# Double Encoder Conditional GAN for Facial Expression Synthesis

**Conference Paper** · July 2018

DOI: 10.23919/ChiCC.2018.8483579



**5 authors**, including:

Xuanji He
School of Information Science and Engineering, Central South University
**3** PUBLICATIONS **60** CITATIONS

SEE PROFILE

Proceedings of the 37th Chinese Control Conference
July 25-27, 2018, Wuhan, China

# Double Encoder Conditional GAN for Facial Expression Synthesis

Mingyi Chen, Changchun Li, Ke Li, Han Zhang, Xuanji He
School of Information Science and Engineering, Central South University, Changsha, 410083, China
E-mail: lcczndx@csu.edu.cn

**Abstract:** Photorealistic facial expression synthesis from single face image is already a highly challenging research work, in part due to a paucity of labeled and paired facial expression samples. Most existing facial expression synthesis works attempt to learn the transformation between expression domains and thus would require the paired samples as well as the labeled query image. In this paper, we propose the Double Encoder Conditional GAN (DECGAN) for facial expression synthesis. Generative Adversarial Networks (GANs) have demonstrated to successfully approximate complex data distributions. And cGANs, which contain external information, can determine the specific relationship between images. This work inspires us to modify the structure of GAN, and use the target facial expression feature as a condition. In this work, we propose two encoders to encode the original expression and the target expression, respectively, to extract the latent vectors and conditional labels features of the real image. In the meantime, associative learning is used to associate unpaired original emoticons with target emoticons in the database and to share identities.

**Key Words:** Synthesis, Facial Expression, GAN, Double Encoder, Associative Learning

## 1 Introduction

Facial expression synthesis aims at showing a face image different from a neutral expression, but still preserve personalized features of the face. While preserving identity information, the synthesis of facial expressions that represent photorealistic photos from a single static face will have a major impact on the area of emotional computing. There are still many interested researchers involved in this field, although this issue is facing great challenges. The lack of a tagged facial expression database makes this problem challenging. Researchers have strict requirements on datasets, i.e. face images of the same person at different expressions, and some even need long-term paired samples, which are very difficult to collect. For example, the Cohn-Kanade AU-Coded Facial Expression Database [1] only includes 486 sequences from 97 subjects. And each sequence begins with a neutral expression and proceeds to a peak expression. Although the database contains a large variety of image facial expressions, it is difficult to discern facial expressions and identity information due to their limited number of subjects. Given the training data, we need to divide it into different expressions and learn the conversion relations among the groups. Therefore, it is necessary to have a labeled picture to locate the picture correctly.

The methods of face expression synthesis are mainly divided into two categories [2]. One category handles the problem by warping images [3], instead of generating them from the latent vector. These methods [4, 5] mainly generate facial expressions by affecting parts of the face, not the entire image. Bitouk et al. [7] warp the original images by learning a mapping from a set of images similar. Recently , This idea is applied to a Variational Auto-Encoder (VAE) [6] to learn the flow field. However, pairing datas with different expressions on the subject are required to train the model. The other category generates target facial expressions by using synthetic techniques. In this category, deep learning-based methods [8, 9] are mainly used. Peng et al. [10] propose a synthesis CNN to generate non-frontal view from a single frontal face. Li et al. [11] apply a temporal restricted Boltzmann machines based model to emotional facial expression transfer.

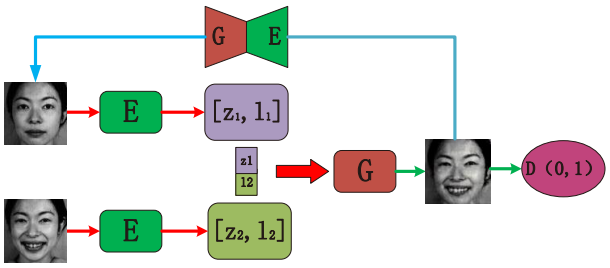



Fig. 1: Illustration of the DECGAN. The input faces are coded as $z_1$ and $z_2$ by an encoder, which represents the personality. An encoder extracts the emotional feature label $l_1$ and $l_2$. The generator $G$ takes as input both the latent vector $z_1$ and the label vector $l_2$. Associative learning is achieved through reconstruction.

