FACIAL-TO-ANIMAL IMAGE MORPHING WITH GAN

Sicheng Liu & Xiang He & Yihao Wang

SID: 520096356, 530205863, 540297164
UniKey: silu3452, xihe6010, ywan0206

ABSTRACT

Image morphing techniques can be used in many fields such as film-making, and computer game production. However, existing methods are designed based on computer graphics algorithms which require sophisticated manual setup. This project proposes two Generative Adversarial Networks (GAN) based method for image morphing. We test our method on Animal Faces-HQ and a Human Face Dataset. Experiments show that we got a competitive result with limited computing resources. The code and data of this project can be found at: https://github.com/SichengLeoLiu/COMP5405—Digital-Media-Computing/tree/bottleneck.

1. INTRODUCTION

In such a fast-moving digital media environment, image morphing has been a standout technique where the hard technology of art metamorphoses images through a seamless flow of visually narrated stories. This project is about how recent advances in deep learning and neural networks can be used to develop an application for human facial image morphing into animal forms. This is a technology that opens the lock of entertainment, mixing human characteristics with animal features to get commercial versatility in tools for design and marketing and to enhance the digital interactive experience.

This technical backbone of the project is very sophisticated image-processing techniques, mainly focusing on generative adversarial networks, and most recently, on diffusion models. Both of these allow the making of very realistic and lively images, keeping the essence of the original but adding something totally new and imagined. The vast power of GANs in image generation and processing has led to big development in synthetic media. When such networks are trained on big datasets of human and animal images, they create a lot of morphed images of realistic quality without falling into the uncanny valley, typical of digital creations of this sort.

This project moreover uses diffusion models. These belong to a more modern class of generative models that have been highly successful in generating high-quality images. Essentially, it learns how to reverse a diffusion process that starts from random noise and denoises it progressively to eventually form a coherent image. In our application, it will allow a smooth transition from the face of a human to the visage of an animal, yielding transformations that look much more fluid and natural as shown in Figure 1. Technically, this model is richer in art inasmuch as one can now witness one's features subtly change across the species barrier. This is all made possible by a two-tiered strategy in the morphing process: identification and image generation. The classification part is first, where the best match for a given human facial image from a pre-stored animal image library is done by an EfficientNet, as it is efficient and scalable. This all depends on the matching of the high-dimensional features pulled from the images, capturing the most important visual information without being disturbed by the excess data. Finally, once a match is identified, image generation takes place by using either GANs or diffusion models, based on the complexity of the features and quality of output one wants.

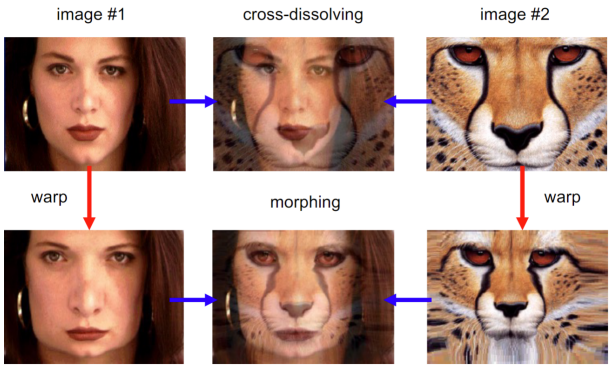

Fig. 1. Image Morphing [1]

These technical means are further backed up with strong machine learning algorithms and neural networks learning from many pictures. The system is configured to improve user interaction by allowing one to set the intensity of the morphing. Users can make adjustments on the amount by which animalistic features are integrated into the image of a person, which is control on user side regarding the morphing activity and the output itself. This is one important feature in professional applications related to design and marketing, where a brand identity may need to have a given set of animal traits to express a given attribute or message. The challenges include ensuring that computational demands are well managed with

the training of sophisticated models such as GANs and diffusion models; ensuring privacy and ethical considerations in using the facial image; and getting a good level of user satisfaction in terms of quality and realism of morphing. This is mitigated with good design, thorough testing, and development iterations that are consistently refining the algorithms and processing techniques. It will not just entertain but will affect digital marketing, social media, personal entertainment, and educational applications, in which such advanced image morphing can go a long way toward better user engagement and finding new ways to interact with digital content.

