Fuzzy Systems and Data Mining VI
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA200707

Realistic Face Image Generation System Based on GANs

Han Zhike, Yang Bin [1], Du Yiren, Du Xingyu, Xing Hao and Cheng Qinnan
Zhejiang University City College, Hangzhou, Zhejiang, 310015, China

Abstract. The purpose of this paper is to study the help of generative adversarial networks (GAN) for face generation, and to explore whether the network can have an effect on complex face generation. Training an image translation neural network model based on a generative adversarial network with the help of a large number of real human face data sets. Using the CV2-based face tagging algorithm and the HED-based face edge extraction algorithm to obtain input information, and then based on the translation neural network model Developing a face generation system through Tensorflow, Torch and other frameworks to realize the function of generating real faces through sketches or "changing faces" through existing faces. Finally, this model provides training configuration and training information.

Keywords. Generative Adversarial Network, Face Generation, Face Tag

1. Introduction

Oral memory portrait method is that the investigators describe the characteristics of the criminals or suspects stated by the victims during the detection of the case, and paint the facial portraits of the criminal suspects with a brush. Moreover, the generative adversarial networks-based image generation application can improve the authenticity and Accelerating efficiency can increase the similarity and narrow the criminal search. In addition, a deeper application can achieve a simple textual description of the eyewitness and then depict the face. The past depiction method is obviously not high enough in terms of time efficiency. No matter how powerful a painter can paint a work, it must be faster than a computer. Using technology to describe a face has a great advantage in terms of timeliness. The precise computer resolution band. The accuracy is also self-evident. Solving a case seeks speed. One minute and one second can affect the rate of arrest of criminals. Therefore, this technology can greatly improve the rate and efficiency of criminal investigation. This topic can be extended to the generation of images, such as buildings and animals. With the support of the data set, the demand for generating real pictures by giving sketch outlines can be realized.

In image processing, many problems in computer graphics and computer vision that require image processing can be regarded as "translating" the input image into the corresponding output image. We define graph and graph transformation as the problem of translating a possible scene expression into another and give enough training data. A major difficulty in language translation is that the mapping between languages is rarely

[1] Corresponding Author: Yang Bin, Zhejiang University City College, Hangzhou, Zhejiang, 310015, China; E-mail: sevenyoung7@foxmail.com.

one-to-one. Similarly, the difficulty of most image translations is many-to-one; such as computer vision: images and edges, semantics, semantic labels or one-to-many of computer graphics: labels or the mapping of all user input to real images. These tasks are traditionally separated item by item and processed purposefully, despite the fact that the processing methods are all predicted pixel by pixel. In order to solve this problem, it is proposed to use generative adversarial networks model to effectively apply to this field [1]. Domestic research has proposed a face reduction algorithm based on generative adversarial networks. This algorithm adds a part of supervised learning on the basis of unsupervised learning, called semi-supervised learning (GANs + CNN) [2]. Reference is made to the previously proposed semi-supervised learning ideology [3], and reference to the co-adapted fixed learning rate [4] is used to set the learning rate.

This paper aims to realize the translation of the input contour through a machine learning translation model into a realistic face image. The basic principle is to use generative adversarial networks to achieve machine learning capabilities and meet the conditions for real face generation. Moreover, through the pix2pix translation framework to achieve image translation, the training set required for model training and prediction is processed by the CelebA face database. For users, only the simple operations of mobile terminal or PC terminal for registration; and login, uploading and input can be used to generate the real and touching faces needed.

2. The Problem and Proposed Solution

2.1. Symbol and term definition

Image Domain: The image domain is more inclined to the attributes of the image content, such as the spatial domain of the image, also known as the image space, which is essentially a space composed of image pixels. The variable takes the space as the variable, the horizontal axis is the x axis, and the vertical axis is the x axis. The axis is the y axis.

Discriminant Function: The purpose of setting the concept of discrimination function is to unify the judgment conditions of classification.

Canny Algorithm: The Canny edge detection operator is a multi-level edge detection algorithm developed by John F. Canny in 1986. The edge detection of the traditional canny operator is divided into five steps. First, the Gaussian filter is used to reduce the noise and make the picture smoother. Then the gradient intensity and direction of a given pixel are calculated, and then the obtained data is used. Use non-maximum suppression to eliminate spurious responses, use dual thresholds to detect edges, and finally complete the detection process by suppressing weak edges.

Labeling Method: On the premise of having a face contour, the translation model needs to mark the image features. This algorithm for marking the features of a face image is called a face marking algorithm.

The Table 1 is the relevant symbol definition of GAN. And the Table 2 is the symbol definition of the input and output of the pix2pix framework.

Table 1. The relevant symbol definition of GAN.

real data	data	real data	data
real data distribution	p_{data}	generator output	p_g

real data	data	real data	data
input data (noise)	z	generate mapping	G ()
distribution of raw noise	p_z	discriminant mapping	D ()

Table 2. The symbol definition of the input and output of the pix2pix framework.

symbol	significance	symbol	significance
G	generator	z	random vector
D	discriminator	D_k^n	Discriminator features of layer n
x	input map	M	total floors
y	output graph	N_n	Number of elements in layer n

2.2. The problem

Area of Research:

Training set: Training a model that generates high-definition images will inevitably require a suitable training set. The training set requires high resolution and a sufficient number.

Edge extraction algorithm: Since the contour of the face needs to be used as both semantic identification and mapping input, it is particularly important to choose an appropriate algorithm to meet these two requirements.

Translation model: In image processing, computer graphics and computer vision, many problems that require image processing can be regarded as "translating" the input image into the corresponding output image. Therefore, a reasonable translation model needs to be selected for training.