With the recent development of Generative Adversarial Networks (GANs) [12], GANs have been able to generate high quality samples in natural environments, such as handwriting fonts, face images, landscapes. In addition, GANs have been successfully applied to face image synthesis [13–15]. Unlike VAE, GANs are particularly interesting because they are optimized directly to produce the most reasonable and realistic data. GANs can use conditional extended GANs to explicitly control the generated image features. These efforts typically use the encoders of GANs to find low-dimensional representations of face images in latent space and then decode them by manipulating potential vectors to generate new images. If we control the incoming parameters of potential vectors during training, when generating potential vectors, we can change these parameter values in order to manage the necessary image information in the picture, which is called conditional Generative Adversarial Network model [17].

In this paper, we study the synthesis and transformation of facial expressions from the perspective of image generation model. Recently, the use of GANs for image-to-image conversion has been extremely successful, and we use GANs for Face Expression synthesis. We propose DECGAN to learn the diversity of faces. Specifically, the learning by the game generator and the discriminator generates an image similar to the original sample. As illustrated in Fig. 1, the generator consists of encoder and decoder. The input of the encoder is face image, and the face image is first mapped to a latent vector through convolutional neural network, and then the vector is projected to the face of human being with emotion feature through deconvolution generator. Discriminator is a convolutional neural network. During training the network, we don't use the image database of the same subject but different expressions. Associative learning is used to associate unpaired original emoticons with target emoticons in the database and to share identities. We propose two encoders that map the identity information of the original images and the emotion feature information of the target images into latent vectors. The training of generator and discriminator can generate face images of the same theme with different facial expressions while greatly preserving identity information.

## 2 Related work

### 2.1 Facial Expression Synthesis

The study of emoticons and their generation method is an indispensable task for emotional interaction. The facial expression generation problem is divided into two parts, face modeling and expression synthesis. The traditional face modeling method normalizes the facial expression generation problem to the deformation operation of the image. The basic idea is similar to that of finite element mesh. For example, Matthews et al. [18] implement 2D face modeling by using complex meshes and applying statistics-based methods. Raouziou et al. [19] propose an animated frame of different faces, which requires 84 points, and have done a lot of work in this area. Facial expression synthesis can be achieved through virtual animations and realistic face expressions. Generally there are two methods of expression synthesis. One method is to convert the expression. The main method to extract the geometric features of face from RGB-D space. Bitouk et al. [7] propose a powerful method in which the author automatically selects a set of images similar to the appearance of the pose and forms the source image. Another method is based on physical modeling techniques, mainly by generating facial skin and muscle movement simulation [20, 21], such as Facial Action Coding System (FACS) [20].

Nowadays, the rapid development of neural network brings a new approach to image generation, because facial motions are based on real-world facial motion data. In contrast to physics-based modeling, neural networks can potentially produce more realistic and natural emotions by using large facial expression databases. For example, a time-limited Boltzmann machine model [11] is applied to emotion facial expression transfer. The method can encode a complex non-linear mapping from the movement of one neutral facial expression to another emotional facial expression. Zhang et al. [22] use a Conditional Adversarial Autoencoder to learn a flow map, which can distort the source image with a facial expression. Different labels of the target image lead to different new faces of the image. In addition, researchers use GAN to change facial features such as hair, mouth, and eyeglasses or to artificially age a face.

### 2.2 Generative Adversarial Network

Strictly speaking, a GAN framework, at a minimum, has two components, one for generation model $G$ and one for discriminant model $D$. During the training, a sample generated by the model and a real sample will be randomly sent to the discriminant model $D$ (or a batch). The goal of discriminant model $D$ is to identify the true sample as correctly as possible (output $true$, or 1) and to identify the resulting false sample as correctly as possible (output $false$, or 0). The goal of generation model is the opposite of the discriminant model. In this way, $G$ and $D$ form a min-max game, during which both parties are constantly optimizing themselves until they reach equilibrium. The above process can be expressed as follows:

$$\begin{aligned}\min_G \max_D V(D,G) =& E_{x\sim p_{data}}[\log D(x)]+ \\ & E_{z\sim p_z}[\log(1-D(G(z)))]\end{aligned} \tag{1}$$

where $z$ is a vector noise sampled from a known simple distribution $p(z)$ .

