USC

CSCI 599: Applied Machine Learning for Games

TalkingHead

Technical Paper

Team Members:

Yuming Gu

Jiahui Ding

Hongli Chen

Guangyun Zhou

Chaofan Zhai

Mengyu Zhang

Table of Contents

Abstract

TalkingHead is a machine learning project that makes realistic and personalized neural talking head models. The reasoning behind applying machine learning algorithms to our TalkingHead project is to efficiently reenact a talking facial model of different game characters for the movie-licensed video game from a few image view of a character, potentially even a single image, improve the game character model with detailed information and furthermore change the current longtime low-quality stereotype of the cinematic-inspired games.

Background and Overview of Movie-Licensed Video Game

Throughout the 2000s, game publishers worked alongside Hollywood to ship games based on the biggest blockbuster movies, day and date with the film's release in theaters. With rare exceptions, these tie-ins were not very good games: The tight schedules of film production meant that the games had to be produced very quickly and couldn't be delayed for further polish, and the fact that Hollywood was pulling the strings meant that the games' developers didn't have much, if any, creative freedom. And yet even though they were never any good, movie-games were seen as an inevitable fact of life for the gaming world, since consumers still bought them based solely on the strength of the license. Though a lack of quality had always been a problem with movie-games, ever since the days when Atari cranked out a horrendous 8-bit version of E.T. in six weeks, people still purchased them. When movie games were popular in the early 2000s, consumers were generally unaware of video game brands such as Medal of Honor or Call of Duty. A major movie license that they had heard of would be the thing most likely to catch their eyes, and their wallets.

But things changed a lot in recent years. Gamers are expecting higher quality of the game rather than only the famous movie name. Thus tons of movie licensed games with low quality disappeared in the market. In 2018, the top ten highest grossing films worldwide only had one video game adaptation between them (the Lego version of the Incredibles 2). Compared to 2008, where there were six video game adaptations of the ten highest grossing films (Kung Fu Panda, Madagascar: Escape 2 Africa, Quantum of Solace, Iron Man, Wall-E and The Chronicles of Narnia: Prince Caspian). The change being that publishers have slowly begun to realise that video games that have their own development cycle and creative ideas outside of the film tend to be received far more positively and then subsequently perform far better. Batman: Arkham City released a few months prior to the Batman film: The Dark Knight Rises, had its own development separate to the film, so didn't have to be rushed out to match the film's release date and was an incredible success, selling over ten million units. Marvel's Spider-Man for PS4 sold an enormous nine million copies. These facts show that the high-quality movie licensed games could still be successful.

Our TalkingHead project is able to help the game producers to improve the quality of the movie licensed game, especially in the game character modeling. And the machine learning model will efficiently reduce the time and cost to acquire a high-quality face model for game characters, which can help the game developers to fit in the intensive development cycle of the movie licensed game. TalkingHead applies machine learning algorithms by taking the character's face from movies to create the face model in the related video game. It will also make the perfect face expression of the game character in motion. Our project is not about a specific game but a technique which is able to be put into use for the development of various movie licensed video games.

Related Work

The traditional method of face reenactment are based on the use of explicit 3D modeling of human faces (Blanz and Vetter 1999) where the 3DMM parameters of the driver face and the target face are extracted from one-shot image, and combined to implement the motion (Thies et al. 2015; Thies et al. 2016). Another well-known approach is image warping, which uses the estimated flow extracted from 3D models (Cao et al. 2013) or sparse 2landmarks(Averbuch-Elor et al. 2017) to get the target image. The recent popular approach also exploring image-to-image translation architectures(Isola et al. 2017) on the success of neural networks, such as the works of Xu et al. (2017) and that of Wu et al.(2018), which included the cycle consistency loss (Zhu et al. 2017). Combined with these two methods, many studies proposed new approaches. Training an image translation network, Kim et al. (2018) maps reenacted render of a 3D face model into a photo-realistic output. Architectures, which are a hybrid of the target's style information and driver's spatial information, have been studied recently. AdaIN (Huang and Belongie 2017; Huang et al. 2018; Liu et al. 2019) layer, attention mechanism (Zhu et al. 2019; Lathuiliere et al. 2019; Park and Lee 2019), deformation operation (Siarohin et al. 2018; Dong et al. 2018), and GAN-based method (Bao et al. 2018) are widely adopted as well. Other ideas such as the use of image-level (Wiles, Koepke, and Zisserman 2018) and feature-level (Siarohin et al. 2019) warping, and AdaIN layer in conjunction with a meta-learning (Zakharov et al. 2019) also make greatly progress. These approaches either require a dataset with image pairs that may be hard to acquire or an independent model per person.

What We Actually Built

Several recent works have shown how highly realistic human head images can be obtained by training convolutional neural networks to generate them. Combined with the practical scenarios of the movie-licensed video game industry, our project's final goal is to provide a personalized talking head model which can be learned from a few image views of a game character. Here, we present three methods to gradually achieve it: 1) generate highly realistic and personalized talking head models of human and portrait paintings in a few- and one-shot setting; 2) build a mapping function to translate human face to game character face; 3) extract the landmark from the human face and use a function to map the facial expression and the head position to the game character model in Unity 3D.

Talking Head Models of Unseen People and Portrait Paintings

We achieve this by introducing two novel sub-networks: LD-Net and FD-GAN. Landmark Disentangling Network (LD-Net) learns to disentangle identity from head poses/expression, predicting facial landmarks that preserve the identity of the target while combining expressions and poses from another driving subject. Feature Dictionary-based Generative Adversarial Network (FD-GAN) learns to transform landmark positions into a personalized video portrait of someone given a single target image, which allows subject agnostic reenactment of a portrait that preserves a recognizable identity of the target and can be applied to unseen identities without subject-specific training. To summarize, we use the following novel techniques to : (1) We introduce a novel one-shot learning method that enables portrait reenactment using the identity from a sin- gle image and expression/poses from videos of another subject. (2) We present a novel network for disentangling the identity from 2D face landmarks for cross-subject portrait reenactment. (3) We also propose a feature dictionary-based generative network to synthesize a high-fidelity face image, which is applicable to new subjects. We evaluate each sub-network as well as the full method extensively via quantitative measurements and qualitative comparisons with the state-of-the-art methods and demonstrate the performance of our method in preserving the identity and its ability to generalize to unseen subjects for cross-subject face reenactment.

Landmark Disentanglement Network(LD-Net)

This network is based on Encode-decode architecture. And we have two stages to do this part.

