Unsupervised Face Frontalization GAN Driven by 3D Rotation and Symmetric Filling

Guanfang Dong(✉) and Anup Basu

University of Alberta,Edmonton, Canada
{guanfang,basu}@ualberta.ca

Abstract. Face frontalization has been partially solved by deep learning methods, such as Generative Adversarial Networks (GANs). However, due to the lack of paired training datasets, current generative models are limited to specific poses. Similarly, current unsupervised frameworks do not utilize properties of human faces, which burdens the neural network training. To improve and overcome current challenges, we design a novel self-supervised method that takes full advantage of human face modeling and facial properties. With our proposed method, single-view images collected in the wild can be utilized in training and testing. Also, the synthesized images are robust to input faces with large variations. We utilize the symmetric properties of human face to texture unseen parts in a human face model. Then, a GAN is used to fix undesired artifacts. Experiments show that our method outperforms many existing methods.

Keyword: Image processing

1 Introduction

Face frontalization is widely used in face recognition, face editing, and virtual and augmented reality. The problem of face frontalization is described as rotating various head poses into the frontal view of a face. As a fundamental challenge in computer vision, traditional methods do not perform well for this problem. This is because of the variations in facial details [1]. Most traditional methods struggle to find corresponding patterns and translate a face with various poses into the frontal view. However, the advent of the Generative Adversarial Network (GAN) brings a different aspect to solving the problem of face frontalization. Specifically, numerous input-output pairs are used for training to find the pattern of the face rotation for a generator and a discriminator. It is no coincidence that many large-scale face datasets [2,3,5–7] allow deep Convolutional Neural Networks to train for pose-invariant recognition. Compared to pose-invariant datasets, collecting pose-variant datasets (target person is photographed by different angles) is more expensive. Also, existing pose-variant datasets are not comparable with pose-invariant datasets in terms of the number of images [8,9]. For example, the

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
S. Berretti and G.-M. Su (Eds.): ICSM 2022, LNCS 13497, pp. 59–74, 2022.
https://doi.org/10.1007/978-3-031-22061-6_5

Multi-PIE face dataset [9] only contains images from 337 people, which is far less than a wild face dataset [6]. Also, these datasets are usually constrained to a controlled environment. The trained model only reflects the result from a specific domain. This kind of model lacks generalization ability. As a result, current methods which directly apply pose-variant datasets to GANs have difficulty in achieving promising results [10–14].

To avoid the problems caused by insufficient training datasets, we propose a novel unsupervised learning structure that allows face frontalization from single-view images. With this new unsupervised learning structure, all known face datasets can be used for training. This measure also vastly increases the generalization ability of our trained model. Compared to inpainting-based unsupervised face frontalization methods [16–18], our method further exploits the symmetric information of a face in a 3D space. This better preserves facial details and identity information for a GAN. We call our method URSF-GAN (Unsupervised Rotation and Symmetric Filling Driven GAN).

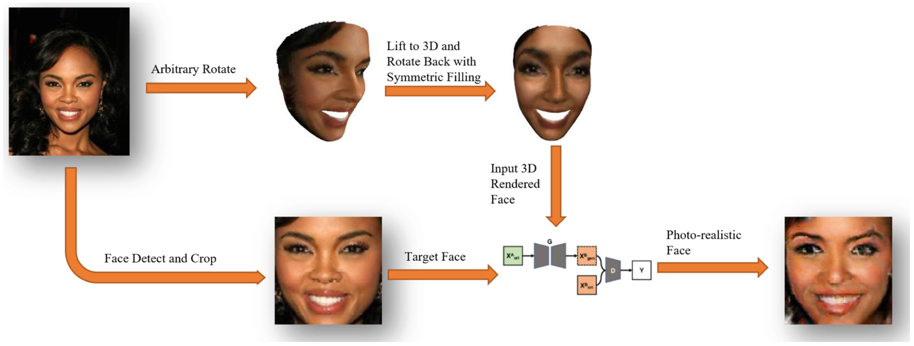

Fig. 1. The pipeline our URSF-GAN (Unsupervised Rotation and Symmetric Filling Driven GAN). Our method supports pose-invariant datasets. In-the-wild single-view facial images can be directly used in our method. First, an input face image is lifted to a 3D Morphable Model (3DMM). Then, the model is rotated by an arbitrary angle. We render the model and lift it back to 3D. A specially designed symmetric repair algorithm fixes missing textures. We rotate the model back to the front and use Pix2Pix GAN to eliminate artifacts.

