PosIX-GAN: Generating Multiple Poses Using GAN for Pose-Invariant Face Recognition

Avishek Bhattacharjee(✉), Samik Banerjee, and Sukhendu Das

Visualization and Perception Lab, Department of Computer Science and Engineering,
IIT Madras, Chennai, India
{avi,samik}@cse.iitm.ac.in, sdas@iitm.ac.in

Abstract. Pose-Invariant Face Recognition (PIFR) has been a serious challenge in the general field of face recognition (FR). The performance of face recognition algorithms deteriorate due to various degradations such as pose, illuminaton, occlusions, blur, noise, aliasing, etc. In this paper, we deal with the problem of 3D pose variation of a face. for that we design and propose PosIX Generative Adversarial Network (PosIX-GAN) that has been trained to generate a set of nice (high quality) face images with 9 different pose variations, when provided with a face image in any arbitrary pose as input. The discriminator of the GAN has also been trained to perform the task of face recognition along with the job of discriminating between real and generated (fake) images. Results when evaluated using two benchmark datasets, reveal the superior performance of PosIX-GAN over state-of-the-art shallow as well as deep learning methods.

Keywords: Face recognition · Pose · GAN · Multi-task Learning

1 Introduction

Deep learning (DL) has attracted several researchers in the field of computer vision due to its ability to perform face and object recognition tasks with high accuracy than the traditional shallow learning systems. The convolutional layers present in the deep learning systems help to successfully capture the distinctive features of the face [19,30]. For biometric authentication, face recognition (FR) has been preferred due to its passive nature. Most solutions for FR fail to perform well in cases involving extreme pose variations as in such scenarios, the convolutional layers of the deep models are unable to find discriminative parts of the face for extracting information.

Most of the architectures proposed earlier deal with the scenarios where the face images used for training as well as testing the deep learning models [3,15,25] are frontal and near-frontal. Further, the recent use of convolutional neural network (CNN) based models [6,7,15,19,23,25,29], which provide very

L. Leal-Taixé and S. Roth (Eds.): ECCV 2018 Workshops, LNCS 11131, pp. 427–443, 2019.
https://doi.org/10.1007/978-3-030-11015-4_31

high accuracies for FR applications even in the wild scenarios, fail to provide acceptable recognition rates in scenarios with pose variations in faces. These models fail to perform well when the face images provided during testing are at extreme poses due to the inability of the models to find discriminative features in the images provided. On the contrary, our model uses a limited number of face images at different poses to train a GAN model (PosIX-GAN), where nine separate generator models learn to map a single face image at any arbitrary pose to nine specific poses and the discriminator performs the task of face recognition along with discriminating a synthetic face from a real-world sample. In the following, we present brief review of work done on face recognition using CNNs, generative adversarial networks (GANs) as well as shallow methods for head pose estimation and face recognition (FR).

The method proposed by [37] learns a new face representation: the face identity-preserving (FIP) features. Unlike conventional face descriptors, the FIP features can significantly reduce intra-identity variances, while maintaining discriminativeness between identities. The work by Zhu *et al.* [38] proposes a novel deep neural net, named multi-view perceptron (MVP), which can untangle the identity and view features, and in the meanwhile infer a full spectrum of multi-view images, given a single 2D face image. Kan *et al.* [14] proposed a multi-view deep network (MvDN), which seeks for a non-linear discriminant and view-invariant representation shared between multiple views. The method proposed by Yin *et al.* [35] study face recognition as a multi-task problem where identity classification is the main task with pose, illumination and expression estimations being the side tasks. The goal is to leverage the side tasks to improve the performance of face recognition. Yim *et al.* [34] proposes a new deep architecture based on a novel type of multitask learning, which achieves superior performance by rotating a face from an arbitrary pose and illumination image to a target-pose face image (target pose controlled by the user) while preserving identity. The method proposed by Wu *et al.* [32] studies a Light CNN framework to learn a deep face representation from the large-scale data with massive noisy labels The method makes use of a Max-Feature-Map (MFM) operation to obtain a compact representation and perform feature filter selection. The method proposed by Tran *et al.* [30] utilizes an encoder-decoder structured generator that can frontalize or rotate a face with an arbitrary pose, even upto the extreme profile. It explicitly disentangles the representation learning from the pose variation through the pose code in generator and the pose estimation in discriminator. It also adaptively fuses multiple faces to a single representation based on the learnt coefficients. The TP-GAN method proposed by Huang *et al.* [13] performs photorealistic frontal view synthesis by simultaneously perceiving global structures and local details. It makes use of four landmark located patch networks to attend to local textures in addition to the commonly used global encoder-decoder network. The method proposed by Liu *et al.* [17] present a novel multi-task adversarial network based on an encoder-discriminator-generator architecture where the encoder extracts a disentangled feature representation for the factors of interest and the discriminators classify each of the factors as individual tasks. Yang *et al.* [33] proposes a novel recurrent convolutional encoder-decoder network that is trained end-to-end on the task of rendering rotated objects starting from a single image.

The method proposed by Gourier *et al.* [10] addresses the problem of estimating head pose over a wide range of angles from low-resolution images. It uses grey-level normalized face images for linear auto-associative memory where one memory is computed for each pose using a Widrow-Hoff learning rule. Huang *et al.* [12] use Gabor feature based random forests as the classification technique since they naturally handle such multi-class classification problem and are accurate and fast. The two sources of randomness, random inputs and random features, make random forests robust and able to deal with large feature spaces. The method proposed by Tu *et al.* [31] localizes the nose-tip of the faces and estimate head poses in studio quality pictures. After the nose-tip in the training data are manually labeled, the appearance variation caused by head pose changes is characterized by tensor model which is used for head pose estimation.

