IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 22, 2020, accepted April 1, 2020, date of publication April 6, 2020, date of current version April 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2986079

Feature-Improving Generative Adversarial Network for Face Frontalization

CHANGLE RONG[1], XINGMING ZHANG[1], AND YUBEI LIN[2]
[1]Department of Computer Science and Engineering, South China University of Technology, Guangzhou 510641, China
[2]Department of Software, South China University of Technology, Guangzhou 510641, China
Corresponding author: Yubei Lin (yupilin@scut.edu.cn)

This work was supported by the Guangdong Provincial Science and Technology Plan of China under Grant 2017B010111003.

ABSTRACT Face frontalization can boost the performance of face recognition methods and has made significant progress with the development of Generative Adversarial Networks (GANs). However, many GAN-based face frontalization methods still perform relatively weak on face recognition tasks under large face poses. In this paper, we propose Feature-Improving GAN (FI-GAN) for face frontalization, which aims to improve the recognition performance under large face poses. We assume that there is an inherent mapping between the frontal face and profile face, and their discrepancy in deep representation space can be estimated. The generation module of FI-GAN has a compact module named Feature-Mapping Block that helps to map the features of profile face images to the frontal space. Moreover, we produce a feature discriminator that can distinguish the features of profile face images from those of ground true frontal face images, which guide the generation module to provide high-quality features of profile faces. We conduct experiments on the MultiPIE, Labeled Faces in the Wild (LFW), and Celebrities in Frontal-Profile (CFP) databases. Our method is comparable to state-of-the-art methods under small poses and outperforms them on large pose face recognition.

INDEX TERMS Face frontalization, face recognition, generative adversarial network.

I. INTRODUCTION

Face frontalization is an interesting problem in both human and machine facial processing and recognition. It is dedicated to addressing pose variations, which are regarded as bottlenecks on face recognition tasks. The methods for face frontalization try to recover frontal face images from profile face images, and then use the recovered face images for recognition.

Earlier efforts [1]–[4] for face frontalization usually utilize 3D geometrical transformations to recover a frontal face image from a profile face image. Their results suffer from deformation under large poses due to severe texture loss. Recently, many face frontalization methods [5]–[9] based on Generative Adverisal Networks (GANs) are proposed. These methods mainly consist of a generator and a discriminator. The generator usually adopts the encoder-decoder architecture, while the discriminator is used for guiding the generator to learn photo-realistic face frontalization. Along with sophisticated loss functions, pairs of face images (frontal and profile) from MultiPIE database [10] are used for training. Although these methods can produce photo-realistic face images under large poses, their performance on face recognition tasks decreases severely when the pose degree achieves 75°. Observing that most methods perform better under smaller poses, we consider that the performance under large poses can be improved by mapping the intermediate features to frontal space in the generative process.

To address the above problem, we propose a face frontalization method named Feature-Mapping Generative Adversarial Network (FI-GAN). There are mainly two differences between the FI-GAN and other GAN-based methods. Firstly, the generation module of FI-GAN has a Feature-Mapping Block, which is similar to DREAM-Block [11]. It can evaluate the discrepancy between the features of profile faces and those of frontal faces. What's more, the discrimination module of our FI-GAN contains a feature discriminator, which competes against the generation module. The former aims to distinguish the intermediate features of profile faces from those of ground true frontal faces, which enforces the latter to produce the high-quality features of profile faces.

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

By mapping the intermediate features of profile faces to frontal space, FI-GAN performs relatively well under large poses.

This paper makes the following contributions:

- We propose the FI-GAN for face frontalization, which aims to achieve state-of-the-art face recognition performance under large poses.
- The generation module of FI-GAN contains a Feature-Mapping Block, which helps to map the features of profile face images to the frontal space.
- A feature discriminator is proposed to improve the features produced by our Feature-Mapping Block further.

The remainder of this paper is organized as follows. In section II, we briefly review some related works. The architecture and details of the FI-GAN are explained in section III. Experimental results and analysis are reported in section IV. In section V, we conclude this paper.

II. RELATED WORK

A. POSE-INVARIANT REPRESENTATION LEARNING

Extracting pose-invariant features is one of the ways to address pose problems on face recognition tasks. Many typical face recognition methods like Light CNN-29 [12] suffer from large poses because their training databases have long-tailed pose distributions. Conventional approaches often leverage metric learning [13] and robust descriptors [14] to tackle pose variance. In contrast, deep learning methods usually handle pose variance by enlarging the training databases [15] or designing special architectures [11], [16]. For example, Wang *et al.* [15] provide a new database named IMDb-Face, which contains more face images under large poses than others. The IMDb-Face empowers them to train strong convolutional networks that can produce pose-invariant features. Cao *et al.* [11] propose a pose-specific module named DREAM-Block, which tries to map the features of profile face images to the frontal space.

B. FACE FRONTALIZATION

Face frontalization is very challenging due to self-occlusion. Existing methods for face frontalization can be divided into three categories: 3D-based methods [1]–[4], statistical methods [17], and deep-learning-based methods [18]–[22]. 3D-based methods usually utilize 3D geometrical transformations to render a frontal face with a mean 3D face model [1], [2] or an identity specific 3D model [3], [4]. These methods perform well under small poses, but their results get much worse under large poses due to severe texture loss. Statistical methods, like [17], use a statistical model for joint frontal view reconstruction and landmark localization by solving a constrained low-rank minimization problem. However, these methods also suffer from poor generalizability under large poses. Deep learning methods usually apply CNNs [18], RNNs [21] or auto-encoders [22] in early time. Yim *et al.* [18] use locally connected convolutional layers for feature extraction and fully connected layer for synthesis. Though these methods get high recognition rates, their synthesized images may lack fine details and tend to be blurry under large poses. Intermediate features instead of synthesized images are used in face recognition tasks. In recent years, some methods based on GANs [5]–[9] are proposed, which are effective in recovering both photo-realistic and identity-preserving face images even under large poses.

In summary, GAN-based methods provide superior performance for face frontalization and recognition under large poses. In this paper, we combine the typical GAN-based model and modules that contribute to providing pose-invariant intermediate features. Our FI-GAN contains the Feature-Mapping Block, which is similar to DREAM-Block. It can map the intermediate features of profile face images to the frontal space. To improve the intermediate features further, we propose a feature discriminator. Our method is explained in section III in detail.

