Face Synthesis and Recognition Using Disentangled Representation-learning Wasserstein GAN

Gee-Sern Jison Hsu, Chia-Hao Tang
National Taiwan University of Science and Technology
Taipei, Taiwan
jison@mail.ntust.edu.tw, m10603423@mail.ntust.edu.tw

Moi Hoon Yap
Manchester Metropolitan University
Manchester, UK
M.Yap@mmu.ac.uk

Abstract

We propose the Disentangled Representation-learning Wasserstein GAN (DR-WGAN) trained on augmented data for face recognition and face synthesis across pose. We improve the state-of-the-art DR-GAN with the Wasserstein loss considered in the discriminator so that the generative and adversarial framework can be better trained. The improved training leads to better face disentanglement and synthesis. We also highlight the influences of imbalanced training data on the disentangled facial representation learning, and point out the difficulty of generating faces of extreme poses. We explore the recently proposed nonlinear 3D Morphable Model (3DMM) to augment the training data, and verify the contributions made by the learning on augmented data. Additionally, we also compare different data normalization schemes and reveal the benefit of using the group normalization. The proposed framework is verified through the experiments on benchmark databases, and compared with contemporary approaches for performance evaluation.

1. Introduction

The approaches for face recognition across pose can be generally split into three categories. One category aims to rotate a non-frontal face to the frontal view for better extraction of facial features and the problem can be solved by comparing the extracted features [10, 17, 30]. Another category aims to learn the pose-invariant features directly from non-frontal faces [3, 13, 19]. The third category aims to learn the disentangled representation so that the identity-preserving features can be disentangled from pose, illumination and other parameters for better representing the identity of the face [15, 25]. The approach proposed in this paper belongs to the third category.

Disentangled representation learning refers to the learning to decompose the representation of an object into multiple independent representations and each independent representation characterizes a specific characteristic of the object. When the object is a face, the independent representations can be composed of the following vectors: $\mathbf{c}_d, \mathbf{c}_p, \mathbf{c}_l, \mathbf{c}_r$, where $\mathbf{c}_d$ characterizes the identity, c_p characterizes the pose, c_l characterizes the illumination and $\mathbf{c}_r$ characterizes other variables independent of the identity, pose and illumination. As the Generative Adversarial Network (GAN) offers an effective tool for extracting disentangled representations, several approaches are proposed recently for better learning of the disentangled facial representations [15, 25]. The approach proposed by Peng et al. combines the multi-source feature embedding, 3D face modeling and reconstruction-based metric learning to disentangle identity and pose features [15]. It demonstrates a competitive performance on the Celebrities in Frontal-Profile (CFP) database [20]. The DA-GAN (Dual-Agent Generative Adversarial Network) [31] uses synthetic profile face images as augmented data to balance the pose variance. It leads to compelling perceptual results under extreme poses and outperforms state-of-the-arts on IJB-A dataset. The DR-GAN (Disentangled Representation learning-GAN) [25] learns a generative and discriminative facial representation which disentangles the face identity from pose so that it can better handle cross-pose recognition. The DR-GAN is built on the common two-player GAN architecture, but its generator explores an encoder-decoder structure, leading to the desired disentanglement. The input to the encoder is a face image of any pose, and the output of the decoder is a synthetic face at a specified pose. The output of the encoder is a latent vector, which connects a one-hot pose code and a noise to serve as the input to the decoder. The discriminator follows the same discriminator design in the Categorical Generative Adversarial (CGA) network [21] which is trained to not only distinguish synthetic (fake) images from real images, but also predict the identity and pose of the input face.

Although the DR-GAN demonstrates a good performance for cross-pose recognition, our experiments reveal

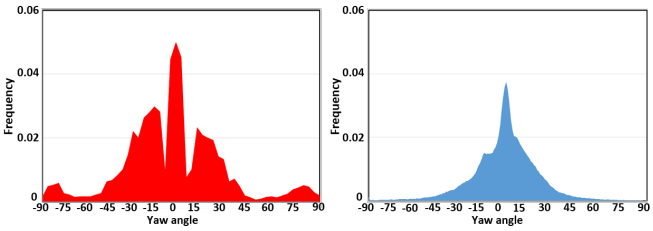

Figure 1. Data distribution across the angle in yaw, (a) IJB-A, (b) CASIA-WebFace.