As Fig. 1 shows, our method consists of two main parts. First, the single-view facial image is tested to see if they are front facing. For front faces, we lift faces from the 2D plane into 3D. Then, we rotate faces by arbitrary angles. The rotated faces are rendered into the 2D plane. For these rendered faces, we lift them into 3D again and rotate them back to the front. Clearly, some facial textures are hidden due to the arbitrary rotations, such as the opposite side of the nose bridge. We repair these missing information using the other side of the face. Until this step we do not use GAN. Nevertheless, the rendered faces are already good. The reason for using GAN is trying to eliminate the artifacts caused by rotation and repairs. The steps above can be described in the following mathematical way.

$$P_b = f_{\text{render}}\ (f_{\text{lift}}\ (P_a) \times R_d)$$
$$P_c = f_{\text{render}}\ (f_{\text{lift}}\ (P_b) \times R_{-d} \oplus P_{-b})$$

Here, P_a means the original front face input. f_{lift} is the function that lifts the face into 3D. R_d is a rotation matrix with degree d. f_{render} means rendering a 3D scene into 2D. P_b is the rotated face, and P_c is the face rotated back. Following this order, $\{P_a, P_c\}$ becomes the training pair that is ready to fit a GAN for eliminating artifacts. As for the selection of a GAN, we modify Pix-to-Pix, which allows it to be more suitable for eliminating artifacts and enhance facial details. More details will be given in Sect. 3.

Remarkably, differing from previous supervised learning models, our proposed structure can fit any existing human face dataset. Our key contributions can be summarized as follows:

1. We propose an unsupervised framework that exploits the concept of 3D lifting and rotation. Single-view face images can be directly used in our framework. This brings unlimited training data and increases the models generalization ability.
2. We apply symmetric repair during rotation. It makes training-free facial images suitable for use. Our approach shifts the training focus of GAN from inpainting faces to eliminating artifacts, making our network easier to train. This structure benefits from GANs to generate more photo-realistic face frontalization results.

2 Related Work

2.1 Traditional Approaches

The problem of face frontalization aims to generate frontalized faces. The classical approach establishes a 3D model from facial images. Then, the 3D model can be rotated to the desired position. In a seminal work, 3D Morphable Model (3DMM) [19] sets a baseline for subsequent studies. 3DMM decomposes facial shape and appearance into PCA spaces. The decomposed information can be further analyzed in studies like face reconstruction or face landmark localization [20]. Based on 3DMM, Claudio et al. proposed an image patch localization and interpolation method by extracting feature descriptors [21]. However, limited by the number of feature descriptors, their method cannot keep enough facial details. Similarly, Zhu et al. proposed a Possion Editing based method to fill the area occluded by 3DMM rotation [22]. Taking advantage of 3DMM fitting, both methods can be used in constrained and unconstrained environments. However, the missing area usually shows more artifacts than deep learning methods.

2.2 Supervised Learning Approaches

The development of deep learning and GAN drew more activity for solving face frontalization problems [10–14,16–18]. Tran et al. proposed a method leveraging a disentangled representation GAN model to learn frontalized face synthesis [10,11]. Also, the encoder-decoder structure is used to estimate variations in different faces. However, there are artifacts in their results due to the lack of effective interpretation from the encoder and decoder. Other than this, Huang et al. proposed a new GAN structure called TP-GAN to solve the problem of face frontalization [12]. This GAN structure involves perceiving and analysis of global structure and local details. They involve a landmark located patch network to assists encoder-decoder preserving local texture. Similarly, Hu et al. proposed a GAN structure that is driven by facial landmark heatmaps, which is called CAPG-GAN [13]. CAPG-GAN also introduces identity information preservation loss to maintain local textures. Besides this, Yin et al. designed a GAN structure that incorporates 3DMM [14]. To avoid undesired effects from occlusion, they also introduce mask symmetry loss to reproduce missing information. Cao et al. proposed a High Fidelity Pose Invariant Model to solve the problem of face frontalization [17]. They designed a new texture wrapping method to bind faces in 2D and 3D surfaces. However, most of the current methods require input-output pairs for training. As we mentioned before, obtaining paired data is very expensive. Thus, most of the current GAN methods are trained on the Multi-PIE dataset. However, the Multi-PIE dataset contains only 337 people, leading to a lack the generalization ability.

2.3 Unsupervised Learning Approaches

Recently, some research has looked into face frontalization for in-the-wild single view images. Deng et al. proposed a GAN structure to repair the missing areas of the facial UV map [16]. Finally, they perform a reconstruction on the repaired UV map. However, without real 3D rotations, an incomplete UV map cannot represent the real missing area, causing risks in robustness. Zhou el at. proposed a GAN inpainting network to fix the missing areas after rotation [18]. However, the convergence speed of their network is slow due to the lack of missing area pre-filling. Our approach both preserves facial details and improves training speed.

