Face Weight Manipulation with Machine Learning (FWM)

Peichen Liu[1]

Abstract: Face Manipulation is constantly discussed in the computer vision and biometric research field. Though many manipulation methods have been proposed, it is still considered an open challenge in many aspects. Since many Face Manipulations use the image-to-image model to implement functions, this does not allow artificial intelligence to learn enough features, so we implemented a text-to-image model based on body weight. Face weight manipulation system. The overall project consists of three parts: the training of the personality text-to-image model, the face weight manipulation, and the evaluation of the different weight-level faces. We successfully trained 97 models that can generate faces of different people and used these models to generate images of faces of five different weight levels corresponding to all 97 people. Finally, we tested the impact of these five different weight levels on the face recognition system. In this paper, we first conduct a simple survey on face manipulation and face recognition; then introduce our design framework and experimental results; finally, evaluate the manipulated face.

Keywords: face weight manipulation, stable diffusion, biometric system, face recognition.

1 Introducation

Face manipulation is an important task in computer graph and biometric systems.[Qi19] There are lots of applications for the manipulation of the face such as style transfer and facial detail transfer.[Le20] Because human body weight transformation requires a long-term process, if you want to use real face photos to analyze the face changes caused by weight transformation, or build a face recognition data set based on this, it may cost a lot of money. long time. This will not only waste time but may increase the impact of age on the face. And not only the weight but also the parameters of various faces need a certain amount of time to change. Addressing the problem is therefore a prime concern for researchers and practitioners. Last few years, several approaches have been proposed in the literature to solve this. In the previous research, the main method of manipulating the face is an image-to-image model. The image generated by this method is only changing some details of the input image.

Researchers and commercial vendors have proposed a number of approaches to manipulate the photo of faces.[Ch18, Le20] These methods start from simply modifying the details of an image to using two images to superimpose to obtain corresponding features to generate a new picture. But these methods are still image-to-image in essence. It is still difficult to study the face of a human and generate a manipulated face from the same person.

We want to propose a text-to-image method to manipulate the human face, especially to simulate the different weights of the human. And even though there have been some

[1] Denmark Technical University, s222475@dtu.dk

studies on manipulating faces, with the development of deep learning and the emergence of new text-to-image models, it is possible to use deep generative models to operate on faces. Our method allows the model to first learn a person's facial features, preserve these facial features, and simulate weight changes in subsequent transformations.

2 Related Work

Face image manipulation GAN(Generative Adversarial Network) has been widely used in computer vision and image processing. In the image manipulation area, GAN can be used in the image-to-image model[Qi19], because it can generate the realistic image. Yunjey Choi proposes a GAN-based image-to-image model, which uses a picture plus a prompt to change some attributes of the original picture[Ch18]. Cheng-Han Lee also proposes a GAN-based method to use two pictures to generate a new image. Image A provides the appearance of the person, and image B provides the expression. In this way, a person with the expression image A in image B is generated[Le20].

Face Recognition System The loss function plays a central role in face recognition, so here we investigate the loss functions of some different face recognition systems to explore the recent year's development of face recognition systems. Rajeev Ranjan proposes the L2-softmax loss which adds an L2-constraint to the feature descriptors which restricts them to lie on a hypersphere of a fixed radius[RCC17]. Weiyang Liu proposes that SphereFace loss can be viewed as imposing discriminative constraints on a hypersphere manifold, which intrinsically matches the prior that faces also lie on a manifold[Li18]. Jiankang Deng proposes an Additive Angular Margin Loss (ArcFace) to obtain highly discriminative features for face recognition, which has a clear geometric interpretation due to its exact correspondence to geodesic distance on a hypersphere[De22].

3 Background

3.1 Text-image Model

A text-to-image model is a machine learning model which takes an input natural language description and produces an image matching that description[Fr21].

3.2 Prompt

The prompt is a descriptive sentence in the text-to-image model. It is like the communication between humans and artificial intelligence. Humans use the appropriate prompt to tell artificial intelligence models what image they want. AI can use these prompts to generate the corresponding images.