System development: After the algorithm design and development are completed, due to the needs of the system user experience. A complete photorealistic image generation system that is convenient for users and meets the basic functions of the system needs to be developed.

Research difficulties:

(1): Training set: using only the open source CelebA face database provided by the Chinese University of Hong Kong is not enough to meet the needs of the project. It needs to be further developed to meet the needs of the project. The specific difficulty is that the pictures in the face library contain Other human parts that are not needed for the subject, thus a large number of pictures need to be edited. In addition, not all the pictures in CelebA meet the requirements of high definition, hence the resolution needs to be adjusted to avoid blurring in the details of the generated image.

(2): Edge extraction: Canny edge extraction is an operator that uses local extreme values to extract edges [5]. The implementation process of the Canny edge detection algorithm mainly includes the following processes: first, Gaussian filtering is used to denoise the image; then the finite difference of the first-order partial derivative is used to calculate the amplitude and direction of the gradient; and finally, it is completed by double thresholds. The algorithm extracts and connects the edges for the first time to obtain the final edge extraction results[6,7,8].The effect of edge extraction using the general Canny algorithm is not very good. It has a problem of many noises and cannot be used as a semantic mark. It needs to be improved to obtain a simpler outline. The idea should focus on semantic segmentation instead of edge extraction. Because after extraction, the obtained edges are input into the model as corpus. An edge extraction model that better meets the needs of the subject is needed.

(3): Translation model: When generating high-resolution images, the effect of the pix2pix framework is not good. For example, when assigning high-resolution input, it is difficult for the network to generate high-resolution output. This shortcoming is reflected in the generation. The details of the picture are very vague. The feeling is not big enough. Therefore, a deeper network or a larger convolution kernel is needed to obtain a larger feeling field. However, simply increasing the depth or breadth will result in overfitting and requires more GPU memory for training.

(4): System development: Since the user group prefers to use the mobile terminal, cross-platform responsive development is required and a large number of adaptations are needed to ensure that the mobile terminal can also obtain a good user experience when accessing the browser.

According to the research difficulties, the main problems we need to solve are: (1) edit the training set, extract key parts and high-definition images; (2) use the edge extraction technology of the appropriate subject; (3) improve the translation model to ensure that it is suitable for high-resolution input and produces high-resolution output; and (4) the overall structure of the website is responsive layout, modular component development and separation of front and back ends.

2.3. GAN and image translation

GAN is called Generative Adversarial Networks, that generates adversarial networks. It was first proposed by Lan J. Goodfelloe et al. in the paper published in October 2014 [9]. This paper proposes a new framework for estimating generative models through an adversarial process, in which two models are trained simultaneously: a generative model G that captures the distribution of data and a discriminant model D that estimates the probability of samples coming from the training data.

Generative adversarial networks are mainly composed of two parts, namely generator and discriminator. The task of the generator is to learn the distribution of the real image to make the generated image more real, and the real image learned through the discriminator test. The task of the discriminator is to judge whether the received picture from the generator is true or false. During the entire training process, the generator continuously makes the generated image more real and the discriminator continuously discriminates whether the image is true or false. This process is similar to a two-player game. During the training, the generator and discriminator continue to confront each other, and finally the two networks approach dynamic equilibrium: the image generated by the generator is close to the real image distribution, and the discriminator could not judge the true and false of the generated image. The probability of being true is infinitely close to 0.5.

real data	data
real data distribution	p_{data}
input data (noise)	z
distribution of raw noise	p_z
generator output	p_g
generate mapping	G()
discriminant mapping	D()

Figure 1. GAN symbol definition.

G represents a generator, which is a multi-layer perceptron. The parameters of the perceptron are θ_g. The generating function G represents mapping noise to the data space, such as G(z;θ_g). This means that the generating mapping function maps noise to the corresponding data space, the same as Just like the discriminant function D. As well, it is considered as a multi-layer perceptron with parameters, such as D(χ;θ_d), which outputs a scalar indicating the probability that the input is real data or the probability of generating data.

$$\min_G \max_D V(D,G) = E_{\chi \sim p_{data}(\chi)}[\log D(\chi)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]. \tag{1}$$

The decoding part consists of another three convolutional layers. However, in this part, it converts the feature vector obtained from the encoding part to restore them to the input size, then we can get the predicted wind speed of each pixel.

The purpose of the discriminator D is to make it possible to distinguish whether the input is a real sample or a generated sample as much as possible, so the goal is to maximize the value of D(x) and minimize the value of D(G(z)), which means Maximize V(D,G). For generator G, the goal is that the discriminator cannot distinguish, that is, the value of D(G(z)) needs to be maximized, that is, V(D,G) is minimized. Through the confrontation of the two models, the final get a global optimal solution. For generator G, the goal is that the discriminator cannot distinguish, that is, the value of D(G(z)) needs to be maximized, that is, V(D,G) is minimized. Through the confrontation of the two models, the final get a global optimal solution.

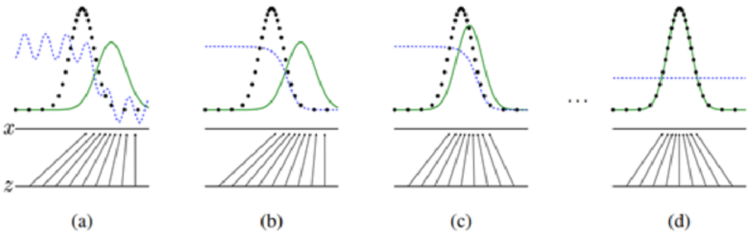


Figure 2. Training process [10].

