Received March 4, 2021, accepted March 22, 2021, date of publication April 9, 2021, date of current version April 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3072277

Detailed Feature Guided Generative Adversarial Pose Reconstruction Network

JINLIN HAO AND XUEYUN CHEN
School of Electrical Engineering, Guangxi University, Nanning 530004, China

Corresponding author: Xueyun Chen (xueyun.xychen@gmail.com)

This work was supported by the National Natural Science Foundation of China under Grant 62061002.

ABSTRACT Face frontalization is a critical and difficult task on face pose reconstruction. Previous researches use simple posture information as guidance, such as pose coding and facial landmarks. To explore the guidance effect of profile faces, we propose detailed features that provide much detailed information. In this paper, a ***D****etailed Feature* ***G****uided Generative Adversarial* ***P****ose* ***R****econstruction Network* (DGPR) is proposed. Firstly, frontal pose coding and profile detailed features are fed into DGPR to generate detailed features of front face. Then, the second generator combines frontal detailed features and profile face to reconstruct front face. Besides, we propose a conditional enhancement loss to strengthen the guiding role of detailed features, and a smoothing loss to reduce edge sharpness in generated faces. Experimental results show that our method generates photorealistic front faces and outperforms state-of-the-art methods on M^{2}FPA and CAS-PEAL. Specifically, DGPR improves the face recognition accuracy under pose angles of ±60°, ±75°, ±90° by 2%, 1%, and 6% respectively over the state-of-the-art methods on M^{2}FPA, achieves the average rank-1 recognition rate to 99.95% and improves it by 0.05% on CAS-PEAL. These results demonstrate the effects of detailed features and corresponding modules.

INDEX TERMS Face frontalization, face pose reconstruction, Generative Adversarial Network, face rotation.

I. INTRODUCTION

Profile faces exist widely in the real world for the application field of face recognition. Face frontalization converts them into front faces improves face recognition accuracy significantly by preserving the rich identification information from the profile faces. Benefiting from convolutional neural networks, such as Generative Adversarial Networks (GAN) [1], Variational Autoencoder (VAE) [2], great progress has been achieved in the field of face frontalization. However, insufficient feature information of profile faces has made a main obstacle to the frontalized face quality, especially under the extreme pose angles (for example, 90°).

Previous researchers have made many outstanding contributions, greatly promoted progress on face frontalization. Some researchers [3]–[5] tried to build a 3D model to generate a reconstruction of front faces. Generally, they reconstructed frontal view faces from estimated 3D surface based on existent 2D profile faces. This type of methods is time-consuming and hard to learn, and vulnerable to an irregular surface.

Some researches (TP-GAN [6], CAPG-GAN [7]) solve this task by 2D-based methods: learning a pose-invariant embedding to reconstruct frontal faces, where model architectures were meticulously considered and finely designed. After encoding faces as high-level features in latent spaces, multiple sub-modules process features in different aspects to generate photorealistic frontal faces.

This type of method focuses on high-level features, neglects detailed information, such as edges, textures and corners which contain rich information useful for the quality improvement of the frontalized faces. They used only simple pose information as guidance, applying one-hot vector to store pose clues, and facial landmark positions to direct the frontalization process. For example, CAPG-GAN [7] utilizes five facial landmarks (eyes, nose, mouth) which cannot store rich pose features on the other areas of the face skin.

Besides, these methods used the common loss functions which had not considered the importance of the detail smoothness, suffered from rough and unrealistic details in their results.

The associate editor coordinating the review of this manuscript and approving it for publication was John See.

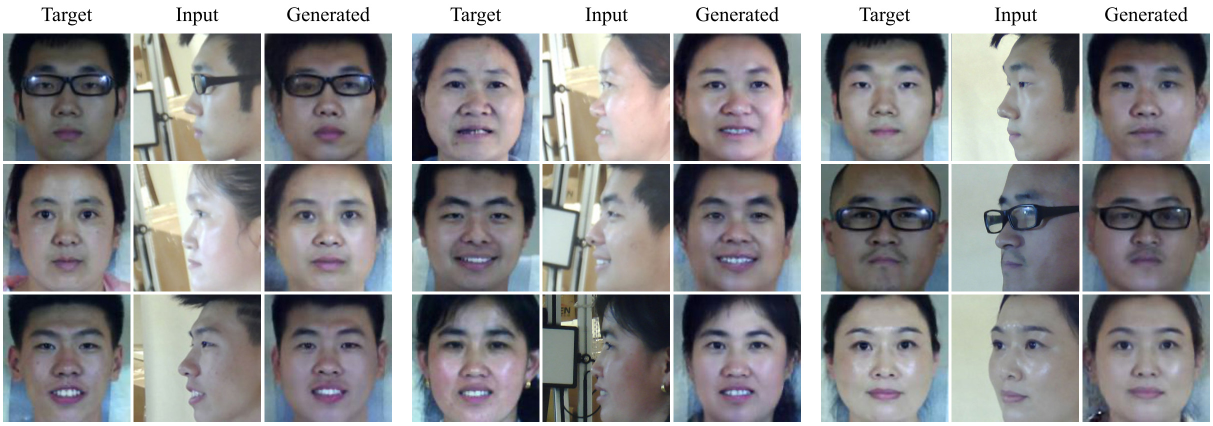


FIGURE 1. The frontal view synthesis by DGPR. There are three groups and each is consist of three columns, the first column is real faces, the second column is input profile faces and the third column is generated frontal faces. The first and second groups illustrate 90° profile faces and corresponding real and generated front faces. The third group displays 75°, 60°, 45° profile faces and corresponding generated results.

Based on these observations, we propose to explore more detailed information from profile faces, aiming to provide more effective information for face frontalization. Specifically, we extract profile sketches as detailed features, which preserve rich identifying information for pose rotation. To eliminate the influence of rough detailed features, smoothing loss is proposed to improve the smoothness of generated detailed features. In addition, we present a conditional enhancement loss to strengthen the detailed feature's guidance effect on generation of target pose.

In this paper, the **D**etailed Feature **G**uided Generative Adversarial **P**ose **R**econstruction Network (DGPR) is proposed, which include two generators and two discriminators: a detailed feature convert generator G_m and a pose convert generator G_t. G_m was used to generate frontal detailed features from profile faces, and G_t was designed to combine profile faces with the corresponding frontal detailed features and eventually generate front faces. Experimental results show that DGPR reaches the state-of-the-art under metric of rank-1 recognition rate. After taking detailed features of profile faces into account, we also reconstruct photorealistic front faces from extreme poses, such as ±90°, which have much less information compared with smaller pose.

The contributions of this paper can be summarized as three-folds: (1) we firstly utilize the rich sketch information of profile faces as guidance to generate front faces, which provides more details, and the improved effects is demonstrated. (2) We proposed a structure of dual generators to combine profile faces and detailed features, which achieves excellent results. (3) Conditional enhancement loss and smoothing loss are proposed to enhances pose consistency and edge naturalness. At the special cases of the extreme horizontal poses (such as ±90°) with little information, our model can also obtain clarifying frontal faces.

The rest of this paper is organized as follows: Section II describes the related works. Section III details our proposed method. Experimental results are shown in Section IV, and Section V summaries this paper.