The works proposed in [27,29,39] mainly deal with multi-stage complex systems, which take the convolutional features obtained from their model and then use PCA (Principal Component Analysis) for dimensionality reduction, followed by classification using SVM. Zhu *et al.* [39] tries to "warp" faces into a canonical frontal view using a deep network, for efficient classification. PCA on the network output in conjunction with an ensemble of SVMs is used for the face verification task. Taigman *et al.* [29] propose a multi-stage approach that aligns faces to a general 3D shape model combining with a multi-class (deep) network which is trained to perform the FR task. The compact network proposed by Sun *et al.* [26–28] uses an ensemble of 25 of these networks, each operating on a different face patch. The FaceNet proposed by Schroff *et al.* [23] uses a deep CNN to directly optimize the embedding itself, based on the triplet loss formulated by a triplet mining method.

Deep Convolutional GAN [20] (DCGAN) first introduced as a convolutional architecture led to improved visual quality in Computer Vision (CV) applications. More recently, Energy Based GANs [36] (EBGANs) were proposed as a class of GANs that aim to model a discriminator $D(x)$ as an energy function. This variant converges in a more stable manner and is both easy to train and robust to hyper-parameter variations. Some of these benefits were attributed to the larger number of targets in the discriminator. EBGAN also implements its discriminator as an auto-encoder with a per-pixel error. While earlier variants of GAN lacked an analytical measure of convergence, Wasserstein GANs [1] (WGANs) recently introduced a loss function that acts as a measure of convergence. However, in their implementation, this comes at the expense of slow training, but with the benefits of stability and better mode of coverage [1]. The BEGAN model [4] utilizes a new equilibrium enforcing method paired with a loss derived from the Wasserstein distance for training auto-encoder based Generative Adversarial Networks. It also provides a new approximate convergence measure, fast and stable training and high visual quality.

Most of the methods of FR/FV discussed above do not show results on Head Pose Image [9] and MultiPIE [11] datasets which have high degree of pose variation in the query faces. Drawbacks of recent GAN based methods are blur, deformities as well as inaccuracy in the synthesis process, as well as instability during training. The contribution of our work on PosIX-GAN model includes synthesis of face images at various poses given an input face image at any arbitrary pose, without much of the aforementioned drawbacks. Apart from this, the proposed model simultaneously performs face recognition with high accuracy. Results are reported using 2 benchmark face datasets with pose variation.

In the rest of the paper, Sect. 2 gives an overview of Generative Adversarial Networks (GANs), Sect. 3 describes the proposed network architecture, along with details about the loss functions used for training. Section 4 provides information about the various datasets used for evaluation of our model. Section 5 reports quantitative as well as qualitative results obtained from experiments performed and observations. Finally, Sect. 6 concludes the paper.

2 Generative Adversarial Network (GAN)

Generative Adversarial Networks (GAN) [8] are based on the adversarial training of two CNN-based models: (i) a generative model (G), which captures the true data distribution, p_{data} and generates images sampled from a distribution p_z, the distribution of the training data provided as input; and (ii) a discriminator model (D), which discriminates between the original images, sampled from p_{data}, and the images generated by G. G maps p_z from a latent space to the data distribution p_{data} of interest, while D discriminates between instances from p_{data} and those generated by G. The adversarial training adopted for GAN, derived from Schmidhuber [22], involves the formulation of an optimization function G to maximize the error in D (*i.e.*, "fool" D by producing novel synthesized instances that appear to have come from p_{data}). Thus the adversarial training procedure followed for GAN, resembles a two player minimax gaming strategy between D and G of a zero-sum game [5] with the value function $V(G, D)$. The overall objective function minimized by GANs [8], is given as:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}}[\log D(x)] + \mathbb{E}_{x \sim p_z}[\log(1 - D(G(z)))] \tag{1}$$

To learn p_z over data x, a mapping to data space is represented as $G(z; \theta_g)$, where G is a differentiable function represented by a CNN with parameter set θ_g. Another CNN based deep network represented by $D(x; \theta_d)$ outputs a single scalar [0/1]. $D(x)$ represents the probability that x is generated from the true data rather than p_z.

The major drawback of such an adversarial system is that GANs fail to capture the categorical information, when all the pixels of the image samples obtained from two distributions, p_{data} and p_z are largely different from each other. We aim to overcome the two drawbacks specified above, in addition to the severe degradation in performance of the FR algorithms under severe pose variations, thus forming the underlying motivation of the work presented in this paper.

3 The Proposed Network

The proposed architecture of PosIX-GAN deals with generating faces at nine different poses from an input face (at any arbitrary pose), along with the task of pose invariant face recognition (PIFR) with the help of nine categorical discriminators which produce an output vector $\in \mathbb{R}^{N+1}$ for every image where N signifies the number of categories and 1 signifies whether the input to the network D_i is real or fake. For experimentation, we resized the images across all datasets to 64×64 pixels, to be provided as input to the generator module of PosIX-GAN. The overall architecture is detailed in Fig. 1, with the individual generator (G) and the discriminator (D) are illustrated in Figs. 2 and 3.