Figure 2 shows the optimization process of the entire confrontation. The black curve corresponds to the true distribution of the samples, while the green curve corresponds to the fake samples generated by the generator. Moreover, the blue curve is the output of the discriminator's judgment on the generated samples. With the improvement of the detector, the distribution of the generated samples is closer to the real distribution. As well, with the help of the output of the discriminator, the generator is further improved. Therefore, the generated distribution and the real distribution almost overlap. At this time, the discriminator cannot make a judgment, and the output is close to 0.5, reached optimization.

When performing the optimized representation, it needs to be divided into two steps. First, the judger determines whether the input sample is a real sample. The logarithm function belonging to the positive class is in the distribution of p_{data} (χ), so that the discriminator is in Obtained p_{data} distribution can accurately obtain D(χ)=1, which is the front of the formula:

$$E_{\chi \sim p_{data}(\chi)}[log\, D\, (\chi)] \tag{2}$$

The second step is the discriminator to determine whether the input sample is a generated sample, which is a logarithmic function of the negative class, that is, the rear part of the formula:

$$E_{z \sim p_z(z)}[log(1 - D(G(z)))] \tag{3}$$

The discriminator needs to maximize the formula. At this time, it is best to judge whether the sample is generated or the real sample. For the generator, it needs to be minimized. The distribution generated at this time is closest to the actual distribution. The entire iterative training process maximizes V(D, G) after the generator outputs, obtains the corresponding D, uses the obtained D and minimizes V(D, G) to obtain a new G. This process can be expressed as:

$$G^* = argmin_G\, V\, (G, D_G^*) \tag{4}$$

When referring to image translation (i.e., image-to-image translation), it is inevitable to first mention the basic encoder-decoder translation model in natural language processing. This model is a description of the framework of a class of models, such as common applications in Chinese and English machines. The most significant feature of translation is the end-to-end learning method of the model, which is mainly two processes of encoding and decoding. This process is required in both Chinese and English translation. The encoding converts the input sequence to a length. The fixed vector and decoding will turn the generated length vector into an output sequence. Such a model is also called seq2seq model, which vividly describes the string sequence that is expected to get semantic translation through a given string sequence.

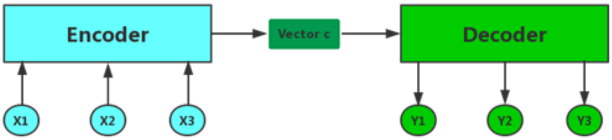


Figure 3. Translation framework.

As shown in Figure 3, the input character sequence X1, X2, X3 is converted into a vector c of fixed length by the encoder, and the decoder converts the vector into a new sequence Y1, Y2, Y3. The most common neural network used by this model is RNN. The first and second two neural networks serve as encoder and decoder respectively.

Through joint training of these two RNN networks, the input sequence can be the conditional probability is maximized and the output sequence is obtained. As a kind of framework rather than a fixed model, decoding and encoding cannot only deal with text, but more widely such as voice, image, video, etc. The same neural network can be CNN or RNN and many more.

On the basis of Encoder-Decoder, it can naturally be applied to the field of images, but there are still great differences between pictures and text. The first is the two different concepts of image content and image domain. Image content can be understood as the semantics of the corresponding text. Although there is still a big difference between the two.

Next is the concept of the image domain. The image domain is more inclined to the attributes of the image content, such as the image spatial domain, also known as the image space, which is essentially a space composed of image pixels. The variable takes the left side of the space as a variable, sitting is taken as the coordinate origin, the horizontal axis is the x axis and the vertical axis is the y axis, as shown in Figure 4.

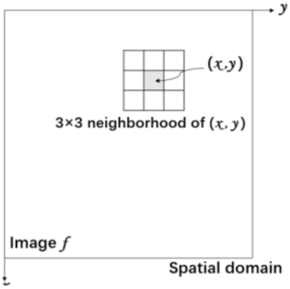


Figure 4. Image domain.

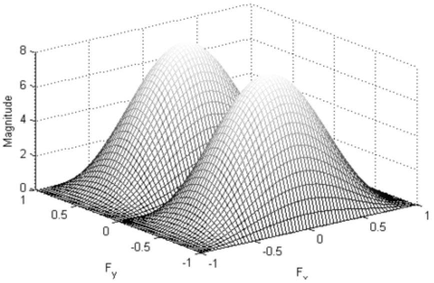


Figure 5. Frequency domain.

The frequency domain is the frequency structure of the signal and the relationship between the frequency and the amplitude of the frequency signal. It often expresses the severity of the grayscale change in the image, that is, the index of the grayscale spatial gradient. For the image, since the edge of the image is abrupt Yes, it belongs to a faster change. Therefore, it appears as a high-frequency component in the frequency domain. As well, the noise in the image represent a high-frequency component in most cases.

Therefore, under the above-mentioned differences, image translation can be understood differently. The first understanding is the translation of the semantics of pictures. The understanding of images is to replace the content of the images, and the specific ones should transform semantic texts into corresponding semantically correct pictures.

In more cases, image translation is understood as the translation of the image domain. One domain α of the image is transformed into another domain β, that is, the original attribute χ of a domain α of the image is removed, and a new attribute δ is given to the domain, namely in the case of $[Y\,,\chi] \subset \alpha$ and $[Y, \delta] \subset \beta$, a transformation f is obtained. The goal of the transformation f is:

$$f([Y\,,\chi]) = [Y, \delta] \tag{5}$$

Due to the fierceness of generative adversarial networks, a large number of generative adversarial networks-based image translation methods have appeared, such as pix2pix:

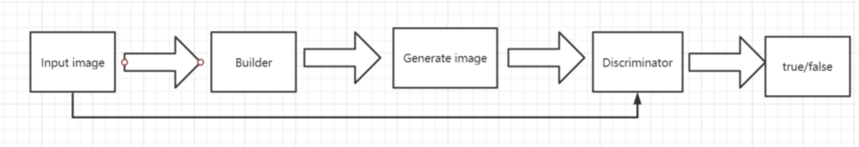


Figure 6. Translation framework pix2pix.