II. RELATED WORKS

Lots of researchers proposed effective algorithms for face frontalization. Manifold learning methods [8], [9] improved feature representation for downstream tasks, but failed in capturing more robust data characteristics in specific tasks compared with deep-learning methods. The existing deep-learning methods can be summarized as two classes: the first type of methods was based on 3D models, which learned a 3D-surface from 2D faces and then rotated and projected to front view. Hassner *et al.* [3] used a single unmodified 3D reference surface to generate the front view. Yin *et al.* [4] used traditional rotation methods to estimate 3D grid to reconstruct front faces in target pose. Cao *et al.* [10] proposed to generate front face by estimating 3D faces surface. They employed two pathway generators, which were used to learn to estimate facial texture map and UV map [11] respectively. Zhou *et al.* [12] generated 3D faces based on 3DDFA [13], and then rendered frontal faces. Zhao *et al.* [14] estimated 3D surface based on 3DMM [15], and devised a two pathway structure to deal with global and local texture.

These kinds of methods relied on a high precision model trained on a 3D fitting database or a strict and accurate 2D to 3D conversion coordinate system. The error accumulates during the process of projecting faces from 2D to 3D or from 3D to 2D easily. In addition, reducing surface smoothness of final synthesized faces occurs frequently, which reduces the naturalness of the reconstructed face image.

The second type of methods generated frontal faces without assistant of the 3D model. There were lots of methods based on Generative Adversarial Network (GAN) [1], Huang *et al.* proposed TP-GAN [6], a two-pathway method that combined two networks with perceive global and local features respectively, enhanced the realism of synthetic front face. Tran *et al.* proposed Disentangled Representation learning GAN(DR-GAN) [16], which used pose coding guide generator and discriminator to decouple the feature representation of poses. Couple-Agent Pose-Guided GAN (CAPG-GAN) [7] equipped five facial landmarks heatmaps as pose guidance, supplemented by a coupled discriminator to constrain pose and structure distribution. The pose guidance in CAPG-GAN was identity irrelevant, which limited the diversity of generated faces. Ganimation [17] used action units to get target face with attention masks.

Yin *et al.* [18] proposed a dual attention structure model DA-GAN. DA-GAN equipped self-attention into generator and enhanced discriminator by facial attention, which was assisted by facial masks. Lu *et al.* [19] devised global and local discriminators to generate clarifying front face. Duan and Zhang [20] proposed a coarse-to-fine architecture BoostGAN. BoostGAN regarded multi-occlusive profile faces as input, and then generated front faces for every profile. Finally, the frontal face was boosted from multi-front faces. Similarly, Banerjee *et al.* [21] reconstructed front faces from occlusive faces, too. Cao *et al.* [22] proposed LB-GAN, which is consist of Face Normalizer and Face Editor. Face Normalizer firstly generated frontal view faces, and Face Editor rotated faces to target pose. Liu *et al.* [23] proposed a novel model PPN-GAN to generate front faces progressively. PPN-GAN started from inferring an intermediate face that had a small view difference to the profile face, and then increased the view difference step by step until the frontal view arrives.

Sagonas *et al.* [24] solved face frontalization with a statistical method. Yim *et al.* [25] proposed a front view synthesis method with multi-task learning. Cole *et al.* [26] decomposed face into a set of sparse landmark points and aligned texture maps, which were combined to generate front faces by image warping. Qian *et al.* proposed Additive Focal Variational Auto-encoder (AF-VAE) [27] that combined VAE with GAN. They introduced a novel random gaussian mixture hypothesis to improve the de-entanglement effect of face content and obtained realistic frontal images. Kan *et al.* [28] proposed Stacked Progressive Auto-encoder (SPAE), which iteratively transformed large poses to virtual smaller pose, until target pose was generated. Yang *et al.* [29] and Feng and Yuen [30] proposed similar methods with virtual poses.

The above mentioned methods all used simple pose coding or landmark points as posture information, the detailed information (such as contours and textures) of profile face had not been extensively used.

III. METHOD

A. OVERVIEW

Recent researches on face frontalization barely concerned about the guidance of profile faces. However, edges contain a lot of information, such as the shape of face, facial structure, and even identity. Therefore, making full use of edges in profile improve the quality of reconstructed frontal faces. We believe that using sketch as detailed features improves the generator's perception of details.

Specifically, we firstly generate detailed features for all faces for preprocessing. The profile faces and corresponding detailed features are the input of our model. Then, we employ the proposed generator to convert profile detailed features to frontal ones. Next, we feed profile face and corresponding frontal detailed features together into another generator. Then the frontal faces are generated finally.

In the following sections, detailed features are introduced for better understanding firstly. Then, we describe the architecture of our proposed DGPR. Finally, we introduce the proposed conditional enhancement loss and smoothing loss. The other losses we used are described, too.

B. DETAILED FEATURE

Posture information plays a vital role to reconstruct front faces from profile faces. The posture with rich details make model reconstruct more realistic target face. Inspired by the study of Yi *et al.* [31], we employ [31] to obtain a sketch image of each profile face.

Compared with one-hot coding, sketches contain almost all edge information of profile faces, which can more efficiently guide model to generate high-quality synthetic faces. Unlike previous methods, which share the same coding for multiple profile persons at one pose, detailed features is one-to-one that improve similarity between synthesized and real front faces. Figure 2 shows the difference between posture guidance we used and CAPG-GAN [7].

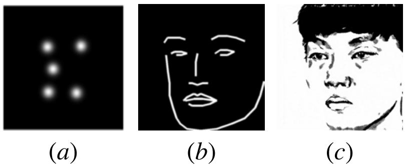


FIGURE 2. The comparison of different guidance. The five facial landmarks heatmap (*a*) is used in CAPG-GAN, (*b*) is 68 facial landmarks with lines, and the sketch image (*c*) is used in our method. It is intuitive that the sketch provides more information than facial landmarks heatmaps.

C. NETWORK STRUCTURE

DGPR can be divided into two parts: The first part is the detailed feature convert generator G_m, which generates a frontal detailed feature. The second part is the pose convert generator G_t, which combines the output of G_m and profile face as inputs to generate front face. Figure 3 shows the overall structure of DGPR based on detailed features.