One of the biggest problems of GAN is that it's hard to control. For example, the training of GAN can easily lose direction and the generated images are often difficult to understand. A prior distribution is needed on the GAN before training. The structure is similar to the structure of a self-encoder such as Variational Auto-Encoder (VAE) [6] and Adversarial Auto-encoders (AAE) [23]. In the past three years, several methods have been proposed to improve the original GAN from different perspectives. For example, GAN can be extended with conditional GAN (cGAN) [17], which is the first job after the original GAN, and the idea is simple: cGAN turns the original build into a build based on some extra information. This work inspires us to modify the structure of GAN, and use the target facial expression feature as a condition.

## 3 Approach

In this section, we first present the struction of the Double Encoder Conditional GAN (DECGAN) that generates realistic face images. As illustrated in Fig. 2, this structure has two independent encoders, which can map the identity information of the original facial expression and the emotion features of the target domain facial expression as latent vectors. In Section 3.1, we give the dual encoder in the framework. In Section 3.2, we propose associative learning that solves the problem of unpaired data training.

### 3.1 Dual encoder

The framework of GANs contains only generators and discriminators, and there is no ability to map a real picture $x$ to its latent vector $z$. We adopt a convolutional neural network as the encoder of the network structure. The convolutional neural network consists of one or more convolutional layers and a top fully connected layer (corresponding to a classical

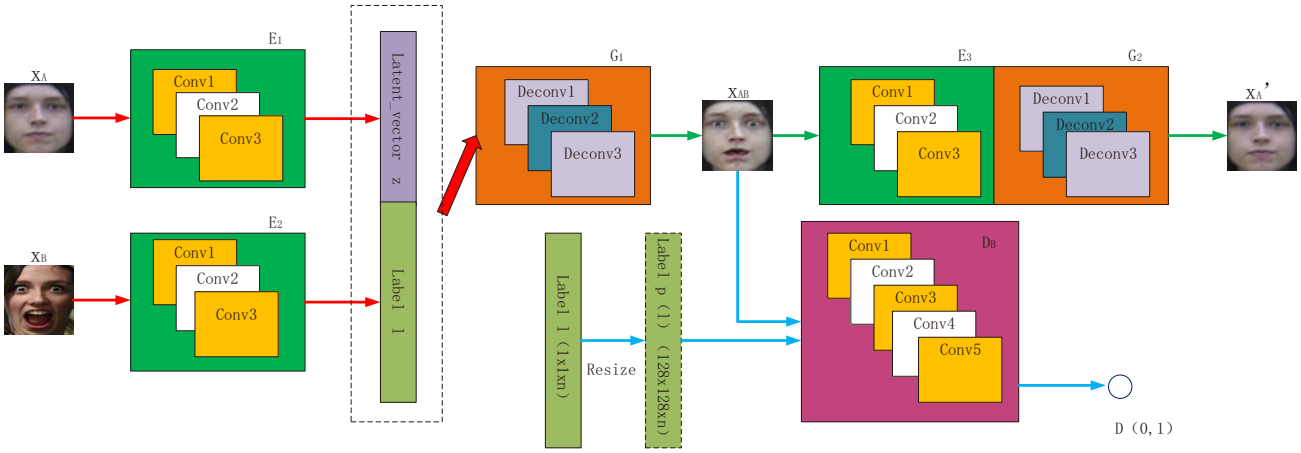



Fig. 2: The system structure of the proposed DECGAN. The Encoder $E_1$ maps the identity information of input face to a latent vector $z$. The Encoder $E_2$ extracts the target domain image emotional feature label $l$. The generator restores the low-level features from the eigenvector by using the deconvolution layer. The generator $G_1$ takes both the latent vector $z$ and the target label vector $l$ as input. Reconstruct the generated image by another generator, to better enhance the association of the target domain and the original domain. The reconstruction loss function is used to describe the difference between the reconstruction effect after the two generators and the original real sample. In the discriminator, $l$ is concatenated in the first convolutional layer, and determines whether the generated image is the original image or the generated image.

neural network), as well as associated weights and pooling layers. This structure allows us to have a latent representation $z$ from a real image $x$, by changing the latent vector $z$ to guide the change of the generated image $x$.

