Real Time Face Swapping with One Target Image

(Group Report)

Yunqiu Xu (z5096489) Qihai Shuai (z5119437) Shaoshen Wang (z5082727)

I. Introduction

In recent years, many approaches have been developed to track, render and animate objects, which can be used in a wide range of area. The aim of this project is to design a robust face swapping system that can animate a face from single image and cast it into source video sequence. A 3D model will be built and fit to get rendered face textures after extracting keypoints, then the it will be treat as the input of a generative adversarial network module to infer face with more details. To improve the clarity, a sub-pixel convolutional neural network module is added to transfer images into super resolution before blending. Our project will be able to perform face swapping with only one target image efficiently, which can represent more details such as teeth and those do not appear on target image.

II. Related work

There has been some implementations [1-3] of face swapping and morphing in image. An simple example [1] is to detect facial keypoints for both source and target faces, align and blend two images. This method is easy to achieve, while sometimes its performance is not good especially when we rotate the head, it is hard to handle via 2D image alignment. In [2] a morphable 3D model is fitted to both the input and target face images, then source face is rendered with parameters estimated from target image. Although fitting a 3D mesh can handle faces in different directions, it's hard to obtain subtle facial changes such as teeth and wrinkles which do not appear in target image. To swap faces automatically, [3] built a common coordinate system from a large face library. For each input image, faces are detected and aligned to coordinate system to find similar candidates. Although this method can swap faces automatically in a rather fast speed, it requires large face dataset to find candidates, which is unrealistic for common use.

In terms of facial performance replacement in video, some efforts [4, 5] have been made in recent years. In [4] a 3D multi-linear model is built to track the facial performance in both videos, then the source is warped to target face and source video is retimed to match target performance. In [5], facial expressions of both source and target video are tracked and reenactment is then performed by fast and efficient deformation transfer between them. Although these works achieve amazing performance, both source video and target video are needed as input. In other words, it will be difficult to apply these methods if there is only one target image instead of obtainable target video.

Generative Adversarial Models [6] have proved to be capable of generating realistic images. Till now different variations of GANs have been used for a wide range of applications including image-to-image translation [7], face generation [8] and completion [9], image inpainting [10] and style transfer [11]. In [12] Pix2Pix GAN is applied to reimplement Face2Face. Although the result is not so good compared with original implementation due to high noise and bad preprocessing, new target faces can be generated from the facial keypoints extracted from source video, which gives us insight for this project.

III. Approach

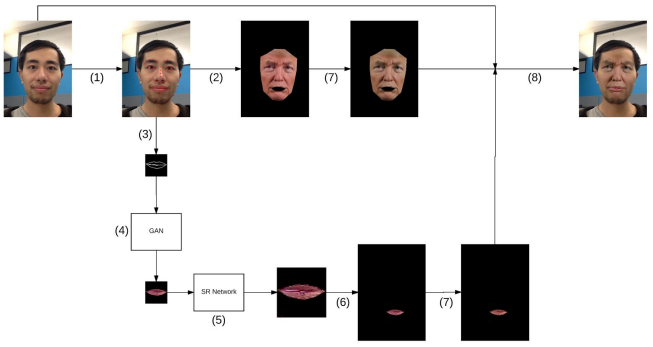

Fig. 1. *Workflow*

Fig.1. shows our workflow. The first frame of source video is treated as model frame, and a 3D model will be built with it as well as "target face". During processing, facial keypoints will be detected and tracked from source frame (1) first , then the 3D model will be fitted from these keypoints and rendered as facial texture (2) .Simultaneously, the mouth part of landmarks will be cropped and connected as polygons (3), which will be treated as the input of Generative Adversarial Network (4). The generated mouth will then be processed via super resolution module to achieve higher resolution (5) , and remapped to original position (6). Color transformation (7) will be performed to both the facial texture and generated

mouth, finally these 2 parts will be blended with original frame to get swapped face (8).