Compared with generative adversarial networks[11], the input to the generator is no longer noise, but an image, which is more biased towards the translation of image semantics. More striking is the improvement of the traditional encoder decoder framework by this model, which uses U-Net [12].

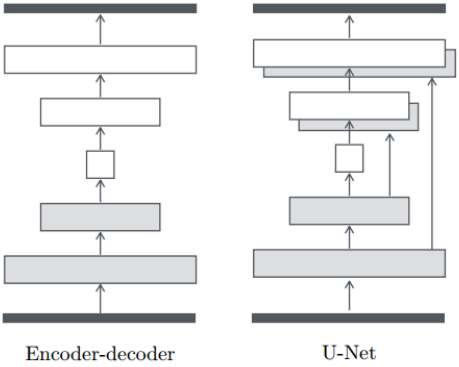


Figure 7. Neural Network U-Net[12].

Compared with the ordinary Encoder-decoder, U-Net in the decoder connects the output before the conv layer with the corresponding mirror layer in the decoder. Through this move, doubling the number of channels will not affect The output dimension of the convolution.

Another translation model, CycleGAN, prefers the conversion of the image domain. The purpose is to convert the images of the two domains with each other, which is different from the traditional generative adversarial networks one-way propagation. As shown in Figure 7, the neural network model is ring-shaped. Two mirror-symmetric generative adversarial networks construct a ring. Each of them has a discriminator. As well, a generator is used to obtain a consistent loss through the loop.

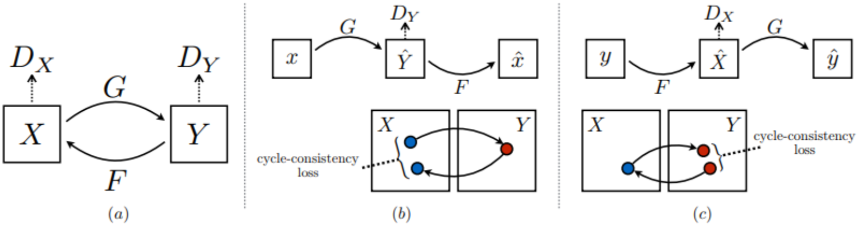


Figure 8. Translation model CycleGAN[11].

3. Experiment and Result

3.1. Face Generation

For user input, two methods are provided. The first is to manually input the contour. Due to the consideration of the two needs of the PC and the mobile, the user needs to implement two methods of manual input and camera shooting, and the manual input method is selected. The way of drawing board, the camera is realized by the front end of the web page.

Canvas is a tag in html that uses scripts to draw graphics, and the specific idea of hand-painting is to use javascript to monitor the mouse. Through the monitoring of mouse events, use the drawcircle method to draw the recorded image. First initialize the canvas, set the function to the brush state, record and display the pressed mouse point, then stop recording when released. For the problem of moving too fast, connect the points to record the line segment. At the back end, cv2 is used to flip the color to meet the needs of the facial feature identification of the translation framework. Based on the principle of image gray value, white to black are divided into multiple gray levels in the form of logarithm. Generally, black is defined as 0 and white is defined as 255. By flipping the gray value of the pixel, the color is flipped. Adjust the resolution of the input image to 1024*1024 pixels that meet the translation framework. Use the dlib face recognition library to perform face recognition on the image, in order to ensure that the input picture meets the facial features and avoids incorrect input.

For the input of a real human face, the file upload function is selected on the pc side. For the mobile side, the camera of the mobile device is first turned on, and the image recorded on the camera is converted into a file and uploaded to the server. Use the dlib face recognition library to perform face recognition on the image to ensure that the input picture meets the facial features and avoids incorrect input. Due to the different resolutions of the uploaded pictures of different cameras, first find the face, intercept the picture, and adjust the resolution of the input image after the interception. According to the original length and width of the image, adjust the length and width to meet the translation frame of 1024 * 1024 pixels.

3.2. Face Tag

Due to the training of machine learning models, a large amount of data is required to form a data set, and the corresponding face translation requires a large number of real faces as a training set. Since the human face has a clear distribution of features, facial features and other features, it is necessary to mark facial features and facial features. Moreover, the edge extraction of human face is divided into two parts: data set and extraction algorithm.

The original data set comes from the celebA face database [13] of the Chinese University of Hong Kong. The face database is open data and contains 202,599 images from 10,177 people at home and abroad. However, these data cannot be directly applied, and preliminary data cleaning is required, as well as the resolution is unified for training. The first step is to perform face detection through the dlib library, determine the position of the face, intercept the picture in proportion, and then transfer it to a 1024*1024 pixel high-definition picture to save it.

Figure 9. Original celebA figure.

Figure 10. HD picture after processing.

After processing, 30,000 faces are obtained as the data set, which is used as the input data of the image translation model.

After having enough initial data, we extract the outline of the picture and initially we use the edge detection of the traditional canny operator, which is divided into five steps. First, the Gaussian filter is used to reduce the noise and make the picture smoother. Next, we calculate the gradient intensity and direction of each pixel. Then we use the obtained data to eliminate the spurious response using non-maximum suppression, we use double thresholds to detect edges, and finally we complete the detection process by suppressing weak edges.

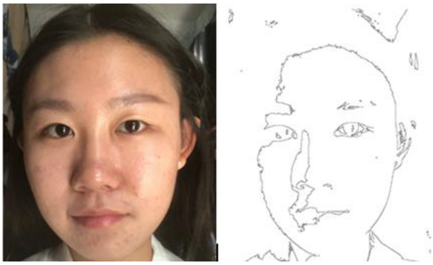


Figure 11. Use canny operator to extract results.