3.3 Stable Diffusion

Stable Diffusion is a deep-learning text-to-image model. It is released in 2022 and then quickly became popular in a short time. Images can be generated in seconds by using this model.[sd] It is a deep generative neural network. Its principle is to generate the target image by continuously denoising random noise until the set number of steps is reached[sd]. It can generate different styles of the picture including realism, cartoon, etc. There are a number of AI drawing commercial software based on stable diffusion.

3.4 DreamBooth

DreamBooth is a personality prompt trainer. Dreambooth can be utilized to train a personality prompt by a few images(3-5 images). For example, five photos of the same dog and an existing stable-diffusion model are used as input, then after training, Dreambooth can generate a model. While this model contains all the functions of the original model, the newly added dog will also become a new prompt. Figure 1 shows the example of the DreamBooth workflow and performance.

Fig. 1: Example for DreamBooth[Ru23]

3.5 Performance of Biometric System

Every biometric has a different accuracy rate. There are several different standards to evaluate the different bio-recognition systems. There are some important values that are useful in the evaluation of the systems especially use in this paper.

- FMR(False-Match-Rate): proportion of the completed biometric non-mated comparison trials that result in a false match[Bu].
- FNMR(False-Non-Match-Rate): proportion of the completed biometric mated comparison trials that result in a false non-match[Bu].
- DET(detection error trade-off) curve: A DET curve is a graphical plot of error rates for binary classification systems, plotting the false rejection rate vs. the false acceptance rate.

- EER(Equal-Error-Rate): Intersection of $y = x$ and DET curve.
- FNMR@FMR=0.1%: The value of the FNMR when the FMR equals 0.1%.

3.6 MagFace

MagFace proposes a new face feature expression that can be applied to multiple tasks. It uses angle and module length two-dimensional information for face recognition, has good recognition performance, and can also analyze face quality well. There are two different score systems in the MagFace, Users can choose the Euclidean distance or cosine similarity as the evaluation basis. And users can also modify the program to use their own functions.

4 Methodology

4.1 Design Framework

In order to simulate the different weight level faces, we divide the weight into five different levels. The five different levels are from extremely thin to extremely fat i.e. extremely thin, thin, medium, fat, and extremely fat. After formulating the different levels, we use the DreamBooth to train the personality model for the different people. Specifically, we use DreamBooth to train 100 different prompts to map 100 different people in the FRGC-v2. After training these models, we use the stable diffusion to generate the different weight level faces of these 100 different people in the dataset. Last but not least, we use MagFace as the evaluation of the different weight-level faces and draw the plot to present the results. A brief introduction of the design framework shows in Figure 2.

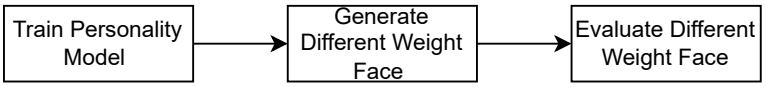

Fig. 2: Design Framework

4.2 Training

In the Training section, we use Dreambooth as the tool to train the personality model (personality prompt). We choose 100 different persons' faces to generate the personality model. The workflow of the training process is as follows.

Firstly, we randomly choose 100 people in the dataset to ensure the average body weight. Secondly, we train the different person's personality models one by one. The input of the Dreambooth trainer is the person's face and the original model and a new prompt of the person's face. We choose the stable-diffusion-v1-5 as the original model in this part. Finally, the Dreambooth trainer will output a checkpoint file that has the original model

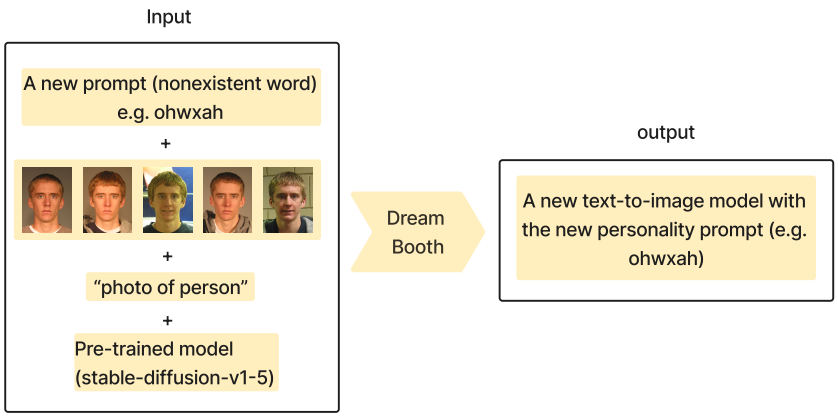

Fig. 3: Workflow of using DreamBooth Train the personality model

function and the new prompt which means the person's face we add to the input. The workflow of training the new prompt model shows in Figure 3.