These technical means are further backed up with strong machine learning algorithms and neural networks learning from many pictures. The system is configured to improve user interaction by allowing one to set the intensity of the morphing. Users can make adjustments on the amount by which animalistic features are integrated into the image of a person, which is control on user side regarding the morphing activity and the output itself. This is one important feature in professional applications related to design and marketing, where a brand identity may need to have a given set of animal traits to express a given attribute or message. The challenges include ensuring that computational demands are well managed with the training of sophisticated models such as GANs and diffusion models; ensuring privacy and ethical considerations in using the facial image; and getting a good level of user satisfaction in terms of quality and realism of morphing. This is mitigated with good design, thorough testing, and development iterations that are consistently refining the algorithms and processing techniques. It will not just entertain but will affect digital marketing, social media, personal entertainment, and educational applications, in which such advanced image morphing can go a long way toward better user engagement and finding new ways to interact with digital content.

2. RELATED WORKS

2.1. GAN-based Image Generation

Generative Adversarial Nets (GAN) have been widely used and improved since it first proposed [2]. It can generate high-quality images by training generative and discriminative networks together. [3] proposed a cycle GAN which overcomes the drawback of needing paired data. Cycle-GAN uses a cycle consistency loss to train two generative models simultaneously and force them to learn how to convert distribution A to distribution B or vice versa. However, the generative images can not be controlled. [4] proposed Condition-GAN which can receive both noise and text as input, and generate the image under the control of input. Stytle-GAN, another controllable model, made many features in generative images being able to be controlled [5]. They split the generative model into several parts and did detailed experiments to evaluate which parts of the latent vector could control which features.

2.2. Image Morphing

Image deformation is an important research area in computer vision and graphics. Early works usually use computational graphics algorithms to manipulate the images. [6] shows that cross-dissolve, once used in image transition, can achieve good results in image morphing. However, traditional algorithms may produce a "ghost" effect. More recently, researchers have started to use deep learning based methods to tackle image morphing task. [7] improved StyleGAN by adding Transformer layer to make the model can be used on image morphing task. However, this method requires inversion for each image which can be time-consuming. [8] first apply Diffusion Model for image morphing by solving the interpolating problem in Diffusion Model. They use two LoRAs to capture the semantic meaning of two images and inteporate the parameters in LoRAs to combine the features in a smooth way. [9] map the image features into text embedding to use Diffusion Model to get smooth changes between images.

However, these methods require large computations, making it difficult to deploy on local machines. Our approach implemented image morphing through a lightweight method which not require much computational resources and can run on local machines.

3. MORPHING WITH CYCLEGAN

3.1. Problem Definition

Given two image dataset $S_X = \{x_1, x_2, \ldots, x_m\}$ and $S_Y = \{y_1, y_2, \ldots, y_n\}$, where m and n are the number of the images in dataset respectively. For any two images in S_X and S_Y, we want to generate the intermediate images with smooth transformation by the generative model, i.e. $G(x_i, y_j) = \{img_1, \ldots, img_i, \ldots, img_l\}$, where l is the total number of generated images we want.

3.2. Overall Structure

We designed our model based on CycleGAN [3]. The overall structure is shown in Fiure 2. The model contains two generators G_{XY} and G_{YX} and two discriminators D_X and D_Y. Generators are designed based on ResNet [10] which contain residual blocks to help parameter updating. We use the same discriminator that is used in PatchGAN [11].