We represent detailed features as m in this paper. x represents a face image at any pose in the dataset X, and we expect to obtain a synthetic face f corresponding to the target pose c_y. The generated f is expected proximity to real frontal face infinitely. DGPR can be expressed as formulas (1) and (2):

$$m_f = G_m\left(m_x, c_y\right) \tag{1}$$

$$f = G_t\left(x, m_f\right), \tag{2}$$

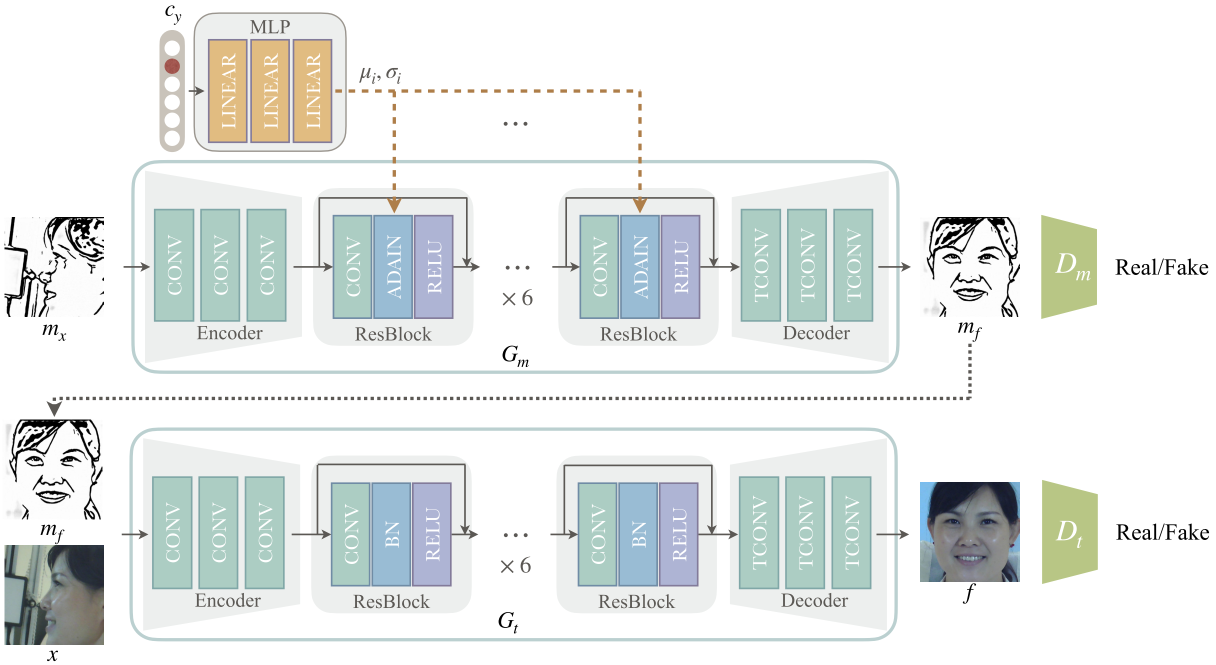


FIGURE 3. The overall structure of DGPR. It consists of 2 generators and corresponding discriminators: the detailed feature convert generator, the pose convert generator, detailed feature discriminator, and faces discriminator.

where c_y is one-hot coding of target pose, m_x is detailed feature of profile face x. G_m reconstruct detailed feature m_f. And f is front face generated by G_t.

In addition, we cycle the training stage by revising input and output at each iteration. The cycle process increases the robustness of model. Specifically, the results obtained are fed back to the model to reconstruct the original input. At the same time, cycle loss is used to optimize the two generators and further improve the performance of our model.

1) DETAILED FEATURE CONVERT GENERATOR G_m

In practice, detailed features of frontal face cannot be obtained. Thus, we design detailed features convert generator G_m to convert detailed features into a frontal one. The structure of G_m is shown in Table 1. The residual blocks *ResBlocks* are composed of *ReflectionPad2d* and convolutional layer with 3 × 3 kernel size. The normalization layers in residual blocks adopt Adaptive Instance Normalize (AdaIN) [32], as shown in formula (3).

$$AdaIN(x, y) = \sigma(y)\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \mu(y) \tag{3}$$

There are differences in the data distribution under different poses. In order to make G_m adaptively learn the difference, we employ Multi-Layer Perceptron (MLP) to learn the parameters μ and σ in Adaptive Instance Normalization layers. The structure of MLP is shown in Table 2, where N is the total number of parameters in Adaptive Instance Normalization.

TABLE 1. The structure of detail feature convert generator G_m.

Layer	Filter/Stride	Output Size
ReflectionPad2d	3	134 × 134 × 3
Conv11	7 × 7/1	128 × 128 × 64
Conv21	3 × 3/2	64 × 64 × 128
Conv31	3 × 3/2	32 × 32 × 256
Conv41	3 × 3/2	16 × 16 × 512
ResBlock ×6	3 × 3/1	16 × 16 × 512
ConvTrans61	3 × 3/2	32 × 32 × 256
ConvTrans71	3 × 3/2	64 × 64 × 128
ConvTrans81	3 × 3/2	128 × 128 × 64
ReflectionPad2d	3	134 × 134 × 64
Conv91	7 × 7/1	128 × 128 × 3

TABLE 2. The structure of MLP. ***N*** is the total number of parameters in adaptive instance normalization layers.

Layers	Input channels	Output channels
Linear	13	256
Linear	256	256
Linear	256	N

2) POSE CONVERT GENERATOR G_t

Frontal detailed features m_f is generated by G_m and is input to G_t together with profile face x. Then, G_t output the final front face f. The x contains face identity and texture information. Meanwhile, frontal detailed feature m_f provides posture and structure information. DGPR combines these two kinds of content together to generate a photorealistic face.

The structure of G_t is different from G_m in the first layer. It adds an additional learnable parameter β for combing m_f and x: $x + \beta m_f$. After down-sampling, a self-attention mechanism is equipped to force model to focus more attention on effective features. In the following *ResBlocks*, G_t adopt Batch Normalization [33] instead of Adaptive Instance Normalize. The settings of other layers are same as G_m.

3) DISCRIMINATORS

The discriminator D_m tries to distinguish generated and real detailed features, while D_t tries to distinguish generated and real images. The adversarial training procedure reduces the difference between generated and real images. In order to make generator achieve better performance and refine the details of generated images at the same time, both discriminators based on PatchGAN [34] are adopted. The structure of discriminators are shown in Table 3.

TABLE 3. The structure of discriminators.

Layer	Filter/Stride	Output Size
Conv11	4 × 4/2	64 × 64 × 64
Conv21	4 × 4/2	32 × 32 × 128
Conv31	4 × 4/2	16 × 16 × 256
Conv41	4 × 4/1	15 × 15 × 512
Conv51	4 × 4/1	14 × 14 × 1

D. LOSS FUNCTIONS

1) ADVERSARIAL LOSS

Judging the authenticity of results can prompt generator to obtain frontal faces with distribution close to real. We use a modified adversarial loss proposed by WGAN-GP [35], [36] to distinguish real or fake images, which adds a gradient penalty to the original adversarial loss. Specifically, the input of G_m is profile detailed feature m_x and frontal pose coding c_y, the output is frontal detailed feature m_f. Adversarial loss L_m can be expressed as (4):

$$L_m = \mathbb{E}_{m_x \sim \mathbb{P}_{m_x}}\left[D_m\left(m_f\right)\right] - \mathbb{E}_{m_x \sim \mathbb{P}_{m_\chi}}\left[D_m\left(m_x\right)\right] + \lambda_1 \mathbb{E}_{\tilde{I} \sim \mathbb{P}_I}\left[\left(\left\|\nabla_{\tilde{I}} D_I(\tilde{I})\right\|_2 - 1\right)^2\right], \quad (4)$$

where P_{m_x} is the data distribution of profile detailed features, $P_{\tilde{I}}$ is random difference distribution, λ_1 is penalty coefficient.