III. PROPOSED APPROACH

The overall framework of the Feature-Improving Generative Adversarial Network (FI-GAN) is described in Fig. 1, which mainly consists of the generation module G and the discrimination module D. G aims to synthesize a photo-realistic and identity-preserving frontal view image x^s from a profile face image x^p. It contains an encoder G_{enc}, a pose estimator G_{pose}, a Feature-Mapping Block G_{fm}, and a decoder G_{dec}. G_{enc} can transform face images into intermediate features. The features of profile face and those of ground true face are denoted as f^p and f^g, respectively. G_{pose} aims to calculate the yaw angle α of the profile face. The G_{fm} is used for mapping f^p to the frontal space along with pose degree α. The features produced by G_{fm} are denoted as $f^{p'}$. G_{dec} can transform the features $f^{p'}$ into the frontal face image x^s. Discrimination module D aims to guide the generation module G to produce high-quality results. D contains a feature discriminator D_f and an image discriminator D_i. D_f aims to distinguish the features $f^{p'}$ from f^g, which enforces G_{fm} to reduce the discrepancy between $f^{p'}$ and f^g. The image discriminator distinguishes synthesized image x^s from ground true image x^g, which encourages the generation module to produce photo-realistic results. Similar to the conventional GAN, our G and D improve each other by competing against each other. D tries to estimate the probability that a sample is produced by G. At the same time, G aims to produce high-quality samples that can confuse D. We use the loss functions introduced in subsection III-B to train our G, D_f, and D_i by turns. In the following subsections, we first introduce the architectures of our G and those of D. Then, we detail all the training loss functions.

A. ARCHITECTURE

1) GENERATION MODULE

The task of the generation module G is synthesizing a frontal-view, photo-realistic, and identity-preserving face image x^s from a profile face image x^p. In the following subsection, we introduce the components of G, including the encoder

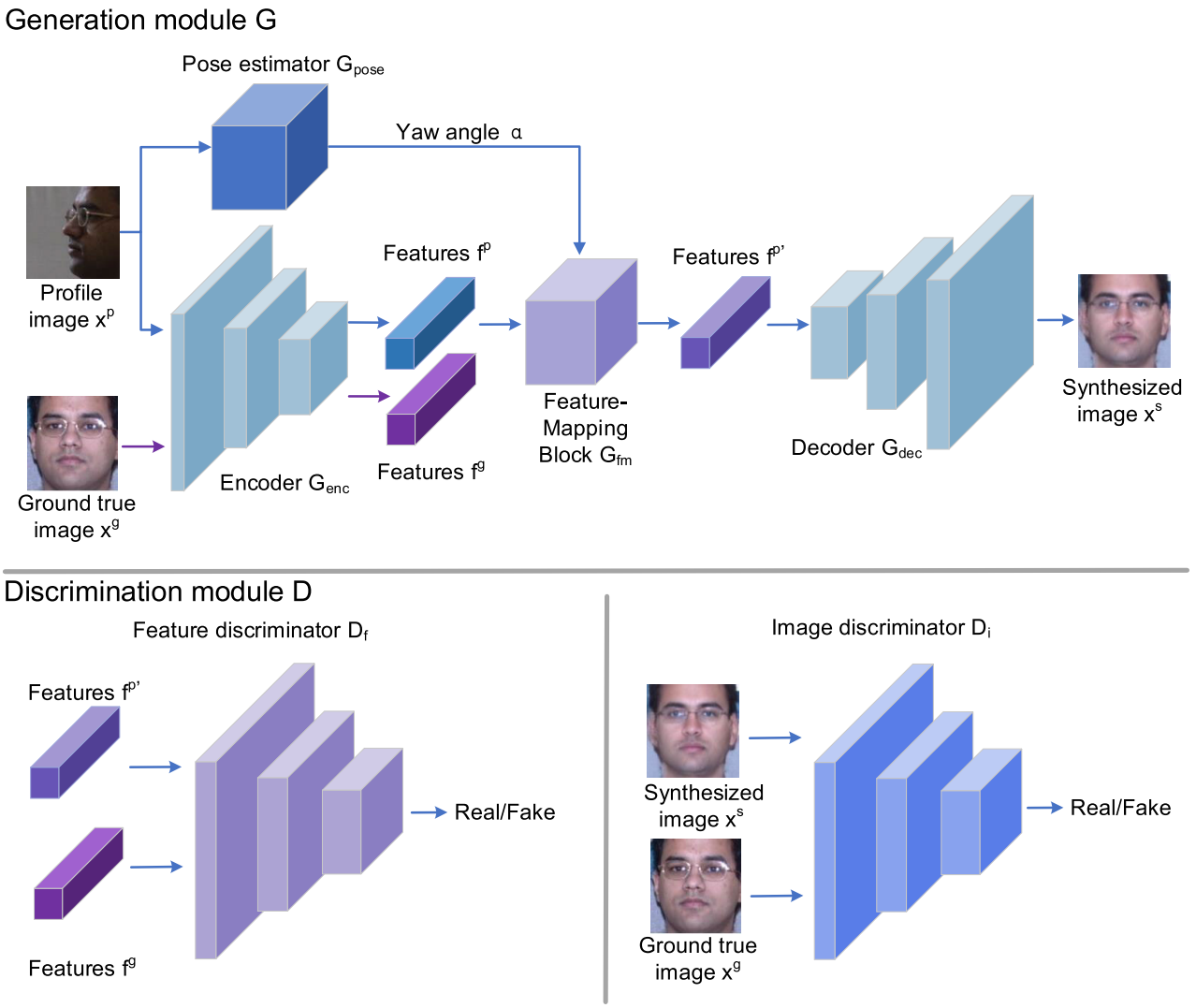


FIGURE 1. The overall framework of FI-GAN. Our FI-GAN mainly consists of the generation module and the discrimination module.

G_{enc}, the pose estimator G_{pose}, the Feature-Mapping Block G_{fm}, and the decoder G_{dec}.

G_{enc} aims to extract the features of input images. It is a typical convolutional neural network (CNN), the network structure of which is shown in Table 1. The size of the input images is fixed as $128 \times 128 \times 3$. Our G_{enc} outputs feature vectors with 256 dimensions. In our G_{enc}, each convolution layer is followed by one residual block [23] and activated by Rectified Linear Unit (ReLU) [24]. The fully connected layer fc1 is activated by maxout [25].

G_{pose} aims to calculate the yaw angle α of the profile face image x^p. The output α can be served as the prior knowledge, which helps to estimate the discrepancy between the profile faces and frontal faces. Ignoring the discrete camera label of the profile face images from the MultiPIE database, we employ the off-the-shelf method introduced in [26] to obtain the yaw angle. We fix the parameters of our pose estimator in our experiments. To simplify the process of training our generation module, we can precalculate the yaw angle of all images in the training set and flip the face images horizontally when the yaw angle is negative.