Our approach consists of training two separate encoders, $E_1$ and $E_2$. Both encoders use the convolutional neural networks to extract the features of the input images, where the output of the encoder $E_1$ is the high level identity feature latent vector $z$ of the original domain image and the output of the encoder $E_2$ is the emotion feature label $l$ of the target domain images. The double encoders solve the problem of the random adoption of $z$. The generator $G_1$ takes as input both the latent vector $z$ and the target label vector $l$, and generates a face with specific personality.

### 3.2 Associative learning

Associative learning is proposed in this structure, which associates unpaired data with feature similarity. Different domains have different image data, but they have a consistent set of intrinsic properties, so we correlate the data from the target domain to the original domain by reconstructing.

We start from the approach of learning by association [24], which is geared towards semi-supervised training. There are two fields of data, the original domain data set $X = x_i$, the target domain data set $Y = y_j$, and two samples $x$, $y$ which belong to the source and target domains. Extracting features by encoders $E_1$ and $E_2$, and mapping them to latent vector yields: $A_i = E_1(x_i)$, $B_j = E_2(y_j)$. The similarity of these two samples from different domains can be expressed as the scalar product of $A_i$ and $B_j$. The transition probability of sample $X_i$ and sample $Y_j$ as formula:

$$P_{ij}^{ab} = P(B_j|A_i) \tag{2}$$

Further, the probability of the sample of the source domain associated with the target domain sample can be got.

The basis of associative similarity is the two-step roundtrip probability. The first step is starting from the latent vector $A_i$ of the labeled source domain to the latent vector $B$ of target domain and the second step is returning to another latent vector $A_j$ via the target domain latent vector $B$.

Through reconstruction:

$$P_{ij}^{ab} = (P^{ab} \cdot P^{ba}) \tag{3}$$

The automatic encoder is adopted as the network structure during reconstruction. In order to maintain circular consistency [15], the category of the reconstructed face images must be the same as the original domain category.

Specifically, the input images from the source domain $X$ map to the target domain $Y$ through two generators and convert them into corresponding images. In addition, the target domain and the original domain need to share features that can be used to map this output image back to the input image. Therefore, another generator must be able to map this output image back to the original domain for associative learning.

### 3.3 Loss function

The loss function is designed according to the actual purpose. The loss function contains the following sections. First, the discriminator allows all the corresponding categories of original images, corresponding 1 to the output set. Second, the discriminator rejects all generated images that fool over the pass, and the corresponding output is set to 0. Third, the generator fools the discriminator to allow fooling through all the generated images. Finally, the generated image must retain the characteristics of the original image. Generator $G_1$ generates a fake picture, and another generator $G_2$ can restore the original image. This process must satisfy the circular consistency [15].

**Adversarial Loss.** An adversarial loss [16] is adopted for matching the distribution of generated images to the data distribution in the target domain. The purpose is to make the

generated image indistinguishable from the real image.

$$L_{adv} = E_x[\log D(x)] + E_{x,l}[\log(1 - D(G_1(x, l)))] \quad (4)$$

Where the generator $G_1$ takes both the latent vector $z$ and the target label vector $l$ as input. while $D$ tries to distinguish between real images $y$ and fake images $G_1(x)$. In this formula, $G$ tries to minimize this function, while $D$ tries to maximize this function.

**GAN genetator loss.** With a given input image $x_A$ and a target domain label $l$, we translate $x$ into the output image $y$. The generator loss function is defined as

$$L_{G_1} = E_{x,l}[-\log D(G_1(x_A, l))] \quad (5)$$

The final generator is able to increase the output of the discriminator to the generated image. The function of the generator is to make the discriminator output value of the generated image as close to 1 as possible, so the generator $G_1$ tries to minimize the loss.

**Reconstruction Loss.** We adopt a reconstruction loss [15] for describing the difference between the generated image and the original image obtained with another generator.

$$x_{AB} = G(x_A, l) \quad (6)$$

Where the generator $G_1$ takes both the input image $x$ and the target label vector $l$ as input. And the generator $G_2$ tries to reconstruct the original image $x_A^{'}$.

$$\begin{aligned} x_A^{'} &= G_2(x_{AB}, l^{'}) \\ &= G_2(G_1(x_A, l), l^{'}) \end{aligned} \quad (7)$$

$d$ is minimized to make the mapping of the original domain and the target domain meaningful. And $d$ is a regular function of $l_1$.