Giving two image sets $S_X \sim \mathcal{X}$ and $S_Y \sim \mathcal{Y}$, for input image $x \in S_X$, the generator G_{XY} transfers it to a fake image and another generator trays to recover it and vice versa.

$$y_{fake} = G_{XY}(x) \tag{1}$$

$$x_{rec} = G_{YX}(y_{fake}) \tag{2}$$

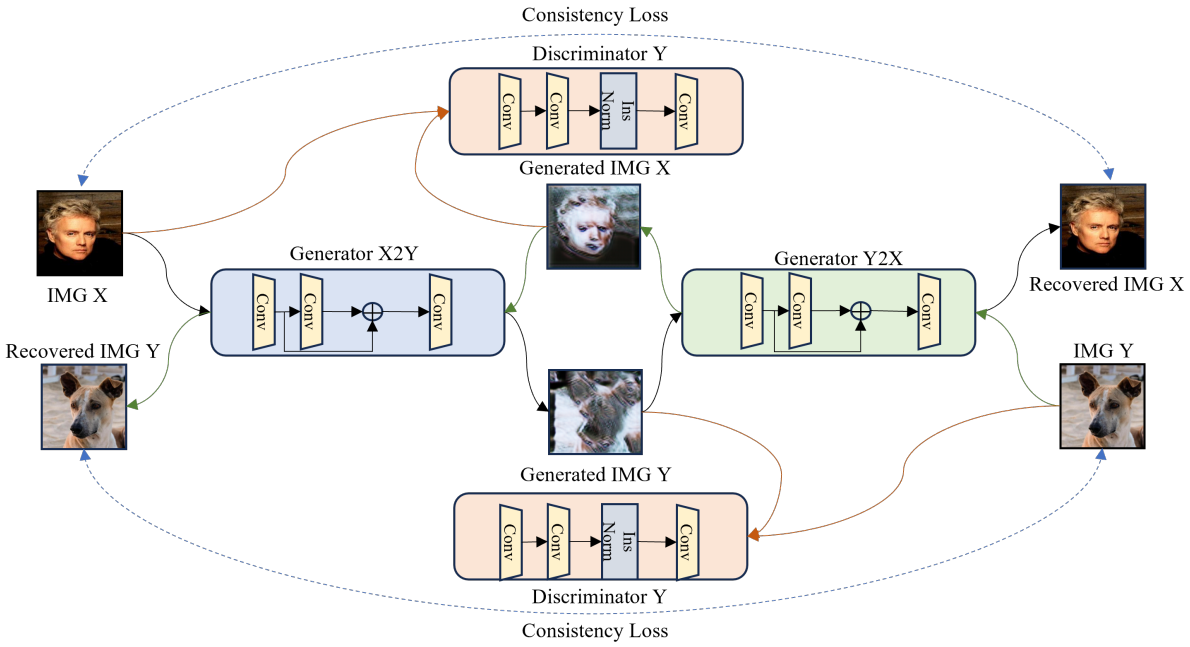

Fig. 2. Overall structure

We hope that the generated image y_{fake} obeys the $\mathcal{Y}$ distribution, and the recovered image x_{rec} back to $\mathcal{X}$ distribution.

The discriminator is used to predict whether an image is generated or not. For the generated $fake_B$, we got the predicted score by using D_Y.

3.3. Loss Function

3.3.1. Adversarial Loss

For generator G_{XY}, it wants to map images from distribution $\mathcal{X}$ to distribution $\mathcal{Y}$ and fool the discriminator D_Y. The model can be trained by the following loss function.

$$\mathcal{L}_{G_{XY}} = \text{MSE}(D_Y(y_{fake}), 1) \tag{3}$$

where MSE is Mean Square Error that is calculated by:

$$\text{MSE}(x, y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2 \tag{4}$$

Similarly, to updating G_{YX}, we have following:

$$\mathcal{L}_{G_{YX}} = \text{MSE}(D_X(x_{fake}), 1) \tag{5}$$

where x_{fake} is the fake image generated by G_{YX}

3.3.2. Cycle Consistency Loss

As CycleGAN does not require paired data, it adds consistency loss to ensure the main content of the input image will not be changed. The main idea is we want the generated image can be recovered by another generator. Therefore, both generators can learn the difference between two distributions in terms of style or features.