Because the result of using the canny operator [14] is very bad, the training effect is also very poor. Therefore, a lot of noise and discontinuous edges [15] cannot get the ideal results, thus another algorithm is used for edge extraction. The HED [16] network model is based on the structure of the VGG16 network model [17]. The core idea is to use the pooling layer to reduce the constant width of the input image layer by layer at a rate of 1/2, which is a multi-scale. The hierarchical network structure uses CNN to carry out end-to-end edge detection [18]. Based on this algorithm, edge extraction is performed after being reproduced in the Caffe deep learning framework.

As shown in Figure 12, the side back put of the convolution layer adds an output layer to the edge output layer, and depth supervision is performed on the edge output layer to make the generated result close to the edge extraction. With this process, the side output layer gradually becomes smaller, and as the side output layer becomes smaller, the feeling gradually becomes larger. At the end of this process, a weighted fusion algorithm is used to convolve the layer to obtain outputs at different scales. At the same time, it is assumed that there are M-layer edge output layers, each layer is accompanied by a classifier, and the corresponding weight is w, and the objective function is set as:

$$L_{side}(W,w) = \sum_{m=1}^{M} \alpha_m l_{side}^{(m)}(W, w^{(m)}). \tag{6}$$

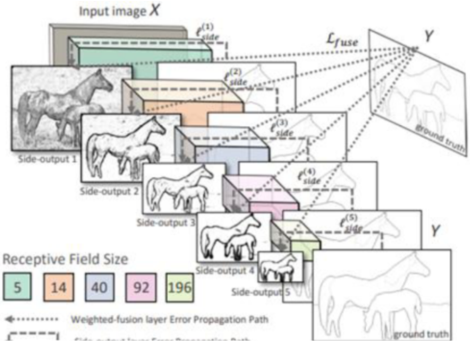


Figure 12. Schematic diagram of network structure.

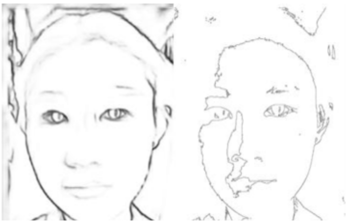


Figure 13. Schematic diagram of network structure.

As shown in Figure 13, compared with the canny operator, a great improvement has been achieved, with extremely continuous edges and less noise.

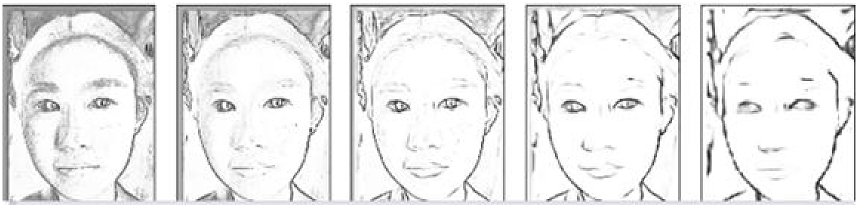


Figure 14. Extraction process using HED algorithm.

The entire extraction process is represented in Figure 14, and different results obtained by performing different edge detections according to different side outputs as output.

Since the images processed by celebA after preliminary processing did not carry out face annotation, and under the condition that the feasibility of real-life annotation is not large and the annotation information does not meet the requirements of the translation model, the face marking algorithm is designed.

In addition to providing a mature face detection and labeling interface, Dlib provides a trained model for face detection and labeling. The model is used to mark the face contours according to 1-17, 18-22 to mark the left eyebrow, 23-27 to mark Right eyebrow, 28-36 nose, 37-42 left eye, 43-48 right eye, 49-60 lip contour, 61-65 upper lip and 66-68 to mark lower lip. A total of 68 points were marked on the face contour and facial features.

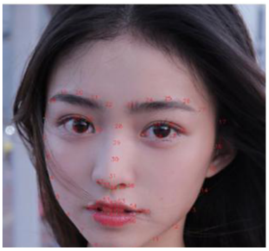


Figure 15. 68 points annotation of face

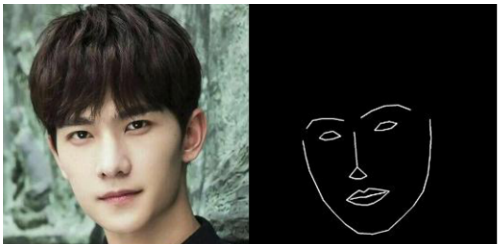


Figure 16. Face feature labeling based on cv2.

Under the premise of having a face contour, the translation model needs to mark the image and the requirements cannot be met only in the form of feature points.

Continuous lines are required to mark the face contour and complete image area for facial features. 68 mark points for further processing. Use cv2 to connect the marked points, and connect the outline, left eyebrow, right eyebrow, left eye, right eye, nose, upper lip and lower lip to obtain the labeled image, as shown in Figure 16.

Due to this labeling method, the distinction between left and right eyebrows, left and right eyes, between eye color patches, lips and nose color patches, is lacking Therefore, the algorithm is improved by first labeling different parts with different colors, and then eye, nose, upper and lower lips fill in the color blocks to get a new marking method.

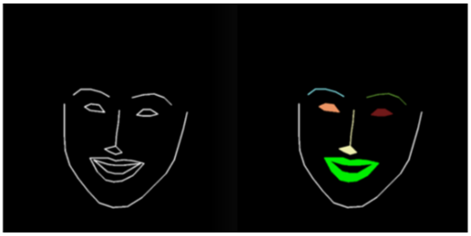


Figure 17. facial feature tags before and after improvement

symbol	significance
G	generator
D	discriminator
x	input map
y	output graph
z	random vector

Figure 18. Input and output instructions.