$$L_{const} = d(G_2(G_1(x_A, l), l^{'}), x_A) \quad (8)$$

$$L_{const} = E_{x,l,l'}[||x_A - G_2(G_1(x_A, l), l^{'})||_1] \quad (9)$$

# 4 Experimental Results

## 4.1 Datasets

**CK+.** The Extended Cohn-Kande Dataset (CK+) [1] is a complete facial expression dataset containing the expression label and the label of the Action Units. This data set consists of 123 subjects ages ranging from 18 to 50 and 593 image sequences. Most of the images in the data set are grayscale images with a positive view of the size of 640×490. Each sequence begins with a neutral expression picture, ending in an extreme expression picture. If one person has several sequences under the same expression type, we choose only one sequence for this person. We extract the first and last frames of each sequence as the training data according to the label information. We construct 7 domains using the following attributes: Anger, Neutral, Disgust, Fearness, Happiness, Sadness and Surprise.

**RAF-DB.** Real-world Affective Faces Database (RAF-DB) [25] is a large facial expression database containing around 30K different facial images downloaded from the Internet. Based on the crowdsourcing annotation, each image is individually labeled by about 40 annotators. The RAF-DB contains 29672 number of real-world images, including 7 kinds of basic emotions. In order to meet the training requirements of deep learning, the database is divided into a training set and a test set. And the size of training set is five times larger than test set. We align the faces using the detected landmark [26], then crop the images and resize them to 256 × 256. Finally, we normalize the pixel values to the range of [-1,1], and we use random flip to enhance training data to ease overfitting.

## 4.2 Implementation Details

The input image first goes through a three-layer convolutional encoder $E_1$, which maps the raw face image to a regularized latent space $z$. The latent vector $z$ and the target label vector $l$ which is extracted from the target domain are then concatenated to generate the target face through a three-layer deconvolutional decoder $G_1$. And the activation function between each layer is Leaky ReLU. We train the networks by using the Adam optimizer [27], with learning rate of 0.0002, $\beta_1$ = 0.5 and $\beta_2$ = 0.999. The output image size used as a baseline is 64×64. To speed up training, we store all the generated images for each of the previous domains and use only one image at a time to calculate the error. First, we fill the image library one by one to make it complete, and then randomly replace the image in a library with the latest generated image and use this replacement image as training for that step. See the Table 1 and Table 2 for more details about the network architecture.

Table 1: Detailed Encoder and Generator Architecture

| Layer name | Kernel | Stride | Filters | Activation |
|---|---|---|---|---|
| Conv1 | 7×7 | 1×1 | 64 | Leaky ReLU |
| Conv2 | 3×3 | 2×2 | 128 | Leaky ReLU |
| Conv3 | 3×3 | 2×2 | 256 | Leaky ReLU |
| Concatenation | Concatenate z and l on 1st dimension | | | |
| Deconv1 | 3×3 | 2×2 | 256 | ReLU |
| Deconv2 | 3×3 | 2×2 | 128 | ReLU |
| Deconv3 | 7×7 | 1×1 | 64 | ReLU |
| Deconv4 | 7×7 | 1×1 | 8 | Tanh |

Table 2: Detailed Discriminator Architecture

| Layer name | Kernel | Stride | Filters | Activation |
|---|---|---|---|---|
| Conv1 | 7×7 | 1×1 | 64 | Leaky ReLU |
| Concatenation | Replicate p(l) and concatenate to conv1 | | | |
| Conv2 | 7×7 | 2×2 | 128 | Leaky ReLU |
| Conv3 | 7×7 | 2×2 | 256 | Leaky ReLU |
| Conv4 | 7×7 | 1×1 | 512 | Leaky ReLU |
| Conv5 | 7×7 | 1×1 | 1 | Sigmoid |

The network architectures of DECGAN are shown in Table 1 and Table 2. The two encoders use a 3-layer convolutional layer with activation function leaky ReLU on each layer. The generator uses four deconvolution layers, and the activation function for each layer is ReLU except the last layer. We use instance normalization for the encoder and generator network. The discriminator network uses five layers of convolutional layer and we use Leaky ReLU with a negative slope of 0.01. The activation function of the last

Fig. 3: A comparison of neutral and smiling face expressions from ck+ database with the expressions generated by our structure. For each input, we show three rows of images. Neutral expression from ck+ database(top), smile from ck+ database(midde), smiley expressions generated by our model(bottom).

layer is sigmoid. In the end, the model is implemented using Tensorflow [28].