A. Facial Keypoints Detecting

Ensemble regression trees (ERT) [13] will be used to extract facial keypoints from source image. This framework is based on gradient boosting for learning ensemble of regression trees, where a two-level boosted regression will be built to make it less sensitive to initialization. Eq.1-Eq.3 shows how the update vector S from the image is predicted and updated by regressor r. Regressor r will be trained via gradient tree boosting algorithm with a sum of square error loss, this process is iterated until a cascade of T regressors r(0)…r(T-1) are learnt which when combined give a sufficient level of accuracy. By using shape-indexed features and a correlation-based feature selection method, ERT can estimate the face's landmark positions directly from a sparse subset of pixel intensities in real-time, which is much effective and efficient than model based methods.

$$\hat{S}^{(t+1)} = \hat{S}^{(t)} + r_t(I, \hat{S}^{(t)}) \quad (1)$$

$$\hat{S}_i^{(t+1)} = \hat{S}_i^{(t)} + r_t(I_{\pi_i}, \hat{S}_i^{(t)}) \quad (2)$$

$$\Delta S_i^{(t+1)} = S\pi_i - \hat{S}_i^{(t+1)} \quad (3)$$

B. Image Rendering

The method described in [14] will be utilized to perform image rendering. Before processing all frames, a 3D model is initialized whose input consists of model frame, target image and some parameters loading from CANDIDE [15]. After getting facial landmarks, 2D keypoints will be transferred to 3D and fit the model. The model fitting can be accomplished by minimizing the difference between the projected shape and the localized landmarks via Gauss Newton method [16]. Eq.4-Eq.5 shows this process, where r, β are column vectors, and J is entry for the Jacobian matrix. Then the model will be projected into 2D space to get texture coordinates, as shown in Eq.6.

$$\beta^{(s+1)} = \beta^{(s)} + (J_f^T J_f)^{-1} J_f^T r(\beta^{(s)}) \quad (4)$$

$$(J_r)_{ij} = \frac{\partial r_i(\beta^{(s)})}{\partial \beta_j} \quad (5)$$

$$s = \alpha P(s_0 + \sum_{i=1}^{i=n} w_i * S_i) + t \quad (6)$$

C. Mouth Generating

In this part, Pix2Pix GAN will be applied to enable end-to-end synthesis of facial textures conditioned on the target identity and source facial expressions. The architecture is shown in Fig.2. Similar to conditional GAN [17], the network consists of a discriminator D and generator G. D learns to classify between real and synthesized pairs while G learns to generator fake image to fool the discriminator. The loss function is shown in Eq.7, where x is the input face texture obtained from last step, y is target output and z is noise to generate "fake images" based on x. In addition, L1 constraint Eq.8 is used to make sure the generated image is similar to both output image and input image, and the final object is shown in Eq.9. In order to reduce noises, skip connections are added between encoder layers and decoder layers.

$$L_{GAN}(G, D) = E_{y \sim p_{data}(y)}[logD(y)] + E_{x \sim p_{data}(x), z \sim p_z(z)}[log(1 - D(G(x, z)))] \quad (7)$$

$$L_{L_1}(G) = E_{x,y \sim p_{data}(x,y), z \sim p_z(z)}[||y - G(x, z)||_1] \quad (8)$$

$$G^* = argmin_G max_D L_{cGAN}(G, D) + \lambda L_{L_1}(G) \quad (9)$$

Compared with [12], which use all facial keypoints to generate the whole image, we treat only mouth part as input and output to make inference more accurate. The architecture of original GAN [18] is simplified by removing 2 encoders and decoders. In this way the network can treat 64 * 64 images as input instead of large images (256 * 256). The modified architecture of Pix2Pix GAN is shown in Fig.2. , which makes the generating process much faster with acceptable decrease of accuracy.