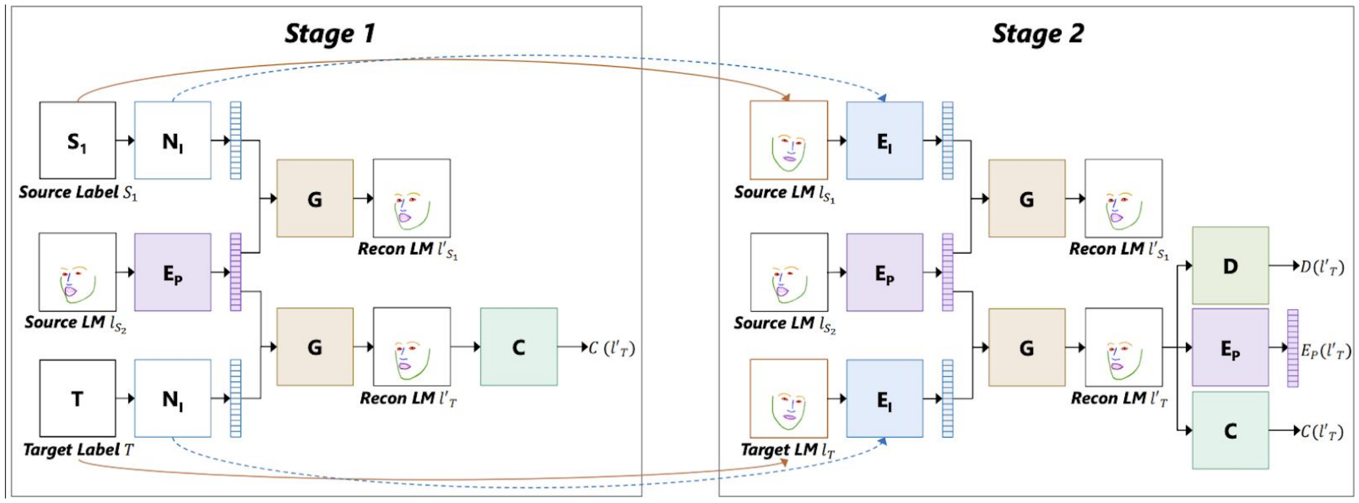


Figure 1. Landmark Disentanglement Network

In Stage 1, similar to the previous anime style space exploration (Xiang et al. 2018), the network consists of four modules: (1) a pose/expression encoder EP that computes a code from the input landmarks l that encodes only pose/expression without information about identity; (2) a one-layer network NI that maps the input one-hot identity label k to an identity code; (3) a generator G that combines a pose/expression code and an identity code to reconstruct landmarks l′ = G(EP (l), NI (k)); and (4) a classifier C that tries to classify the generated landmarks based on their identity.

To achieve this, the classifier C tries to classify l′ = lT′ as being a landmark of S1 while the pose/expression encoder EP , generator G and identity encoder NI tries to prevent the classifier from doing so. C is trained with the classification loss of the form:

$$\mathcal{L}_C = \mathbb{E}[-\log P(S_1|l')].$$

Meanwhile, EP , G and NI jointly optimized the reconstruction loss minus the identity classification loss as in Eq. 2. The reconstruction loss is defined as per-point squared Euclidean distance using landmark coordinates.

$$\mathcal{L}_G = \lambda_{\text{rec}}\mathbb{E}[||l - l'||_2^2] + \lambda_C\mathbb{E}[\log P(S_1|l')],$$

where λrec = 1000 and λC = 0.1.

All expectations are taken over l ~ p(l),k ~ p(k) where p(l) and p(k) are training distributions of landmarks and identities where l and k may be from different subjects. Different from the network architecture in [1], all convolutional networks are replaced with Multi-layer Perceptrons (MLP) in LD-Net, since instead of images we operate on landmark coordinates.

In Stage 2, in order to generalize to novel subjects, we introduce an identity encoder to Stage 2. A notable deficiency of [1] is that it does not include any encoder for the labeled data (i.e., identity in our task) and thus it is limited to generating new samples only for the labeled classes in the training data. In Stage 2, we replace the one-layer network NI with a full-fledged identity encoder EI that accepts landmarks as input and encodes them into an identity code.

The full training network of Stage 2 is shown in Figure 1, also involving two branches similar to Stage 1, that is, lS1 and lS2 are from the same subject while lS2 and lT belong to different subjects.

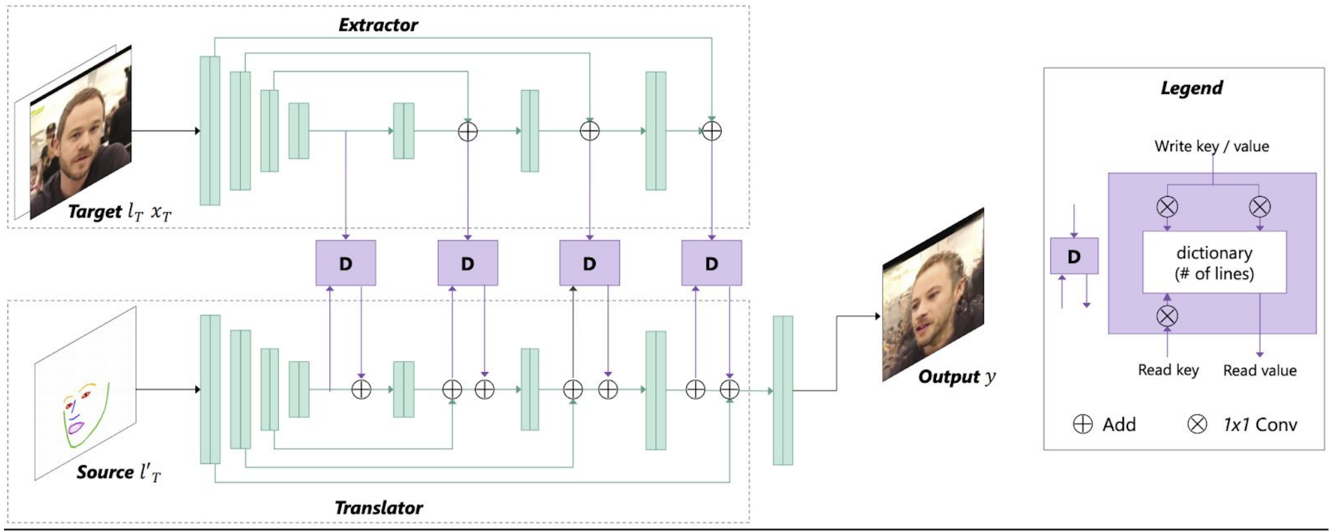


Figure 2. Feature Dictionary-based Generative Adversarial Network

For the second input Ls2 and Lt in the second branch, due to unavailability of ground truth, we train a discriminator D and a classifier C to constrain the reconstructed landmarks l'T. We use least square loss for the discriminator D following to minimize:

$$\mathcal{L}_D = \mathbb{E}[(D(l_{S_2}) - 1)^2 + (D(l'_T) + 1)^2].$$

The classifier C is trained with an adversarial loss on both input landmarks lT and generated landmarks:

$$\mathcal{L}_C = \mathbb{E}[-\log P(k|C(l_T))] + \mathbb{E}[-\log(1 - P(k|C(l'_T))],$$

With expectations taken over lS2 , lT ~ p(l), k ~ p(k). k is the identity label of lT . In addition, a content consistency loss term is defined between the generated pose/expression code and its ground truth EP (lS2):

$$\mathcal{L}_{\text{cont}} = \mathbb{E}[||E_P(l_{S_2}) - E_P(l'_T)||_2^2].$$

Thus, the identity encoder EI and generator G are jointly optimized to min- imize a weighted sum of the above losses:

$$\begin{aligned}\mathcal{L}_G = {} & \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} \\ & + \lambda_{\text{cont}}\mathbb{E}[||E(l_{S_2}) - E(l'_{S_1})||_2^2] \\ & + \lambda_D\mathbb{E}[D(l'_T)^2] \\ & + \lambda_C\mathbb{E}[-\log P(k|C(l'_T))],\end{aligned}$$

where λrec = 1000, λcont = 0.01, λD = 0.1, and λC = 0.1.