After training these 100 people, we get 100 different text-to-image models. They retain the original functions and the new prompts to express the corresponding faces.

4.3 Face Weight Manipulation

In order to generate the different weight-level faces, we use the stable-diffusion-webui(an open-source project using the stable diffusion model)[AU]. We use the model trained in the DreamBooth which is mentioned in the secition4.2 and use five different text prompts to generate different weight level faces. The correspondence of prompts and weight level shows in Table1.

weight level	positive prompt
extremely thin	a photo of a person's face, extremely skinny, fragile, [personality prompt], Front view
thin	a photo of a person's face, just a little thin, [personality prompt], Front view
medium	a photo of a person's face, medium weight level, [personality prompt], Front view
fat	a photo of a person's face, just a little fat, [personality prompt], Front view
extremely fat	a photo of a person's face, extremely fat, overweight face,[personality prompt], Front view

Tab. 1: Prompts for the different weight levels

And appropriately adding adjectives in the negative prompt is necessary, if the style of the generated image does not meet expectations. Other parameters in the stable diffusion are shown in Tab2.

Image size	512*512
CFG Scale	7
Sampling steps	40
Sampling method	Euler a

Tab. 2: Other parameters in the stable diffusion

The Image size means the height∗weight(The unit is a pixel). This size of image is the highest quality image of the stable diffusion webui with the model trained by DreamBooth. The classifier-free guidance scale (CFG Scale) is one of the additional settings found in the Stable Diffusion model. The CFG scale adjusts how much the image looks closer to the prompt and input image.[Dr] The sampling steps are the level of detail of the image, the higher the sampling steps mean the more details. Euler is the simplest sampler and is mathematically equivalent to Euler's method for solving ordinary differential equations. The whole process of image generation shows in Figure 4.

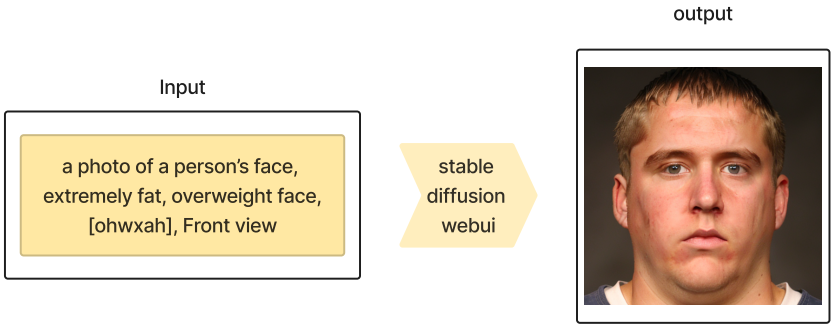

Fig. 4: Workflow of generating the different face

4.4 Feature Extraction and Comparison Score Calculation

In the face recognition section, we use the model in the MagFace project named "magface epoch 00025" to extract the feature. These features are stored in a file in the form of vectors. After calculating the features, we use these feature vectors to calculate the mated and nonmated scores. So, in the score calculation section, we also use MagFace to calculate the score of the baseline and the different weight levels. MagFace uses a modified cosine distance to calculate the score of the face similarity. Specifically, for example, there are

two different vectors named V1 and V2. These two vectors are the feature vector of two images. The first step is that calculate the dot of these two vectors.

$$dot = \sum_{i=1}^{n} V1 * V2 \tag{1}$$

After that, calculate the L2 norm as well as the Euclidean distance of these two vectors. Then multiply the Euclidean distance of the two vectors to calculate the norm.

$$norm = |V1|_2 * |V2|_2 \tag{2}$$

The next step is to calculate the cosine similarity, the cosine similarity is calculated by dividing the dot by the norm. And the value needed in the range[-1,1].