the following issues with the DR-GAN: 1) It is difficult to stabilize the training. In a few stable cases, the mode collapse often takes place, producing degenerate images; 2) It can hardly generate faces of extreme poses, e.g., yaw $> 60°$. This issue can be caused by the data imbalance across pose, i.e., very few training data are with nearly profile or extreme poses. This issue can be one of the reasons that make the DR-GAN report the face recognition performance on the MPIE database [8] up to $60°$ only, but the MPIE offers up to $90°$. To circumvent the above issues, we propose the following improvements: 1) Replacement of the minimization of the Jensen-Shannon divergence considered in the DR-GAN by the minimization of the Wasserstein-1 loss considered in the Wasserstein GAN (WGAN) [1]; and 2) Data augmentation by the face images synthesized using the nonlinear 3D Morphable Model (3DMM) [24] to augment the training data for better pose distribution. In addition to these improvements, we also study the benefits of replacing the batch normalization in the DR-GAN and other GANs by the recently proposed group normalization [28].

Although it is commonly known that learning based on imbalanced data would result in biased estimation, the influence of imbalanced data on the learning of disentangled representation has not received much attention so far. Pointed out in a recent work by Masi et al. [13], several common databases all exhibit imbalanced pose distribution. Two examples, the ARPA Janus Benchmark A (IJB-A) [11] and the CASIA WebFace [29], are illustrated in Fig. 1. Note that most of the faces are within $40°$ in yaw, and almost none with profile or nearly profile poses. As our experiments reveal that the training based on the imbalanced CASIA WebFace leads to undesired face images made by the generator and deteriorate the learning, we explore the recently proposed nonlinear 3DMM [24] for making the face images with poses needed to augment the training dataset. In the following, we first give a brief review to the disentangled facial representation learning in Sec. 2. The proposed Disentangled Representation-learning Wasserstein GAN (DR-WGAN) and the nonlinear 3DMM based data augmentation for handling the imbalanced data are presented in Sec. 3. The experiments to verify the proposed framework are reported in Sec. 4, followed by a conclusion in Sec. 5.

2. Related Work

Deep generative models can represent high dimensional data by using a low dimensional representation, which is often referred to as a *code*. The relationship between the data and the code can be described by a conditional probability distribution parametrized by a deep neural network, but it is difficult to interpret the relationship in a semantically meaningful way [7]. Disentangled representation learning offers a good way to better interpret the relationship. To better handle cross-pose recognition, a few approaches are proposed recently for the disentangled facial representation learning [15, 25]. The approach proposed by Peng et al. combines the 3D face modeling, multi-source feature embedding, and reconstruction-based metric learning [15]. They first augment the data by generating non-frontal views of a frontal face using the time-consuming conventional 3DMM. The augmented data is used to encode identity and non-identity features by multi-source supervision. A feature reconstruction metric learning is developed to disentangle identity and pose by demanding alignment between the reconstructed features through various combinations of identity and pose features.

The DR-GAN is built on a modified version of the CASIA-Net [29]. It learns an identity representation $I(x)$ for a face image x by using an encoder-decoder structured generator, i.e., $G = [G_e, G_d]$, where the representation is the output of the encoder G_e and the input of the decoder G_d so that it can synthesize various poses of faces for the same identity, making $I(x)$ a generative representation. Besides, a separate pose code is entered to G_d during training, and G_e is trained to disentangle the pose variation from $I(x)$, making $I(x)$ a discriminative facial representation for the identity. Two objectives are pursued by the DR-GAN. One is to learn a G_e to transform a face x into a pose-invariant $I(x)$, and to learn a G_d which takes $I(x)$ together with a given pose code c_p and a noise z as input to synthesize a face image $\hat{x}$ that has the same identity as x but in the pose assigned by c_p. The other objective is to learn a multi-task D to distinguish the synthesized $\hat{x}$ from the real x, and to identify the identity and pose of x and $\hat{x}$. Given a real face x, D aims to estimate its identity and pose; while given a synthetic face $\hat{x}$ made by the generator, D aims to classify $\hat{x}$ as fake. The objective considered in training D has two parts, one is to maximize the probability of x being classified to the correct identity and pose, and the other is to maximize the probability of $\hat{x}$ being classified as fake. The goal of G is to fool D to accept $\hat{x}$ to be real, and classify $\hat{x}$ to the same identity as of input x with the target pose assigned by c_p.