2.4 Image-to-Image Translation GAN

In recent years, many GAN structures try to solve the problem of image-to-image translation, which means the input images will be encoded in a low-dimensional space and stylized for the output [23–26]. Almost all such architectural solutions include decoders and encoders. However, l_2 loss often blurs output images during decoding. Thus, l_1 loss is usually used for the Image-to-Image translation GAN structure. Corresponding loss functions are adapted for specific problems. For classical network architectures, Pix2Pix focuses on mapping the training pairs by traditional l_2 loss [23]. It is applicable and easy to use for many image-to-image translation tasks. Also, collecting paired datasets is expensive. Unsupervised Image-to-Image GAN structures take advantage of cycle consistency to stylize the input images [26]. As a classical example, CycleGAN can translate images without feeding it paired data. Although we have the same goals, face frontalization requires the maximum preservation for identity information and facial details. Thus, we modify the Pix2Pix method to eliminate artifacts and enhance facial details.

3 Proposed Approach

3.1 Summary of Our Framework

Our approach can be conceptually divided into two parts. The first part aims to model, rotate and infer for single-view face images to generate a training dataset. The second part is to de-fake the inferred faces by an Image-to-Image translation GAN. We will then explain the implementation details and ideology behind each part.

3.2 Basic Concepts on Lifting Faces into 3D Space

Due to the needs of our framework, we model the face while preserving almost all of the facial textures. Currently, many deep learning-based studies are available to fit faces as 3D parameters. Like many models in 3D space, 3D face models are defined by their vertices. The mathematical representation is as follows:

$$V_{\text{face}} = [v_1, v_2, v_3, \ldots, v_n] \quad \text{while} \quad v_i = [x_i, y_i, z_i]^T$$

V_{face} is the collection of facial vertices. Based on this 3D feature, 3D Morphable Models (3DMMs) allow fitting 3D facial models with given 3D parameters [19]. Recall the fitting formula:

$$S = \bar{S} + A_{id}\alpha_{id} + A_{exp}\alpha_{exp}$$

$$T = \bar{T} + A_{tex}\alpha_{tex}$$

Here, S has a similar meaning as V, i.e., 3D vertex coordinates. $\bar{S}$ represents the mean facial shape from 200 young adults (100 male and 100 female). A_{id} denotes shape basis. A_{exp} means expression basis. Here, $\bar{S}$, A_{id} and A_{exp} are constant values. With given 3D parameters α_{id} and α_{exp}, the input 2D facial images can be fitted into a 3D model. Similarly, $\bar{T}$ means mean texture, and A_{tex} represents the texture basis. However, from the perspective of fitting accuracy, the results of texture fitting are poor. Thus, we restore the texture information by recording the vertex RGB information for the corresponding 2D face image. The process of obtaining vertex RGB information can also be called vertical projection. It can be mathematically expressed as:

$$\mathbb{C}_i = \mathcal{H}_{\tilde{x}_i,\tilde{y}_i}\left(\widetilde{\mathbb{C}}_i\right) \quad \text{while} \quad \mathbb{C}_i = [r_i, g_i, b_i]^T$$

Here $\mathcal{H}_{\tilde{x}_i,\tilde{y}_i}(\tilde{\mathbb{C}}_i)$ are the vertical projections of vertices at 2D position $\tilde{x}_i, \tilde{y}_i$. Suppose we do not perform the rotation matrix operation after 3DMM modeling. In this case, we are able to ignore the depth information in 3D space and get all facial textures if the input image is a frontal face. In other words, if we are only fitting a face as 3D Morphable Model, the face's pitch, yaw, and roll will remain the same as the original 2D facial image. However, if the input image is not a frontalized face, the problems of occlusion and texture persist. We will discuss this issue in a later section (Figs. 2, 3, 4 and 5).

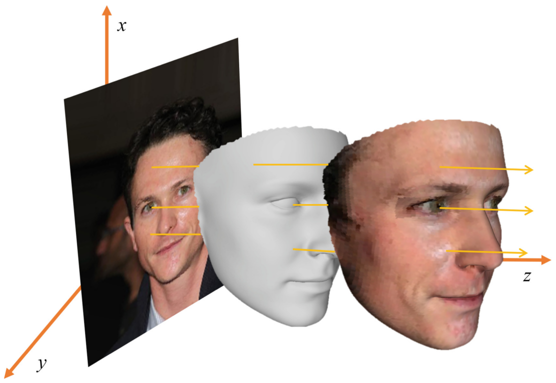

Fig. 2. The 3DMM fitting and texturing pipeline. The face is fed into 3DDFAv2 to estimate 3DMM parameters. Then, we lift face images to the 3DMM model. For frontal faces, we can use the vertical projection directly to obtain the facial texture.

Note that α_{id}, α_{exp} are two 3D parameters that we need to calculate for a unique face. We use 3D Dense Face Alignment Version 2 (3DDFAv2) [27], as one of the most efficient, stable and accurate methods, to model the facial images. The most crucial core of this method is the 3D-assisted short video synthesis method, which can simulate in-plane and out-of-plane face movement to convert a static image into a short video. Then, the model can be trained based on the short video. Thus, it makes 3D Dense Face Alignment possible for only one input static image, which matches our need.