Feature Dictionary-based Generative Adversarial Network (FD-GAN)

With the predicted landmarks rasterized into a landmark image, our next goal is to translate it to a realistic face image. The translation procedures are as follows: a local patch around each location in the landmark image indicates "what facial part should be here". And for each location, we want to translate it into "how should it look like". Therefore, we are able to use a novel feature dictionary-based generative adversarial network(FD-GAN) to capture this intuition.

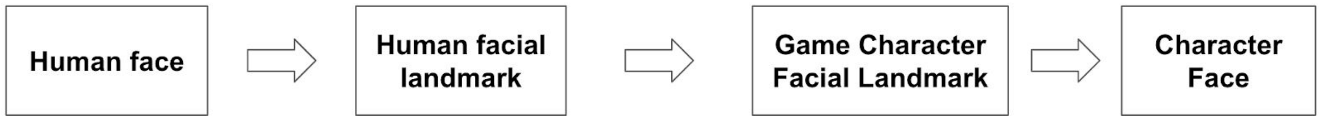
The architecture of our model is illustrated in Figure 2, which consists of an extractor and a translator. Given a target image and its corresponding landmark image, we will train an extractor that constructs a "feature dictionary" in the module D which is essentially a mapping from an annotation in the landmark image to its appearance in the target image. Concurrently, given another landmark image and the feature dictionary, we train a translator that retrieves relevant facial features from the dictionary based on the landmarks and composes a face image.

We designed a python-based system which could batch processing the images input. This system included : Training, Testing and evaluation. For each part, we did serious experiments and details, and we package it as an exe file in the future. Also for performance, our method takes approximately 0.08s for FD-GAN to generate one image and 0.02s for LD-Net to do landmark disentangling on a single NVIDIA TITANX GPU.

Talking Head Models of Game Characters

The general idea of this method is to build a mapping function that translates human face to game character face. An end to end solution is to build an image-to-image translation generative network whose input is human face and output is game character face. But getting pairs of images of human face and game character face is hard, and we need to build a special dataset for each person even if we do not change the target game character. So we propose another solution.

The keypoint of our solution is to use facial landmarks. Each human face has facial landmarks such as eyes, nose and mouse. Each game character's face also has similiar facial landmarks. So our key point is to find the relationship between human facial landmark and game character facial landmark. To be more specific, the process of our method is shown in the chart below. We will then have a detailed discussion about how we handle each part of this process.



Human face to human facial landmark

Dlib was used to extract the facial landmarks, which will be demonstrated in detail later.

Game character facial landmark to game character face

This problem can be solved by the Conditional Generative Adversarial Network(CGAN) (Mirza and Osindero, 2014). GAN is a kind of unsupervised learning method that can capture the underlying distribution of the dataset images and generate realistic images that has never been seen by the model before. CGAN is a special kind of GAN which belongs to supervised learning. Each image in the training set has several labels so the model can learn to generate images based on the information provided by the label.

We need to generate game character faces based on the information provided by the game character facial landmarks. If we consider the game character facial landmarks as an image, we can use pix2pix which is a special kind of CGAN that can translate an image to another image. Figure 3 is from the official website of pix2pix and shows its ability.

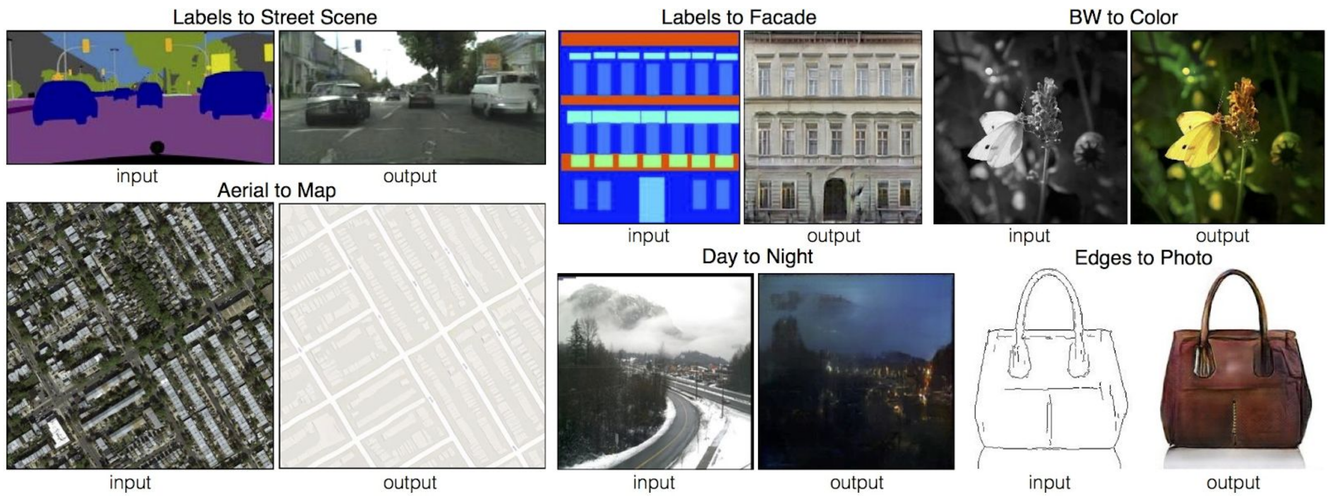


Figure 3. Example results of pix2pix on several image-to-image translation problems

We build a training set of 9000 pairs of images below (Figure 4). The left image is the game character landmark which is the input of our generator network. The right image is its corresponding game character face which is the output of our generator network. The mario face training data is produced automatically by our Unity3D mario model.

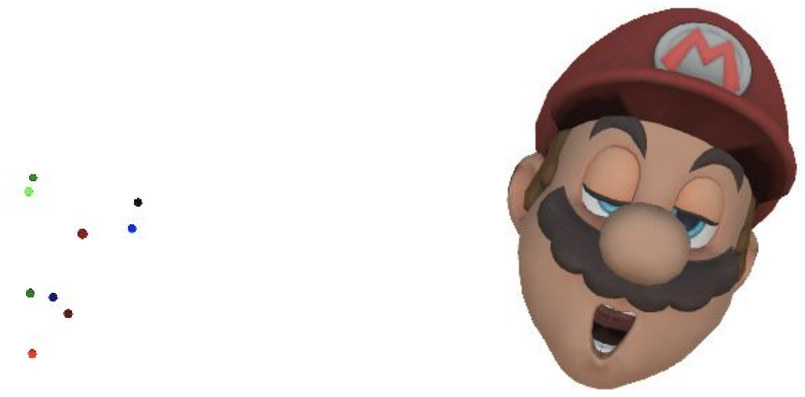


Figure 4. Example of Our Training Set

The general idea of pix2pix is shown in Figure 5. Different from the original GAN, the generator produces the target image based on an source image that describes the target image. The Discriminator judges whether the source image and the target image is a pair.