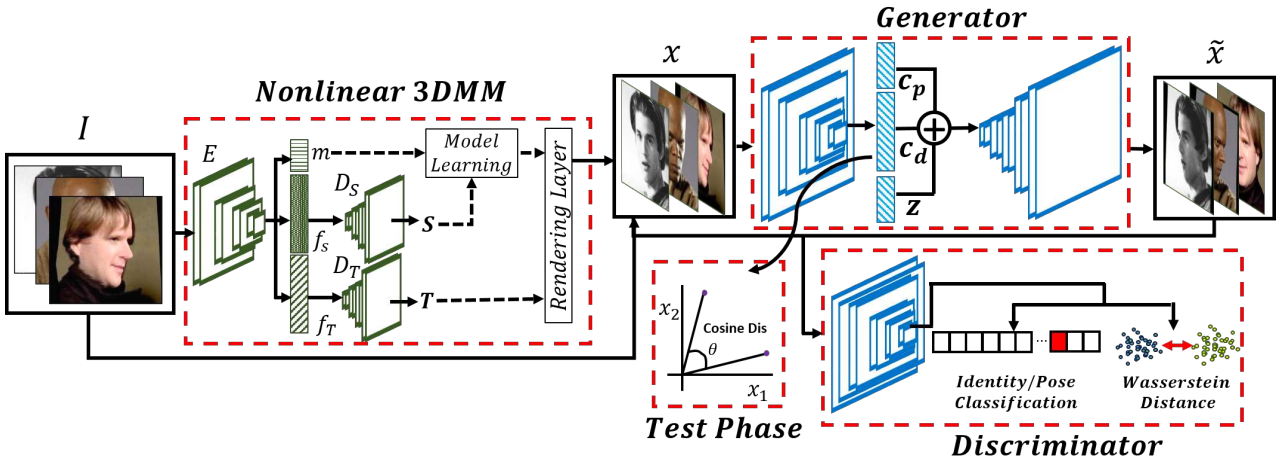

Figure 2. The proposed framework composed of the DR-WGAN for disentangled representation learning and the nonlinear 3DMM-based face profiling for data augmentation. The cosine distance is used as the metric for feature matching in test phase.

3. Proposed Framework

The proposed framework is shown in Fig. 2, which includes the Disentangled Representation-learning Wasserstein-GAN (DR-WGAN) and the training data augmentation module made of the nonlinear 3DMM. The DR-WGAN is revised from the DR-GAN with the following amendments: 1) The discriminator is built upon the Wasserstein loss (in contrast to the cross-entropy loss computed by the softmax function in the DR-GAN) for better training properties; and 2) the batch normalization in the DR-GAN replaced by the group normalization [28] for better feature extraction across the convolution layers. In the following, we first briefly review the GAN and the Wasserstein-GAN (WGAN) in Sec. 3.1, then present the proposed DR-WGAN in Sec. 3.2, and then the nonlinear 3DMM for data augmentation in Sec. 3.3.

3.1. Wasserstein Generative Adversarial Network

The Generative Adversarial Networks (GANs) are a family of deep learning networks for constructing generative models based on the two-player game theory. GANs are designed for achieving two objectives. One objective is to train a generator network $G(\cdot, \theta_G)$ to learn the best parameter θ_G^* such that the network can transform a noise distribution $p_z(z)$ to the desired model distribution $p_g(\hat{x})$, where $\hat{x} = G(z, \theta_G^*)$, and make $p_g(\hat{x})$ as close as possible to the real data distribution $p_d(x)$. The other objective is to train a discriminator network D to distinguish the G-generated *fake* $\hat{x}$ from the real x, and in turn, $G(z, \theta_G^*)$ is trained to fool D into accepting $\hat{x}$ as real. This game between G and D can be written as a min-max objective:

$$\min_{\theta_G} \max_{D} \mathbf{E}_{x\sim p_d} [\log D(x)] + \mathbf{E}_{z\sim p_z} [\log (1 - D(\hat{x}))] \quad (1)$$

where $\hat{x} = G(z, \theta_G)$. It is known that the training of GANs is difficult and suffers from mode collapse and diminishing gradients. To partially circumvent these issues, Arjovsky et al. [1] propose to revise the cost function based on the Wasserstein-1 distance $W(p_d, p_g)$ to convert the problem to the cost of transporting the mass of p_g to that of p_d, and call their framework Wasserstein GAN (WGAN). The following min-max objective is considered in the optimization of D.