G_{fm} aims to map the features of profile faces to the frontal space. Since most face frontalization methods perform better under smaller poses, we consider that the performance under large poses can be improved by mapping the intermediate features to frontal space. Cao *et al.* propose the DREAM-Block [11] to estimate the discrepancy between features of profile faces and those of frontal faces, which helps to evaluate pose-robust features. Inspired by their work, we propose a similar module named Feature-Mapping Block (G_{fm}), whose working process is shown in Fig.2. The features of the profile face produced by the encoder G_{enc} are denoted as f^p. The yaw angle of the profile face estimated by the pose estimator is denoted as α. In order to map f^p to the frontal space, we design another two branches. The first additional branch contains a compact neural network FC, which has two fully-connected

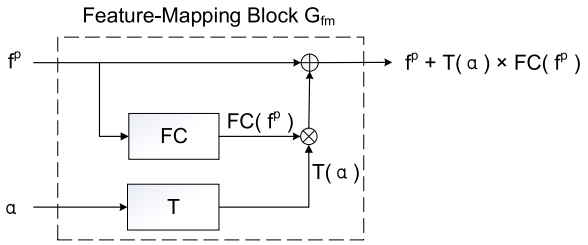


FIGURE 2. The working process of the Feature-Mapping Block. f^p and α refer to features of the profile face and pose degree of profile face, respectively. *FC* and *T* refer to a fully-connected network and a trigonometric function. FC aims to evaluate the residuals of f^p while T can provide the coefficient of the residuals. The output $f^p + T(\alpha) \times FC(f^p)$ refers to the features produced by our Feature-Mapping Block.

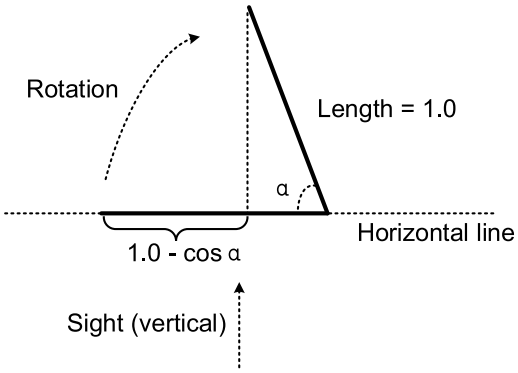


FIGURE 3. The rotation process of the line segment. The length of the line segment is set to 1.0. After the line segment rotates α degree, the length of its projection on the horizontal line decreases $1.0 - \cos(\alpha)$.

layers with Parametric Rectified Linear Unit (PReLU) [27] as the activation function. It aims to evaluate the residuals of the features f^p. The output of this branch, denoted as the $FC(f^p)$, is a 256-dims vector, the size of which is the same as f^p. The second additional branch contains a function T. The input α refers to the yaw angle of the face images. The output $T(\alpha)$ is defined as the coefficient of the residuals $FC(f^p)$ evaluated by the first branch. Cao *et al.* set the function T to $sigmoid(\alpha/45 - 1.0)$, which is a monotonous nonlinear function that maps the input to a positive value within the range of (0, 1). In this paper, we propose a more suitable function, which is defined as follows:

$$T(\alpha) = 1.0 - cos(\alpha) \tag{1}$$

Since the yaw angle of most face images ranges from 0° to 90°, function T can map it to a positive value within the range of [0, 1]. When α is equal to 0°, the coefficient $T(\alpha)$ is zero, which means our G_{fm} does not affect the features of frontal faces. When the pose degree increases, the coefficient $T(\alpha)$ also increases, which means our G_{fm} has more effects on features f^p. The reason why we adopt this trigonometric function is that it can describe the rotation roughly. In Fig.3, the length of the line segment is set to 1.0, then the length of its projection on the horizontal line is equal to 1.0. After the line segment rotates α degree, the length of its projection decreases $T(\alpha)$. As we all know, frontal faces contain most

TABLE 1. The network structure of the encoder G_{enc}.

Layer	Input	Filter/Stride	Output size
conv0	x^p	7 × 7/1	128 × 128 × 64
conv1	conv0	5 × 5/2	64 × 64 × 64
conv2	conv1	3 × 3/2	32 × 32 × 128
conv3	conv2	3 × 3/2	16 × 16 × 256
conv4	conv3	3 × 3/2	8 × 8 × 512
flatten,fc1	conv4	-	512
maxout	fc1	-	256

TABLE 2. The network structure of the decoder G_{dec}. The symbol $f^{p'}$ refers to features produced by G_{fm}.

Layer	Input	Filter/Stride	Output size
fc2, reshape	$f^{p'}$	-	8 × 8 × 64
dc11	fc2	4 × 4/4	32 × 32 × 32
dc12	dc11	2 × 2/2	64 × 64 × 16
dc13	dc12	2 × 2/2	128 × 128 × 8
dc21	fc2	2 × 2/2	16 × 16 × 512
dc22	dc21	2 × 2/2	32 × 32 × 256
dc23	dc22, dc11	2 × 2/2	64 × 64 × 128
dc24	dc23, dc12	2 × 2/2	128 × 128 × 64
conv5	dc24, dc13	5 × 5/1	128 × 128 × 64
conv6	conv5	3 × 3/1	128 × 128 × 32
conv7	conv6	3 × 3/1	128 × 128 × 3

identity information, which gradually loses with the faces rotate. The coefficient of the residuals $FC(f^p)$ is set to $T(\alpha)$.

Above all, one branch of G_{fm} produces the residuals $FC(f^p)$, while another branch evaluates the coefficient of $FC(f^p)$. The outputs of the above two branches are multiplied and added to the features of profile images. The features produced by G_{fm} are denoted as $f^{p'}$, which equal to $f^p + T(\alpha) \times FC(f^p)$. In subsection IV-D, we compare our Feature-Mapping Block with DREAM-Block on face frontalization and recognition tasks.

G_{dec} aims to recover frontal-view, photo-realistic, identity-preserving face images x^s from features produced by G_{fm}. The network structure of our G_{dec} is shown in Table 2. Note that the input features named $f^{p'}$ are the output of our G_{fm}. Our decoder consists of three parts. The first part is a simple deconvolution structure to upsample the features $f^{p'}$. The second part consists of deconvolution layers stacked for reconstruction, and each of them is followed by one residual block. The third part involves some convolution layers for recovering the frontal face images. In our G_{dec}, the final layer *conv7* is activated by the hyperbolic tangent function (tanh), while other layers are activated by ReLU.

2) DISCRIMINATION MODULE

Our discrimination module D is used for guiding G to produce high-quality results. D consists of a feature discriminator D_f and an image discriminator D_i.