3.3. Face Generation Model Training

The image translation framework uses the pix2pix framework, first defining the input and output.

The output is undoubtedly the image output generated by the generator and the input is not a simple random vector, but a picture. In order to ensure the matching of the output, the discriminator needs some generator characteristics, thus the model loss function is in the generative adversarial networks loss function on the basis of [14] adjusted to:

$$L_{cGAN(G,D)=E_{x,y}[log\,D(x,y)]+E_{x,y}[log\,(1-D(G(x,z))]}. \tag{7}$$

In order to meet the needs of comparison, generative adversarial networks that only judges the real image is needed, and the loss function of the generative adversarial networks is:

$$L_{GAN(G,D)=E_y[log\,D(y)]+E_{x,z}[log(1-D(G(x,z))]}. \tag{8}$$

For the task of face translation, the input and output of the generator actually share feature information and the side information will affect the similarity. Therefore, the L1 paradigm Loss is added to the original generative adversarial networks loss, in order to ensure the similarity.

$$L_{L1}(G) = E_{x,y,z}[||y - G(x,z)||_1]. \tag{9}$$

In consequence, the aggregate loss function of this model is adjusted to:

$$G^{*} = arg\,\underset{D}{min}\,\underset{G}{max} L_{cGAN}(G,D) + \lambda L_{L1}(G). \tag{10}$$

For contour translation, the resolution of the input contour is adapted, and the noise is added to the input image and then generated by the generator in the trained model. The entire process is represented in Figure 19.

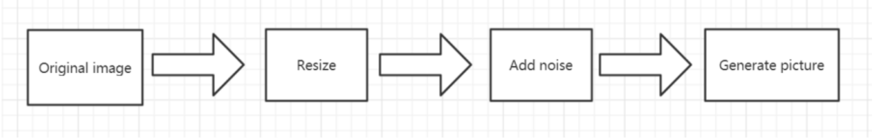


Figure 19. Contour translation process.

Under the premise of satisfying the face identification, there is a better generation situation, as shown in Figure 20.

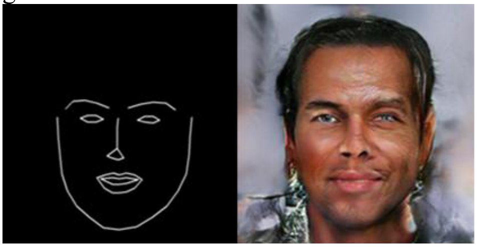


Figure 20. Contour translation.

For image translation, the resolution of the input contour is adapted. As well, after extracting the face mark, the original image and the mark are input into the generator together, the original image is used as the role of noise, and then generated by the generator in the trained model. The entire process is represented in Figure 21.

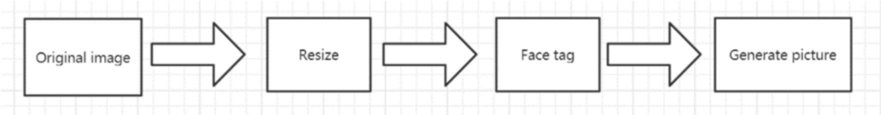


Figure 21. Image translation process.

Under the premise of satisfying the face identification, there is a better generation situation, as shown in Figure 22.

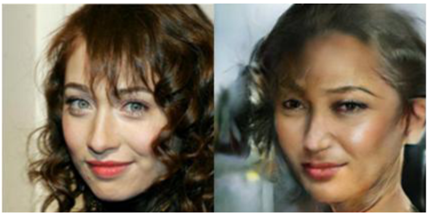


Figure 22. Image translation to generate images.

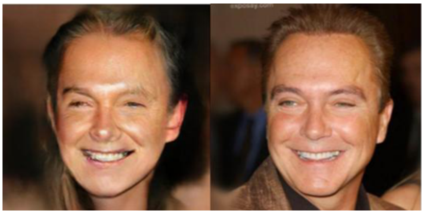


Figure 23. Generate a comparison chart before and after.

The key to improve the quality of the generated pictures is to enhance the resolution of the pictures. The idea is to gradually increase the training generator and discriminator, starting from low resolution, gradually increase the convolutional layer and gradually improve the image details during the training process [19]. In training, first obtain the general structure of the image distribution and gradually increase the details, rather than learning all the distributions at the same time. By improving the neural network and training methods, high-resolution images can be obtained more reliably and good results have been achieved, as shown in Figure 23.

In the resolution process performed by the multiplier generator and discriminator (by adding convolution) to smooth the entire process, the neural network is adjusted for the change of α weight, as shown in Figure 24, which is a 16*16. The process of pixel growth regards the processing of higher-resolution convolutional layers as residual blocks (α weights show a linear increase from 0 to 1), correspondingly doubling or halving the nearest neighboring filter and convolution pool. RGB conversion refers to the mutual conversion of vector and RGB color information.

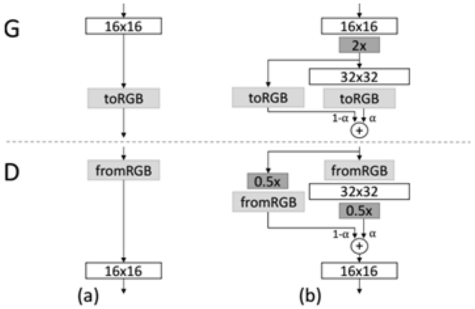


Figure 24. Neural network structure.