In this project, we use L1 loss to measure the difference between original images and recovered images. L1 calculates the difference of absolute value.

$$\mathcal{L}_X = \text{L1}(x, x_{rec}) \tag{6}$$
$$\mathcal{L}_Y = \text{L1}(y, y_{rec}) \tag{7}$$

In addition to directly comparing recovery image with input image, we add a perception loss to compare the high-level features. Specifically, we use a pre-trained ResNet to extract the features from both input image and recovery image. We then calculate the difference between feature embedding. The perception is calculated as follow:

$$\mathcal{L}_{P_X} = \text{L1}(ResNet(x), ResNet(x_{rec})) \tag{8}$$
$$\mathcal{L}_{P_Y} = \text{L1}(ResNet(y), ResNet(y_{rec})) \tag{9}$$

The overall loss for generators can be calculated as:

$$\mathcal{L}_G = \mathcal{L}_{G_{YX}} + \mathcal{L}_{G_{XY}} + \mathcal{L}_X + \mathcal{L}_Y + \mathcal{L}_{P_X} + \mathcal{L}_{P_Y} \tag{10}$$

3.3.3. Discriminator Loss

The discriminators also need to learn together with the generator. Discriminators aim to distinguish whether the image is generated. In this project, the discriminators do not directly

estimate the authenticity of images but rather the authenticity of each image patch. Therefore, we use MSE to train the model. The loss can be calculated as follows:

$$\mathcal{L}_{D_X} = \text{MSE}(D_X(x), 1) + \text{MSE}(D_X(x_{fake}), 0) \quad (11)$$

$$\mathcal{L}_{D_Y} = \text{MSE}(D_Y(y), 1) + \text{MSE}(D_Y(y_{fake}), 0) \quad (12)$$

3.4. Interpolation

The whole inference process is shown in Figure 3. We use linear interpolation between two input images to get 10 images containing information from both images as the model input. Then we use the trained generative to generate images with smooth transformation.

Model input can be calculated by Equation 13.

$$input_i = (i/10) * img_x + (1 - i/10) * img_y \quad (13)$$

Then we got output images by using G_{XY}.

$$img_i = G_{XY}(input_i) \quad (14)$$

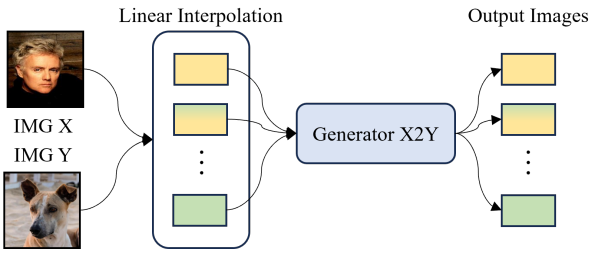

Fig. 3. Inference Process

4. MORPHING WITH BOTTLENECK

We also propose a new method for image morphing. Since GAN can generate images with noise input, we try to directly use the generator in GAN with a bottleneck structure for the morphing task. Specifically, the generator contains a down-sampling block and an up-sampling block. Down-sampling block tries to capture the features in the input image. After down-sampling, the size of feature embedding becomes smaller than the input image size and we call it the bottleneck feature. It can force the model to learn the features in the input image rather than directly memorize it. We use an up-sampling block to generate the image based on the bottleneck features. We apply a MSE loss to help the model learn how to recover the image from bottleneck features. The overall structure of the model is shown in Figure 4.

We combine two datasets S_X and S_Y into one dataset S. Therefore, during training, the model can learn to generate images in both domains.

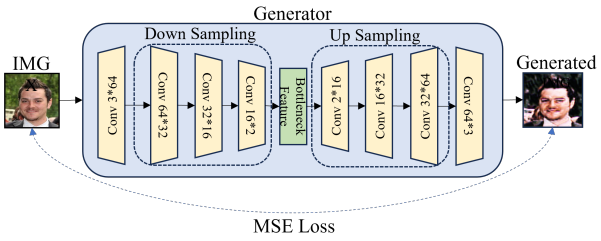

Fig. 4. Generator with bottleneck

5. EXPERIMENT

5.1. Evaluation Metric

To evaluate the quality of generated images, we use two evaluation metrics.