D_f can guide the generation module to produce high-quality intermediate features. We denote the features of profile face mapped by Feature-Mapping Block G_{fm} and features of ground true frontal face produced by the encoder G_{enc} as $f^{p'}$ and f^g, respectively. D_f tries to distinguish $f^{p'}$ from f^g.

It competes with G which tries to produce high-quality $f^{p'}$ to confuse D_f. With D_f being more powerful, G is enforced to minimize the discrepancy between $f^{p'}$ and f^g. By guiding G_{fm} to map $f^{p'}$ to frontal space, D_f can improve the performance of our method. D_f has two fully-connected layers with Parametric Rectified Linear Unit (PReLU) [27] as the activation function.

D_i can guide the generation module G to produce photo-realistic face images. It tries to distinguish the synthesized frontal face image x^s from the ground truth frontal face image x^g. Note that in our work, all frontal face images in training set are regarded as real samples, whereas all synthesized images are considered as fake samples. Moreover, we limit real faces to frontal views only, which encourages the generation module to produce frontal face images. By competing with each other, the generation module G and image discriminator D_i are both improved. With D_i being more powerful, G is enforced to produce more photo-realistic face images to confuse G. With G being more powerful, D_i is enforced to improve its ability to distinguish generated face images from real face images. In the early stages, when synthesized faces may be profile, D_i makes the real or fake decision based on pose, which guides G to synthesize frontal face images. In the later stages, when most of the synthesized faces are frontal, D_i focuses on subtle details, which helps G to produce more photo-realistic face images. The network structure of D_i is shown in Table 3. Each layer in our D_i is activated by ReLU.

TABLE 3. The network structure of the image discriminator.

Layer	Input	Filter/Stride	Output size
conv0	x^s or x^g	$4 \times 4/2$	$64 \times 64 \times 64$
conv1	conv0	$4 \times 4/2$	$32 \times 32 \times 128$
conv2	conv1	$4 \times 4/2$	$16 \times 16 \times 256$
conv3	conv2	$4 \times 4/2$	$8 \times 8 \times 512$
conv4	conv3	$4 \times 4/2$	$4 \times 4 \times 1024$
conv5	conv4	$4 \times 4/1$	$1 \times 1 \times 1$

In summary, D and G improve each other by competing with each other. The training details are described in subsection IV-A.

B. TRAINING LOSS FUNCTIONS

In this subsection, we introduce the training losses, including the feature loss $L_{feature}$, the pixel-wise loss L_{pixel}, the symmetry loss L_{sym}, the identity-preserving loss L_{ip}, and the adversarial losses L_{adv1} and L_{adv2}.

We employ the feature loss $L_{feature}$ to improve the features produced by Feature-Mapping Block,

$$L_{feature} = \|G_{fm}(G_{enc}(x^p), \alpha) - f^g\|^2 \quad (2)$$

where x^p and α represent the profile face image and its yaw angle, respectively. $G_{fm}(G_{enc}(x^p), \alpha)$ refers to features of x^p and is evaluated by our G_{fm}. f^g refers to the features of corresponding ground true frontal face image and evaluated by our encoder. $L_{feature}$ aims to minimize the Euclidean distance between features $G_{fm}(G_{enc}(x^p), \alpha)$ and f^g.

We adopt the pixel-wise loss,

$$L_{pixel} = \frac{1}{W \times H \times C} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{C} |G(x^p)_{i,j,k} - x^g_{i,j,k}| \quad (3)$$

where W and H represent the width and height of image, respectively. C is the number of image channels. G and x^p refer to our generation module and profile image, respectively. $G(x^p)$ and x^g refer to synthesized image and ground truth image. L_{pixel} aims to reconstruct the ground truth with minimal error.

Based on the prior knowledge that most human faces are bilateral symmetry, we impose a symmetry loss function to minimize the Manhattan distince between synthesized image and its symmetry image.

$$L_{sym} = \frac{1}{W/2 \times H \times C} \sum_{i=1}^{W/2} \sum_{j=1}^{H} \sum_{k=1}^{C} |G(x^p)_{i,j,k} - G(x^p)_{W-i,j,k}| \quad (4)$$

where W and H represent the width and height of image, respectively. C is the number of image channels. Imposing the symmetric constraint on the synthesized images contributes to alleviating the self-occlusion problems. So we can improve performance under large poses by using the symmetry loss.

For recognition via generation framework, preserving identity is an essential task. We adopt the idea from perceptual loss [28] to design the identity-preserving loss based on the activations of the last two layers of a pre-trained face expert network F:

$$L_{ip} = \sum_{n=1}^{2} \frac{1}{W_n \times H_n \times C_n} \sum_{i=1}^{W_n} \sum_{j=1}^{H_n} \sum_{k=1}^{C_n} |F(G(x^p))^n_{i,j,k} - F(x^g)^n_{i,j,k}| \quad (5)$$

where W_n and H_n denote the width and height of the feature maps in the last nth layer, respectively. C_n is the number of channels in the last nth layer. The $G(x^p)$ and x^g denote synthesized image and ground true image, respectively. The identity-preserving loss enforces the synthesized image to have a small distance with the ground truth face image in the deep feature space. We employ the Light CNN-29 [12] as our face expert network.

We propose two adversarial loss functions named L_{adv1} and L_{adv2}, respectively. L_{adv1} is defined as follows:

$$L_{adv1} = logD_f(f^g) + log(1.0 - D_f(G_{fm}(G_{enc}(x^p)))) \quad (6)$$

where the f^g and $G_{fm}(G_{en}(x^p))$ refer to features of ground true face and those of profile face, respectively. When training the generation module G, we minimize this loss function to guide our G to produce high-quality intermediate features to confuse D_f. When training D_f, we maximize this loss function to improve the ability to distinguish features of profile face from those of ground true face. The L_{adv2} is formulated as follows:

$$L_{adv2} = logD_i(x^g) + log(1.0 - D_i(G(x^p)) \quad (7)$$

where the x^g and $G(x^p)$ refer to ground true image and synthesized image, respectively. When training the generation module G, we minimize this loss function to guide our G to produce a visually pleasing synthesized image to confuse D_i. When training D_i, we maximize this loss function to improve the ability to distinguish synthesized image from ground true image.