At this time, the input sample X is expressed as:

$$X = X_{16\text{pixel}} * \left(1\text{-}\alpha\right) + X_{32\text{pixel}} * a. \tag{11}$$

The ultimate goal is to achieve the mapping of latent vectors to 1024 pixels. It is difficult to achieve such a mapping network using generative adversarial networks alone, thus procedural training is used, starting with low-quality samples (4*4 pixels), then multiplying the increase of the resolution (4*4 to 8*8 to 16*16, and finally reach 1024*1024) to achieve the detail of the picture. The whole training process is represented in Figure 25.

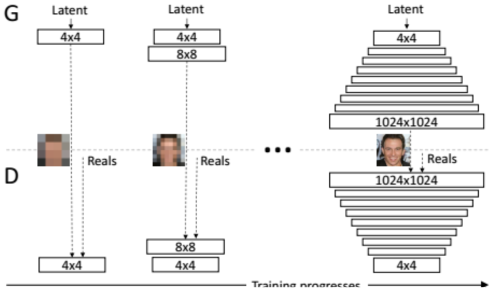


Figure 25. Training process.

symbol	significance
D_k^n	Discriminator features of layer n
M	total floors
N_n	Number of elements in layer n

Figure 26. Symbol Description.

The high resolution requires the discriminator to have a large receptive field, which will lead to overfitting problems and excessive storage space requirements due to extremely deep networks and extremely large convolution kernels. Therefore, in order to avoid high resolution, the instability brought by the huge receptive field of the residual block uses different discriminators to process images with different resolutions, where these discriminators have the same structure. Suppose there are discriminators D1, D2, D3, D4, then their loss function is expressed as:

$$\underset{G}{min}\ \underset{D1,D2,D3,D4}{max} \sum\nolimits_{n=1,2,3,4} L_{GAN}(G,D_k). \qquad (12)$$

At the same time, feature matching is added to stabilize the generation requirements of generators with different resolutions:

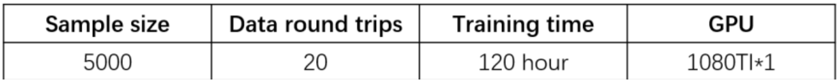
$$L_{FM}(\mathrm{G},\mathrm{D}_k)=\mathrm{E}_{(s,x)} \sum\nolimits_{n=1}^{m} \frac{1}{N_n}[||D_k^n(s,x) - D_k^n(s,G(s))||_1]. \qquad (13)$$

The objective function of the model at this time is:

$$\underset{G}{min}((\underset{D1,D2,D3,D4}{max} \sum\nolimits_{n=1,2,3,4} L_{GAN}(G,D_n)) + \lambda(\sum\nolimits_{n=1,2,3,4} L_{FM}(G,D_n)). \qquad (14)$$

Under the input information, the edges extracted by the HED algorithm are used as the edge distribution and the semantic labels of the labeled faces as the common input, and input to the generation network. After conversion by the encoder, the image data is input, and the edge distribution in the image is matched with the semantic labels. After the matching, a clustering algorithm is used to obtain the feature code of the semantic category. When the model is inferred, a cluster center is randomly selected as the coding feature, matched with the corresponding label map and input to the generator to obtain the generated map.

3.4. Training Information and Version Configuration



Sample size	Data round trips	Training time	GPU
5000	20	120 hour	1080TI*1

Figure 27. Model training information.

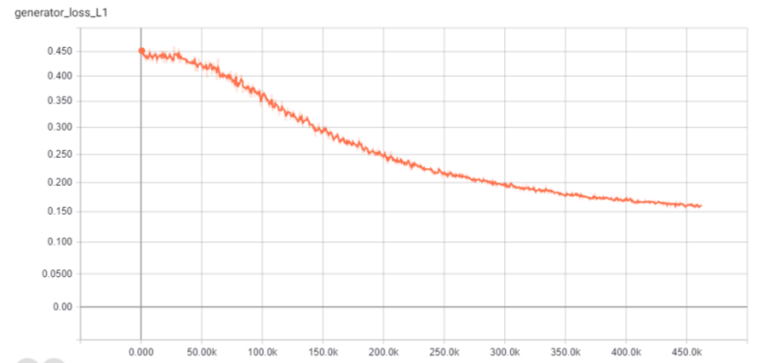


Figure 28. Generator L1 Loss.