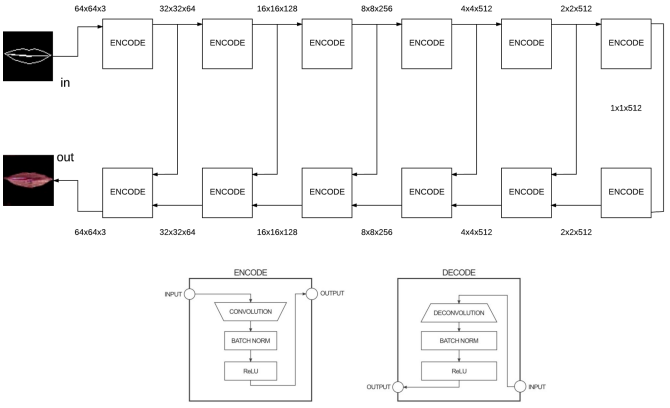

Fig. 2. The architecture of modified Pix2Pix GAN

D. Super Resolution Processing

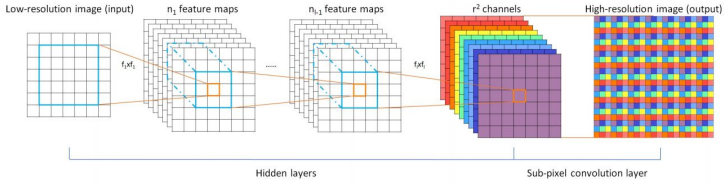

Fig. 3. The architecture of ESPCN

One drawback of GAN is that the generated images are in low resolution. In order to handle this, Efficient Sub-Pixel Convolutional Neural Network (ESPCN) [19] is applied after GAN. As shown in Fig.3, compared with previous methods, where the low resolution(LR) input image is upscaled to the

high resolution(HR) space using a single filter before reconstruction, this framework extracted feature maps in LR space. In addition, an efficient sub-pixel convolution layer is introduced to learn an array of upscaling filters to upscale the final LR feature maps into HR output. By doing this the framework can transform images to higher resolution in rather fast speed.

E. Blending

The differences in skin-tone and lighting between images will lead to discontinuity around the edges of the overlaid region. Color transformation will be performed before blending to make sure color among facial texture, generated mouth and original frame is similar. Then alpha blending [20] will be applied to merge 3 images.

IV. Experiment Details

The training process of models is performed on deep learning work station with 2 Nvidia Tesla K40 GPUs, and the combined system is tested on macbook pro 2015.

In our project, a dlib package implementation of ensemble regression tree [21] is applied to get facial keypoints. For Pix2Pix GAN, we build a training dataset with 300 images from some videos of Trump Weekly Talk as well as a testing dataset with 30 images. After getting frames, mouth landmarks are cropped and resized to 64 * 64, then input images will be generated by connecting these points as polygons, as shown in Fig.4. The model is trained by 500 epochs (150000 iterations) and stored per 100 epochs. In terms of super resolution processing, the pytorch implementation of ESPCN [22] is trained on BSD300 dataset [23] with scale factor set as 3 (i.e. the resolution of output will be 3 times as large as the input), then this part will be modified to make it suitable for our project.

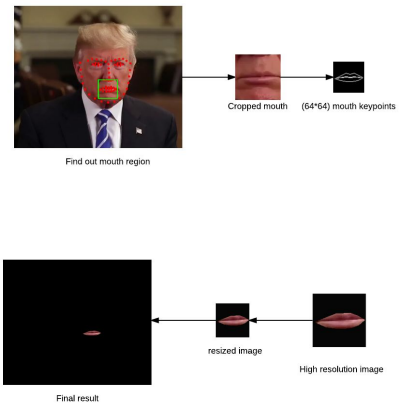

Fig. 4. Mapping and remapping

During testing, following optimisations will be taken to improve the performance:

1) The input frame from webcam is resized to half (320 * 480)
2) Instead of performing super resolution on the whole image, this part is only used for generated mouth.
3) Skip frame is used per 3 frames
4) GAN and SR are used per 6 frames, which can speed up the process significantly, and make the face “more stable”

V. Results and Discussion

Fig.5. shows 68 facial keypoints detected by ERT and Fig.6. shows the rendering result. Compared with face alignment which is only performed on 2D level, 3D model rendering achieves more real result especially when we rotate the face. Due to the lack of in-mouth information for target image, the mouth part for rendered image is empty, which is the motivation to apply GAN.