$$\min_{\theta_G} \max_{D\in \mathbf{D}_L} \mathbf{E}_{x\sim p_d} [D(x)] - \mathbf{E}_{\hat{x}\sim p_g} [D(\hat{x})] \quad (2)$$

where $\mathbf{D}_L$ is the set of 1-Lipschitz functions. When the discriminator D is being optimized, the parameter update involved in $\min_{\theta_G}$ leads to the minimization of $W(p_d, p_g)$, which yields a *critic* function whose gradient behaves better than does the gradient involved in (1). To meet more requirements, Arjovsky et al. [9] improve the WGAN with a gradient penalty (GP) added in, resulting in WGAN-GP. They impose a constraint on the gradient norm of the discriminator's output with respect to its input, and constitute the following objective:

$$\begin{aligned}\min_{\theta_G} \max_{D\in \mathbf{D}_L} \quad & \mathbf{E}_{x\sim p_d} [D(x)] - \mathbf{E}_{\hat{x}\sim p_g} [D(\hat{x})] + \\ & \lambda \mathbf{E}_{\tilde{x}\sim p_{\tilde{x}}} \left[(||\nabla_{\tilde{x}} D(\tilde{x}) - 1||_2)^2 \right] \quad (3)\end{aligned}$$

The last term is the penalty on the gradient norm computed at random samples $\tilde{x} \sim p_{\tilde{x}}$. $p_{\tilde{x}}$ is implicitly defined as the distribution of the uniform samples along the straight lines between the pairs of the data sampled from the p_d and p_g.

3.2. Disentangled Representation-learning WGAN

The proposed Disentangled Representation-learning Wasserstein GAN (DR-WGAN) is composed of a generator G and a discriminator D, and both are built on the same structure of a base network. We choose the modified CASIA Net [29], same as that used in the DR-GAN, as the

base network, denoted by N_0. The modified CASIA Net N_0 is developed on a relatively simple architecture but offers a comparable performance to the DeepFace [23] and DeepID2 [22] for face recognition. It consists of 5 convolution blocks, including 1 double-convolution block and 4 triple-convolution blocks, followed by an average pooling (AvePool) layer for feature code extraction. The extracted feature code c_d is processed differently in G and in D.

In D, the AvePool layer is connected to two separate fully connected (FC) layers, one for handling the classification of identities and poses using the cross-entropy loss computed by the softmax function, and the other for discriminating the real from fake (generated) face images using Wasserstein loss function. The discriminator can therefore be written as two parts, i.e., $D = [D_{dp}, D_r]$, where D_{dp} is for identity and pose classification and D_r for real/fake discrimination. Note that the G and D in the DR-GAN all use the batch normalization to stabilize training.

The generator G is composed of an encoder G_e and a decoder G_d, i.e., $G = [G_e, G_d]$. We follow the design for making G in the DR-GAN. Given a face image x, the encoder's output code $c_d = G_e(x) \in R^{N_c}$ from the AvePool layer is concatenated with a pose code $c_p \in R^{N_p}$ and a noise $z \in R^{N_n}$ to form $[c_d, c_p, z]$, which is used as the input to G_d. G_d is a deconvolutional neural network that transforms $[c_d, c_p, z]$ to a decoded face image, i.e., $\hat{x} = G_d([c_d, c_p, z])$. G aims to make D_{dp} classify $\hat{x}$ as the same identity as x but in the desired pose c_p, and to fool D_r into determining $\hat{x}$ to be real. Therefore, in the DR-WGAN, the loss L_g considered for training G is evaluated via the softmax function with cross-entropy loss and the Wasserstein loss considered in the discriminator $D = [D_{dp}, D_r]$.

To make D_r, we connect the output of the average pooling layer in N_0 to a scalar output. When training D_r with a given image x, the images are entered in real-fake pairs $(x, \hat{x})$, and the interpolated data $\tilde{x}_s = \eta x + (1-\eta)\hat{x}$ can be determined by choosing $\eta \sim U[0,1]$. Given x, $\hat{x}$ and $\tilde{x}_s$, the loss $L_r(x, \hat{x})$ as shown in (3) can be computed.