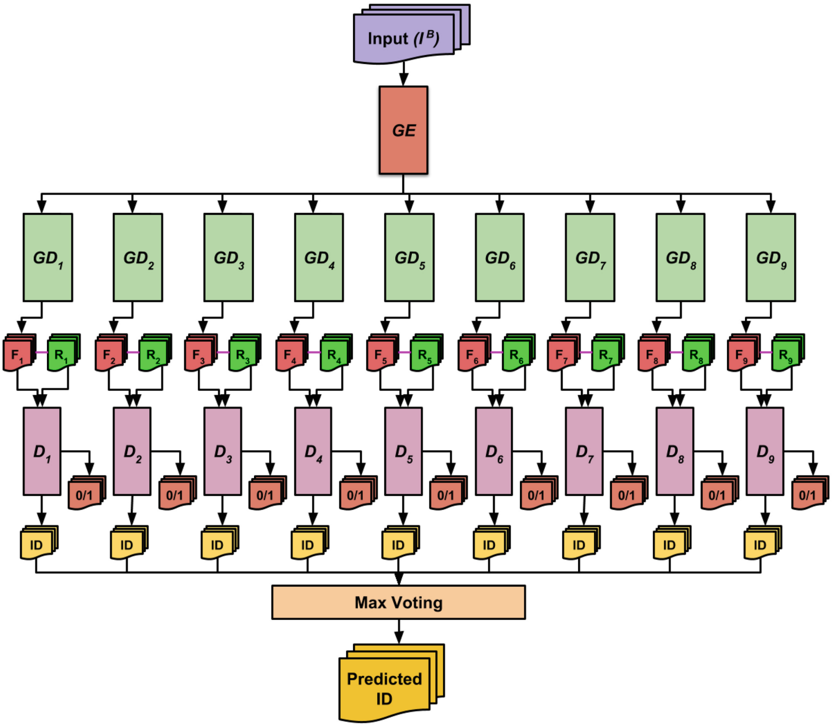

Fig. 1. The proposed architecture of PosIX-GAN, used in our work for PIFR. *GE* denotes the shared encoder of the generator network, GD_i; $i = \{1, 2, \ldots, 9\}$ denotes nine decoder networks connected to *GE*. F_i and R_i refers to the set of fake and original images. ID refers to the class IDs generated by the set of nine discriminator D_i; $i = \{1, 2, \ldots, 9\}$ which also generates 0/1 to indicate a real or fake image. (best viewed in color)

3.1 Architecture Details

Figure 1 shows an overview of the architecture of PosIX-GAN. The network consists of two parts, generator and discriminator. The generator itself has two components: a shared encoder network GE and nine decoder networks GD_i; $i = \{1, 2, \ldots, 9\}$ and G_i is defined as $(GE+GD_i)$. The components are described as follows:

The encoder is a deep-CNN based architecture, shown in Fig. 2(a), which takes input images with resolution of 64×64 pixels and outputs a vector $\in \mathbb{R}^{256}$. This encoder architecture has been adopted from that proposed in BEGAN model [4]. The encoder maps the input images to a latent space to produce an encoded vector, which acts as an input to each of the nine decoder networks.

The proposed PosIX-GAN model consists of nine decoder modules (Fig. 2(b)) which are attached to a single encoder network. The output from the encoder is fed as the input to each of the decoder networks. The decoder output F_i is then used along with a separate batch of real images R_i with distinct poses angles (different for every decoder network), while also preserving class information, to evaluate and minimize the patch-wise MSE loss described later in Algorithm 1. This helps the decoder module to learn to generate images at a specific pose given any image with an arbitrary pose.

The proposed model also consists of nine separate discriminator networks D_i; $i = \{1, 2, \ldots, 9\}$, shown in Fig. 3, which performs two tasks, recognizing fake images (F_i) generated by GD_i from original images (R_i) along with classifying input images into separate categories. Thus, the discriminator minimizes three loss components, the loss occurred when an original image is classified as a fake image, loss incurred due classification of generated image as real image and the categorical cross entropy loss which ensures correct classification of the input images.

It may be noted that as the model consists of nine discriminator modules, it may produce nine class-ids for the same input image. Thus, to evaluate the final class-id for a given image during test time, we deploy the Max-Voting mechanism [16]. It is to be noted that as the images provided to the decoder networks have a small variation in pose and the model has already been trained sufficiently to discriminate between different classes with varied tilt and pan angles, a group of decoder networks always vote for the same class, which helps to perform max-voting.

3.2 Loss Functions

The loss functions which have been employed in the proposed PosIX-GAN model are defined as follows:

Patch-Wise MSE Loss. Patch-wise MSE (PMSE) loss is derived from the mean-squared error between two images. Let p_1 and p_2 be the two patches

Algorithm 1. Conceptual steps for Patch-wise MSE loss evaluation

```
function PMSE($im_1, im_2$)
    // PATCH_SIZE $\leftarrow 11 \times 11$ px
    // stride $\leftarrow 3 \times 3$ px
    $mse^{(1)} \leftarrow 0; mse^{(2)} \leftarrow 0; mse^{(3)} \leftarrow 0$
    foreach $patch$ $\{p_1, p_2\}$ $in$ $\{im_1, im_2\}$ do
        $mse^{(1)} := mse^{(1)} + \mathrm{MSE}(p_1^{(1)}, p_2^{(1)})$
        $mse^{(2)} := mse^{(2)} + \mathrm{MSE}(p_1^{(2)}, p_2^{(2)})$
        $mse^{(3)} := mse^{(3)} + \mathrm{MSE}(p_1^{(3)}, p_2^{(3)})$
        // $\alpha^{(i)} \leftarrow i^{th}$ channel of $\alpha$
        // $MSE(a, b)$: as in equation 2
        $loss := 0.2989 \times mse^{(1)} + 0.5870 \times mse^{(2)} 0.1141 \times mse^{(3)}$
    $P\_avg_{loss} := \frac{loss}{|p_1|}$
    // $|p_i| \leftarrow$ total number of patches
    return $P\_avg_{loss}$
```