Fig. 5. Facial keypoints detected by ERT

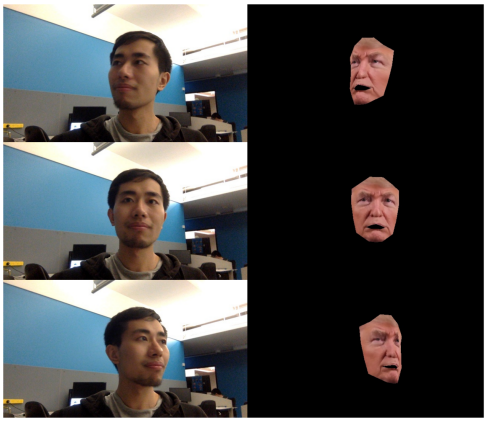

Fig. 6. Facial texture in different directions

The test result of GAN is shown in Fig.7. By comparing mouths generated by models with different epochs, we can find that the model trained by 500 epochs achieves best performance, that the result is rather similar to ground truth. We can also find that the L1-loss keeps decreasing, which means the performance of model can be better by training more epochs. The test result of super resolution is shown in Fig.8., where the output images are clearer than input ones. In terms of speed, it’s slow to perform SR on large image and we just apply it on mouth part.

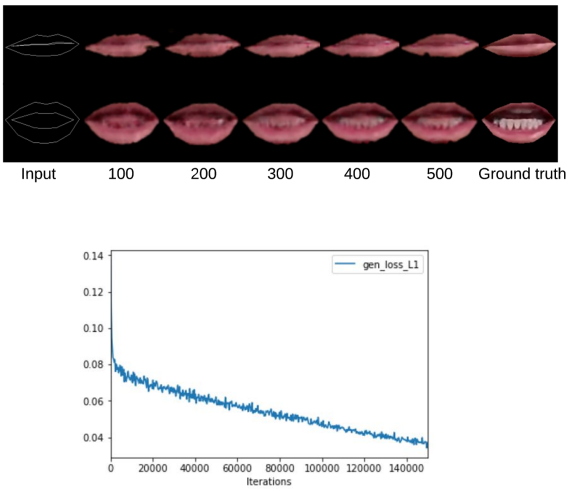

Fig. 7. Results for GAN with different training epochs

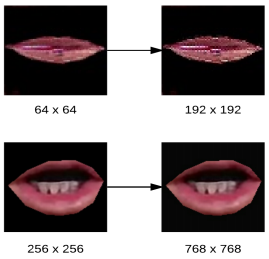

Fig. 8. Results for ESPCN

The overall performance is shown in Table 1, that our fastest model can achieve about 30 FPS, which satisfies the demand of real-time (24 FPS)

Input size	Mouth size (GAN)	Scale factor (SR)	Skip Frame	FPS
960 * 640	-	-	-	~23
960 * 640	256 * 256	-	-	~2
480 * 320	256 * 256	-	-	~3
480 * 320	128 * 128	-	-	~8
480 * 320	64 * 64	-	-	~20
480 * 320	64 * 64	4	-	~13
480 * 320	64 * 64	3	-	~16
480 * 320	64 * 64	3	3	~30

Table.1. *Performance of project with different parameters*

VI. TASK SUBDIVISION

Table 2 shows task subdivision of our group:

Group Member	Task
Yunqiu Xu	Build the framework
	Modify and train GAN
	Super resolution
	Image blending
Shaoshen Wang	Build dataset for GAN
	Mouth mapping and remapping
Qihai Shuai	Facial keypoints detection
	Image rendering

Table.2. *Task subdivision*

VII. CONCLUSION AND FUTURE PERSPECTIVES

We successfully build a face swapping system with following features:

1) This system takes only one target face as input.
2) It is able to handle face rotation
3) It is able to infer in-mouth information which does not appear in target face.
4) Face swapping can be performed on a general laptop efficiently with real-time performance.