The identity-pose discriminator D_{dp} has two parts $D_{dp} = [D_d, D_p]$, where $D_d(x) \in R^{d_d}$ is for identity classification and $D_p(x) \in R^{d_p}$ for pose identification. To make D_{dp}, we connect the output of the average pooling layer in N_0 to a $(d_d + d_p)$-dimensional fully connected layer with softmax outputs. The loss L_{dp} is the sum of the cross-entropy losses from the two parts,

$$\begin{aligned} L_{dp}(x) &= L_{d,d}(x) + L_{d,p}(x) \\ &= \mathbf{E}[\log D_d(x)] + \mathbf{E}[\log D_p(x)] \end{aligned} \quad (4)$$

Given the above losses considered in D, the loss considered for training G can be written as $L_g = L_r + L_{dp}$.

It is pointed out by Gulrajani et al. [9] that the WGAN-GP does not work with batch normalization (BN), which changes the discriminators processing from mapping a single input to a single output to mapping from a batch of inputs to a batch of outputs. Although the layer normalization (LN) is recommended for the WGAN-GP, we have found that the group normalization [28] performs better. The group normalization (GN) was proposed as a simple alternative to BN. GN divides the input channels into groups and computes the mean and variance within each group for normalization. However, different from BN, the GN computation is independent of the batch size, and the obtained parameters are stable over a wide range of batch sizes. Although the WGAN-GP processes single inputs, there are multiple channels for each single input, making the GN an appropriate choice for normalization. Note that the normalization discussed above is for the real/fake discriminator D_r with the WGAN-GP built in. The normalization for the G and D_{dp} can still be BN or GN, and we report the performance comparison of both in Sec. 4.

3.3. Nonlinear 3D MM

We explore the nonlinear 3DMM (3D Morphable Model) [24] for synthesizing the novel views of a face sample for data augmentation. The nonlinear 3DMM framework, proposed by Tran and Liu [24], has three deep networks, namely the encoder E, the shape decoder D_S and the texture decoder D_T. The encoder $E : \mathbf{I} \rightarrow \mathbf{m}, \mathbf{f}_S, \mathbf{f}_T$ estimates the projection parameter $\mathbf{m}$, the 3D shape parameter $\mathbf{f}_S \in R^{l_S}$, and the texture parameter $\mathbf{f}_T \in R^{l_T}$ for a given image $\mathbf{I}$. The 3D shape decoder $D_S : \mathbf{f}_S \rightarrow \mathbf{S}$ decodes the shape parameter $\mathbf{f}_S$ to a 3D shape $\mathbf{S}$. The texture decoder $D_T : \mathbf{f}_T \rightarrow \mathbf{T}$ decodes the texture parameter $\mathbf{f}_T$ to a realistic texture $\mathbf{T} \in R^{U \times V}$.

For reconstructing an input face image, the three deep networks work together with a geometry-based rendering layer. The problem can be formulated as follows: Given a set of 2D face images $\{\mathbf{I}_i\}_{i=1}^N$, we need to learn the three deep networks E, D_S, D_T with the objective that the rendered image with $\mathbf{m}, \mathbf{S}$, and $\mathbf{T}$ can approximate the original image well. The objective function considered is:

$$\underset{E,D_S,D_T}{\operatorname{argmin}} \sum_{i=1}^{N} \|\mathcal{R}(E_m(\mathbf{I}_i), D_S(E_S(\mathbf{I}_i)), D_T(E_T(\mathbf{I}_i))) - \mathbf{I}_i\|_1 \quad (5)$$

where $\mathcal{R}(\mathbf{m}, \mathbf{S}, \mathbf{T})$ is the face image rendering layer. In the following, we summarize the nonlinear 3DMM in three sections: first is the shape and texture representations in Sec. 3.3.1, followed by the making of face image rendering layer in Sec. 3.3.2, and then the network loss functions in Sec. 3.3.3. We use the code available at [12] in our experiments.