extracted from a pair: $image_1$ and $image_2$. The PMSE between $image_1$ and $image_2$, is calculated as:

$$\mathcal{L}_{pmse}(image_1, image_2) = \sum_{i=1}^{|C|} \frac{\lambda_i}{|p|} \sum_{j=1}^{|p|} \| p_1^{(i,j)} - p_2^{(i,j)} \|^2 \tag{2}$$

where, $|C|$ & $|p|$ specifies the number of channels and patches in the image, while the subscript k in p_k represents the image from which the patch is extracted and λ_i's are the weights of each channel in the image ($\lambda = \{0.2989, 0.5870, 0.1141\}$ as used in our experimentations). A weighted linear combination of the three MSE's components is then used to estimate the overall MSE for each patch as given in Algorithm 1. PMSE is the average MSE over all the patches.

Categorical Cross Entropy Loss. Categorical cross-entropy [24] is a loss function used effectively in the field of deep learning (DL) for multi-class classification problems and sigmoid output units. The loss function is given as:

$$\mathcal{L}_{cce}(X, Y) = -\frac{1}{n} \sum_{i=1}^{n} y^{(i)} \ln a(x^{(i)}) \tag{3}$$

where, $X = \left\{x^{(1)}, \ldots, x^{(n)}\right\}$ is the set of input examples in the training dataset, and $Y = \left\{y^{(1)}, \ldots, y^{(n)}\right\}$ is the corresponding set of labels for those input examples. The function $a(x)$ represents the output of the neural network (perceptron) given input x.

3.3 Training using Multi-objective Adversarial Loss Function

Adversarial training has been used for the training of PosIX-GAN, which minimizes the loss functions within generator ($\mathcal{L}_G^{adv}$) and discriminator ($\mathcal{L}_D^{adv}$). The encoder network transforms the input image into a 256-dimensional vector which is then fed to each of the nine decoder networks, producing nine 64 × 64 px. images at different poses. Thus, the adversarial loss corresponding to G is given as:

$$\mathcal{L}_G^{adv} = \mathcal{L}_{cce}(D(G(I^B)), \mathbf{1}) + \mathcal{L}_{cce}(D(G(I^B)), y) + \mathcal{L}_{pmse}(I_r^B, G(I^B)) \tag{4}$$

where, I^B are the real-world face images and B indicates the batch, while r represents nine specific pose angles (see Sect. 4 for further details.), y are the class labels. $\mathcal{L}_{pmse}$ and $\mathcal{L}_{cce}$ are defined earlier in Eqs. 2 and 3. The three components of Eq. 4 are described subsequently. To ensure that the generated images are similar to the original image, the first component is formulated as:

$$\mathcal{L}_{cce}(D(G(I^B)), \mathbf{1}) = \sum_{i=1}^{9} \mathcal{L}_{cce}(D_i(G_i(I^B)), \mathbf{1}) \tag{5}$$

The following loss function captures the class information and helps to ensure that the generated images resemble the real images of the same class:

$$\mathcal{L}_{cce}(D(G(I^B)), y) = \sum_{i=1}^{9} \mathcal{L}_{cce}(D_i(G_i(I^B)), y) \tag{6}$$

Finally, the PMSE loss function is given as:

$$\mathcal{L}_{pmse}(I_r^B, G(I^B)) = \sum_{i=1}^{9} \mathcal{L}_{pmse}((I_r^B)_i, G_i(I^B)) \tag{7}$$

It is to be noted that $(I_r^B)_i$ corresponds to R_i as shown in Fig. 1, for each decoder GD_i. Each of the decoder modules of the generator is fed with the generated images. The decoder modules make use of corresponding real images of the same class at a predefined pose along with these generated images to find a mapping between any arbitrary image and the generated image at a certain pose, while also preserving class-information by minimizing the patch-wise MSE loss which is described in Sect. 3.2.

Post generation of images by G_i, the discriminators D_i (shown in Fig. 3) perform two tasks: (a) it discriminates the images that are generated by G from the original images, and (b) also classifies the images to provide their class-IDs. The training of D is also based on the adversarial loss, described as:

$$\mathcal{L}_D^{adv} = \mathcal{L}_{cce}(D(I_r^B), \mathbf{1}) + \mathcal{L}_{cce}(D(G(I^B)), \mathbf{0}) + \mathcal{L}_{cce}(D(G(I^B)), y) \tag{8}$$

where, $\mathcal{L}_{cce}(D(G(I^B)), y)$ is the categorical cross-entropy loss. The three terms of Eq. 8 are further described subsequently. The first loss component helps the