Another important concept in our work is rendering. Rendering means projecting the 3D model into 2D space. Before rendering, the model can be rotated by a certain Euler angle to the desired position. Mathematically, the step of rendering can be represented as:

$$V_{2d}(\mathbf{p}) = f * \mathbf{Pr} * \mathbf{R} * \left(\bar{\mathbf{S}} + \mathbf{A}_{id}\alpha_{id} + \mathbf{A}_{exp}\alpha_{exp}\right) + \mathbf{t}_{2d}$$

Here, $V_{2d}(\mathbf{p})$ is the projection result. f is the scale factor. $\mathbf{Pr}$ represents the orthographic projection matrix. $\mathbf{R}$ is the rotation matrix which contains the Euler angles of pitch, yaw and roll. $\mathbf{t}_{2d}$ is a translation matrix. $(\bar{\mathbf{S}} + \mathbf{A}_{id}\alpha_{id} + \mathbf{A}_{exp}\alpha_{exp})$ is the 3DMM fitted model.

3.3 Unsupervised Training Strategy

Recall the overview in the introduction section. The purpose of lifting 2D face images to 3D space and rotating, filling, and rendering is to create effective training data pairs. The whole set of operations described here (lifting, adding texture, rotating and rendering) can be replaced by the mathematical notations $\{S_a, T_a, R_\theta, V\}$. In order to create the input and output datasets, we have to rotate the face model to the front face position after rotating it by an arbitrary angle. This means that the above process will be performed twice. In terms of notation sets, $\{S_a, T_a, R_\theta, V_b, S_b, R_{-\theta}, T_{-a}, V_c\}$ can represent the entire process. Among them, since the input face image must be a frontal face, the facial texture information would be completely obtained. However, when faces are rotated by an Euler angle, there will be some missing texture patches on the model. Thus, to solve this problem, we first need to find the location of missing patches.

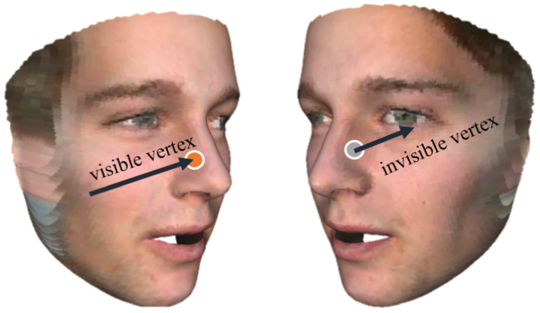

Fig. 3. For vertices that have different depths, only the vertex in the surface can obtain the correct texture when rendering a 3D model. In this example, the orange vertex is visible, and the gray vertex is invisible.

Recall the vertical projection $\mathbb{C}_i = \mathcal{H}_{\tilde{x}_i,\tilde{y}_i}(\tilde{\mathbb{C}}_i)$ *while* $\mathbb{C}_i = [r_i, g_i, b_i]^T$. Now, we consider the depth information z_i. Due to the properties of the projection, the texture information is valid only when it first crosses the surface of the object. So, within the same coordinate, only the uppermost z value will yield the correct texture information. The remaining vertices are invisible due to occlusion. Thus, we modify the original vertical projection formula and categorize the vertices into two classes (with and without visible texture).

$$v_i = \begin{cases} v_{visib} = \mathbb{C}_i = \mathcal{H}_{\tilde{x}_i\tilde{y}_i}\left(\widetilde{\mathbb{C}}_i\right) \text{ while } \mathbb{C}_i = [r_i, g_i, b_i]^T \text{ and } i = \operatorname{argmax}(x, y, \forall z_i) \\ v_{\text{invisib}} = \mathbb{C}_i = \{x, y, \forall z_i\} \setminus \operatorname{argmax}(x, y, \forall z_i) \end{cases}$$

Here, $\mathcal{V}_{visib}$ can only be vertices that are located at the surface of the object. For the remaining vertices with lower depth, we consider them as invisible vertices and do not add texture on them.

As an intuitive idea to filling the $\mathcal{V}_{invisib}$, the perpendicular distance between an invisible vertex to the face midline can be calculated. Based on this distance, we can find the vertex that is perpendicular to the midline at the other half of the face. Thus, we can find a one-to-one correspondence for the left and right face vertices. Mathematically, it can be represented as:

$$\boldsymbol{d}\left(\boldsymbol{v}_{\boldsymbol{i}\sim\text{midline}}\right) = \left| V_i \perp \text{ midline } \right|$$
$$V_{i'} = -\boldsymbol{d}\left(\boldsymbol{v}_{\boldsymbol{i}\sim\text{midline}}\right) \perp \text{ midline}$$

Here, $V_{i'}$ is the corresponding vertex. $\boldsymbol{d}\left(\boldsymbol{v}_{\boldsymbol{i}\sim\text{midline}}\right)$ is the Euclidean distance between the current vertex to the point perpendicular at midline. Since we compute the distance, we can find the corresponding vertex. Finally, the texture information (RGB value) for invisible vertices can copy and paste vertices from another half of the face.

