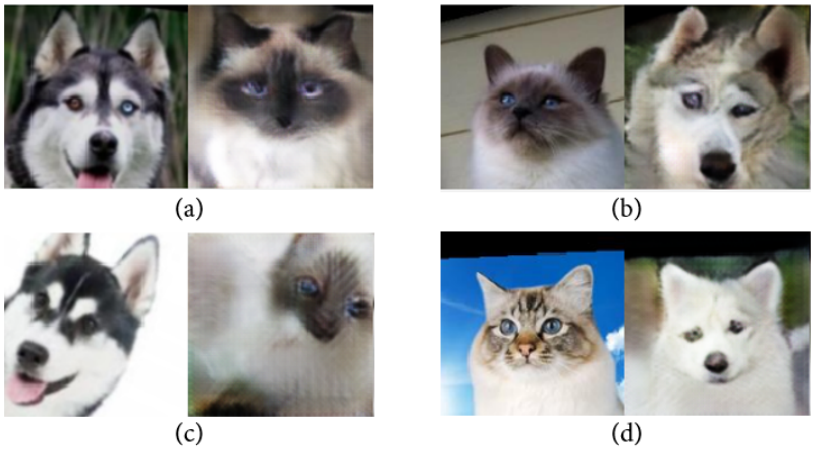

Figure 10: MS-SSIM + Style Matching
Left: True Right: Fake

In the second ablated setting, the functionality of spatial matching loss is examined by cutting out the weight of spatial matching loss in the total loss function. The weight originally belongs to spatial matching loss is allocated to style matching loss, so the weights are set as $\lambda_{adv} = 0.33, \lambda_{style} = 0.33, \lambda_{cyc-L1} = 0.1, \lambda_{cyc-SSIM} = 0.24$. The other hyperparameters also leave unchanged.

Since the feature maps extracted from deeper convolutional layers only contain the information of general location of the object, the location, orientation and posture of the original image could not be preserved after the spatial matching loss is eliminated. Fig.10 shows the result in this setting. In Fig.10(a), the location of the generated cat is higher than the original dog. (c) is a more representative example, where the location of the generated cat is totally uncorrelated with the true image. In (b) the original cat is looking to the left but the generated one seems to look to the left, and in (d), the posture of the cat is also not preserved. It is noticeable that all of the generated images could still preserve a good relative position of each element of the face, which is a side proof of the functionality of style loss.

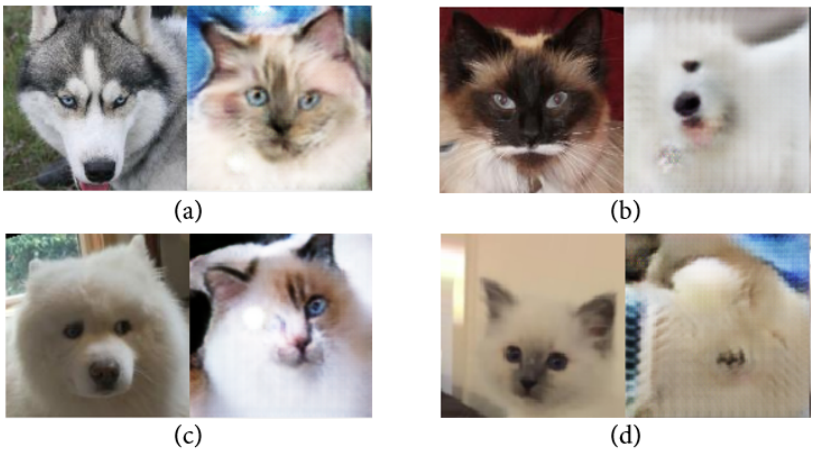

Figure 11: Spatial Matching + Style Matching
Left: True Right: Fake

In the last setting, we cut out the MS-SSIM loss to test its functionality and its compatibility with the two perceptual loss tested above. Since MS-SSIM loss is a reconstruction loss, so the weight is assigned to the L1 loss. Now, the reconstruction loss is degraded to the reconstruction loss function used in CycleGAN. The overall weights are set as $\lambda_{adv} = 0.33, \lambda_{spatial} = 0.07, \lambda_{style} = 0.26, \lambda_{cyc-L1} = 0.34$.

Fig.11 shows the result without the MS-SSIM loss. The generated face is twisted or could even hardly be recognized as a face because of the lack of some basic elements. Besides, without the MS-SSIM loss term, the perceptual loss can not work normally alone. It is not enough if we want to do the shape deformation on the object only with the L1 norm loss as reconstruction loss.