$$similarity = min(max(\frac{dot}{norm}, -1), 1) \tag{3}$$

Finally, the cosine distance is calculated, which is defined as the result of $1 - cosinesimilarity$. However, in the implementation of MagFace, what is calculated is the normalized cosine distance, which is the value obtained by dividing arccos(similarity) by π.

$$cosdist = \frac{arccos(similarity)}{\pi} \tag{4}$$

However, the cosine distance in the MagFace is different from the normal score method. In the MagFace the higher score means the lower similarity. And the range of the score is [0,1]. So, we calculate the final score by Eq5

$$finalscore = 1 - cosdist \tag{5}$$

5 Experimental evaluation

5.1 Dataset

The data set in the project is the FRGC-v2 dataset. The FRGC(Face Recognition Grand Challenge)-v2 dataset is a wild used open source dataset. The data for FRGC consists of 50,000 recordings divided into training and validation partitions.[Ph] The size of the image in the dataset is 720∗960. The dataset also includes a file that presents the region, gender, and age.

5.2 Implementation Detail

The first step is to train the personality model with DreamBooth. We use Google colab as the training platform and use an open-source DreamBooth project[Ru22] to train 100 personality models. We choose the Tesla V100 as the training GPU. The average time of one model training is 30 minutes. Secondly, we use these 100 different models to generate the different weight-level photos. The image generated is 512*512 pixels which is the best performance of the stable diffusion model. In the process of face generating, there are 3 models that have some problems which can not generate the right face, the example and analysis of the mistake show in Section 5.4.

The next task is to do face recognition and calculate the mated score and non-mated score. In this part, we choose the trained model first. We use the $magface\_epoch\_00025$ as the recognized model which is trained by using $MS1MV2$. Because the training dataset in the $magface\_epoch\_00025$ model is $112 * 112$, so we alter the image from 512∗512 to 112∗112 to adapt to the system. We use Brime[Br](an online image modification website) to change the size of the image generated from the last step.

We use the MagFace face evaluation function to calculate the baseline and the different performances for the different weight levels. We modify the MagFace program to output the cosine distance as a file. And we create a shell script to calculate 6 different groups' mated and non-mated scores. The result shows in Section 5.4.

5.3 Pair list for baseline and the weight-manipulation group

We create the pair list to calculate the mated score and non-mated score. In the field of machine learning, a pair list is usually used to store the information of paired samples. In this project, we use a pair list to represent the two pictures for comparison during testing. The format of the pair list shows in Tab3

probe image	reference image	whether the same person

Tab. 3: format of the pair list

So, the pair lists can be divided into two different types: the mated score and the nonmated score of baseline, and the mated score and the nonmated score of the five different weight-level. We use the 100 persons' faces used in the DreamBooth training to build the baseline pair list. In the baseline situation, the mated score is calculated by comparing face images of the same person in the probe set and the reference set, as we mentioned in Section 5.1. On the other hand, the nonmated score is calculated by comparing facial images of different people in the probe set and the reference set. So, the pair list of the baseline is created by these standards.

5.4 Results of the face weight manipulation

After training the personality model, the output is a checkpoint(".ckpt") file. And there are 100 different checkpoint files after training. We generate five different weight-level and the example of one person whose prompt is "ohwxah" shows in Figure 5. There are also some

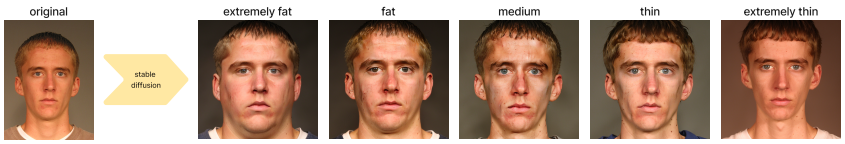

Fig. 5: Example of the face after manipulate

situations that lead to the mistake of image generation. There are four mistake situations during our experiment: A style is an animated image, an image that is not a front view, a full body image of a person, and an image that is not a person. The example of these three mistake situations show in Figure 6 The first three mistakes can be solved by adding

Fig. 6: Example of the mistaken image

some positive or negative prompts. For the animated style image, we add "cartoon" and "animated" as negative prompts. For the non-front view image, we add the "side view" as

a negative prompt. For the images to present the full body, we add the ”face of human” as the positive prompt and the ”full body of human” as the negative prompt. But for the last mistake situation, we can not solve it by changing the prompts. This situation happens because of a training error. There are three possible reasons for this error: 1. the quality of the input image; 2. the error input of the new prompt; 3. adding two human faces to train one model. During our experiment, only 3 models can not generate a human face (the last mistake), so, in the end, we get 97 different persons with different weight level faces as well as 485 face-weight manipulated images.