The total loss is a weighted sum of the above losses. The generation module G, feature discriminator D_f and image discriminator D_i are trained by turns to optimize the following min-max problem:

$$\max_{\theta_{D_f},\theta_{D_i}} \min_{\theta_G} L_{total} = L_{pixel} + \lambda_1 L_{sym} + \lambda_2 L_{ip} + \lambda_3 L_{adv1} + \lambda_4 L_{adv2} + \lambda_5 L_{feature} \quad (8)$$

where θ_{D_f}, θ_{D_i}, and θ_G refer to parameters of D_f, D_i, and G, respectively. The λ_1, λ_2, λ_3, λ_4, λ_5 are the weights of each loss function. Inspired by similar work [6], we empirically set $\lambda_1 = 0.2$, $\lambda_2 = 0.003$, $\lambda_3 = 0.001$, $\lambda_4 = 0.001$, and $\lambda_5 = 0.00005$.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The FI-GAN aims to recover photo-realistic and identity-preserving frontal view face images from the input face images. In this section, we evaluate the FI-GAN from two aspects, including visual quality and identity-preserving ability. For the former, the face images synthesized by FI-GAN are shown. For the latter, we calculate Rank-1 recognition rates on synthesized images. The databases we used for evaluating are the MultiPIE database [10], the Labeled Faces in theWild (LFW) database [29], and the Celebrities in Frontal-Profile (CFP) database [30]. In subsection IV-A, we introduce databases and implementation details. In subsection IV-B, we compare visualized results produced by the FI-GAN with those produced by state-of-the-art methods. In subsection IV-C, we quantitatively evaluate face recognition performance on our frontalized images, compared to other frontalization methods. In subsection IV-D, we compare FI-GAN with its variants, which proves the effectiveness of our proposed modules and that of our loss functions.

A. DATABASES AND IMPLEMENTATION DETAILS

1) DATABASES AND TESTING PROTOCOLS

The MultiPIE database is one of the largest databases for evaluating face frontalization and pose-invariant face recognition in the controlled setting. It consists of about 750,000 images from 337 subjects under pose, illumination and expression changes. Inspired by testing protocols in [6], we utilize two settings to evaluate the methods. In Setting1 [16], we use the face images in Session1, which contains faces of 250 subjects. We only consider the face images with neutral expression under 13 poses and 20 illuminations. Therefore, 65000 face images are chosen (250 * 13 * 20 = 65000). The face images of the first 150 subjects are used for training whereas the rest are used for testing. In Setting2 [18], we use the face images in all four sessions, which contain faces of 337 subjects. We include images with neutral expression under 20 illuminations and 13 poses within $\pm$ 90°. The first 200 subjects are used for training, and the rest are used for testing. Note that, in both settings, there are not overlap subjects between the training and testing sets. For each testing subject, one frontal face image with normal illumination is added to the gallery set whereas the rest images are added to the probe set. In the testing process, firstly, we apply FI-GAN for frontalizing the face images with arbitrary poses in the testing set. Then we use a face recognition network named Light CNN-29 [12] to extract the identity features of synthesized face images. Finally, we calculate the rank-1 recognition rates.

The LFW database is one of the most popular databases for face verification in the uncontrolled setting. It consists of 13,233 images of 5,749 subjects. Since the face images are collected from the Internet and under various expression, pose, illumination changes, conducting face frontalization experiments on the LFW database is challenging work. In the verification protocol [29], the face images on the LFW database are divided into ten parts, each with 300 same-person pairs and 300 different-person pairs. For each pair, we need to judge whether two face images come from the same person. In this experiment, firstly, we trained our FI-GAN on the MultiPIE database following Setting2. Then, we frontalize the face images and evaluate face verification performance on the LFW database.

The CFP database is another database for face verification in the uncontrolled setting. It consists of 7,000 images of 500 subjects, where each subject has ten frontal and four profile face images. The verification protocol [30] includes frontal-frontal (FF) and frontal-profile (FP) face verification, each having 3500 same-person pairs and 3500 different-person pairs. In this experiment, firstly, we trained our FI-GAN on the MultiPIE database following Setting2. Then, we frontalize the face images and evaluate face verification performance on the CFP database.

2) IMPLEMENTATION DETAILS

To pre-process the face images, we apply the algorithm introduced in [31] for face alignment. The size of all face images is fixed as $128 \times 128 \times 3$. Before training, we apply our pose estimator to precalculate the yaw angle α of face images in the training set. The FI-GAN is implemented with Tensorflow [32]. We train the FI-GAN empirically, with the batch size set to 12 and the learning rate set to 0.0008. All training weights are initialized from a zero-centered normal distribution with a standard deviation of 0.02. We use the loss function L_{total} which is defined in (8) to train the generation module G, the feature discriminator D_f, and the image discriminator D_i by turns, *e.g.*, one step for optimizing G, one for D_i and one for D_f. When the value of each loss function remains relatively stable, we can stop the training. Our encoder G_{enc}, decoder G_{dec}, and D_f, which contain conventional layers, have about 0.3, 0.8, 0.05 billion FLOPs, respectively. The graphic processing unit of the computer

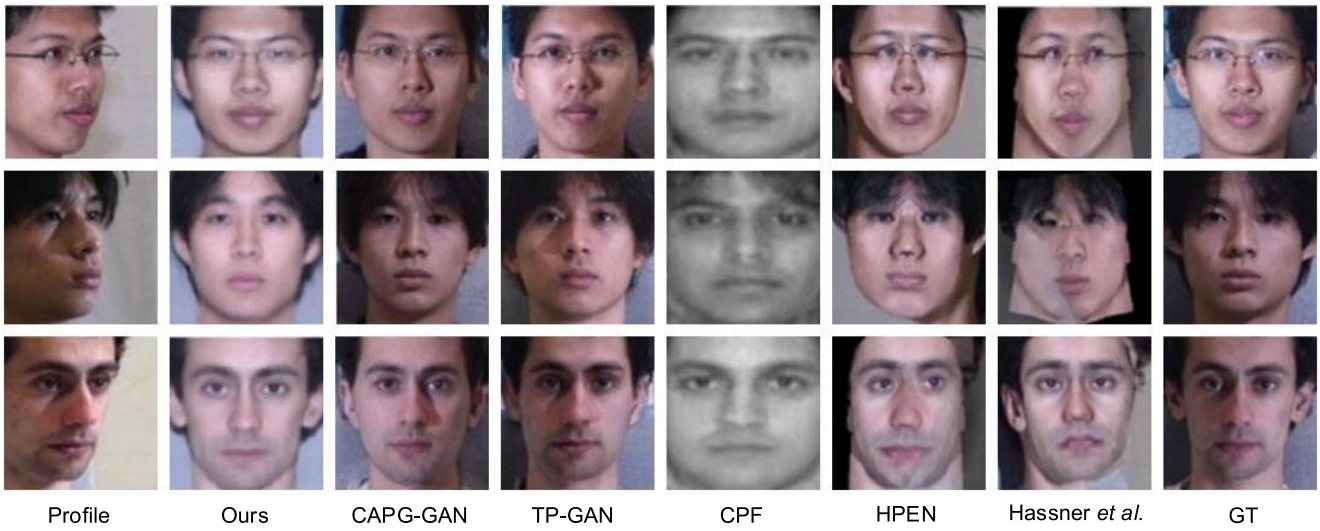


FIGURE 4. Comparison with state-of-the-art synthesis methods on the MultiPIE database under the pose of 45° (first two rows) and 30° (last row).

used for training is NVIDIA GeForce GTX 1080 TI. The training of our method lasts for about half-day.