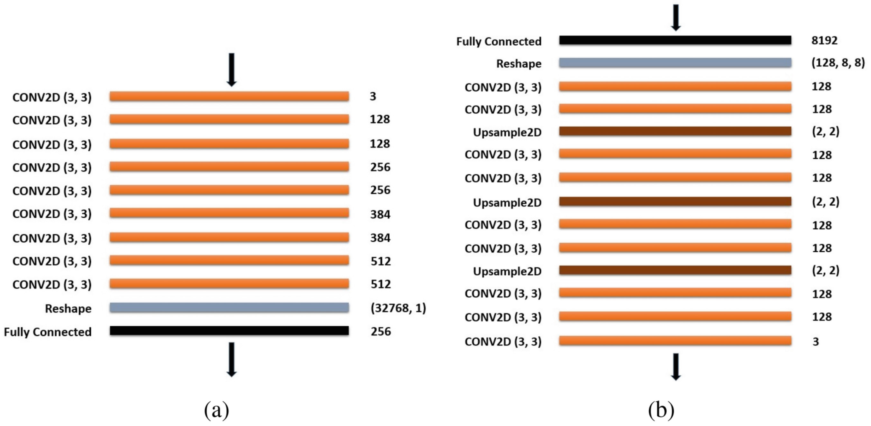

Fig. 2. The architecture of the Generator module G_i which contains a (a) shared encoder module, and a set of nine (b) decoder modules (best viewed in color). (Color figure online)

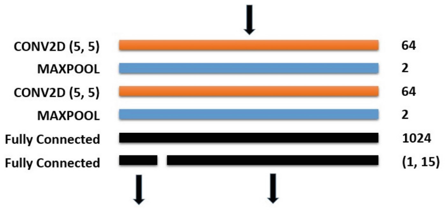

Fig. 3. The architectural details of a single Discriminator module (D_i) (see Fig. 1) (best viewed in color). (Color figure online)

discriminator network to learn to correctly classify the original images as real, given as:

$$\mathcal{L}_{cce}(D(I_r^B), \mathbf{1}) = \sum_{i=1}^{9} \mathcal{L}_{cce}(D_i(I_r^B), \mathbf{1}) \tag{9}$$

The following loss component ensures that the images generated by the generator network are correctly identified as fake images by the discriminator network:

$$\mathcal{L}_{cce}(D(G(I^B)), \mathbf{0}) = \sum_{i=1}^{9} \mathcal{L}_{cce}(D_i(G_i(I^B)), \mathbf{0}) \tag{10}$$

Finally, the last term of the loss function (Eq. 8) given below helps the discriminator network to correctly classify each face image.

$$\mathcal{L}_{cce}(D(G(I^B)), y) = \sum_{i=1}^{9} \mathcal{L}_{cce}(D_i(G_i(I^B)), y) \tag{11}$$

The introduction of the class-IDs reinforces stabilization during adversarial training of PosIX-GAN, facilitating a faster convergence [21] of the network to an equilibrium. As the nine discriminators produce nine labels for the same input image, we use max voting mechanism [16] to ascertain the class label for a certain image during testing.

4 Datasets Used for Experimentation

We have used two datasets for experimentation purposes, viz. Head pose Image Database (HPID) [9] and the Multi-PIE dataset [11]. The dataset statistics are described below:

4.1 Head Pose Image Database (HPID)

The head pose image database is a benchmark of 2790 monocular face images of 15 persons with variations of pan and tilt angles from −90 to +90°. Two series of images were captured for each person, having 93 images (each having a distinct pose) in each series. The purpose of having 2 series per person is to be able to train and test algorithms on known and unknown faces. People in the database wear glasses or not and have various skin color. Background is willingly neutral and uncluttered in order to focus on face operations. Figure 4 shows the pose variations present in the dataset where the values represented as $(\cdot, \cdot)$ on top of the images indicate the `(tilt, pan)` angles.

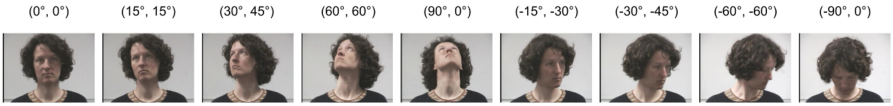

Fig. 4. A few samples from the Head Pose Image Dataset (HPID) [10] showing the pose variations that are present in the dataset (best viewed in color). The values represented as $(\cdot, \cdot)$ on top of the images indicate the `(tilt, pan)` angles.

For the purpose of our experimentation, we make use of the complete series 1 and 40% of series 2 for training the PosIX-GAN and use the rest for testing purposes. For the patch-wise MSE loss, we separate a few images per subject from the training set itself and group them into nine sets by the nine tilt angles present across the dataset while clubbing together images at different pan angles under each category, as shown above in Fig. 4. These faces are then utilized by the decoder network to evaluate the patch-wise MSE loss.

4.2 Multi-PIE Dataset

To systematically capture images with varying poses and illuminations a system of 15 cameras and 18 flashes connected to a set of Linux PCs was used. Thirteen cameras were located at head height, spaced at 15° intervals, and two additional cameras were located above the subject, simulating a typical surveillance view. During a recording session 20 images were captured for each camera: one image without any flash illumination, 18 images with each flash firing individually, and then another image without any flash. Taken across all cameras a total of 300 images was captured within 0.7 seconds. Figure 5 shows a few samples from the Multi-PIE dataset where the values on top of the images indicate the corresponding pan angles.

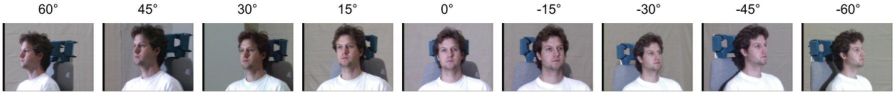

Fig. 5. A few samples from the Multi-PIE dataset [11] showing the pose variations present in the dataset (best viewed in color). The values on top of the images indicate the corresponding pan angles. (Color figure online)

Subjects were seated in front of a blue background in close proximity of the camera. The resulting images are 3072 × 2048 in size with the inter-pupil distance of the subjects typically exceeding 400 pixels. The part of the dataset with neutral expression was only used for experimentation purposes.