G_t also uses the adversarial loss L_t proposed by WGAN-GP, which is shown in (5). P_x is the data distribution of X. $P_{\tilde{o}}$ is random difference distribution and λ_2 is penalty coefficient.

$$L_t = \mathbb{E}_{x \sim \mathbb{P}_x}\left[D_t(f)\right] - \mathbb{E}_{x \sim \mathbb{P}_x}\left[D_t(x)\right] + \lambda_2 \mathbb{E}_{\tilde{o} \sim \mathbb{P}_{\tilde{o}}}\left[\left(\left\|\nabla_{\tilde{I}} D_o(\tilde{o})\right\|_2 - 1\right)^2\right] \quad (5)$$

2) RECONSTRUCTION LOSS

With the thought of cycle consistency, we adopt a reconstruction structure that we feed final results back into the model to reconstruct inputs. The performance of generators would benefit from cycle reconstruction loss. We adopt reconstruction loss functions in both G_m and G_t to optimize the generator parameters. Formula (6) is reconstruction loss for G_m, and formula (7) for G_t.

$$L_{rm} = \left\|m_f - \hat{m}_f\right\|_2 \quad (6)$$

$$L_{rt} = \|f - \hat{f}\|_2 \quad (7)$$

3) SMOOTHING LOSS

In the beginning, we used the same method to train G_m and G_t, but due to the different image types processed by two generators, G_t gets good generation while G_m reaches collapse easily. Inspired by research [17], smoothing loss L_s and $\hat{L}_s$ promote to reduce sharpness of detailed features and avoid collapse. L_s and $\hat{L}_s$ is formulated as equations 8 and 9 respectively.

$$L_s = \lambda_3 \mathbb{E}_{m_x \sim \mathbb{P}m_x} \times \left[\sum_{i,j}^{H,W}\left[\left(\mathrm{M}_{i+1,j} - \mathrm{M}_{i,j}\right)^2 + \left(\mathrm{M}_{i,j+1} - \mathrm{M}_{i,j}\right)^2\right]\right] \quad (8)$$

$$\hat{L}_s = \lambda_4 \mathbb{E}_{m_f \sim \mathbb{P}m_f} \times \left[\sum_{i,j}^{H,W}\left[\left(\widehat{\mathrm{M}}_{i+1,j} - \widehat{\mathrm{M}}_{i,j}\right)^2 + \left(\widehat{\mathrm{M}}_{i,j+1} - \widehat{\mathrm{M}}_{i,j}\right)^2\right]\right] \quad (9)$$

where $\mathrm{M} = G_m\left(m_x \mid c_y\right)$, $\widehat{\mathrm{M}} = G_m\left(m_f \mid c_x\right)$, $\mathbb{P}_{m_f}$ is the data distribution of generated frontal detailed features. λ_3 and λ_4 are the parameters of smoothing loss.

4) CONDITIONAL ENHANCEMENT LOSS

To enhance the constraint of pose conditions on generated results, DGPR employs a fully connected layer to classify the poses of features that output from *Conv*41 in G_t. The cross entropy loss is calculated as conditional enhancement loss L_{c1} and L_{c2}. This loss contains two items: the first item calculates the difference between generated and target pose and constrains generated face to match the target pose. The second item uses real images as reference to learn how to make pose judgments from real distribution.

For G_m, we expect to enhance the influence of guidance information c_y. The conditional enhancement loss for G_m is as follows:

$$L_{c1} = L_{CrossEntropy}\left(D_{c1}\left(G_m\left(m_x \mid c_y\right)\right), c_y\right) + L_{CrossEntropy}\left(D_{c1}\left(m_x\right), c_x\right), \quad (10)$$

For G_t, we expect to enhance the influence of detailed feature m_f. The conditional enhancement loss for G_t is as follows:

$$L_{c2} = L_{CrossEntropy}\left(D_{c2}\left(G_t\left(x \mid m_f\right)\right), m_f\right) + L_{CrossEntropy}\left(D_{c2}(x), m_x\right), \quad (11)$$

5) IDENTITY LOSS

To enhance the identity consistency of synthetic faces, we need to determine whether the identity between profile and generated front faces are identical. The identity loss L_{m-id} for G_m, which can not only punish the differences between profile and reconstructed frontal faces, but also force G_m to preserve identity.

$$L_{m-id} = \mathbb{E}_{mf \sim \mathbb{P}_{mf}}\left[\left\|G_m\left(G_m\left(m_x \mid c_y\right) \mid c_x\right) - x\right\|_1\right] \quad (12)$$

For G_t, the identity consistency loss L_{t-id} is adopted to maintain the identity of generated images. In addition to L1 loss, we also constrained identity consistency at feature level. The FaceNet [37], which is pre-trained on VGGFace [38], is employed to extract the identity features. Then the mean square error (MSE) distance between test and registered features is computed as L_{t-id}. All data input to FaceNet are scaled to 160 × 160 in practice. L_{t-id} is as follows:

$$\begin{aligned} L_{t-id} = {} & \mathbb{E}_{x\sim\mathbb{P}_x}\frac{1}{n}\sum\left(F_f - F_x\right)^2 \\ & + \mathbb{E}_{f\sim\mathbb{P}_f}\left[\left\| G_f\left(G_f\left(x \mid m_f\right) \mid m_x\right) - x\right\|_1\right], \end{aligned} \tag{13}$$

where F_f is the identity features of generated front faces and F_x for profile faces.

6) MULTI-SCALE PIXEL-LEVEL LOSS

Inspired by CAPG-GAN [7], we apply a multi-scale pixel-level loss L_{pix} for f to constrain its content consistency, accelerate optimization and reconstruct more detailed information.

$$L_{pix} = \frac{1}{S}\sum_{S=1}^{S}\frac{1}{W_S H_S C}\sum_{w,h,c=1}^{W_{S,H_S,C}}\left|\hat{I}^{b}_{s,w,h,c} - I^{b}_{s,w,h,c}\right|, \tag{14}$$

where S is the number of scales, W_s and H_s represent the width and height of images for each scale respectively, and C is the number of image channel.

7) TOTAL LOSS

The total loss for G_m is shown in Equation (16):

$$\begin{aligned} L_{G_m} = {} & \lambda_m L_m + \lambda_{rm} L_{rm} + \lambda_s(L_s + \hat{L}_s) \\ & + \lambda_{c1} L_{c1} + \lambda_{m-id} L_{m-id}, \end{aligned} \tag{15}$$

and the loss for G_t:

$$L_{G_t} = \lambda_t L_t + \lambda_{rt} L_{rt} + \lambda_{c2} L_{c2} + \lambda_{t-id} L_{t-id} + L_{pix}, \tag{16}$$

where $\lambda_s, \lambda_{c1}, \lambda_{c2}, \lambda_{m-idt}, \lambda_{f-idt}, \lambda_{pix}$ are hyperparameters that control weight of each loss item. Their values are shown in Table 4.

TABLE 4. The values of loss coefficients.