B. VISUAL QUALITY

In this subsection, we compare the frontalized face images of the FI-GAN with those of other methods. Face images in the training set of Setting2 from the MultiPIE database are used for training.

For the experiment in the controlled setting, we frontalize the face images in the test set of Setting2 from the MultiPIE database. Since most of the face frontalization methods can only deal with the face images under small poses, we firstly display the frontalized images of different methods, including our FI-GAN, CAPG-GAN [5], TP-GAN [6], CPF [18], HPEN [4], and Hassner *et al.* [2], under the pose of 30° and 45° in Fig. 4. The synthesized face images of our methods look better than those of many techniques in terms of both global structure and local texture. In the last two years, some GAN-based methods, which can address face frontalization problem under large poses, have been proposed. We compare the synthesized images of the FI-GAN with those of state-of-the-art GAN-based methods, including CAPG-GAN [5], and TP-GAN [6], under the pose of 75° and 90° in Fig. 5. We observe that all methods perform well under large poses. The face images produced by our FI-GAN are photo-realistic and comparable to those of state-of-the-art models.

To further prove the effectiveness of our FI-GAN in the uncontrolled setting, we show synthesized images of compared methods, including PIM [9], TP-GAN [6], and Hassner *et al.* [2], on the LFW database in Fig. 6. Note that all GAN-based methods are only trained on the MultiPIE database; Obtaining good visual results on the LFW database is challenging since the LFW contains other variations, such as low resolution and occlusion. As is shown in Fig. 6, the synthesized images of [2] deviate from original

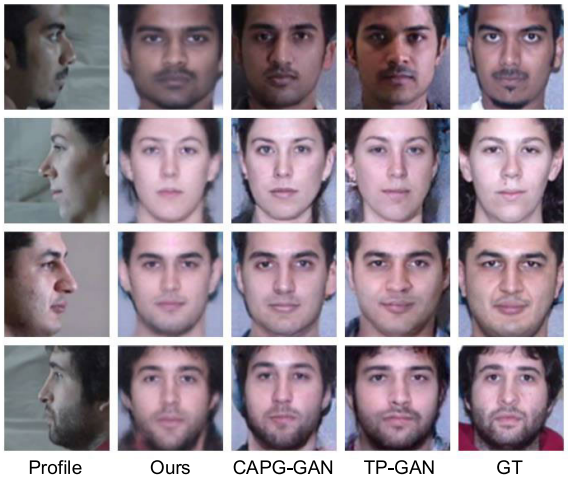


FIGURE 5. Comparison with state-of-the-art synthesis methods on the MultiPIE database under the pose of 75° (first two rows) and 90° (last two rows).

appearance seriously. The recovered face images of TP-GAN [6] are severely blurry. Comparatively, our FI-GAN obtains relatively well visual results. Not only global face shapes but also local details are well recovered.

Fig. 7 shows our synthesized images on the CFP database in challenging cases. Our FI-GAN can preserve observed face attributes in the original input image in most cases, e.g., eyeglasses, expression, and black skin (there are few blacks on our training database). More importantly, the synthesized face images of our FI-GAN are photo-realistic.

C. IDENTITY-PRESERVING PROPERTY

In order to quantitatively evaluate the identity-preserving ability of different methods, we conduct face recognition on the MultiPIE database and face verification on the LFW

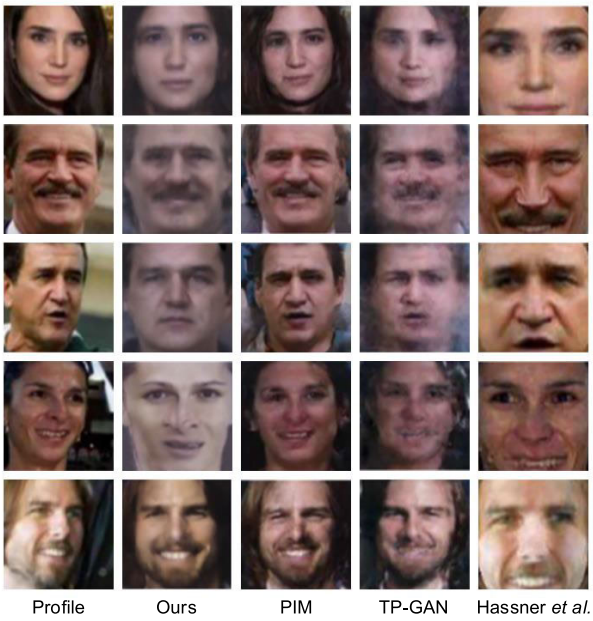


FIGURE 6. Comparison with state-of-the-art synthesis methods on the LFW database.

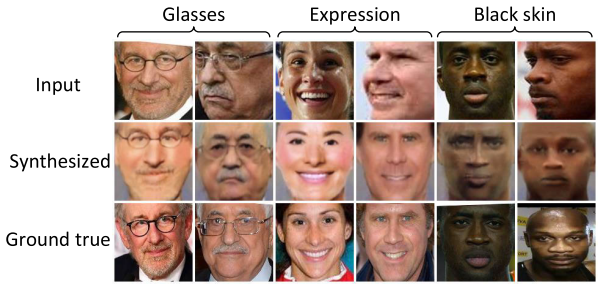


FIGURE 7. The synthesized results of our FI-GAN on the CFP database in challenging cases. Here, facial attributes include eyeglasses, expression, and black skin (there are few blacks in the training database). Moreover, the input faces in odd columns are frontal, whereas those in even columns are profile.

TABLE 4. Benchmark comparison of identification rate (%) across poses on the MultiPIE database under Setting1.