Images across Sessions 1–4 with neutral facial expressions was used for experimentation purposes. As our method does not deal with low illumination images, we only used well illuminated face images (file names ending with 06–09) at all pose variations, except Sections 08_1 and 19_1, for experimentation. The filtered data, thus obtained, was randomly partitioned into training (70%) and test (30%) data. As in the case of HPID, a few samples were seperated out from the training set (specifically, from the following nine Sections 04_1, 05_0, 05_1, 08_0, 09_0, 13_0, 14_0, 19_0, 20_0; for the nine decoder networks) and divided into the nine pan angles as shown above in Fig. 5. These nine set of images were then provided as input to the nine decoder networks in G for evaluation of the PMSE loss, thus enabling each of the networks to learn a mapping between any arbitrary pose to a predefined pose (separate for every decoder GD_i).

5 Experimental Results and Observations

The experimentations are performed on a machine with Dual-Xeon processor and 256 GB RAM, having 4 GTX-1080 Ti GPUs. The implementations are all coded in *Keras* platform using *tensorflow*-backend. The model weights were all randomly initialized and was trained on GPU for 5–6 hours. The batch size was kept to 10 and the input size of the network is kept to 64 × 64 pixels.

In the following sections, we report the quantitative results (using Rank-1 recognition rates) as well as qualitative results (faces generated at various poses). We also provide with results where patch-wise MSE loss is not incorporated using the training phase to show the effectiveness of the PMSE loss to obtain crisp result.

Table 1. Rank-1 Recognition Rate for different state-of-the-art methods on the Multi-PIE [11] dataset (in %). Results in bold shows the best performance. All the results except the last row (ours) have been directly reported from [13,30,32].

Sl.	Algorithm	±15°	±30°	±45°	±60°	±75°	±90°
1	Zhu *et al.* [37]	90.7	80.7	64.1	45.9	-	-
2	Zhu *et al.* [38]	92.8	83.7	72.9	60.1	-	-
3	Kan *et al.* [14]	100	100	90.6	85.9	-	-
4	Yin *et al.* [35]	99.2	98.0	90.3	92.1	-	-
5	Yim *et al.* [34]	76.64	79.1	78.4	79.2	-	-
6	Wu *et al.* [32]	100	99.9	99.6	95.0	32.3	9.0
7	Tran *et al.* [30]	94.0	90.1	86.2	83.2	-	-
8	Liu *et al.* [17]	95.3	89.7	87.9	84.1	-	-
9	Huang *et al.* [13]	98.7	98.0	95.4	87.7	77.4	64.6
10	Ours	**100**	**100**	**100**	**97.8**	**85.3**	**80.6**

5.1 Quantitative Results

Table 1 reports the experimental findings of our proposed method, compared with eight state-of-the-art methods, using the Multi-PIE dataset [11]. All the images were cropped using Chehra [2] to discard the background. The Rank-1 recognition rates of the methods listed in Table 1 have been directly reported from [13,30,32] for the dataset. The missing values in the table are not reported by the respective authors in their paper. Although, for lower pose variations the method proposed by Wu *et al.* [32] performs the 2^{nd} best, but it fails at major pose variations like $\pm 75-90°$, where the TP-GAN [13] performs the 2^{nd} best. Comparing all the results reported in Table 1, it may be noted that our method outperforms all other techniques by a considerable margin.

Experiments have also been carried out on the Head Pose Images Dataset [9] where the dataset partition strategy mentioned in Sect. 4.1 has been followed for evaluating the proposed method. The preprocessing procedure in this case remains the same as that done for Multi-PIE dataset. Table 2 reports the Rank-1 recognition rates of the proposed method along with a few classical methods on this dataset. From Table 2, it can be seen that our method outperforms all other compared methods by a large margin. The method proposed by Huang *et al.* [13] again provides the 2^{nd} best performance.

5.2 Qualitative Results

In this section, we show a few synthetic images generated by PosIX-GAN and also compare our performance with a hybrid BEGAN [4] model implemented without the PMSE loss. Figure 6 shows the generated result by our proposed model PosIX-GAN. The second set of images shown in Fig. 7, which are generated without the use of PMSE loss, exhibit lack the crispness compared to that in

Table 2. Rank-1 Recognition Rate for different state-of-the-art methods on the Head Pose Image Dataset [9] dataset. Results in bold shows the best performance. All the results except the last row (ours) have been directly reported from [12].

Sl.	Algorithm	Classification accuracy
Gourier experimental settings see Table 2 of [18]		
1	Human Performance [10]	59.0
2	Associative Memories [10]	43.9
3	VRF+LDA [12]	66.9
Jilin Tu experimental settings see Table 2 of [18]		
4	High-order SVD [31]	54.8
5	PCA [31]	57.9
6	LEA [31]	50.6
7	VRF+LDA [12]	62.6
Proposed experimental settings (see Sect. 4.1)		
8	Huang et. al [13]	81.8
9	Ours	**92.1**

Fig. 6. The first set of images have good clarity and are closer to the ground truth compared to the second of images which are blurry with aliasing effects throughout. The numerical values at the end of each row in Figs. 6 and 7 indicate the average PSNR/SSIM values for those in each row.

Further, we also compare the synthesis results of DR-GAN [30], MTAN [17] and RNN [33] methods with the proposed PosIX-GAN. Figures 8 and 9 show a few face images generated along with the corresponding PSNR and SSIM values for every image (grayscale version used here) compared *w.r.t.* ground truth given in top row.