λ_m	λ_t	λ_{rm}	λ_{rt}	λ_s	λ_{c1}	λ_{c2}	λ_{m-id}	λ_{f-id}	λ_{pix}
0.5	0.5	10	10	1e−10	0.01	0.1	0.05	10	0.01

The final objective function can be formulated as (17), where generators try to minimize the objective function and discriminators try to maximize it. The algorithm flow of DGPR is illustrated as Algorithm 1.

$$\mathcal{L} = \arg\min_{G}\max_{D}\left(L_{G_m} + L_{G_t}\right) \tag{17}$$

Algorithm 1 Algorithm of Proposed Detailed Feature Guided Generative Adversarial Pose Reconstruction Network

```
Require:
  Input image $x$, detailed feature $m_f$, target pose one-hot coding $c_y$
Ensure:
  Frontal face image $f$
  for $epoch = 1$ to $MAX_{EPOCH}$ do
    $m_f = G_m\left(m_x, c_y\right)$
    $f = G_t\left(x, m_f\right)$
    Update discriminators $D_m$,$D_t$
    Update generators $G_m$ and $G_t$, The gradient is $\nabla\left(L_{G_m}\right) = \nabla\left(L_m\right) + \lambda_s\nabla\left(L_s + L_{\hat{s}}\right) + \lambda_{c1}\nabla\left(L_{c1}\right) + \lambda_{m-idt}\nabla\left(L_{m-idt}\right)$
  end for
  return $f$
```

IV. EXPERIMENTS

We implemented our method by PyTorch [39] and all experiments were conducted on a single GeForce GTX 2080Ti GPU with 11G memory. In the following sections, we will introduce the dataset used in our experiments, qualitative and quantitative analysis of DGPR compared with the state-of-the-art methods, also the ablation study of DGPR.

A. DATASETS

To demonstrate the effect of DGPR in the task of face frontalization, we conduct experiments on the M^{2}FPA [40] and CAS-PEAL [41]. M^{2}FPA is provided by the Institute of Automation, Chinese Academy of Sciences. It is a great comprehensive multi-view public dataset currently that available for facial pose analysis research. M^{2}FPA contains a total of 397,544 pictures of 229 subjects, with 62 poses, 4 attribute changes, and 7 lighting changes. Each subject's 62 poses were captured simultaneously, including 13 yaw angles (−90° ∼ 90°), 5 pitch angles (−30° ∼ 45°) and 44 yaw and pitch angles. In our experiments, we only concern about face frontalization of different poses of 13 yaw angles, and pitch angles are ignored. Five types of glasses and three kinds of expressions improve diversity. Several previous methods such as DR-GAN [16], TP-GAN [6] and CAPG-GAN [7] are provided effective benchmarks for face frontalization and pose invariant face recognition results on M^{2}FPA. For a fair comparison, we evaluate DGPR on M^{2}FPA dataset with the official train/test split.

The CAS-PEAL [41] is a public Chinese face database with controlled pose, expression, accessory, and lighting. It contains 30,863 images of 1,040 subjects, and all images are grayscale. Following previous researches, we use images with various poses 0°, ±15°, ±30° and ±45°. We randomly split the dataset into train/validation/test sets with 7:1:2.

B. IMPLEMENT DETAILS

All the loss balance coefficients can be seen in Table 4. The pose number is 13 on M^{2}FPA and 11 on CAS-PEAL.

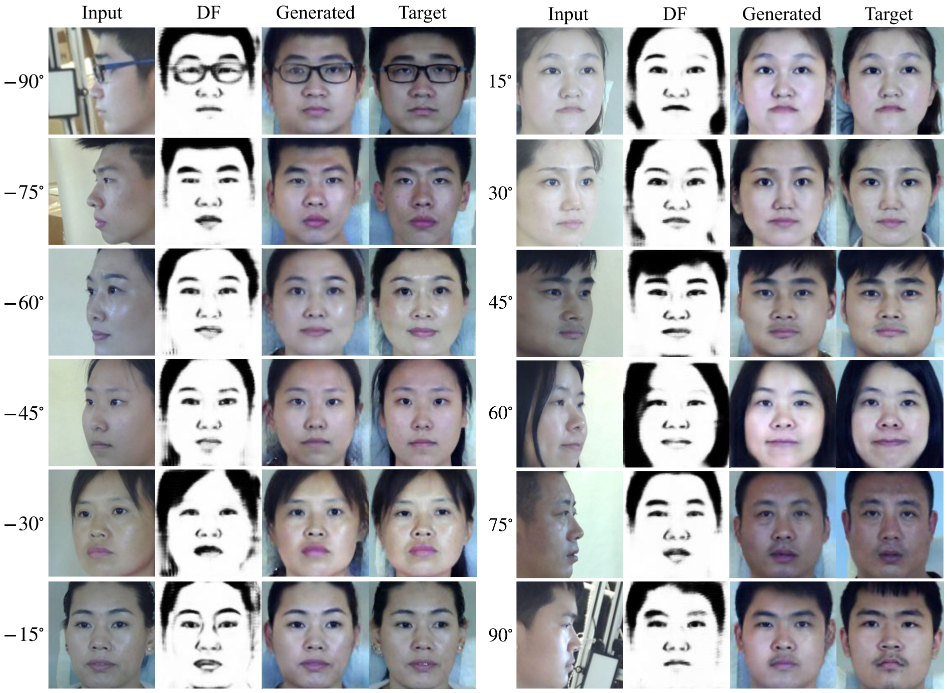


FIGURE 4. Results of face frontalization on M^{2}FPA. Every four columns are a group, the first column is profile face, the second column is generated detailed feature (DF), the third column is generated front face, and the fourth column is real front face. For the first group, from top to bottom, illustrates results when input faces various from −90° to −15°. For the second group, from top to bottom, is the results when input faces various from 15° to 90°. All the intervals are 15°.

The output dimension N of MLP is 24, which is also the parameters number of AdaIN layers. All images are resized to 128×128. Our model is trained for 30 epochs with a learning rate of 2e-5, under ADAM [42] optimizer with batch size 16. The average training time is 2.4 hours per epoch on M^{2}FPA and 30 minutes per epoch on CAS-PEAL, and the average generation time is 0.2 seconds per face on both datasets.

C. EVALUATION METRICS

We evaluate the quantitative results of DGPR with face recognition performance and illustrate generated frontal face for qualitative evaluation. Specifically, we use rank-1 recognition rate as face recognition evaluation metric. For fair comparison, we follow the previous researches' setting in the evaluation. LightCNN [43] is used as face feature extractor. Faces of different yaw angles are fed as input, which can be expressed as ±15°, ±30°, ±45°, ±60°, ±75°, ±90° respectively. Rank-1 recognition rate is calculated by averaging two corresponding poses. For example, rank-1 value of ±15° is the average of +15° and −15°.

D. QUALITATIVE ANALYSIS

The experimental results on M^{2}FPA are illustrated in Figure 4, and results on CAS-PEAL are illustrated in Figure 5. As is shown in Figure 4, DGPR generates detailed features that are similar to real front faces. Facial characteristic can also be generated accurately, such as glasses, bangs, and eyebrows. In addition, DGPR is able to adapt to changes in poses and generate realistic front faces. In Figure 5, faces can be generated with similar characters as input profile faces. These demonstrate the effectiveness of our proposed model.