5.5 Evaluation of the manipulated image

In order to evaluate the performance of the weight manipulates image, we use the DET curve to show the comparison of the different weight level performances. Firstly, we calculate the mated and non-mated scores by using the MagFace. The score of baseline and other levels is shown in Figure7 and Figure 8. It is worth mentioning that all of the other weight levels have the same size of data.

Statistic	Mated	Non-mated
Observations	3716.000000	9315.000000
Minimum	0.347460	0.400980
Maximum	1.000000	0.598140

Fig. 7: baseline data

Statistic	Mated	Non-mated
Observations	615.000000	1455.000000
Minimum	0.456560	0.417780
Maximum	0.851660	0.770320

Fig. 8: other levels data

And use these mated and non-mated scores and Biometric Performance Tutorial to generate the DET curve and calculate the EER and FNMR@FMR=0.1%. The DET curve is shown in Figure 9.

In the DET curve, we can find out that every weight level will affect the performance of the face recognition system. And we can find out the extremely fat level has the worst performance which means it has the largest difference from the original dataset. The fat, medium, and extremely thin levels have a similar performance which means the face in these three groups has the same influence on the recognition system.

To evaluate the system in more detail, we also calculate the EER and FNMR@FMR=0.1%. And EER and FNMR@FMR=0.1% is shown in Tab4.

6 Conclusion

6.1 Overview of the project

In this project, we researched and implemented Face Weight Manipulation based on stable diffusion and DreamBooth. We use MagFace to calculate the performance of the baseline

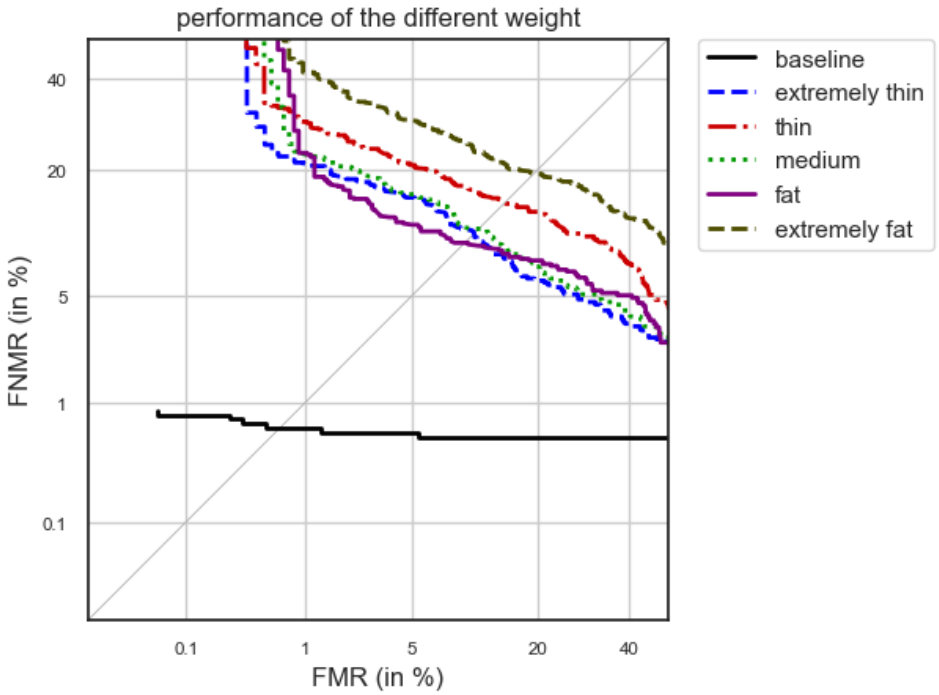

Fig. 9: DET curve for baseline and different levels

	EER	FNMR@FMR=0.1%
baseline	0.48%	59.2%
extremely fat	19.5%	80.65%
fat	9.6%	96.74%
medium	11.2%	97.5%
thin	14.7%	86.5%
extremely thin	10.6%	75.8%