The PSNR/SSIM values estimated for the images indicates the superiority of our method compared to the existing state-of-the-art methods. A noticeable drawback among all methods is their inability to produce crisp images without deformities at extreme poses. The faces generated by PosIX-GAN are devoid of any such deformities and are quite crisp even at extreme pose.

The images generated by the DR-GAN method [30], shown in Fig. 8, exhibit deformities as well as inaccuracies in the generated faces (beard not present in the generated images of Multi-PIE [11], while being present in the ground-truth). The output generated by the MTAN method [17] (Fig. 9; right) is blurry and the quality of synthesis deteriorates with larger values of the pan angle as evident from the PSNR/SSIM values. The RNN method [33] performs the 2^{nd} best in face generation task, which can be verified visually, as well as, from the PSNR/SSIM values reported for each image in Fig. 9 (left). However, this method can only generate images upto a pan angle of ±45°.

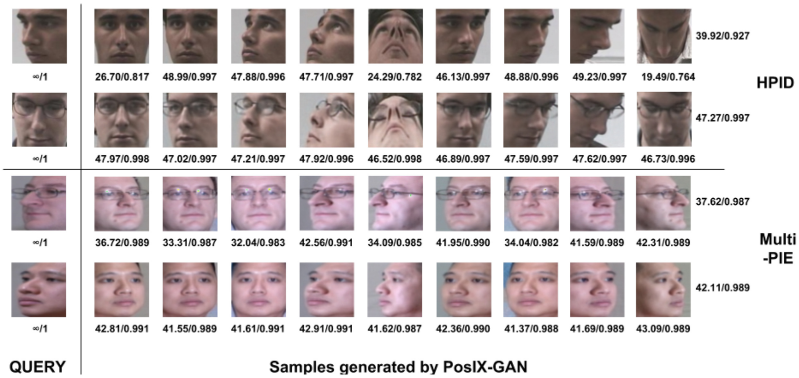

Fig. 6. Images generated by our proposed PosIX-GAN model along with the PSNR/SSIM values (best viewed in color). The numerical values at the end of each row indicate the average PSNR/SSIM values for the complete row. (Color figure online)

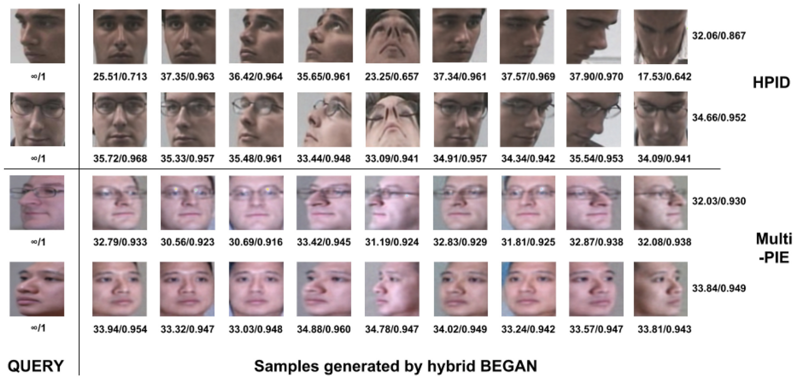

Fig. 7. Images generated by the hybrid BEGAN model (without PMSE loss) along with the PSNR/SSIM values for each image (best viewed in color). The numerical values at the end of each row indicate the average PSNR/SSIM values for the complete row. (Color figure online)

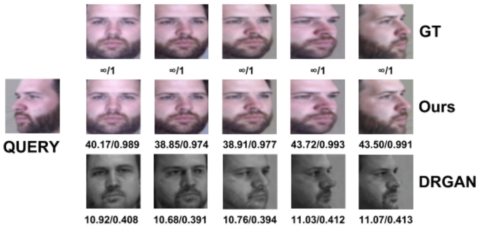

Fig. 8. Comparison of images generated with DR-GAN [30] with the proposed PosIX-GAN for Multi-PIE dataset [11] (best viewed in color). (Color figure online)

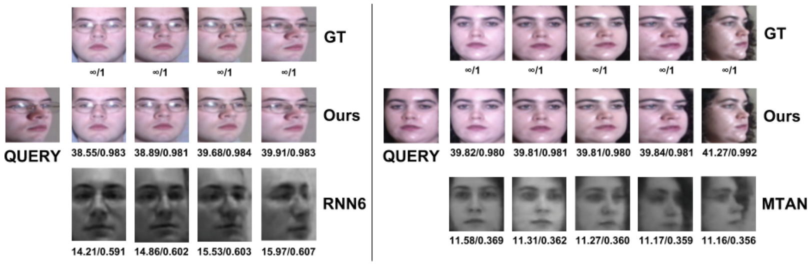

Fig. 9. Comparison of images generated with RNN [33] (left) and MTAN [17] (right) with the proposed PosIX-GAN for Multi-PIE dataset [11] (best viewed in color). (Color figure online)