3.3.1 Shape and Texture Representations

The shape representation $\mathbf{S} \in \mathbb{R}^{3\times Q}$ is a set of Q vertices on the face surface. The shape decoder D_S is a Multi-Layer Perceptron (MLP) with the shape parameter $\mathbf{f}_S$ as input. The texture representation is an unwrapped 2D texture. Assuming that the face mesh has the top pointing up the y axis, the projection of a 3D vertex $\mathbf{v}_S = (x, y, z)$ onto the UV space, denoted as $\mathbf{v}_T = (u, v)$, is computed as:

$$v \rightarrow \alpha_1.\arctan\left(\frac{x}{z}\right) + \beta_1, \quad u \rightarrow \alpha_2.y + \beta_2, \tag{6}$$

where $\alpha_1, \alpha_2, \beta_1, \beta_2$ are scale constants and translation scalars to enclose the unwrapped face into the image boundaries. The texture decoder D_T is a deep network implemented by fractionally-strided convolution layers [16].

3.3.2 Making of Face Image Rendering Layer

The making of the face image rendering layer $\mathcal{R}(\mathbf{m}, \mathbf{S}, \mathbf{T})$ has three steps: In Step 1, the texture value of each vertex in $\mathbf{S}$ is determined by its predefined location in the 2D texture $\mathbf{T}$. In Step 2, the 3D shape/mesh $\mathbf{S}$ is projected onto the image plane by using the following weak perspective projection model:

$$g(\mathbf{m}) = \mathbf{V} = f * \mathbf{Pr} * \mathbf{R} * \mathbf{S} + \mathbf{t}_{2d} = M(\mathbf{m}) * \begin{bmatrix}\mathbf{S}\\ \mathbf{1}\end{bmatrix}, \tag{7}$$

where $g(\mathbf{m})$ gives the 2D positions of the 3D vertices, f is the scale factor, $\mathbf{Pr}$ is the orthographic projection matrix, $\mathbf{R}$ is the rotation matrix, and $\mathbf{t}_{2d}$ is the translation vector. While the projection matrix M has dimensions 24, it has six degrees of freedom, which is parameterized by a 6-dim vector $\mathbf{m}$.

In the last Step 3, the 3D mesh is rendered using a Z-buffer renderer, where each pixel is associated with a single triangle of the mesh, computed as follows

$$\hat{\mathbf{I}}(m, n) = \mathcal{R}(\mathbf{m}, \mathbf{S}, \mathbf{T})_{m,n} \quad = \sum_{\mathbf{v}_S \in \Phi(g,m,n)} \lambda \mathbf{T}_S(\mathbf{v}_S), \tag{8}$$

where $\Phi(g, m, n) = \{\mathbf{v}_S^{(1)}, \mathbf{v}_S^{(2)}, \mathbf{v}_S^{(3)}\}$ is an operation returning three vertices of the triangle that encloses the pixel (m, n) after applying the projection g. In order to handle occlusions, when a single pixel resides in more than one triangle, the triangle that is closest to the image plane is selected. The value of each pixel is determined by interpolating the intensity of the mesh vertices via barycentric coordinates $\{\lambda^{(i)}\}_{i=1}^{3}$.

3.3.3 Network Loss Function

See [24] for the architecture of the network. The network is end-to-end trainable to reconstruct the input images with the following loss function:

$$L = L_{\text{rec}} + \lambda_{\text{adv}} L_{\text{adv}} + \lambda_L L_{\text{L}}, \tag{9}$$

where the reconstruction loss $L_{\text{rec}} = \sum_{i=1}^{N} ||\hat{\mathbf{I}}_i - \mathbf{I}_i||_1$ makes the rendered image $\hat{\mathbf{I}}_i$ close to the input $\mathbf{I}_i$, the adversarial loss L_{adv} makes $\hat{\mathbf{I}}_i$ look real, and the landmark loss L_L makes $\hat{\mathbf{I}}_i$ obey the geometrical constraint. The networks that generate the rendered image $\hat{\mathbf{I}}_i$ form the generator. The discriminator D_A aims to distinguish $\hat{\mathbf{I}}_i$ from the real image $\mathbf{I}_i$. During training, the texture model D_T will be updated with the objective that $\hat{\mathbf{I}}_i$ is being classified as real by D_A. As the global structure of the face image has been handled by the aforementioned face rendering, the adversarial loss is computed on the textures of local facial regions by using the patchGAN [6] in D_A.