### 4.3 Facial Expression Synthesis

We conducted the spontaneous human face expression synthesis experiment on the ck+ database and RAF-DB. For ck+ database , each sequence contains images with expressions ranging from neutral to extreme. If one person has several sequences under the same expression type, we choose only one sequence for this person. The ck+ dataset has different expressions from the same person. For RAF-DB, we classified facial expressions based on the image label information. Because both databases have limited images from neutral to extreme, we crawled images from Internet to expand the databases.

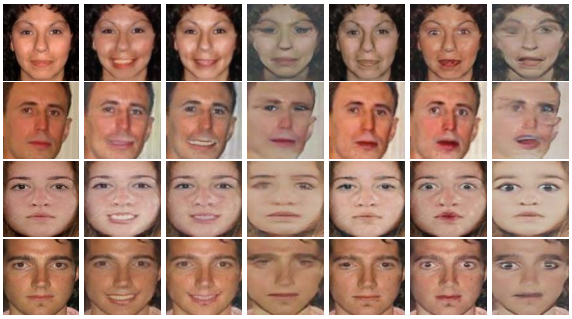
Some example results are shown in Fig. 3. Because of the paired data training, the ground truth images(Target expression) and the results(Generation expression) from the DECGAN are shown in the second and third row. By comparison, we can see our model is suitable for ck+ database. In additon, the generated images and the actual images are very similar in expression and identity.

Fig. 4 shows the model's ability to transfer the expression. Unlike ck+ databases, there is no paired dataset for RAF-DB collected and expanded over the web. In the database, there is no expression from the same person. For example, identities and expressions are different. In this model, we introduce two encoders to encode the original expression and the target expression. Two encoders extract the latent vectors and conditional labels features of the real image. In the meantime, associative learning is used to associate unpaired original emoticons with target emoticons in the database and to share identities.

Fig. 5 shows the results of facial expression synthesis on RAF-DB. Compared to IcGAN, our model is better to preserve the identity feature of the input face and smaller image distortion. We conjecture that this is because we use two encoders to extract emotional features and identity features, and share the feature information of the original domain and image domain by reducing the reconstruction loss. Just a low-dimensional feature vector is used as latent representation in IcGAN.

subA exB subA+exB subA exB subA+exB

Fig. 4: Facial expression transfer of unpaired images. The left column and the middle column are not the same subject. The model can learn the emotional characteristics of expressions from middle column, and generate images with subA and expression B(right column).



$Ne$ $H_1$ $H_2$ $Sa_1$ $Sa_2$ $Su_1$ $Su_2$

Fig. 5: The results of facial expression synthesis on RAF-DB. The illustrations show three emotions from four subjects. Each kind of emotion is divided into two columns. For each input ($Ne$), we compare the synthetic facial expression images of DECGAN($H_1, Sa_1, Su_1$) and IcGAN($H_2, Sa_2, Su_2$). ($Ne$: neutral, $H$: happiness, $Sa$: sadness, $Su$: surprise)

## 5 Conclusion

In this paper, we have proposed the Double Encoder Conditional GAN (DECGAN), which can synthesize new face expressions. It solves the problem that GANs lack to generate the specified target expression, meanwhile, it also solves the problem of unpaired data training. The main innovation of this paper is that the model has two encoders and adopts associative learning to share features. The Encoder $E_1$ maps the input face to a vector $z$. The Encoder $E_2$ extracts the target domain image emotional feature label $l$. The generator $G_1$ takes as input both the latent vector $z$ and the target label vector $l$. Reconstruct the generated image by another generator. Training two independent encoders has proven to be the best option in our experiments. In our experiments, we obtained satisfactory results in the ck+ database and the expanded RAF-DB. True face expressions can be synthesized in paired and unpaired databases. In summary, the model generates face images of the same theme with any desired facial expression while greatly preserving identity information.

## References


[1] P.Lucey. et al, The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression, *in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, 2010, pp. 94-101.

[2] F. Pighin and J. P. Lewis, Facial motion retargeting, *In ACM*

*SIGGRAPH 2006 Courses, SIGGRAPH 06*, New York, NY, USA, 2006.ACM.

[3] S. Schaefer, Mcphail T, Warren J, Image deformation using moving least squares, *ACM SIGGRAPH ACM*, 2006:533-540.