However, this intuitive idea does not produce good results. There are two main reasons. First, during face photography, different parts of the face are exposed to different light levels. This also results in a small area of the face being affected by the shadow. At the same time, such an effect is difficult to completely remove by normalization and other methods. In regular modeling, the shaded part of the face will feel natural because of the smooth transition. However, if we force the corresponding positions of the left and right faces to be swapped, the vertex textures at the missing positions will make the overall look feel abrupt. Second, the human face is not perfectly symmetrical from left to right. If we do not use a landmark to determine the position of the face's organs, direct left-right swapping usually does not perfectly match the corresponding positions. Thus, using an intuitive idea would lead to massive artifacts, which inevitably affect the analysis of GAN in the next step.

Considering the above, we perform a pre-component before the symmetric copy. By landmark detection of the base 3DMM face model, we perform a regional division for the left and right faces. The region can be divided as eyes, nose, mouth and other parts for each left and right faces. For each classification, if there are vertices that are not visible in this classification, we swap and copy directly from the other half of the face that is visible. Furthermore, suppose more than two classifications have invisible vertices. This case indicates that we have a large missing range and we directly copy the other half of the face to the invisible part. The advantage of this approach is that only a small amount of artifacts appear in the marginal parts of each category. Also, these artifacts are patterned. This weakens the difficulty when analyzing input-output pairs in the next step of GAN.

$$\mathcal{P}_{\text{left}} = \mathcal{P}_{\text{right}} \text{ while } V_i \in \mathcal{P}_{\text{right } \vee \text{ left}} \tag{1}$$

Here, $\mathcal{P}$ represents the parts for eyes, nose, mouth, and other parts of one side of the face. The texture value can be found in the corresponding part if the current part has invisible vertices.

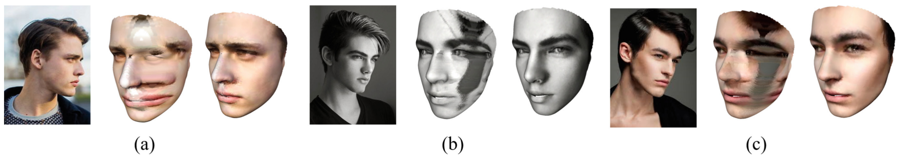

Fig. 4. Comparison of texturing with and without restoration. The left image is the original facial image. The middle image is texturing result without restoration. The right image is the texturing result with our restoration method. When there is no repair, the occluded vertices will also get texture information from the corresponding 2D position for the image. This causes the model texture to be normal on one side of the face and distorted on the other side. We added detection and repair for texture. The model obtained is almost free of artifacts. The results are already in a usable state even before processing by GAN.

3.4 Photo-Realization by URSF-GAN

In order to achieve photo-realistic results, we need to eliminate artifacts by applying an Image-to-Image translation network. Again, compared to other GAN-based methods for encoding 2D facial images or inpanting missing parts, we have less burden on the network. Thus, we choose Pix2Pix as the base network to translate the images. We have made small modifications to this network. We trained the network on RTX 2080*2. Eventually, after three hours of training, the network can generate realistic facial images.

URSF-GAN uses U-Net as its Generator. The reason is that our task needs to translate from one facial image to another. That is, the input and output need to be roughly aligned. Although the input and output differ in surface detail, they have the same underlying general structure. U-Net can be employed to capture both underlying structure and high-level semantic information.

To train our network, we apply the conditional GAN loss defined as:

$$\begin{aligned}\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{x,y}[\log D(x,y)] + \\ \mathbb{E}_{x,z}[\log(1-D(x,G(x,z))], \arg\min_{G}\max_{D}\mathcal{L}_{cGAN}(G,D).\end{aligned}$$

Here, $D(x,y)$ is the probability of the target facial image being real. $(1-D(x,G(x,z))$ is the probability of the generated facial image being real. Thus, the generator tries to minimize the loss, and the discriminator tries to maximize