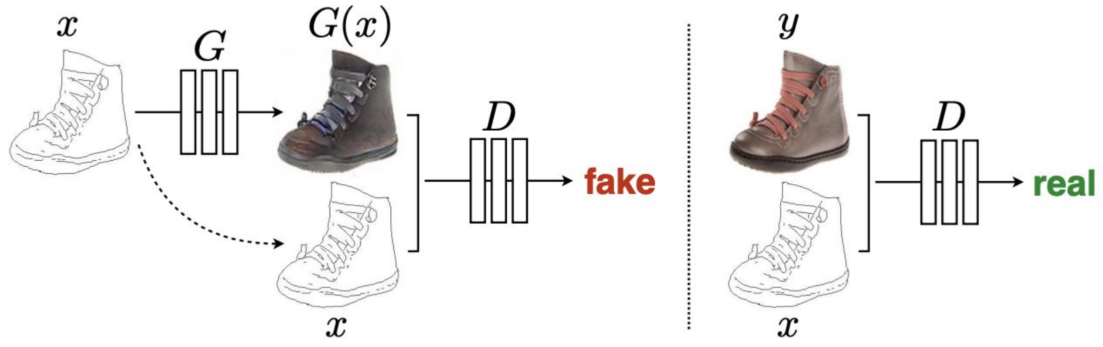


Figure 5. The General Idea of pix2pix

Same with the original paper, we use random crop and flip for data augmentation and we use L1 loss and cross entropy loss as the loss function and Adam to train the network. We also use the same network architecture for the Discriminator. But we made a small modification to the generator. For the generator, the original paper uses U-Net which is an encoder-decoder with skip connection. The encoder and decoder has the same number of filters w.r.t.

corresponding layer. However, in our dataset, the source image is only the landmark of the game character face, so we believe that the source image contains much less information than the target image. So we cut the number of filters in each layer of the encoder to half and achieve a higher speed of training and prediction without any quality loss. The final architecture of our generator network is shown in Figure 6, where the number in each layer is its number of filters. The generator can be seen as a translator from game character facial landmark to game character face.

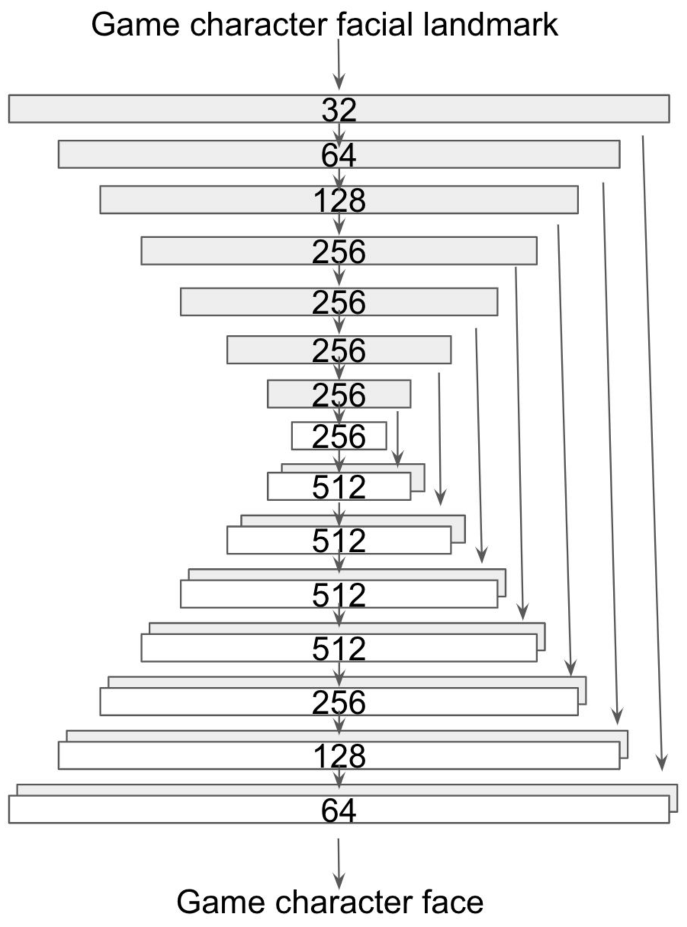


Figure 6. The Final Architecture of Our Generator Network

Human facial landmark to game character facial landmark

The last part of the puzzle is to build a translator from human facial landmark to game character facial landmark. In Figure 7, the left part is the human facial landmark which is our input. The right part is the game character facial landmark which is our output.

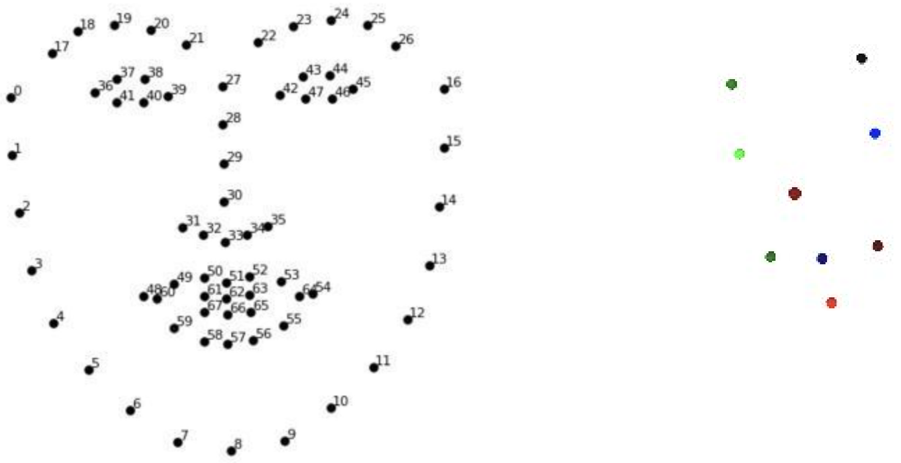


Figure 7. Human Facial Landmark to Game Character Facial Landmark

A possible solution is to use machine learning to train a model for this mapping, but it needs a lot of training data and it is time consuming to tune the model. So we use a more straightforward method, which is to manually design the mapping function. The mario facial landmark contains 9 points, each eye has a top and a bottom point, nose has 1 point, mouth has 4 points to represent its top, bottom, left and right. For each point in the mario facial landmark, we design a linear combination of the human facial landmark points to represent its position.

$$\begin{aligned} LeftEyeTopPosition &= 5p_{38} - 4p_{40} \\ LeftEyeBottomPosition &= p_{40} \\ RightEyeTopPosition &= 5p_{43} - 4p_{47} \\ RightEyeBottomPosition &= p_{47} \\ NosePosition &= 0.5p_{28} + 0.5p_{29} \\ MouseTopPosition &= p_{62} \\ MouseBottomPosition &= p_{66} \\ MouseLeftPosition &= 0.5p_{48} + 0.5p_{61} \\ MouseRightPosition &= 0.5p_{54} + 0.5p_{63} \end{aligned}$$

Where p represents a point in the human facial landmark. Most points are simple to map, but we have special designs for eyes. Mario eyes are much bigger than human eyes(about 5 times the size of human eyes), so we set the bottom of mario eye to the same as the human eye, but stretch the top higher. Although the mapped mario facial landmark is not the same as the real mario facial landmark, the generator can still produce realistic mario faces because the discriminator forced the generator to produce realistic mario faces in the training stage.

We also tried to produce the kung fu panda face with 700 images manually labeled by ourselves. Kung fu panda has richer facial expressions compared to mario and the images in kung fu panda movies have richer background settings. Our manually designed facial landmark mapping function for kung fu panda is shown below.

$$\begin{aligned}
LeftEyeTopPosition &= 6p_{37} - 4p_{41} - p_{38} \\
LeftEyeBottomPosition &= 2p_{41} - p_{40} \\
RightEyeTopPosition &= 6p_{44} - 4p_{46} - p_{43} \\
RightEyeBottomPosition &= 2p_{46} - p_{47} \\
NosePosition &= p_{33} \\
MouseTopPosition &= p_{62} + p_{30} - p_{29} \\
MouseBottomPosition &= p_{66} + p_{30} - p_{29} \\
MouseLeftPosition &= 2p_{48} - p_{67} + p_{30} - p_{29} \\
MouseRightPosition &= 2p_{54} - p_{65} + p_{30} - p_{29}
\end{aligned}$$

Talking Head Models of Game Characters in Unity3D

The general idea of this method is to extract the landmark from the human face and use a function to map the facial expression and the head position to the game character model, such as Mario, in Unity 3D.