Method	90°	75°	60°	45°	30°	15°
Hassner *et al.* [2]	-	-	44.81	74.68	89.59	96.78
HPN [1]	29.82	47.57	61.24	72.77	78.26	86.23
c-CNN Forest [16]	47.26	60.66	74.38	89.02	94.05	96.97
TP-GAN [6]	64.03	84.10	92.93	**98.58**	**99.85**	99.78
CAPG-GAN [5]	77.10	87.40	93.74	98.28	99.37	**99.95**
Light CNN-29 [12]	9.00	32.35	73.30	97.45	98.80	99.78
Ours	**81.17**	**90.24**	**96.68**	98.12	99.49	99.60

database. In both experiments, we apply our face expert network named Light CNN-29 [12] to extract the identity features of our frontalized face images. Then we evaluate the face recognition performance with a cosine distance metric.

Table 4 shows the rank-1 accuracy rates of different methods under Setting1 [16] of MultiPIE. The results of Light CNN-29 [12] serve as our baseline. We compare FI-GAN with Hassner *et al.* [2], HPN [1], c-CNN Forest [16], TP-GAN [6], and CAPG-GAN [5]. The performance of all methods gets worse when the pose degree increases because more facial information gets lost. However, the rank-1 accuracy rates of our approach decrease slower than those of competitors. We think the reason is that our Feature-Mapping Block maps the features of the profile face images to the frontal space, so our FI-GAN performs relatively well under large poses. Besides, when the yaw angle of the face images is small, the coefficient of residual produced by Feature-Mapping Block is relatively small. Then our Feature-Mapping Block has limited effects on immediate features, so the synthesized results of our FI-GAN are similar to other GAN-based methods under small poses.

TABLE 5. Benchmark comparison of identification rate (%) across poses on the MultiPIE database under Setting2.

Method	90°	75°	60°	45°	30°	15°
CPF [18]	-	-	61.9	79.9	88.5	95.0
DR-GAN [7]	-	-	83.2	86.2	90.1	94.0
FF-GAN [8]	61.2	77.2	85.2	89.7	92.5	94.6
TP-GAN [6]	64.64	77.43	87.72	95.38	98.06	98.68
CAPG-GAN [5]	66.05	83.05	90.63	97.33	**99.56**	**99.82**
Light CNN-29 [12]	5.51	24.18	62.09	92.13	97.38	98.59
Ours	**77.03**	**88.24**	**96.22**	**97.37**	98.52	98.80

TABLE 6. Face verification accuracy (ACC) and area-under-curve (AUC) results on the LFW database.

Method	ACC(%)	AUC(%)
Hassner *et al.* [2]	93.6	88.4
HPEN [4]	96.3	99.4
FF-GAN [8]	96.4	99.5
DR-GAN [7]	96.9	**99.6**
Ours	**98.3**	**99.6**

Table 5 shows the rank-1 accuracy rates of different methods under Setting2 [18] of MultiPIE. Compared with Setting1, Setting2 is more difficult because the number of test subjects increases. Once more, the results of the Light CNN-29 [12] serve as our baseline. We compare our FI-GAN with some state-of-the-art methods, including DR-GAN [7], FF-GAN [8], TP-GAN [6], CAPG-GAN [5]. As is shown in Table 5, our method greatly improves the performance of Light CNN-29, especially under large poses. Moreover, our method outperforms other frontalization methods under most of the poses, which proves that our synthesized images are identity-preserving.

Table 6 shows the face verification accuracy and area-under-curve results on the LFW database. We compare our FI-GAN with some state-of-the-art methods, including Hassner *et al.* [2], HPEN [4], FF-GAN [8], and DR-GAN [7]. Although FI-GAN is not trained on the LFW database, it achieves high verification rates. Since most of the testing face images on this database have small poses, our proposed Feature-Mapping Block has limited effects on intermediate features. The results of our methods are not very different from those of others.

Table 7 shows the face verification accuracy and area-under-curve results on the CFP database. We compare

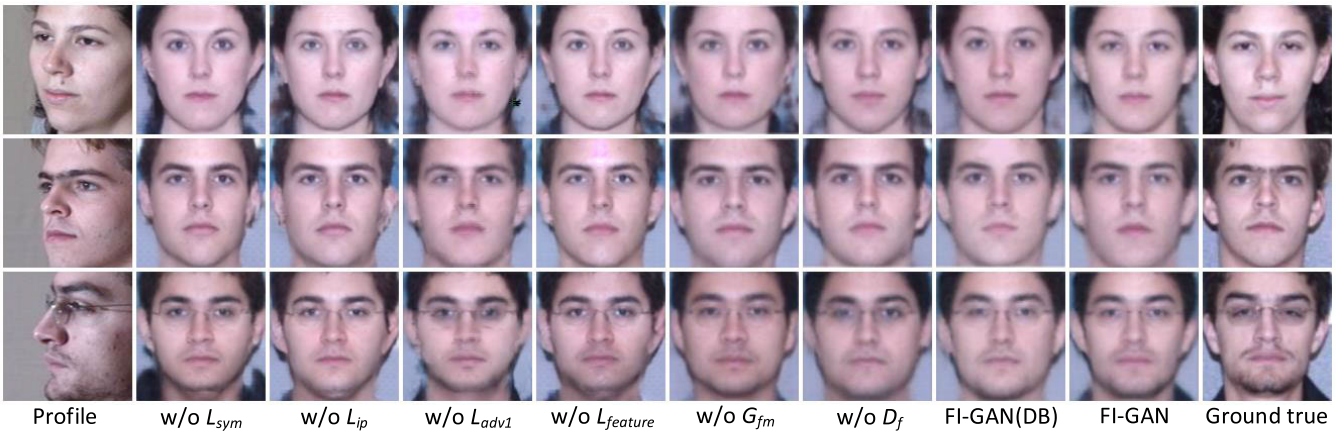


FIGURE 8. Synthesized images of FI-GAN and its variants on the MultiPIE database under the pose of 30° (first row), 60° (second row), and 90° (third row).

TABLE 7. Face verification accuracy (ACC) and area-under-curve (AUC) results on the CFP database.

Method	Frontal-Frontal		Frontal-Profile	
	ACC	AUC	ACC	AUC
Sengupta *et al.* [30]	96.4	99.4	84.9	93.0
DR-GAN [7]	97.1	-	91.0	-
PIM [9]	**99.4**	**99.9**	93.1	97.7
Ours	98.9	99.7	**94.2**	**98.9**

TABLE 8. Model comparison: Rank-1 recognition rates (%) of our FI-GAN and its variants on the MultiPIE database.