Tab. 4: EER and FNMR@FMR=0.1%

and the different weight levels and draw the DET curve to visualize the performance. From the obtained DET curve, it can be concluded that all changes in weight levels will have an impact on the results of face recognition, and extremely fat level has the greatest impact on face recognition.

This new text-to-image modifies the details of the face, which changes the traditional image-to-image method. It does not need to input pictures, and the personality model can be trained in advance to generate different weight levels with different prompts.

6.2 Limitation and Future work

Our experiments still have some shortcomings. When training the personality model, we randomly selected the faces in the original database to ensure that the generated faces can reflect the average change. However, the establishment of a data set in this way cannot accurately reflect the impact on the face recognition system from a certain weight level to another weight level in the subsequent performance evaluation. But we were unable to continue this work due to the dataset and time constraints. In future work, this is a place worthy of study.

More, we believe that we can continue to develop an integrated system in the future, only need to input the face that needs to be trained or scan the face in real-time, and then output the manipulated face according to the user's needs.

References

[AU] AUTOMATIC1111: , stable diffusion webui. https://github.com/AUTOMATIC1111/stable-diffusion-webui.

[Br] Brime: , Brime. https://www.birme.net/?target$_w idth = 112 target_h eight = 112$.

[Bu] Busch, Christoph: , ISO/IEC JTC SC33. Harmonized Biometric Vocabulary.

[Ch18] Choi, Yunjey; Choi, Minje; Kim, Munyoung; Ha, Jung-Woo; Kim, Sunghun; Choo, Jaegul: StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). June 2018.

[De22] Deng, Jiankang; Guo, Jia; Yang, Jing; Xue, Niannan; Kotsia, Irene; Zafeiriou, Stefanos: ArcFace: Additive Angular Margin Loss for Deep Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(10):5962–5979, oct 2022.

[Dr] DreamBooth: , What is CFG and how to use it. https://decentralizedcreator.com/cfg-scale-in-stable-diffusion-and-how-to-use-it/.

[Fr21] Frolov, Stanislav; Hinz, Tobias; Raue, Federico; Hees, Jörn; Dengel, Andreas: Adversarial text-to-image synthesis: A review. Neural Networks, 144:187–209, 2021.

[Le20] Lee, Cheng-Han; Liu, Ziwei; Wu, Lingyun; Luo, Ping: MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). June 2020.

[Li18] Liu, Weiyang; Wen, Yandong; Yu, Zhiding; Li, Ming; Raj, Bhiksha; Song, Le: , SphereFace: Deep Hypersphere Embedding for Face Recognition, 2018.

[Ph] Phillips, P. Jonathon: , FRGC (Face Recognition Grand Challenge). https://paperswithcode.com/dataset/frgc.

[Qi19] Qian, Shengju; Lin, Kwan-Yee; Wu, Wayne; Liu, Yangxiaokang; Wang, Quan; Shen, Fumin; Qian, Chen; He, Ran: Make a Face: Towards Arbitrary High Fidelity Face Manipulation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). October 2019.

[RCC17] Ranjan, Rajeev; Castillo, Carlos D.; Chellappa, Rama: , L2-constrained Softmax Loss for Discriminative Face Verification, 2017.

[Ru22] Ruiz, Nataniel; Li, Yuanzhen; Jampani, Varun; Pritch, Yael; Rubinstein, Michael; Aberman, Kfir: DreamBooth: Fine Tuning Text-to-image Diffusion Models for Subject-Driven Generation. 2022.

[Ru23] Ruiz, Nataniel; Li, Yuanzhen; Jampani, Varun; Pritch, Yael; Rubinstein, Michael; Aberman, Kfir: , DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation, 2023.

[sd] stable diffusion: , Stable Diffusion Online. https://stablediffusionweb.com/.