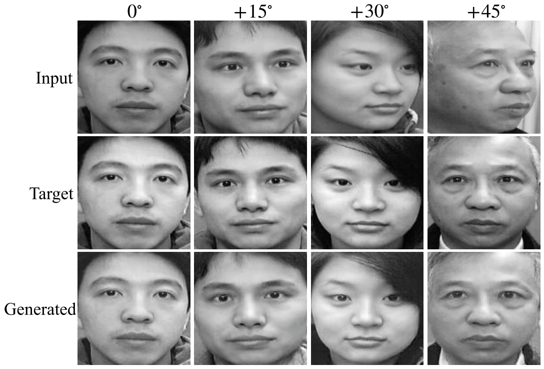


FIGURE 5. Results of face frontalization on CAS-PEAL. From left to right columns, we illustrate generated results under 0°, 15°, 30°,and 45°.

In order to prove that DGPR can preserve the identity of one person from diverse poses, faces of same person from different poses are fed into DGPR. The comparison about generated frontal faces as shown in Figure 6. It shows that DGPR can reconstruct photorealistic synthetic faces and preserving identity consistency when poses are closer to 0°. While the pose is larger than 60°, the identity consistency between reconstructed and real images decreases slightly compared with smaller poses. Although the contour of face

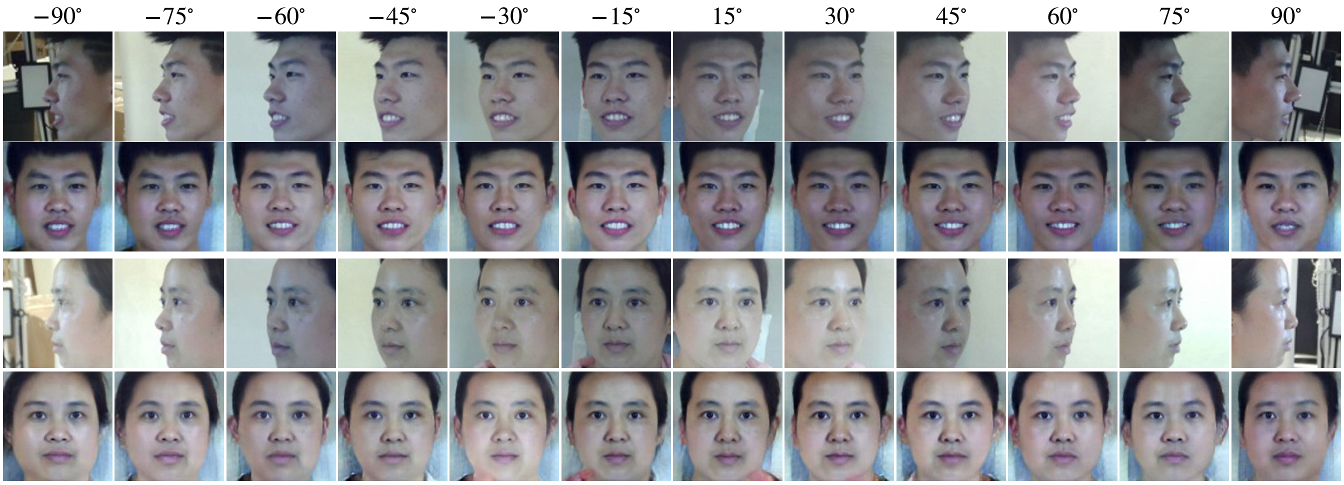


FIGURE 6. The generated front faces from diverse poses. For each group of two rows, the first row is input profile faces, and the second is reconstructed front faces. From left to right, the pose changes from −90° to 90° differs by 15°.

TABLE 5. Rank-1 recognition rate (%) of different models on M^2FPA. First row indicate the profile poses. It can be seen that DGPR achieves state-of-the-art rank-1 recognition rate.

Model	±15°	±30°	±45°	±60°	±75°	±90°
Original	100	100	99.8	98.6	86.9	51.7
TP-GAN	99.9	99.8	99.1	97.3	87.6	62.1
DR-GAN	98.9	97.9	95.7	89.5	70.3	35.5
CAPG-GAN	99.9	99.7	99.4	96.4	87.2	63.9
Ours	**100.0**	**100.0**	**99.9**	**98.6**	**88.1**	**68.1**

TABLE 6. Rank-1 recognition rate (%) of different models on CAS-PEAL at pitch 0°.

Model	±15°	±30°	±45°	Avg
CR-GAN	97.61	95.80	89.73	94.38
TP-GAN	100.00	99.94	98.71	99.55
DA-GAN	100.00	100.00	99.70	99.90
Ours	**100.00**	**100.00**	**99.84**	**99.95**

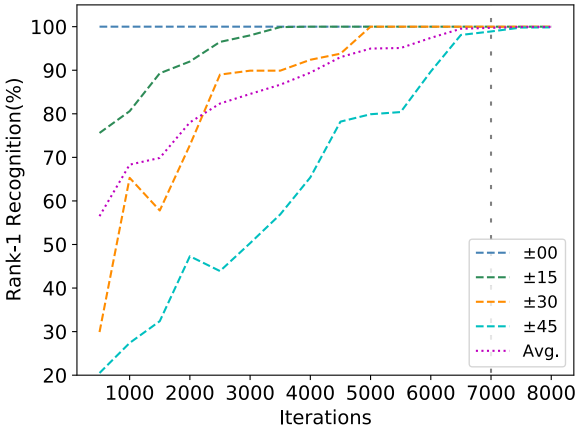


FIGURE 7. Rank-1 recognition rate at different training iteration on CAS-PEAL. The rank-1 recognition rate under different poses tends to stabilize around 7000 iterations of training.

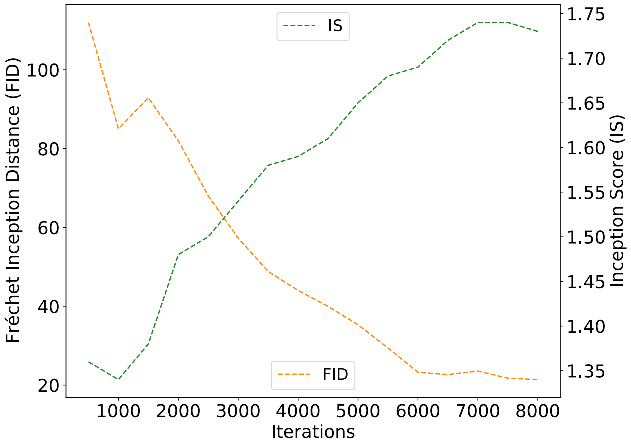


FIGURE 8. FID and IS score at different training iteration on CAS-PEAL.

is slightly deformed, the shape of face remains normal and the identity still maintains a high consistency.

E. QUANTITATIVE ANALYSIS

Table 5 shows rank-1 recognition rate of generated frontal faces on M^2FPA under poses ±15°, ±30°, ±45°, ±60°, ±75°, ±90°. It can be seen that rank-1 recognition rate of different poses has been improved. While profile poses are at ±60°, ±75° and ±90°, the rank-1 recognition rate is increased by 2%, 1% and 6% respectively, which compared with the state-of-the-art methods. It demonstrated that DGPR significantly explore the effect of detailed features, enrich reconstructed frontal faces from even large poses.