Based on the results of this ablation study, we can conclude the functionality of these 3 important loss terms in our model. The style matching loss enables the generated object to reproduce detailed stylistic features of a real object, like the relative positions among different parts of the face. The spatial matching loss can help the generated images to well preserve the location and orientation information of the object. The MS-SSIM is the most important one to ensure the basic structure of the object in the generated image.

3.5 Failure Results

There are also some failure results in our training process. As is shown in Fig.12, the model collapses after a few epochs. In this situation, there is only a single output image for each category. The reason for this collapse is shown in the training curve as is shown in Fig.13. Fig.13(a) shows the accuracy of the discriminator and (b) shows the loss of the generator. The accuracy of the discriminator is very high during the later epochs, the value is near 98% and hardly goes down, while the loss of generator is high with the trend even growing higher as the training process goes on. This is because the capacity of the discriminator is

much stronger than the generator and could almost distinguish every image generated by the generator from the true image. This will make the gradient of the generator extremely low and it is hard for the generator to get updated, a.k.a. gradient vanishing problem in training GANs.

Figure 12: Failure results
Left: True Right: Fake

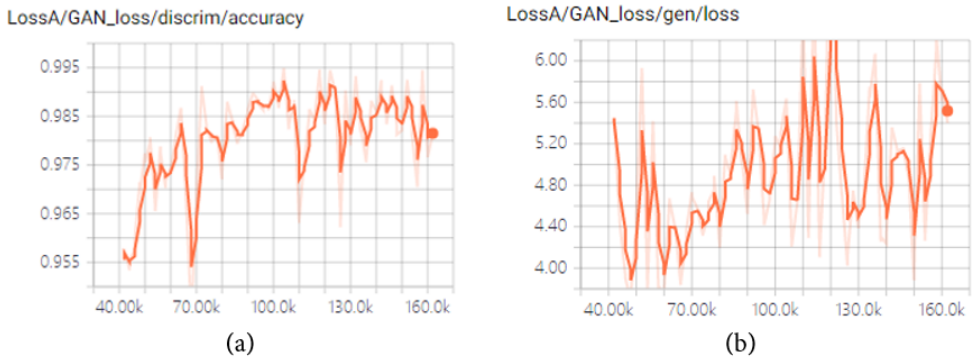

Figure 13: Training Curve

To solve this problem, several methods could be employed. First, we could adopt early stop methods to get the result earlier before the model collapse. Then, we could take steps to weaken the capacity of the discriminator. For example, we could reduce the channel numbers in the discriminator, and we could also adopt the WGAN model with the gradient penalty method to further improve our training process[15]. We could also add adaptive Gaussian noise to the input image to further confuse the discriminator according to the accuracy of discriminator[16]. Due to the limited time, we will leave this improvement to future work.

4 Conclusion

In our project, a new CycleGAN-based model is proposed and get satisfying results on the object transfiguration task with shape deformation. First, we find an image dataset

containing faces of cats and dogs and adopt the pipeline of CycleGAN and some novel ideas in GANimorph. After finding some drawbacks of GANimorph, the perceptual loss composed by spatial matching loss and style matching loss is introduced to further improve the deformation effect. The result of our model is compared with CycleGAN and GANimorph to show the advantage of our modification in this object transfiguration task. Finally, the functionality of each loss function is analyzed by the ablation study.

References

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

[2] 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*, pp. 2223–2232, 2017.

[3] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681–4690, 2017.

[4] 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*, pp. 1125–1134, 2017.

[5] K. Nazeri, E. Ng, and M. Ebrahimi, "Image colorization using generative adversarial networks," in *International conference on articulated motion and deformable objects*, pp. 85–94, Springer, 2018.

[6] Y. Cao, Z. Zhou, W. Zhang, and Y. Yu, "Unsupervised diverse colorization via generative adversarial networks," in *Joint European conference on machine learning and knowledge discovery in databases*, pp. 151–166, Springer, 2017.

[7] A. Gokaslan, V. Ramanujan, D. Ritchie, K. In Kim, and J. Tompkin, "Improving shape deformation in unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 649–665, 2018.

[8] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[9] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, and M.-H. Yang, "Drit++: Diverse image-to-image translation via disentangled representations," *International Journal of Computer Vision*, pp. 1–16, 2020.

[10] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[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*, pp. 2794–2802, 2017.

[12] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in neural information processing systems*, pp. 2234–2242, 2016.

[13] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*, pp. 694–711, Springer, 2016.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] I. Gulrajani, "Improved training of wasserstein gans," *Advances in neural information processing systems 2017*, 2014.

[16] T. Karras, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.