6 Conclusion

This paper proposes a single-encoder, multi-decoder based generator model as a modified GAN boosted by multiple supervised discriminators for generating face images at different poses, when presented with a face at any arbitrary pose. The supervised PosIX GAN can act as a pre-processing tool for 3-D face synthesis. The qualitative as well as the quantitative results reveal the superiority of our proposed technique over few recent state-of-the-art techniques, using two benchmark datasets for PIFR. The PosIX model is capable of handling extreme pose variations for generation as well as recognition tasks, which most of the state-of-the-art techniques fail to achieve. This method also provides a basis for multiple image 3D face reconstruction, which can be explored in the near future for generating faces with dense set of pose values.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. arXiv preprint arXiv:1701.07875 (2017)
2. Asthana, A., Zafeiriou, S., Cheng, S., Pantic, M.: Incremental face alignment in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1859–1866 (2014)
3. Banerjee, S., Das, S.: Mutual variation of information on Transfer-CNN for face recognition with degraded probe samples. Neurocomputing **310**, 299–315 (2018)
4. Berthelot, D., Schumm, T., Metz, L.: Began: Boundary equilibrium generative adversarial networks. arXiv preprint arXiv:1703.10717 (2017)
5. Binmore, K.: Playing for Real: A Text on Game Theory. Oxford University Press, Oxford (2007)
6. Chen, J.C., Zheng, J., Patel, V.M., Chellappa, R.: Fisher vector encoded deep convolutional features for unconstrained face verification. In: IEEE International Conference on Image Processing (ICIP), pp. 2981–2985 (2016)
7. Chen, M., Xu, Z., Weinberger, K., Sha, F.: Marginalized denoising autoencoders for domain adaptation. arXiv preprint arXiv:1206.4683 (2012)
8. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (NIPS), pp. 2672–2680 (2014)

9. Gourier, N., Hall, D., Crowley, J.L.: Estimating face orientation from robust detection of salient facial features. In: ICPR International Workshop on Visual Observation of Deictic Gestures (ICPRW) (2004)
10. Gourier, N., Maisonnasse, J., Hall, D., Crowley, J.L.: Head pose estimation on low resolution images. In: Stiefelhagen, R., Garofolo, J. (eds.) CLEAR 2006. LNCS, vol. 4122, pp. 270–280. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-69568-4_24
11. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-PIE. Image Vis. Comput. (IVC) **28**(5), 807–813 (2010)
12. Huang, C., Ding, X., Fang, C.: Head pose estimation based on random forests for multiclass classification. In: International Conference on Pattern Recognition (ICPR), pp. 934–937 (2010)
13. Huang, R., Zhang, S., Li, T., He, R., et al.: Beyond face rotation: Global and local perception GAN for photorealistic and identity preserving frontal view synthesis. arXiv preprint arXiv:1704.04086 (2017)
14. Kan, M., Shan, S., Chen, X.: Multi-view deep network for cross-view classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4847–4855 (2016)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (NIPS), pp. 1097–1105 (2012)
16. Lam, L., Suen, S.: Application of majority voting to pattern recognition: an analysis of its behavior and performance. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **27**(5), 553–568 (1997)
17. Liu, Y., Wang, Z., Jin, H., Wassell, I.: Multi-task adversarial network for disentangled feature learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3743–3751 (2018)
18. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **31**(4), 607–626 (2009)
19. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: British Machine Vision Conference (BMVC), vol. 1, p. 6 (2015)
20. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: International Conference on Learning Representations (ICLR) (2015)
21. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. In: Advances in Neural Information Processing Systems (NIPS), pp. 2234–2242 (2016)
22. Schmidhuber, J.: Learning factorial codes by predictability minimization. Neural Comput. **4**(6), 863–879 (1992)
23. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 815–823 (2015)
24. Shore, J., Johnson, R.: Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. IEEE Trans. Inf. Theory **26**(1), 26–37 (1980)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
26. Sun, Y., Liang, D., Wang, X., Tang, X.: Deepid3: Face recognition with very deep neural networks. arXiv preprint arXiv:1502.00873 (2015)

27. Sun, Y., Wang, X., Tang, X.: Deep learning face representation from predicting 10,000 classes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1891–1898 (2014)
28. Sun, Y., Wang, X., Tang, X.: Deeply learned face representations are sparse, selective, and robust. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2892–2900 (2015)
29. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: DeepFace: closing the gap to human-level performance in face verification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1701–1708 (2014)
30. Tran, L., Yin, X., Liu, X.: Disentangled representation learning GAN for pose-invariant face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 3, p. 7 (2017)
31. Tu, J., Fu, Y., Hu, Y., Huang, T.: Evaluation of head pose estimation for studio data. In: Stiefelhagen, R., Garofolo, J. (eds.) CLEAR 2006. LNCS, vol. 4122, pp. 281–290. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-69568-4_25
32. Wu, X., He, R., Sun, Z., Tan, T.: A light cnn for deep face representation with noisy labels. IEEE Trans. Inf. Forensics Secur. **13**(11), 2884–2896 (2018)
33. Yang, J., Reed, S.E., Yang, M.H., Lee, H.: Weakly-supervised disentangling with recurrent transformations for 3D view synthesis. In: Advances in Neural Information Processing Systems (NIPS), pp. 1099–1107 (2015)
34. Yim, J., Jung, H., Yoo, B., Choi, C., Park, D., Kim, J.: Rotating your face using multi-task deep neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 676–684 (2015)
35. Yin, X., Liu, X.: Multi-task convolutional neural network for pose-invariant face recognition. IEEE Trans. Image Process. (2017)
36. Zhao, J., Mathieu, M., LeCun, Y.: Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126 (2016)
37. Zhu, Z., Luo, P., Wang, X., Tang, X.: Deep learning identity-preserving face space. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 113–120 (2013)
38. Zhu, Z., Luo, P., Wang, X., Tang, X.: Multi-view perceptron: a deep model for learning face identity and view representations. In: Advances in Neural Information Processing Systems (NIPS), pp. 217–225 (2014)
39. Zhu, Z., Luo, P., Wang, X., Tang, X.: Recover canonical-view faces in the wild with deep neural networks. arXiv preprint arXiv:1404.3543 (2014)