Method	90°	75°	60°	45°	30°	15°
w/o L_{sym}	71.63	84.83	94.29	96.78	97.20	98.43
w/o L_{ip}	49.88	57.34	74.92	86.20	91.75	96.36
w/o L_{adv1}	70.62	84.21	94.79	95.64	97.29	98.19
w/o $L_{feature}$	65.57	78.28	91.71	94.62	96.87	98.52
w/o G_{fm}	62.33	75.92	88.24	93.74	96.54	98.69
w/o D_f	71.34	84.72	93.94	95.41	98.33	98.60
FI-GAN (DB)	75.39	86.32	95.34	97.23	**98.76**	98.42
FI-GAN	**77.03**	**88.24**	**96.22**	**97.37**	98.52	**98.80**

our FI-GAN with some state-of-the-art methods, including Sengupta *et al.* [30], DR-GAN [7], and PIM [9]. Our FI-GAN achieves comparable performance as its competitors under the frontal-frontal setting. Moreover, our FI-GAN consistently outperforms other state-of-the-art under more challenging frontal-profile setting, which proves that our Feature-Mapping Block has more effects on synthesized results under large poses. The experimental results on this database suggest the identity-preserving ability of the FI-GAN in the uncontrolled environment.

D. ABLATION STUDY

In this subsection, we go over different architectures and loss function combinations to gain insight into their respective roles in face frontalization. We train seven partial variants of our FI-GAN, e.g., w/o L_{sym} (L_{sym} is removed), w/o L_{ip}, w/o L_{adv1}, w/o $L_{feature}$, w/o G_{fm}, w/o D_f (D_f or L_{adv2} is removed), FI-GAN (DB) (G_{fm} is replaced by DREAM-Block [11]). We report both qualitative visualization and quantitative recognition results.

Fig. 8 illustrates the perceptual performance of these variants. The synthesized images of our FI-GAN look more like ground true images than those of partial variants, especially under large poses. As expected, inference results without L_{ip}, $L_{feature}$, or G_{fm} deviate from the true appearance seriously, and those without L_{adv1} tend to be blurry. The synthesized results of other partial variants look similar to those of our FI-GAN.

The rank-1 recognition rates on MultiPIE under Setting2 are reported in Table 8. We observe that our FI-GAN performs better than its variants under most poses. In particular, the accuracy decreases severely if L_{ip}, $L_{feature}$, or G_{fm} is removed, especially under large poses. Although not as much apparent, L_{sym}, L_{adv1}, and D_f help to improve recognition performance. Moreover, when we replace DREAM-Block with our Feature-Mapping Block, the accuracy increases sightly, which proves that Feature-Mapping Block is more suitable for our FI-GAN than DREAM-Block.

Both visualization and recognition results prove that each proposed component or loss function is essential for FI-GAN during synthesis.

V. CONCLUSION

In this paper, we propose the Feature-Improving GAN (FI-GAN) for face frontalization. FI-GAN uses the Feature-Mapping Block for mapping the features of the profile face images to the frontal space, which improves the face recognition rates under large poses. To improve the features of profile face images further, we propose a feature discriminator which enforces the features of the profile faces to have a small distance with those of ground true frontal faces. Experimental results demonstrate that our approach can synthesize photo-realistic and identity-preserving results, which are comparable to those of the state-of-the-art. At present, our FI-GAN only concentrates on the variation of yaw angle. In the future, we will investigate how to deal with face images with large pitch angle.

REFERENCES

[1] C. Ding and D. Tao, "Pose-invariant face recognition with homography-based normalization," *Pattern Recognit.*, vol. 66, pp. 144–152, Jun. 2017.

[2] 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.

[3] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1701–1708.

[4] X. Zhu, Z. Lei, J. Yan, D. Yi, and S. Z. Li, "High-fidelity pose and expression normalization for face recognition in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 787–796.

[5] 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.

[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," 2017, *arXiv:1704.04086*. [Online]. Available: http://arxiv.org/abs/1704.04086

[7] 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. 1415–1424.

[8] 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. 1–10.

[9] 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.

[10] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," *Image Vis. Comput.*, vol. 28, no. 5, pp. 807–813, May 2010.

[11] K. Cao, Y. Rong, C. Li, X. Tang, and C. C. Loy, "Pose-robust face recognition via deep residual equivariant mapping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5187–5196.

[12] 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.

[13] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, no. 9, pp. 207–244, 2009.

[14] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.

[15] F. Wang, L. Chen, C. Li, S. Huang, Y. Chen, C. Qian, and C. Change Loy, "The devil of face recognition is in the noise," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 765–780.

[16] C. Xiong, X. Zhao, D. Tang, K. Jayashree, S. Yan, and T.-K. Kim, "Conditional convolutional neural network for modality-aware face recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3667–3675.

[17] 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.

[18] 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.

[19] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Multi-view perceptron: A deep model for learning face identity and view representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 217–225.

[20] 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.

[21] J. Yang, S. E. Reed, M.-H. Yang, and H. Lee, "Weakly-supervised disentangling with recurrent transformations for 3D view synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1099–1107.

[22] Y. Zhang, M. Shao, E. K. Wong, and Y. Fu, "Random faces guided sparse Many-to-One encoder for pose-invariant face recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2416–2423.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[24] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[25] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," 2013, *arXiv:1302.4389*. [Online]. Available: http://arxiv.org/abs/1302.4389

[26] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4511–4520.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

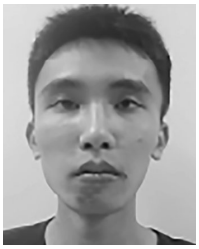
[28] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 694–711.

[29] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Proc. Workshop Faces Real-Life'Images, Detection, Alignment, Recognit.*, 2008.

[30] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs, "Frontal to profile face verification in the wild," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–9.

[31] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial Landmarks)," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1021–1030.

[32] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. OSDI*, vol. 16, 2016, pp. 265–283.



CHANGLE RONG received the B.Sc. degree in computer science from the South China University of Technology, Guangzhou, China, in 2017, where he is currently pursuing the M.Sc. degree.

His research interests include image processing and information processing.

XINGMING ZHANG received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 1996.

He is currently a Professor and Ph.D. Supervisor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He is a member of the Standing Committee of the Technical Committee of Education, China Computer Federation, and an Executive Director of the Computer Federation of Guangdong Province. His research focuses on image processing, video coding, and surveillance.

YUBEI LIN received the M.Eng. degree in computer software and theory from the Sun Yat-sen University and the Ph.D. degree in computer application technology from the South China University of Technology, Guangzhou, China, in 2005 and 2016, respectively. She joined The Hong Kong Polytechnic University as a Research Assistant, in 2004. She has been an Engineer with the Computing Center of South China University of Technology, since 2005. Her research interests include video coding, image processing, and information processing.