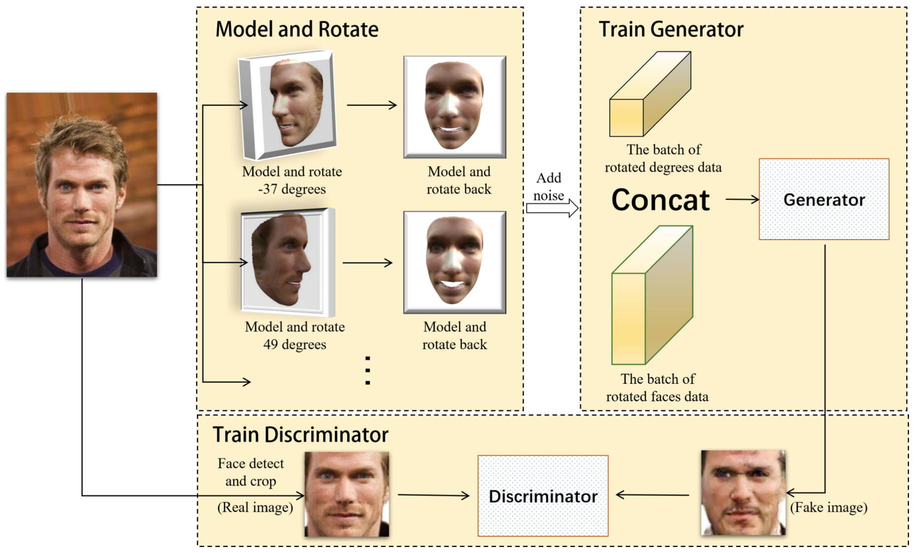

Fig. 5. The GAN structure of our URSF-GAN method. We have two differences from the original Pix2Pix. First, we embed the degrees when the model is rotated back. Second, we add random Gaussian noise to the input face image to avoid over-fitting.

the loss. Also, current research show that l_2 loss empirically leads to blurry output [28]. We involve the l_1 loss to avoid blurred images. This is expressed as:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}\left[\|y - G(x,z)\|_1\right]$$

Eventually, the final loss function for our URSF-GAN generator is:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda\mathcal{L}_{L1}(G)$$

Before training, random Gaussian noise is added to the input facial images to avoid over-fitting. Then, to standardize the face size, we use a face detection algorithm on the image dataset to crop the images. Besides this, we also embed the 3D model rotation degree since we found that higher degree tends to have a wider face and lower light. Embedding the rotation degree helps the network analyze the repairing environment. The whole network setup is shown below.

Our model is tested differently than the way it is trained. For testing, the face image is lifted to a 3D model. The same method of obtaining textures requires the occluded texture to be repaired. Then, the model is rotated to a frontal face. Finally, the rendered model is used as the model input for the GAN. There is no need to add random noise during testing.

4 Experiments and Evaluations

4.1 Experimental Settings

We use the CelebA dataset for preprocessing and training. CelebA contains 202599 face images [29]. Among all facial images, we pick the first 60000 for training purpose, and use 5000 images (index from 60000 to 65000) for testing. Before training, we perform facial detection based on the Dlib C++ Library to standardize the face size. Then, each face image is modeled after rotation by 10 different arbitrary angles. After this, the face models are rotated back, filled, and rendered. The abovementioned steps allow us to create training input-output pairs. These data is run for 50 epochs. After that, the training data is discarded, and new training data needs to be generated by performing the abovementioned steps again.

Our model is trained in the Pytorch 1.6 environment with 2 * RTX 2080 settings. The size of input and output is $100 * 100 * 3$. During the preprocessing, we add Gaussian noise to input images with mean 0 and standard deviation 1. We choose the Adam optimizer to optimize the network with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate is 0.0001. The model is well-trained after 400 Epochs.

4.2 Evaluation

Within the field of face frontalization, most of the methods do not have publicly available source code. Thus, we directly use the demonstration images of various papers for comparison. Figure 6 shows the result of comparison between our URSF-GAN and other state-of-the-art methods. Since we cannot find the original facial images, we just show screenshots from other papers. Thus, the input resolution in our method is very low. However, although our method has low resolution, it still generates good results.

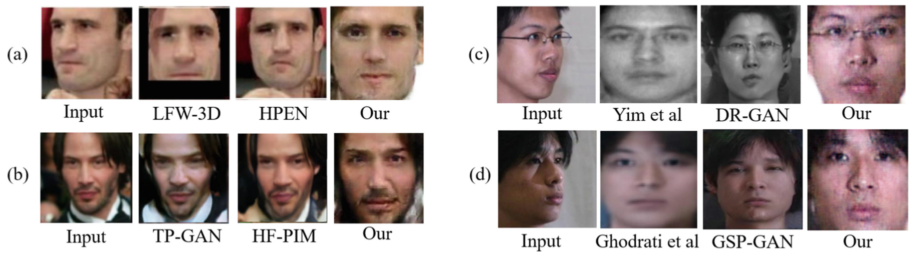

Fig. 6. Results of comparison with other popular methods.