[4] S. Kshirsagar and N. M. Thalmann, Visyllable based speech animation, *Computer Graphics Forum*, 22.3(2010):631-639.

[5] J. Ma, R. Cole, B. Pellom, W. Ward, and B. Wise, Accurate automatic visible speech synthesis of arbitrary 3d model based on concatenation of diviseme motion capture data, *Computer Animation and Virtual Worlds*, 15.5(2004):485-500.

[6] DP. Kingma , M. Welling. Auto-Encoding Variational Bayes, *arXiv preprint arXiv:1312.6114*, 2013.

[7] D. Bitouk, N. Kumar, S. Dhillon, S. Belhumeur, and S. K. Nayar, Face swapping: Automatically replacing faces in photographs, *ACM SIGGRAPH*,ACM,2008:39.

[8] A. Zhmoginov and M. Sandler. Inverting face embeddings with convolutional neural networks, *arXiv preprint arXiv:1606.04189*, 2016.

[9] E. Richardson, M. Sela, and R. Kimmel. 3d face reconstruction by learning from synthetic data. *Fourth International Conference on 3D Vision*, pages 460-469. IEEE, 2016.

[10] X. Peng, X. Yu, K. Sohn, D. Metaxas, and M. Chandraker. Reconstruction for feature disentanglement in pose-invariant face recognition, *arXiv preprint arXiv:1702.03041*, 2017.

[11] Liu S, Huang D Y, Lin W, et al. Emotional facial expression transfer based on temporal restricted Boltzmann machines[C]// *Asia-Pacific Signal and Information Processing Association, 2014 Summit and Conference* IEEE, 2014:1-7.

[12] Goodfellow I J, Pougetabadie J, Mirza M, et al. Generative Adversarial Networks[J]. *Advances in Neural Information Processing Systems*, 2014, 3:2672-2680.

[13] Perarnau G, Weijer J V D, Raducanu B, et al. Invertible Conditional GANs for image editing, *arXiv preprint arXiv:1611.06355*, 2016.

[14] Ding H, Sricharan K, Chellappa R. ExprGAN: Facial Expression Editing with Controllable Expression Intensity, *arXiv preprint arXiv:1709.03842*, 2017.

[15] Zhu J Y, Park T, Isola P, et al. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, *arXiv preprint arXiv:1703.10593*, 2017.

[16] Choi Y, Choi M, Kim M, et al. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation, *arXiv preprint arXiv:1711.09020*, 2017.

[17] Mirza M, Osindero S. Conditional Generative Adversarial Nets, *Computer Science*, 2014:2672-2680.

[18] Matthews I, Baker S. Active Appearance Models Revisited[J]. *International Journal of Computer Vision*, 2004, 60(2):135-164.

[19] Raouzaiou A, Tsapatsoulis N, Karpouzis K, et al. Parameterized Facial Expression Synthesis Based on MPEG-4.[J], *Eurasip Journal on Advances in Signal Processing*, 2002, 2002(10):1-18.

[20] Ekman P, Friesen W V. Facial Action Coding System: Manual, *Agriculture*, 1978.

[21] Seidel H P. Geometry-based muscle modeling for facial animation, *Graphics Interface. Canadian Information Processing Society*, 2001:37-46.

[22] Zhang Z, Song Y, Qi H. Age Progression/Regression by Conditional Adversarial Autoencoder, *arXiv preprint arXiv:1702.08423*, 2017:4352-4360.

[23] Makhzani A, Shlens J, Jaitly N, et al. Adversarial Autoencoders, *Computer Science*, 2015.

[24] Haeusser P, Frerix T, Mordvintsev A, et al. Associative Domain Adaptation, *IEEE International Conference on Computer Vision. IEEE Computer Society*, 2017:2784-2792.

[25] Li S, Deng W, Du J P. Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild, *IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society*, 2017:2584-2593.

[26] Zhang, Kaipeng, et al. "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks." *IEEE Signal Processing Letters* 23.10(2016):1499-1503.

[27] Kingma, Diederik P, and J. Ba. "Adam: A Method for Stochastic Optimization." *Computer Science* 2014.

[28] Abadi M, Agarwal A, Barham P, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, *arXiv preprint arXiv:1603.04467*, 2016.