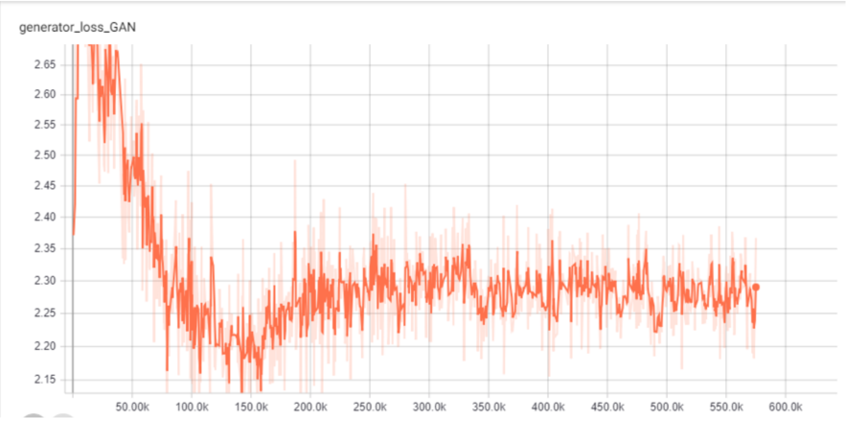


Figure 29. Generator Loss

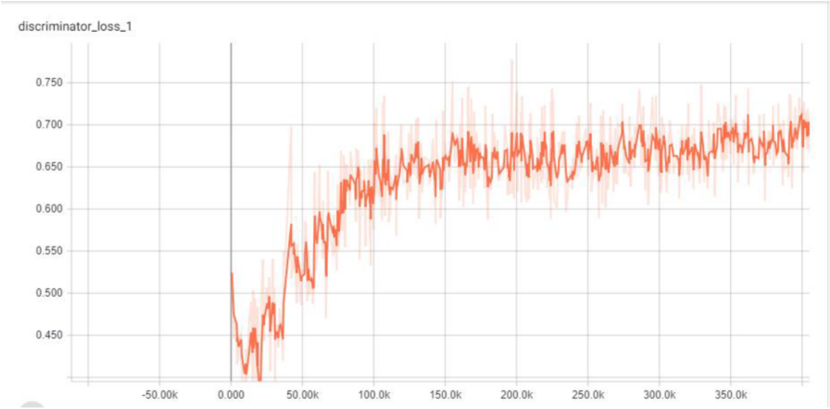


Figure 30. Discriminator Loss

content	platform	CUDA	Cudnn	Python Version
Edge extraction	Caffe	7.5	5.0	2.7
Face tag	Cv2	7.5	5.0	2.7
Pix2pix model	Tensorflow	9.0	7.0	3.6
HD training	Pytorch	9.0	7.0	3.6

Figure 31. Platform and configuration

4. Conclusion

In this study, the application of image translation in human faces was explored. First, we searched for a suitable data set and cleaned it. In the case of a translation framework, edge extraction and face identification are performed on the face image as a reasonable output of the translation framework. After multiple training and testing of the model, the research method is continuously improved, and then perfected. The method of edge extraction and face identification has improved the neural network structure and training method in the translation framework. After obtaining an effective translation model, the front end, database and back end of the system have been developed and tested. The Face generation system is feasible.

In the research of the subject, although there are certain results, there are also certain problems. First of all, the unfamiliarity with the translation framework, the lack of own experience and the lack of information acquisition channels have caused a lot of useless work that could be avoided. Secondly, due to its own strength, it is not possible to obtain a data set of Asian faces, so there is still a lack of applicability. There are also many areas that can be improved in system development.

However, in the research of the subject, it is found that there is huge room for development on the basis of the traditional translation framework. Only on the basis of the existing framework, there are more fields for translation generation. At the same time, the translation framework is no matter the training efficiency or the generalization. Adaptability and other aspects can be improved and developed.

References

[1] Phillip Isola. Image-to-Image Translation with Conditional Adversarial Networks[C].CVPR. CVPR2017.Honolulu:CVPR,2017.

[2] Cao Zhiyi, Niu Shaozhang, Zhang Jiwei. Research on face reduction algorithm based on semi-supervised learning generation adversarial network[J]. Journal of Electronics and Information Technology. 2018, 40(2):323-330. (in Chinese)

[3] RASMUS A, VALPOLA H, et al. Semisupervised learningwith ladder network[J]. CoRR.2015, 7(2672):1-7.

[4] Hinton G E , Srivastava N , Krizhevsky A , et al. Improving neural networks by preventing co-adaptation of feature detectors[J]. Computer Science, 2012, 3(4):212-223.

[5] CANNY J. A compatibility approach to edge detection[J]. Pattern Analysis and Machine Intelligence, 1986, 8(6):679-698.

[6] Li Junshan, Ma Ying, Zhao Fangzhou, etc. Improved Canny image edge detection algorithm [J]. Acta Photonica Sinica, 2011, 40(1): 55-54. (in Chinese)

[7] Chen Hongxi. Edge detection of the canny operator based on edge-preserving smooth filtering [J]. Journal of Lanzhou Jiaotong University, 2006, 25 (1): 86-90. (in Chinese)

[8] Wang Rong, Gao Liqun, Chai Yuhua, etc. Edge detection method combining Canny method and wavelet transform [J]. Journal of Northeastern University, 2005, 26 (12): 1131-1133 (in Chinese)

[9] GOODFELLOW I J, POUGETABADIE J, MIRZA M, et al. Generative adversarial nets[J]. Advances in Neural Information Processing Systems, 2014, 3: 2672-2680.

[10] Goodfellow I J. Generative adversarial nets[C].NIPS.NIPS2014.Montreal: NIPS, 2014.

[11] J Zhu.Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks [C].ICCV . ICCV2017. Venice: ICCV, 2017.

[12] Ronneberger O.U-Net Convolutional Networks for Biomedical Image Segmentation[C].MICCAI.MICCAI2015. Munich: MICCAI, 2015.

[13] Yang S. From Facial Parts Responses to Face Detection A Deep Learning Approach[C].ICCV Santiago: ICCV,2015:3676-3684.

[14] Xu Xinchao, Fu Chen, Xu Aigong. An improved Canny edge extraction method[J]. Remote Sensing Information. 2016, 31(5): 43-46. (in Chinese)

[15] Zhang Sheng, Yang Yan. Image Canny edge extraction improvement based on variance between classes[J]. Information Technology.2018,1(1):60-62. (in Chinese)

[16] Xie S, Tu Z. Holistically-Nested Edge Detection[J]. International Journal of Computer Vision. 2015, 125(1-3):3-18.abs/1701.07875.

[17] Li Y , Xiao N , Ouyang W . Improved Boundary Equilibrium Generative Adversarial Networks[J]. IEEE Access. 2018,1(1):1-1.

[18] Lecun Y L , Bottou L , Bengio Y , et al. Gradient-Based Learning Applied to Document Recognition[J]. Proceedings of the IEEE. 1998, 86(11):2278-2324.

[19] Karras T. Progressive Growing of GANs for Improved Quality, Stability, and Variation[C].ICLR.ICLR2018. Vancouver:ICLR,2018