4.3 Comparison with Other Methods

We analyze the results from different methods. Figure 6 contains four comparison pairs (a), (b), (c) and (d). For (a), as a critical issue, LFW-3D [15] and HPEN [22] do not rotate the face into an exact frontal position. For each method, LFW-3D uses a single, unmodified 3D reference for all input faces to generate frontal images. Apparently, a single reference cannot adapt well to all human faces, which leads to significant misalignment of the results. HPEN also employs a method that combines 3D modeling and interpolation. However, without further artifact elimination, the results have significant distortions at the facial boundary. Compared to other methods, our URSF-GAN generates a true frontal face and maximally preserves facial details.

For (b), since TP-GAN [12] takes a supervised method to train the network, it has chromatic aberration problems in the output. Specifically, their chromatic aberration is biased towards the chromatic aberration used in the MultiPIE dataset [9]. This also demonstrates the lack of generalization capability for supervised methods at the current stage. HF-PIM [17] decomposes the facial rotation problem into dense field estimation and facial texture map recovery. However, both TP-GAN and HG-PIM cannot fully rotate the face into frontal faces. Finally, our method has some problems in the nose tip area, where we recover some parts of the nose tip as bread. We believe this is because of the low input resolution. Our 3D facial model contains 38365 vertices, which is larger than the number of input pixels. Thus, when we fit the model, many vertices will share the same pixel value. This leads to misalignment when we recover the model. Besides this issue, our URSF-GAN can rotate the input faces quite well.

For (c), Yim et al. proposed a method that only utilizes DNN instead of GAN [31]. Their DNN can decode, rotate and reconstruct the final image. However, for image synthesis, DNN is inferior to GAN. Their result has issues in reconstructing glasses. DR-GAN [10] is deficient in terms of retaining identity information. Also, the synthetic human facial image has problems on the face contour and positions. Our method is better at preserving identity information. One small drawback is that our method does not have satisfactory restoration for the right side of the glasses. This is due to the inability of 3DMM in modeling accessories like glasses. Thus, accessories are directly captured as skin texture, which influences the accuracy of positions after rotation. Fixing this problem will be the key focus of our future research in 3D-based modeling.

For (d), Ghodrati et al. proposed a two-stage DNN method to generate and refine the frontal face [32]. Their result is excessively smooth and hardly conveys the effect of light and shade when a person is photographed. The result for GSP-GAN [33] has good resolution. However, it cannot fully preserve the identity information. After comparison, we found that our method performed best in terms of overall performance.

4.4 More Results for CelebA Dataset

Figure 7 shows more testing results for CelebA dataset. In this figure, we roughly divide the results into three categories (faces rotated in minor degree, moderate

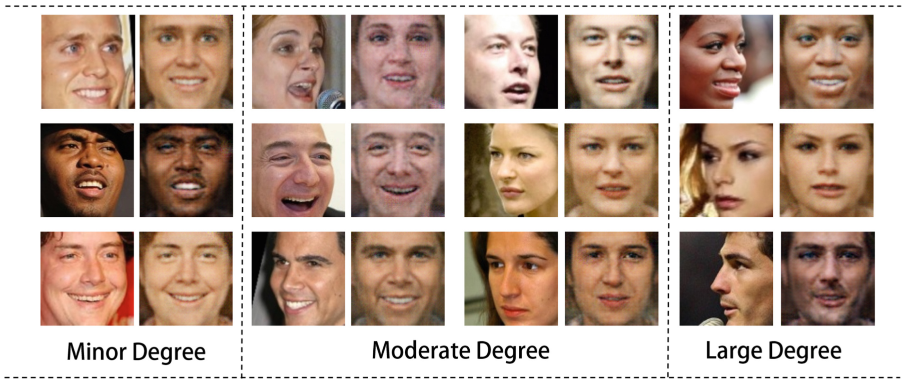

Fig. 7. More results on CelebA dataset.

degree and large degree). The left is the input face and the right is the output. By comparison among three categories, our facial identity feature decays slowly as the rotation degree increases. Benefitting from the strategy of modeling rotation, this strategy also limits the facial details that are generated by GAN, which simplifies the inference processing and leads to the preservation of identity feature. Facial recognition results from Sect. 4.5 also proves this point.

4.5 Face Recognition Results

We also perform the face recognition test on Multi-PIE dataset with baseline of LightCNN [30]. The quantitative results show that our method has greater robustness when rotating faces with larger angles (Table 1).

Table 1. Face recognition correct rate by LightCNN on the Multi-PIE dataset.

Angle	30°	45°	60°	75°	90°
FF-GAN	92.5	89.7	85.2	77.2	61.2
TP-GAN	98.1	95.4	87.7	77.4	64.6
CAPG-GAN	**99.6**	**97.3**	**90.3**	76.4	66.1
URSF-GAN (Our)	91.4	90.7	88.6	**81.3**	**70.8**

5 Conclusion

We designed an unsupervised face frontalization model based on 3D rotation and symmetric filling. Then, image translation GAN was used to fix artifacts. Compared to other unsupervised GAN models, our model reduces the GAN's burden from inpainting to fixing artifacts, which makes our model easy to train. Results show that our model can generate photo-realistic frontal face images. For faces with large angles, our method can preserve more identity information than other state-of-the-art methods. In the future, we want to embed our method with the supervised structure to improve robustness when people wear accessories or are occluded by others.