On CAS-PEAL, we report results in Table 6. We compared our model with state-of-the-art methods CR-GAN [44], TP-GAN [6] and DA-GAN [18]. It can be seen that our method outperforms state-of-the-art methods as the average rank-1 recognition rate is improved by 0.05%. To further demonstrate the effectiveness of our model, we report results under pose ±22° and ±67°, which are not reported in previous researches but exist in dataset. Rank-1 recognition rates are 100% and 98.54% under ±22° and ±67° respectively.

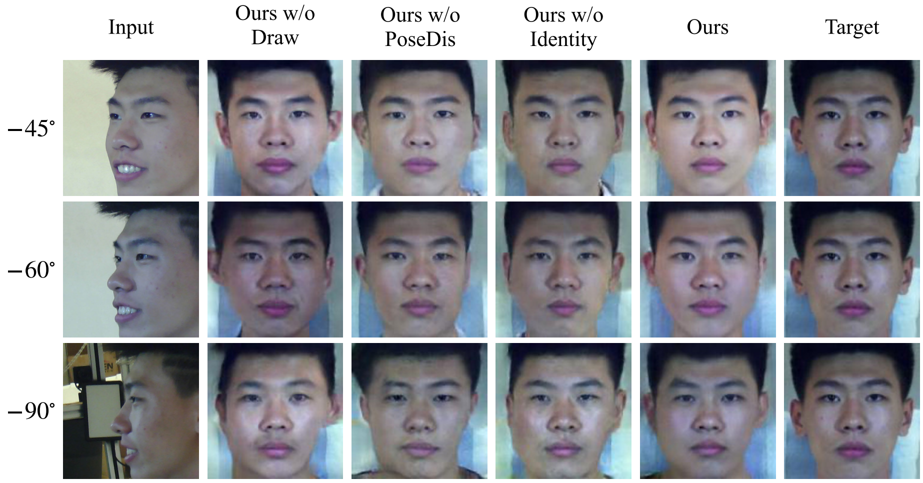


FIGURE 9. Ablation experiment results on M^2FPA dataset.

TABLE 7. Quantitative results under different posture guidances, Compared under rank-1 recognition rate(%). DF denotes detailed features and 68-L denotes 68 Landmarks.

Guidance	±15°	±30°	±45°	±60°	±75°	±90°
68-L	100.0	100.0	99.9	97.8	87.7	60.2
DF	**100.0**	**100.0**	**99.9**	**98.6**	**88.1**	**68.1**

To demonstrate the convergence of our model, we illustrate rank-1 recognition rate at different training iteration on CAS-PEAL in Figure 7. Four lines under poses ±00°, ±15°, ±30° and ±45° are plotted, and the average rank-1 accuracy is plotted as dot-line, too. It can be seen that the model reaches convergence after training 7000 iterations. The results demonstrate our model is well converged. The fluctuating of curves reflect the instability of GAN training. Furthermore, the Fréchet Inception Distance (FID) score [45] and Inception Score (IS) [35] score at different training iteration on CAS-PEAL are shown in Figure 8. The FID score is the lower the better, and IS score is the higher the better. The max value of IS is 2. It can be seen that FID and IS reach stable status at around 7000 iterations.

F. ABLATION STUDY

1) VALIDITY OF DETAILED FEATURES

In order to analysis the effect of different posture guidance on the generation procedure, we compared the front faces generated by DGPR with 68 landmarks and detailed features respectively, which are two kinds of posture guidance. The quantitative comparison results are shown in The Table 7.

It can be seen that regarding detailed features as guidance, resolution and identity of generated front faces significantly improved. Among all posture guidances, the model with detailed features generated better front faces, and rank-1 recognition rate outperforms the model with 68 landmarks. What's more, the rank-1 recognition rate is greatly improved under ±90°, which proves detailed features is valid.

TABLE 8. Ablation study of DGPR with the metric of rank-1 reconstruction rate (%).

Model	±15°	±30°	±45°	±60°	±75°	±90°
Real Data	100.0	100.0	99.8	98.6	86.9	51.7
w/o Draw	99.7	99.4	95.0	79.6	60.6	36.5
w/o PosDis	100.0	100.0	99.8	96.8	87.9	63.7
w/o FaceNet	100.0	99.8	97.6	86.1	67.4	50.0
Ours	**100.0**	**100.0**	**99.9**	**98.6**	**88.1**	**68.1**

2) VALIDITY OF DIFFERENT COMPONENTS

In order to prove the effect of different modules in DGPR, each module was sequentially removed to carry out reconstruction experiments from profile faces to front faces. *w/o Pose Dis* means removing conditional enhancement loss and corresponding pose classifier; *w/o Draw* means removing detailed feature convert generator G_m, then reconstruct front faces without detailed features; *w/o FaceNet* means removing feature-level identity constraint, including the identity consistency loss and corresponding embedding extracted by pre-trained FaceNet.

As shown in Figure 9, with the absence of detailed feature guidance or posture constraints, more obvious facial deformation occurred. The deformation enlarges when pose angle is larger. As is seen in Table 8, due to the absence of detailed features guidance or posture constraints, facial deformation decreases rank-1 recognition rate of frontal images reconstructed by DGPR.

As shown in Table 9, Fréchet Inception Distance (FID) score [45] and Inception Score (IS) [35] of reconstructed front faces are improved after removing the constraint of identity features, while rank-1 recognition rate was significantly improved with feature-level identity constraint. This observation suggests that the identity constrain has a slight negative impact on image quality, but the identity consistency between reconstructed and real images improved significantly. We set to adjust the coefficient λ_{m-id} for identity feature consis-

TABLE 9. Ablation study of DGPR with the metric of FID and IS.

Model	FID↓	IS↑
Real Data	0.000	2.0
Ours w/o Draw	19.683	1.869
Ours w/o PosDis	14.853	1.949
Ours w/o FaceNet	**13.455**	**1.950**
Ours	16.610	1.886

tency loss L_{m-id} to reach a balance between generating more photorealistic images and higher face recognition accuracy. Experiments show that each component of DGPR has various effects, removing any of them will decrease the performance.

V. CONCLUSION

This paper introduces Detailed Feature Guided Generative Adversarial Pose Reconstruction Network (DGPR), which explores the guidance effect of profile faces to generate front faces. DGPR utilizes face sketches as detailed features, which stores rich pose clues and detailed information, and guides generators to obtain realistic frontal faces. DGPR contains a dual generator structure: the detailed feature convert generator and the pose convert generator, which can take advantage of both profile faces and detailed features. In addition, we propose smoothing loss to reduce edge sharpness in generated faces, conditional enhancement loss to constrain pose consistency. The experimental results show that our method outperforms previous methods on face recognition rate on M^{2}FPA and CAS-PEAL datasets. Specifically, our model gains a great improvement for face frontalization from extreme poses, which suggests the effect of our model.

REFERENCES

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 1–7.

[2] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. 2nd Int. Conf. Learn. Represent. (ICLR)*, Banff, AB, Canada, Apr. 2014.

[3] T. Hassner, S. Harel, E. Paz, and R. Enbar, "Effective face frontalization in unconstrained images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4295–4304.