Extract facial landmarks

Dlib is an implementation based on Vahid Kazemi and Josephine Sullivan’s idea. The main idea is to use an ensemble of regression trees to estimate the facial landmark. The detail can be found in the paper One Millisecond Face Alignment with an Ensemble of Regression Trees (Vahid Kazemi and Josephine Sullivan, 2014). In our project, it works so well. A high-quality prediction can be got in milliseconds. In our project, we use the pre-trained model where the iBUG300-W dataset was used.

Head position estimation

In order to achieve our goal, we need to estimate the relative orientation and position with respect to the camera. The head position and rotation problem here can be referred to the Perspective-n-Point problem(PNP).

As the Figure 8 below shows, there are three coordinates, including the world coordinate, image coordinate and the camera coordinate. If the rotation and translation in the world coordinates were known, the three dimensional points in world coordinates could be transformed to three dimensional points in camera coordinates. It would be also easy to project the three dimensional points in camera coordinates onto the image coordinate system with the intrinsic parameters of the camera, such as optical center, focal length, etc.

In our project, some points such as eyes, nose and mouth are written in the script. Dlib gives us the 2D landmark for the human face. We use the existing solution in openCV to solve the PnP problem. Given a set of correspondences between 3D points pi expressed in a world reference frame, and their 2D projections ui onto the image, we could calculate the pose (R and t) with the camera parameters focal length f and optical center c.

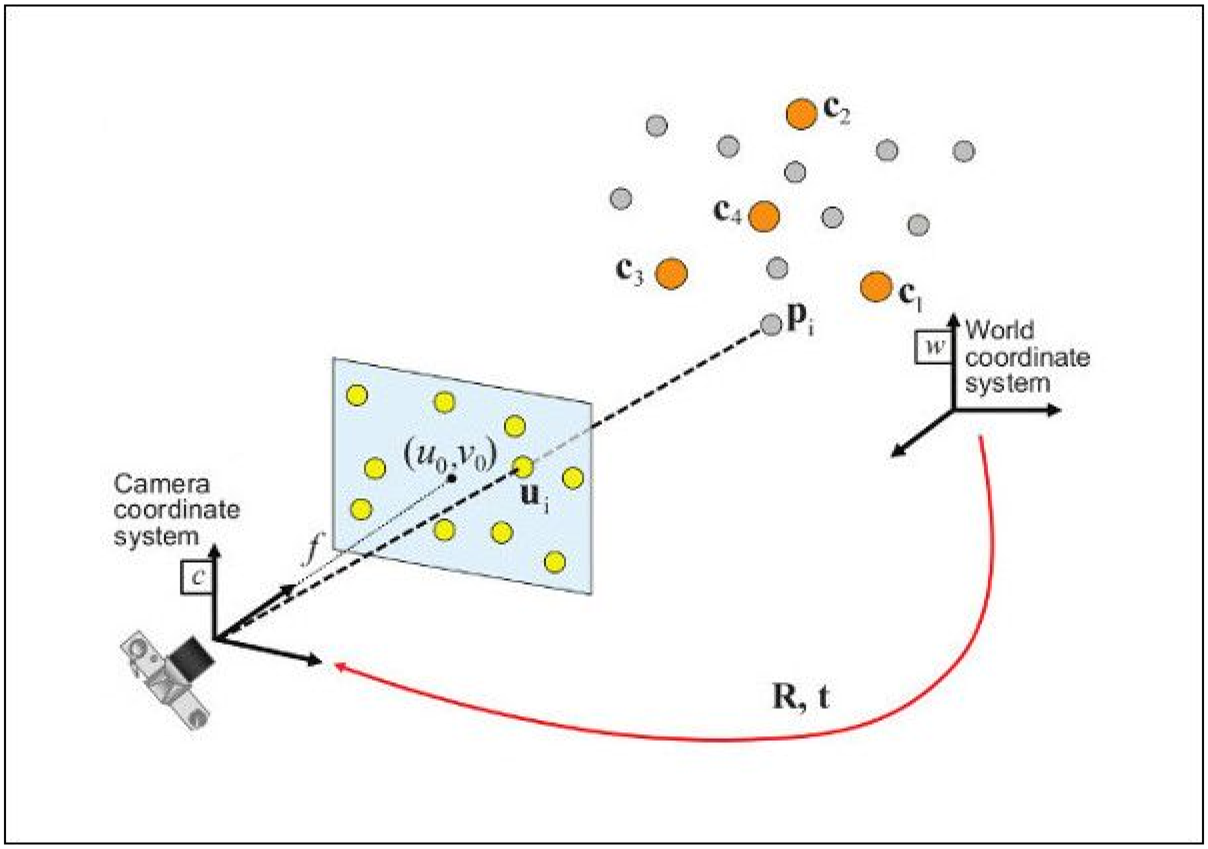


*Figure 8. Perspective-*n*-Point problem*

$$s\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}f_x & 0 & c_x\\0 & f_y & c_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_1\\r_{21} & r_{22} & r_{23} & t_2\\r_{31} & r_{32} & r_{33} & t_3\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

Facial expression estimation

In order to control mario facial expression, we need to detect the movement of eyes and mouth. This is a relatively new problem and there is not any standard solution for it. However, There are various attempts to solve this problem in the video. In our project, we apply eye aspect ratio (EAR), proposed by Soukupva and Cech in 2016.

The idea of EAR is simple and straightforward. EAR could be simply considered as the ratio of the eyes' width and their length.

$$\mathrm{EAR}=\frac{\|p_2-p_6\|+\|p_3-p_5\|}{2\|p_1-p_4\|}$$

Where p1, …, p6 is the points depicted in Figure 9.

The EAR is mostly constant when an eye is open and is getting close to zero while closing an eye.

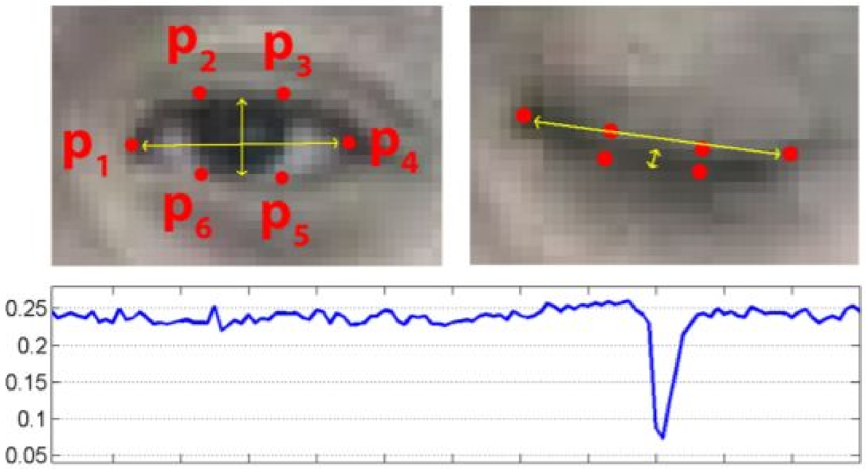


Figure 9. Eye Aspect Ratio

When an eye is open, the EAR is almost constant, however, when people close an eye, the ratio is getting close to zero. However, when a person rotates his head, the width of his eyes will change dramatically by Tianxing (2019). A more robust measure was proposed by Tianxing (see Figure 10). It used the length and width of the face instead of eyes or mouth. The formula can be seen as follow:

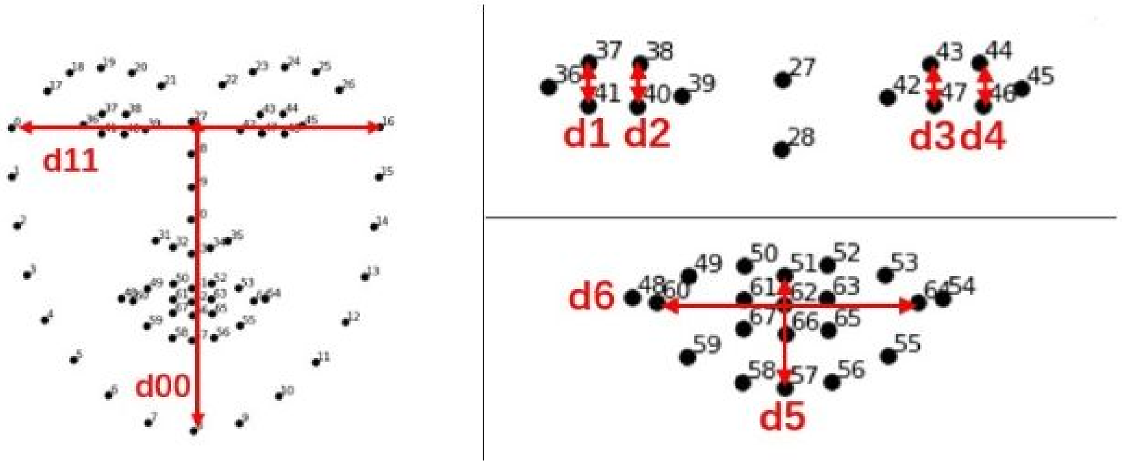


Figure 10. A More Robust Measure Proposed by Tianxing

$$d_{ref} = \frac{d_{00} + d_{11}}{2} = \frac{\|P_{27} - P_8\| + \|P_0 - P_{16}\|}{2}$$

$$d_1 = \|P_{37} - P_{41}\|$$

$$\vdots$$

$$d_4 = \|P_{44} - P_{46}\|$$

$$leftEyeWid = 6 \times (\frac{d_1 + d_2}{2d_{ref}} - 0.02)$$

$$rightEyeWid = 6 \times (\frac{d_3 + d_4}{2d_{ref}} - 0.02)$$

$$mouthWid = 1.27 \times (\frac{d_5}{d_{ref}} - 0.13) + 0.02$$

$$mouthLen = \frac{d_6}{d_{ref}}$$

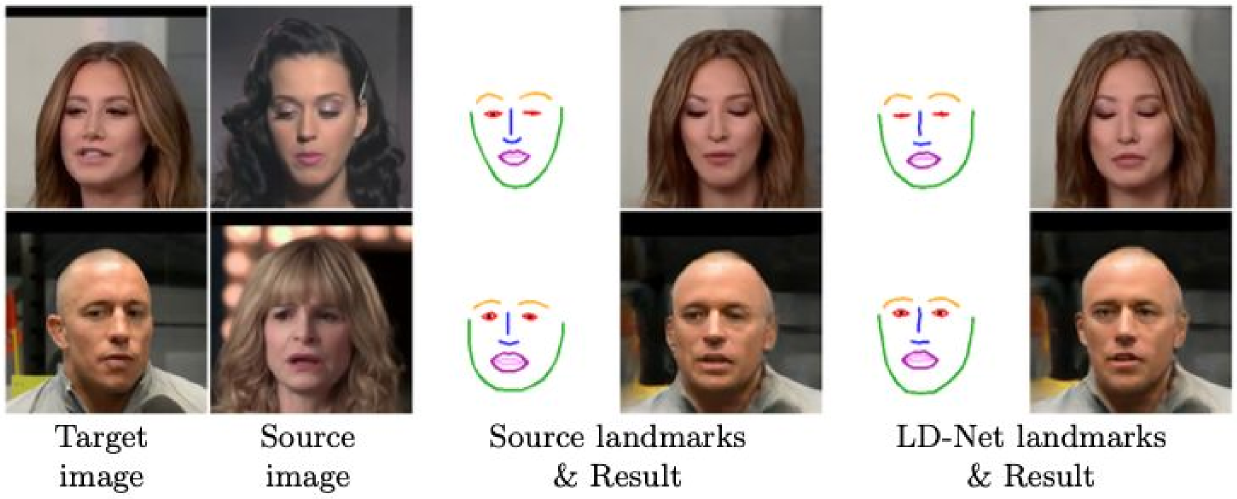
Results and Analysis[1]

1. Talking Head Models of Unseen People and Portrait Paintings

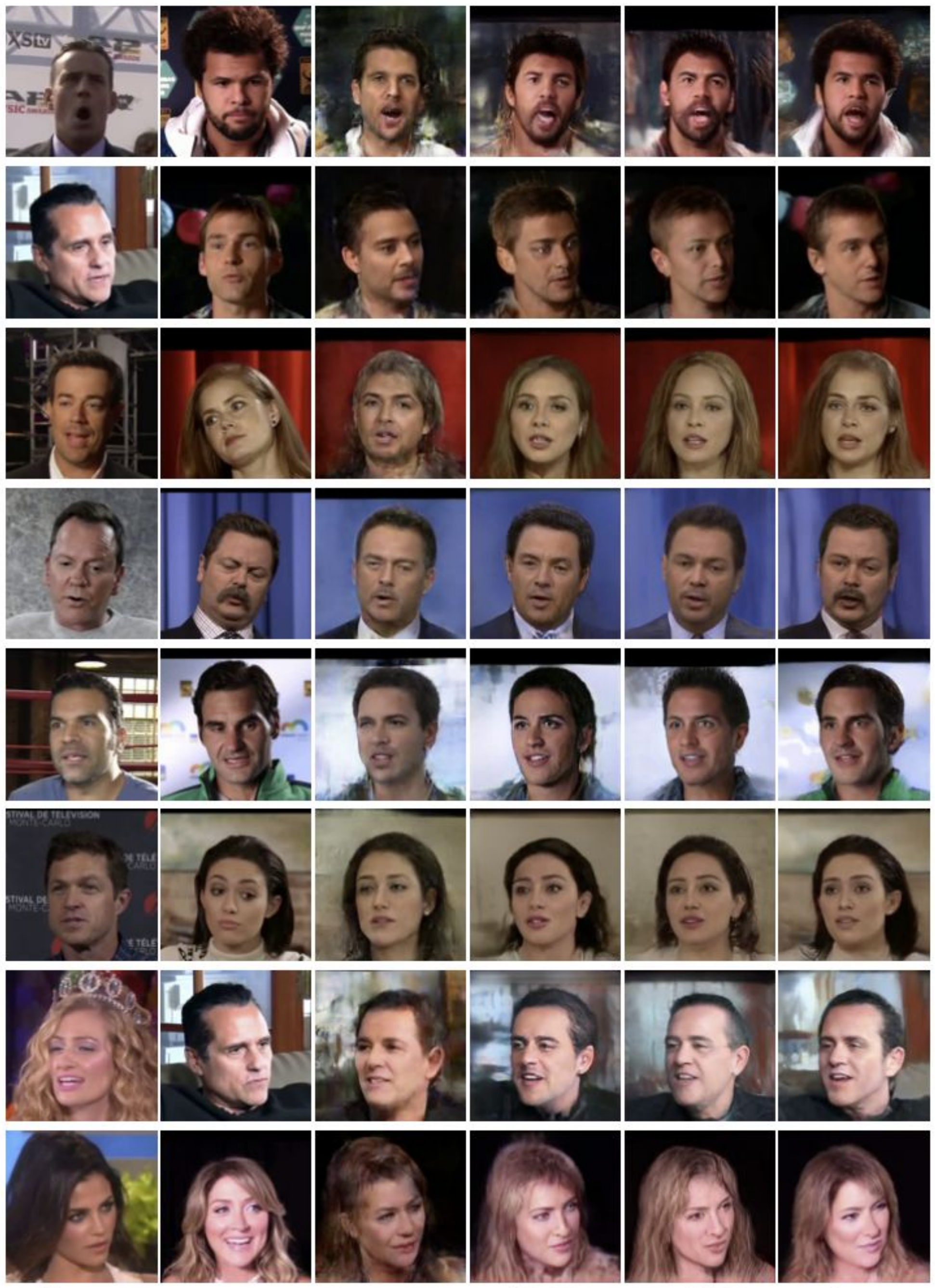
We have demonstrated a technique for portrait reenactment that only requires a single target picture and 2D landmarks of the target and the driver. The resulting portrait is not only photorealistic but also preserves recognizable facial features of the target.