As the fully unsupervised training may lead to degenerate outcomes due to common undesired initialization, the pre-training loss (to be described next) is considered in the beginning phase of the training. The 3DMM shape parameter $\widetilde{\mathbf{S}}$ and projection parameter $\widetilde{\mathbf{m}}$ given in the linear 3DMM [32] are used to create the pseudo ground-truth texture $\widetilde{\mathbf{T}}$ by mapping the pixels in the UV space back to the input face image, and compute the pre-training loss. As the pre-training loss and the landmarks are the only supervision needed, the learning of the whole framework is considered as weakly supervised.

4. Experimental Evaluation

The experiments are designed to highlight the following: 1) The advantages of the real/fake discriminator with Wasserstein loss over that with softmax function with cross-entropy loss; 2) The advantages of the using augmented data for disentangled representation learning; 3) Comparison with other contemporary approaches. The advantages can be shown in terms of 1) the performance of face recognition across pose and 2) the visual quality of the generated (synthesized) face images.

We select the MPIE [8] and CASIA-WebFace for training, and the CFP (Celebrities in Frontal-Profile) database [20] and IJB-A for testing. The MPIE is one of the most popular in-the-house databases, and it contains more than 750,000 images of 337 people recorded in four sessions over the span of five months. Subjects were imaged under 13 view points and 19 illumination conditions while displaying 6 facial expressions. The view points make 13 poses across $-90^\circ \sim 90^\circ$ in yaw with 15° interval. The CASIA-WebFace offers 494,414 face images of 10,575 subjects taken *in the wild*, and a great majority of poses with $< 45^\circ$ in yaw. We removed 1103 subjects out of the CASIA-WebFace because of poor image quality and mislabeling. We use the Face Alignment Network (FAN) [2] to locate the landmarks of each face in the database, and exploit the nonlinear 3DMM [24], as summarized in Sec. 3.3,

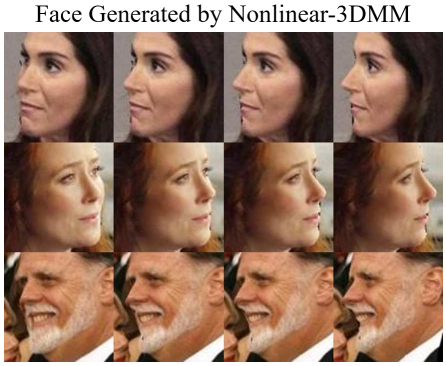

Figure 3. Faces with large orientations generated by nonlinear 3DMM with the original given on the left of each raw.

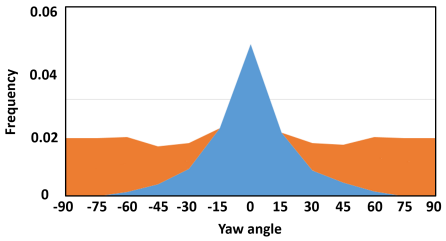

Figure 4. The blue shows the pose distribution of the CASIA-WebFace, and the orange shows the augmented segments generated by using nonlinear 3DMM.

to synthesize face images for data augmentation. We have generated 361,782 additional faces with $30^\circ \sim 90^\circ$ in yaw. Fig. 3 shows a few cases generated by the nonlinear 3DMM, and Fig. 4 shows the pose distribution before and after the data augmentation. Although the pose does not appear uniformly distributed after data augmentation, most of the missing pose segments are partially compensated. The CFP database is composed of 500 subject with 10 frontal images and 4 profile images per subject. The evaluation protocol contains 2 different phases including frontal to frontal(FF) and frontal to profile(FP) face verification, each having 350 intra pairs and 350 extra pairs. The IJBA contains images and videos of 500 subjects captured in the wild. The protocol contains identification (search) and verification (compare) for unconstrained face recognition.

All experiments were preformed with the following settings. All face images were aligned to a canonical view of 100×100 in size. Random sampling of 96×96 regions from the aligned face were cropped for augmenting the data. The image intensity was linearly scaled to the range of $[1, 1]$. All weights in the networks were initialized in a normal distribution with 0 mean and standard deviation 0.02. The Adam optimizer was set with a fixed learning rate 0.0001 and momentum 0.5. The batch size was set to be 64. All experiments were run with GTX 1080 Ti GPU and CUDA 8.0 with cuDNN6.0 on Pytorch.