There are also improvements can be done in the future:

1) Currently we can only handle limited faces due to lack of data. We can generalize our model by using larger dataset such as Facial Action Coding System (FACS) [24]
2) The prediction of GAN can be more accurate by modifying hyper parameters.
3) The system can be more robust to handle occlusion and multi faces.

REFERENCES

[1] https://github.com/matthewearl/faceswap/blob/master/faceswap.py

[2] Blanz V, Scherbaum K, Vetter T, et al. Exchanging faces in images[C]//Computer Graphics Forum. Blackwell Publishing, Inc, 2004, 23(3): 669-676.

[3] Bitouk D, Kumar N, Dhillon S, et al. Face swapping: automatically replacing faces in photographs[J]. ACM Transactions on Graphics (TOG), 2008, 27(3): 39.

[4] Dale K, Sunkavalli K, Johnson M K, et al. Video face replacement[J]. ACM Transactions on Graphics (TOG), 2011, 30(6): 130.

[5] Thies J, Zollhofer M, Stamminger M, et al. Face2face: Real-time face capture and reenactment of rgb videos[C]//Proceedings of the IEEE

Conference on Computer Vision and Pattern Recognition. 2016: 2387-2395.

[6] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]//Advances in neural information processing systems. 2014: 2672-2680.

[7] Pix2Pix Isola P, Zhu J Y, Zhou T, et al. Image-to-image translation with conditional adversarial networks[J]. arXiv preprint arXiv:1611.07004, 2016.

[8] Gauthier J. Conditional generative adversarial nets for convolutional face generation[J]. Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester, 2014, 2014(5): 2.

[9] Li Y, Liu S, Yang J, et al. Generative Face Completion[J]. arXiv preprint arXiv:1704.05838, 2017.

[10] Yeh R, Chen C, Lim T Y, et al. Semantic image inpainting with perceptual and contextual losses[J]. arXiv preprint arXiv:1607.07539, 2016.

[11] Gatys L, Ecker A S, Bethge M. Texture synthesis using convolutional neural networks[C]//Advances in Neural Information Processing Systems. 2015: 262-270.

[12] https://medium.com/towards-data-science/face2face-a-pix2pix-demo-that-mimics-the-facial-expression-of-the-german-chancellor-b6771d65bf66

[13] Kazemi V, Sullivan J. One millisecond face alignment with an ensemble of regression trees[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 1867-1874.

[14] https://github.com/MarekKowalski/FaceSwap

[15] Ahlberg J. Candide-3-an updated parameterised face[J]. 2001.

[16] https://en.wikipedia.org/wiki/Gauss%E2%80%93Newton_algorithm

[17] Mirza M, Osindero S. Conditional generative adversarial nets[J]. arXiv preprint arXiv:1411.1784, 2014.

[18] https://github.com/affinelayer/pix2pix-tensorflow

[19] Shi W, Caballero J, Huszár F, et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 1874-1883.

[20] 15-463: Computational Photography Alexei Efros, CMU, Spring 2010

[21] https://github.com/davisking/dlib

[22] https://github.com/pytorch/examples/tree/master/super_resolution

[23] Movahedi V, Elder J H. Design and perceptual validation of performance measures for salient object segmentation[C]//Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on. IEEE, 2010: 49-56.

[24] Ekman P, Friesen W V. Facial action coding system[J]. 1977.