References

1. Moore, K.L., Dalley, A.F., Agur, A.M.: Moore's clinical anatomy (2010)
2. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
3. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: MS-Celeb-1M: a dataset and benchmark for large-scale face recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 87–102. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_6
4. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database forstudying face recognition in unconstrained environments. In: Workshop on faces in 'Real-Life' Images: Detection, Alignment, and Recognition (2008)
5. Klare, B.F., et al.: Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus benchmark A. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1931–1939 (2015)
6. Nech, A., Kemelmacher-Shlizerman, I.: Level playing field for million scale face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7044–7053 (2017)
7. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3730–3738 (2015)
8. Zhu, X., Lei, Z., Liu, X., Shi, H., Li, S.Z.: Face alignment across large poses: a 3D solution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 146–155 (2016)
9. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-pie. Image vision comput. **28**(5), 807–813 (2010)
10. Tran, L., Yin, X., Liu, X.: Disentangled representation learning GAN for pose-invariant face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1415–1424 (2017)
11. Tran, L., Yin, X., Liu, X.: Representation learning by rotating your faces. IEEE Trans. Pattern Anal. Mach. Intell. **41**(12), 3007–3021 (2018)
12. Huang, R., Zhang, S., Li, T., He, R.: Beyond face rotation: global and local perception GAN for photorealistic and identity preserving frontal view synthesis. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2439–2448 (2017)
13. Hu, Y., Wu, X., Yu, B., He, R., Sun, Z.: Pose-guided photorealistic face rotation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8398–8406 (2018)
14. Yin, X., Yu, X., Sohn, K., Liu, X., Chandraker, M.: Towards large-pose face frontalization in the wild. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3990–3999 (2017)

15. Hassner, T., Harel, S., Paz, E., Enbar, R.: Effective face frontalization in unconstrained images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern recognition, pp. 4295–4304 (2015)
16. Deng, J., Cheng, S., Xue, N., Zhou, Y., Zafeiriou, S.: UV-GAN: adversarial facial UV map completion for pose-invariant face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7093–7102 (2018)
17. Cao, J., Hu, Y., Zhang, H., He, R., Sun, Z.: Learning a high fidelity pose invariant model for high-resolution face frontalization. arXiv preprint arXiv:1806.08472 (2018)
18. Zhou, H., Liu, J., Liu, Z., Liu, Y., Wang, X.: Rotate-and-render: unsupervised photorealistic face rotation from single-view images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5911–5920 (2020)
19. Blanz, V., Vetter, T.: 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 (1999)
20. Li, S., Li, H., Cui, J., Zha, H.: Pose-aware face alignment based on CNN and 3DMM. In: BMVC, p. 106 (2019)
21. Ferrari, C., Lisanti, G., Berretti, S., Del Bimbo, A.: Effective 3D based frontalization for unconstrained face recognition. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 1047–1052. IEEE (2016)
22. Zhu, X., Lei, Z., Yan, J., Yi, D., Li, S.Z.: High-fidelity pose and expression normalization for face recognition in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 787–796 (2015)
23. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)
24. Zhu, J.Y., et al.: Multimodal image-to-image translation by enforcing bi-cycle consistency. In: Advances in Neural Information Processing Systems, pp. 465–476 (2017)
25. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: StarGAN: unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8789–8797 (2018)
26. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2223–2232 (2017)
27. Guo, J., Zhu, X., Yang, Y., Yang, F., Lei, Z., Li, S.Z.: Towards fast, accurate and stable 3D dense face alignment. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12364, pp. 152–168. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58529-7_10
28. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: feature learning by inpainting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2536–2544 (2016)
29. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3730–3738 (2015)
30. Wu, X., He, R., Sun, Z., Tan, T.: A light CNN for deep face representation with noisy labels. IEEE Trans. Inf. Forensics Secur. **13**(11), 2884–2896 (2018)

31. Yim, J., Jung, H., Yoo, B., Choi, C., Park, D., Kim, J.: Rotating your face using multi-task deep neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 676–684 (2015)
32. Ghodrati, A., Jia, X., Pedersoli, M., Tuytelaars, T.: Towards automatic image editing: learning to see another you. arXiv preprint arXiv:1511.08446 (2015)
33. Luan, X., Geng, H., Liu, L., Li, W., Zhao, Y., Ren, M.: Geometry structure preserving based GAN for multi-pose face frontalization and recognition. IEEE Access **8**, 104676–104687 (2020)