[4] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, "Towards large-pose face frontalization in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4010–4019.

[5] Z. Zhang, X. Chen, B. Wang, G. Hu, W. Zuo, and E. R. Hancock, "Face frontalization using an appearance-flow-based convolutional neural network," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2187–2199, May 2019.

[6] R. Huang, S. Zhang, T. Li, and R. He, "Beyond face rotation: Global and local perception GAN for photorealistic and identity preserving frontal view synthesis," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2458–2467.

[7] Y. Hu, X. Wu, B. Yu, R. He, and Z. Sun, "Pose-guided photorealistic face rotation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8398–8406.

[8] F. Luo, B. Du, L. Zhang, L. Zhang, and D. Tao, "Feature learning using spatial-spectral hypergraph discriminant analysis for hyperspectral image," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2406–2419, Jul. 2019.

[9] G. Shi, H. Huang, and L. Wang, "Unsupervised dimensionality reduction for hyperspectral imagery via local geometric structure feature learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 8, pp. 1425–1429, Aug. 2020.

[10] J. Cao, Y. Hu, H. Zhang, R. He, and Z. Sun, "Learning a high fidelity pose invariant model for high-resolution face frontalization," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 1–11.

[11] J. Deng, S. Cheng, N. Xue, Y. Zhou, and S. Zafeiriou, "UV-GAN: Adversarial facial UV map completion for pose-invariant face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp,. 7093–7102.

[12] H. Zhou, J. Liu, Z. Liu, Y. Liu, and X. Wang, "Rotate- and-render: Unsupervised photorealistic face rotation from single-view images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5910–5919.

[13] X. Zhu, X. Liu, Z. Lei, and S. Z. Li, "Face alignment in full pose range: A 3D total solution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 78–92, Jan. 2019.

[14] J. Zhao, L. Xiong, Y. Cheng, Y. Cheng, J. Li, L. Zhou, Y. Xu, J. Karlekar, S. Pranata, S. Shen, J. Xing, S. Yan, and J. Feng, "3D-aided deep pose-invariant face recognition," in *Proc. IJCAI*, 2018, pp. 1–8.

[15] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proc. 26th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, 1999, pp. 187–194.

[16] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning GAN for pose-invariant face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1283–1292.

[17] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. MorenoNoguer, "Ganimation: Anatomically-aware facial animation from a single image," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 818–833.

[18] Y. Yin, S. Jiang, J. P. Robinson, and Y. Fu, "Dual-attention GAN for large-pose face frontalization," in *Proc. 15th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, Nov. 2020, pp. 249–256.

[19] Y. Lu, T. Sun, X. Jiang, K. Xu, and B. Zhu, "Frontal view synthesis based on a novel GAN with global and local discriminators," in *Proc. 12th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, Oct. 2019, pp. 1–5.

[20] Q. Duan and L. Zhang, "BoostGAN for occlusive profile face frontalization and recognition," 2019, *arXiv:1902.09782*. [Online]. Available: http://arxiv.org/abs/1902.09782

[21] S. Banerjee and S. Das, "SD-GAN: Structural and denoising GAN reveals facial parts under occlusion," 2020, *arXiv:2002.08448*. [Online]. Available: http://arxiv.org/abs/2002.08448

[22] J. Cao, Y. Hu, B. Yu, R. He, and Z. Sun, "Load balanced GANs for multi-view face image synthesis," 2018, *arXiv:1802.07447*. [Online]. Available: http://arxiv.org/abs/1802.07447

[23] L. Liu, L. Zhang, and J. Chen, "Progressive pose normalization generative adversarial network for frontal face synthesis and face recognition under large pose," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 4434–4438.

[24] C. Sagonas, Y. Panagakis, S. Zafeiriou, and M. Pantic, "Robust statistical face frontalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3871–3879.

[25] J. Yim, H. Jung, B. Yoo, C. Choi, D. Park, and J. Kim, "Rotating your face using multi-task deep neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 676–684.

[26] F. Cole, D. Belanger, D. Krishnan, A. Sarna, I. Mosseri, and W. T. Freeman, "Synthesizing normalized faces from facial identity features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3386–3395.

[27] S. Qian, K.-Y. Lin, W. Wu, Y. Liu, Q. Wang, F. Shen, C. Qian, and R. He, "Make a face: Towards arbitrary high fidelity face manipulation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10032–10041.

[28] M. Kan, S. Shan, H. Chang, and X. Chen, "Stacked progressive auto-encoders (SPAE) for face recognition across poses," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1883–1890.

[29] J. Yang, S. Reed, M.-H. Yang, and H. Lee, "Weakly-supervised disentangling with recurrent transformations for 3D view synthesis," 2016, *arXiv:1601.00706*. [Online]. Available: https://arxiv.org/abs/1601.00706

[30] G. C. Feng and P. C. Yuen, "Recognition of head-and-shoulder face image using virtual frontal-view image," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 30, no. 6, pp. 871–882, Nov. 2000.

[31] R. Yi, Y.-J. Liu, Y.-K. Lai, and P. L. Rosin, "Unpaired portrait drawing generation via asymmetric cycle mapping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8214–8222.

[32] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1510–1519.

[33] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 448–456.

[34] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.

[35] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: https://arxiv.org/abs/1701.07875

[36] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," 2017, *arXiv:1704.00028*. [Online]. Available: https://arxiv.org/abs/1704.00028

[37] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[38] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age," in *Proc. 13th IEEE Int. Conf. Automatic Face Gesture Recognit. (FG)*, 2018, pp. 67–74.

[39] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," 2019, *arXiv:1912.01703*. [Online]. Available: http://arxiv.org/abs/1912.01703

[40] P. Li, X. Wu, Y. Hu, R. He, and Z. Sun, "M2FPA: A multi-yaw multi-pitch high-quality dataset and benchmark for facial pose analysis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10042–10050.

[41] W. Gao, B. Cao, S. Shan, X. Chen, D. Zhou, X. Zhang, and D. Zhao, "The CAS-PEAL large-scale Chinese face database and baseline evaluations," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 1, pp. 149–161, Jan. 2008.

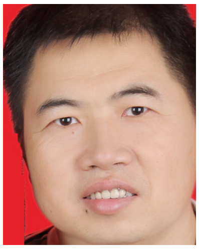
[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015.

[43] X. Wu, R. He, Z. Sun, and T. Tan, "A light CNN for deep face representation with noisy labels," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2884–2896, Nov. 2018.

[44] J. Zhao, Y. Cheng, Y. Xu, L. Xiong, J. Li, F. Zhao, K. Jayashree, S. Pranata, S. Shen, J. Xing, S. Yan, and J. Feng, "Towards pose invariant face recognition in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2207–2216.

[45] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 1–38.

JINLIN HAO is currently pursuing the master's degree with the School of Electrical Engineering, Guangxi University. Her research interests include face pose reconstruction and face liveness detection.



XUEYUN CHEN is currently an Associate Professor and a Doctoral Supervisor with the School of Electrical Engineering, Guangxi University. His research interests include target detection and recognition in remote sensing image, face detection, road detection, and automatic driving.