Our comparison shows significantly improved results compared to state-of-the-art single-image portrait manipulation methods. Our extensive evaluation confirms that identity disentanglement of 2D landmarks is effective in preserving the identity when synthesizing a reenacted face. We have shown that our method can handle a wide variety of challenging facial expressions and poses of unseen identities without subject-specific training.

It is achieved by our generator that uses a feature dictionary to translate landmark features into a photorealistic portrait.



[1] Video results could be found in our project website: https://zhangmengyuhi.wixsite.com/talkinghead

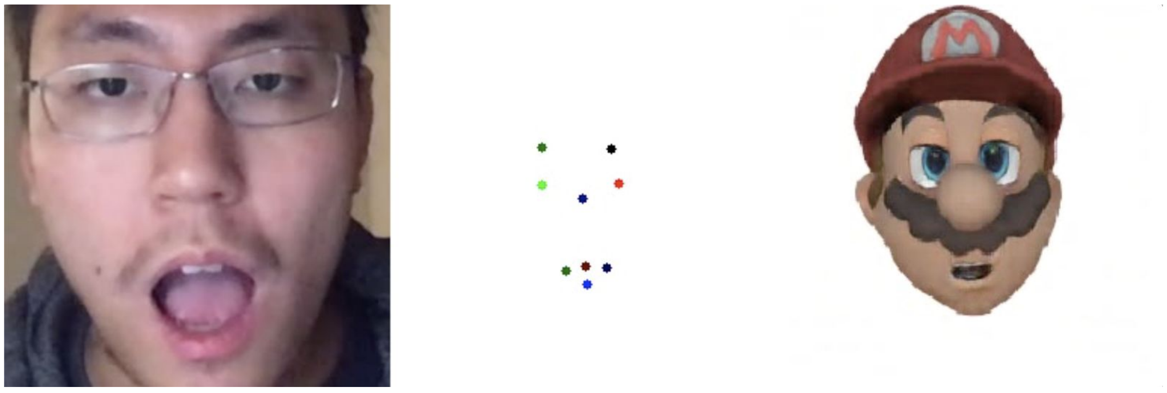


Source Target Pix2pixHD AdaIN FD-GAN-1 Ours

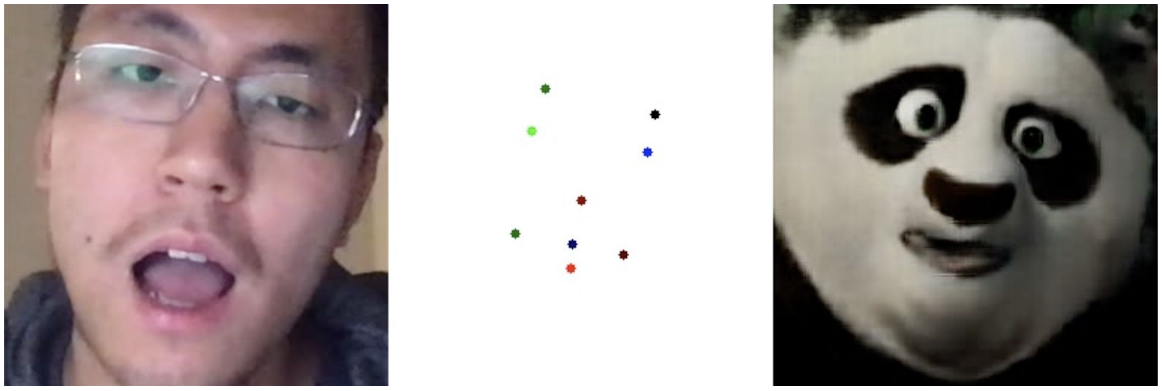
2. Talking Head Models of Game Characters

The general process of this method is to build a mapping function that translates human face to game character face. Firstly, we get the human facial landmark from the human face. We use the same method with the Unity3D method in our project. Secondly, we translate the human facial landmark to mario facial landmark. We design the mapping function manually. The position of each point in mario facial landmarks is a linear combination of human facial landmarks. Finally a Mario face and a panda were generated from their facial landmarks. We use an ancient dark magic called deep learning. The generator network trained by GAN can produce realistic game character faces.