Fréchet Inception Distance (FID) calculates the distance between the features of generated images and ground truth. We use an inception network to extract features from both generated images and ground truth. FID can be calculated by Equation 15.

$$FID(x, g) = ||\mu_x - \mu_g||_2^2 + Tr(\Sigma_x + \Sigma_g - 2((\Sigma_x \Sigma_g)^{\frac{1}{2}}) \quad (15)$$

where μ_x and μ_g are the mean value of the features, and Σ_x and Σ_g are the covariance of the features. Tr calculates the trace of the matrix. FID measure the quality of the generated image without interpolation.

Perceptual Path Length (PPL) can be used to evaluate the consistency of image morphing. We calculate the perceptual loss of two neighbouring generated images and sum them up to get the PPL.

5.2. Results Analysis

We test two methods (CycleGAN and Bottleneck) on the test set. The experiment results show in Table 1. We can see that using generator with bottleneck can have even better result in terms of FID. Since the distribution of human face images and dog face images are largely different, it is difficult for CycleGAN to learn transferring between two distributions. It is not necessary for the generator with bottleneck to learn transferring. It only learn how to recover the images. Therefore, it generate images with higher quality easier. However, this training method can not ensure the image can morph smoothly because the uncertainty is only caused by limitation of features. The results of generator with bottleneck are more like the result after linear interpolation leading to a higher PPL.

We also measure the training time for each method. As CycleGAN needs to train two generators and two discriminators at the same time, it requires much more time to train. In experiment, we train it for 30 epochs and it takes 5 hours. However, training a single generator is much simpler. We

Method	FID ↓	PPL ↓	Training Time(h) ↓
CycleGAN	292	0.0096	5
BottltNeck	227	0.0161	0.25

Table 1. Experiment result

train it for 5 epochs to achieve the best results. Total training time is 15 minutes.

We visualize the morphing results with both methods in Figure 5. We can see that images generated using CycleGAN will be distorted and the result of bottleneck generator have higher resolution. In addition, images generated by CycleGAN can have smoother deformations. This is consistent with previous measurements of FID and PPL.

6. CONCLUSION AND LIMINATION

This project design two light-weight methods for image morphing. It can achieve competitive results under the condition of limited computing resources. We further analyze the effectiveness of two methods and discuss their advantages and disadvantages. However, some shortcomings remain in this project.

- The generated image loses color information, resulting in a large difference from the input image.
- The generated intermediate image still suffer from "ghost effects".

We will continue to address these issues in the future.

7. REFERENCES

[1] Jonas Gomes and Luiz Velho, *Image processing for computer graphics*, Springer Science & Business Media, 1997.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[3] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei 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] Mehdi Mirza and Simon Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[5] Tero Karras, Samuli Laine, and Timo Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[6] George Wolberg, "Recent advances in image morphing," *Proceedings of CG International'96*, pp. 64–71, 1996.

[7] Na Zhang, Xudong Liu, Xin Li, and Guo-Jun Qi, "Morphganformer: Transformer-based face morphing and demorphing," *arXiv preprint arXiv:2302.09404*, 2023.

[8] Kaiwen Zhang, Yifan Zhou, Xudong Xu, Xingang Pan, and Bo Dai, "Diffmorpher: Unleashing the capability of diffusion models for image morphing," *arXiv preprint arXiv:2312.07409*, 2023.

[9] Zhaoyuan Yang, Zhengyang Yu, Zhiwei Xu, Jaskirat Singh, Jing Zhang, Dylan Campbell, Peter Tu, and Richard Hartley, "Impus: Image morphing with perceptually-uniform sampling using diffusion models," in *The Twelfth International Conference on Learning Representations*, 2023.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] Ugur Demir and Gozde Unal, "Patch-based image inpainting with generative adversarial networks," *arXiv preprint arXiv:1803.07422*, 2018.

(a) Result of CycleGAN

(b) Result of Bottleneck Generator

Fig. 5. Visualization of morphing