(1) Mario Face Model
Training set size: 9000 pics

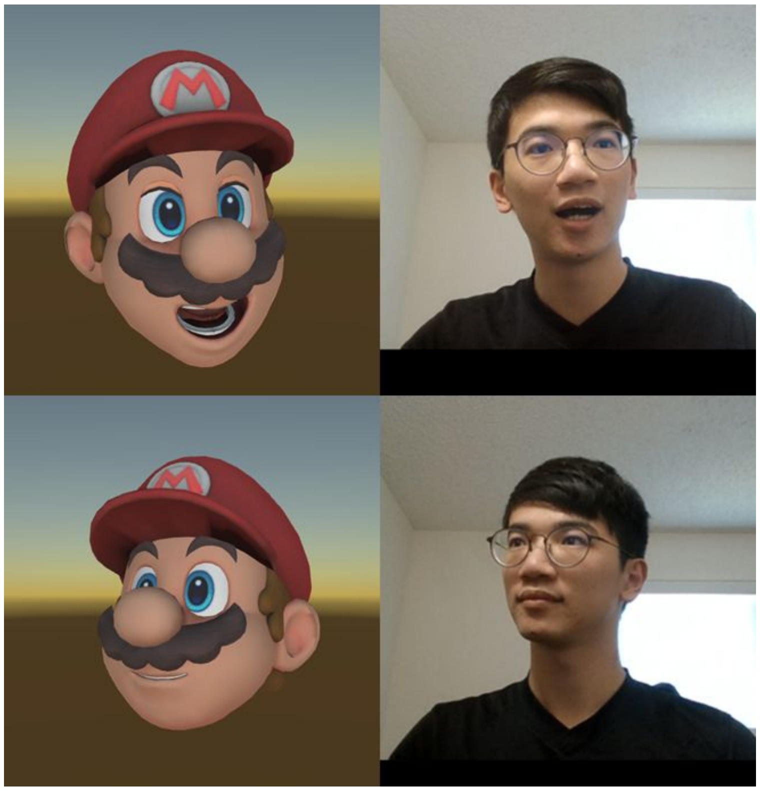


(2) Kung Fu Panda Model
Training set size: 900 pics



3. Talking Head Models of Game Characters in Unity3D

We modify a Mario model and design a function mapping human's facial landmarks to control Mario facial expression. The previous Mario 3D model on the Internet can not meet the requirement of our project. We need to add design weight for each vertex and design the animation for it. After lots of experiments, we figure out the relationship between the human facial landmark and Mario's facial expression. We could control Mario's head to rotate and move forward and backward, also to blink and move his mouth in real time.



Limitations, Conclusions and Future Work

A major limitation of our talking head project is that we have difficulty in reproducing the hair and background faithfully. The appearance of the target photo is transferred to the source landmark image via a feature dictionary, and we rely on the landmark image to determine in which line the feature at each location should be stored and from which line the feature at each location should be read. Since the facial landmarks contain no structural information about hair or background, it is hard to transfer the appearance of these parts of the target portrait accurately. While our method can produce reasonably stable portrait reenactment results from a frame of target and 2D landmarks, the temporal consistency could be further improved by taking into account temporal information of the entire video.

For the talking head models of game characters, currently we can also generate a kung fu panda face which is hard to build a model. We only use less than 1000 images to generate this face. The performance does not meet our expectation, but we anticipate 3 ways that could possibly improve its quality. First and the most promising method is to improve the quality of our training set, such as labelling more facial landmark points of kung fu panda faces. Second, we can use better models, such as vid2vid to improve video stabilization. Third, we can try a larger training set.

For the Unity3D model, it needs lots of time to build the proper model and design the animation for it. When more than two people enter the camera, it is hard to track the right person. Sometimes it will fail to judge whether the eyes are closed or open, due to the rotation. In future works, we can use dual cameras to track a human's facial expression and increase accuracy. We can also do research about how to calculate weights of each vertex, and generate the game character’s facial expression animation automatically.

References

** For some studies that have not been published in official conferences and journals, we didn't include them in the References.*

Averbuch-Elor, H., Cohen-Or, D., Kopf, J., & Cohen, M. F. (2017). Bringing portraits to life. *ACM Transactions on Graphics (TOG)*, *36*(6), 196.

Blanz, V., & Vetter, T. (1999, July). A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (pp. 187-194).

Bao, J., Chen, D., Wen, F., Li, H., & Hua, G. (2018). Towards open-set identity preserving face synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6713-6722).

Cao, C., Weng, Y., Zhou, S., Tong, Y., & Zhou, K. (2013). Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, *20*(3), 413-425.

Dong, H., Liang, X., Gong, K., Lai, H., Zhu, J., & Yin, J. (2018). Soft-gated warping-gan for pose-guided person image synthesis. In *Advances in neural information processing systems* (pp. 474-484).

Dario, N. (2018). Mario (Mario Party 10) - Download Free 3D model by Nintega Dario (@nintegaduigi3) [602c6fb]. *Retrieved from https://sketchfab.com/3d-models/mario-mario-party-10-602c6fbb4f1c4818b9682fb7fd19db25*

Earl, M. (n.d.). Switching Eds: Face swapping with Python, dlib, and OpenCV. *Retrieved from https://matthewearl.github.io/2015/07/28/switching-eds-with-python/*

Huang, X., & Belongie, S. (2017). Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1501-1510).

Huang, X., Liu, M. Y., Belongie, S., & Kautz, J. (2018). Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 172-189).

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125-1134).

Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. *In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1867-1874).*

Kim, H., Garrido, P., Tewari, A., Xu, W., Thies, J., Nießner, M., ... & Theobalt, C. (2018). Deep video portraits. *ACM Transactions on Graphics (TOG)*, *37*(4), 1-14.

Liu, M. Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., & Kautz, J. (2019). Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 10551-10560).

Mallick, S. (2016). Head Pose Estimation using OpenCV and Dlib. *Retrieved from https://www.learnopencv.com/head-pose-estimation-using-opencv-and-dlib/*

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784.*

Park, D. Y., & Lee, K. H. (2019). Arbitrary Style Transfer With Style-Attentional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5880-5888).

Pix2Pix : TensorFlow Core. (n.d.). Retrieved from https://www.tensorflow.org/tutorials/generative/pix2pix

Real Time pose estimation of a textured object. (n.d.). *Retrieved from https://docs.opencv.org/master/dc/d2c/tutorial_real_time_pose.html*

Soukupová, T., & Cech, J. (2016). Real-Time Eye Blink Detection using Facial Landmarks.

Wu, T.(2019). OpenVHead. *Retrieved from https://github.com/TianxingWu/OpenVHead*

Siarohin, A., Sangineto, E., Lathuilière, S., & Sebe, N. (2018). Deformable gans for pose-based human image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3408-3416).

Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., & Sebe, N. (2019). Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2377-2386).

Siarohin, A., Lathuilière, S., Sangineto, E., & Sebe, N. (2019). Attention-based Fusion for Multi-source Human Image Generation.

Shape Predictor 68 face Landmarks, *Retrieved from http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2*

Thies, J., Zollhöfer, M., Nießner, M., Valgaerts, L., Stamminger, M., & Theobalt, C. (2015). Real-time expression transfer for facial reenactment. *ACM Trans. Graph.*, *34*(6), 183-1.

Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., & Nießner, M. (2016). Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2387-2395).

Wiles, O., Sophia Koepke, A., & Zisserman, A. (2018). X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 670-686).

Wu, W., Zhang, Y., Li, C., Qian, C., & Change Loy, C. (2018). Reenactgan: Learning to reenact faces via boundary transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 603-619).

Xu, R., Zhou, Z., Zhang, W., & Yu, Y. (2017). Face transfer with generative adversarial network. *arXiv preprint arXiv:1710.06090*.

Xiang, S., Gu, Y., Xiang, P., He, M., Nagano, K., Chen, H., & Li, H. (2020, April 26). One-Shot Identity-Preserving Portrait Reenactment. *arXiv preprint arXiv:2004.12452*

Xiang, S., & Li, H. (2018). Anime style space exploration using metric learning and generative adversarial networks. *arXiv preprint arXiv:1805.07997*.

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, *23*(10), 1499-1503.

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223-2232).

Zakharov, E., Shysheya, A., Burkov, E., & Lempitsky, V. (2019). Few-shot adversarial learning of realistic neural talking head models. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 9459-9468).

Zhang, J., Zeng, X., Pan, Y., Liu, Y., Ding, Y., & Fan, C. (2019). Faceswapnet: Landmark guided many-to-many face reenactment. arXiv preprint arXiv:1905.11805.

Zhu, Z., Huang, T., Shi, B., Yu, M., Wang, B., & Bai, X. (2019). Progressive pose attention transfer for person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2347-2356).