Table 1. Performance on MPIE, Avg is the average rate for $0^\circ \sim 60^\circ$ and $(\cdot)$ is the average rate for $0^\circ \sim 90^\circ$

Method	0°	15°	30°	45°	60°	75°	90°	Avg
Zhu et al.[33]	95.7	92.8	83.7	72.9	60.1	-	-	79.3
Yim et al.[30]	99.5	95	88.5	79.9	61.9	-	-	83.3
DR-GAN[25]	97	94	90.1	86.2	83.2	-	-	89.2
Peng et al.[15]	-	97.2	96.6	95.6	92.7	85.7	74.9	(90.5)
DR-WGAN$_{LN}$	99.5	97.4	93.8	89.5	86.5	81.9	70.4	92.1 (87.4)
DR-WGAN$_{GN}$	99.5	98.2	95.7	93.3	89.6	84.8	72.2	94.3 (89.5)

In addition to the training with the MPIE and CASIA WebFace and testing on the CFP and IJB-A, we also conducted an experiment on MPIE, which offered samples with poses uniformly distributed (thus no data augmentation performed on MPIE). We selected the first 188 subjects for training and the rest 149 subjects for testing. The trained DR-WGAN was used to generate faces of all needed poses for each face from the gallery of the testing set. The gallery is composed of one image per subject with frontal view and neutral illumination, and the probe set contains the rest of images. The performance is shown in Table 1 along with other contemporary approaches. The proposed DR-WGAN is reported with two normalization settings: 1) DR-WGAN$_{LN}$ is the discriminator D with layer normalization (LN)[1] and the generator G with batch normalization (BN), and 2) DR-WGAN$_{GN}$ is both D and G with group normalization (GN). The DR-WGAN$_{GN}$ outperforms DR-WGAN$_{LN}$, demonstrating the better appropriateness of using GN for normalization. The DR-WGAN$_{LN}$ outperforms DR-GAN and other approaches, showing that the stabilized training induced by the Wasserstein GAN leads to better disentangled representation learning and improves the recognition performance. Given a profile face as input, the faces of other poses synthesized by the DR-WGAN$_{GN}$ are shown in Fig. 5, compared with the faces generated by the DR-GAN. The faces made by DR-WGAN$_{GN}$ look more similar to the ground truth, and better in shape, texture and overall image quality.

Fig. 6 shows the face synthesized by the DR-WGAN and DR-GAN. The faces synthesized by the DR-WGAN again appear better than those made by the DR-GAN in shape, texture and preserving the characteristics of the identities. The recognition performance of the DR-WGAN is given in Table. 2, with several different settings. The DR-WGAN$_{\mathbf{C}}$ refers to training on CASIA WebFace only, and the DR-WGAN$_{\mathbf{C},Aug}$ refers to training on the CASIA WebFace and the augmented data. As mentioned above, the CASIA WebFace is imbalanced in pose distribution, and the pose distribution before and after data augmentation is illustrated in Fig. 4. Table. 2 reveals that the DR-WGAN$_{\mathbf{C},Aug}$ outperforms DR-WGAN$_{\mathbf{C},Aug}$, especially for the Frontal-

[1]Note that the Wasserstein discriminator does not work with BN, and can work with LN or GN.

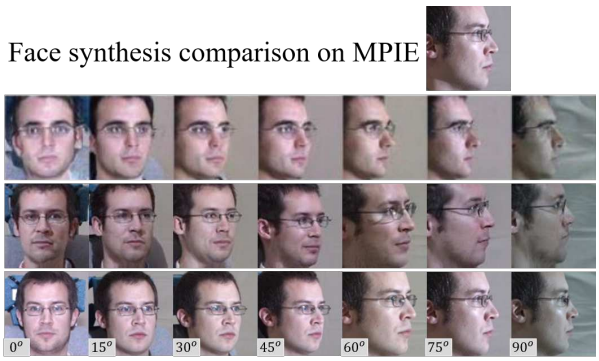

Figure 5. Given a profile face on the top, the top row shows the faces made by DR-GAN, the middle row shows the face made by DR-WGAN, and the bottom row shows the ground truth.

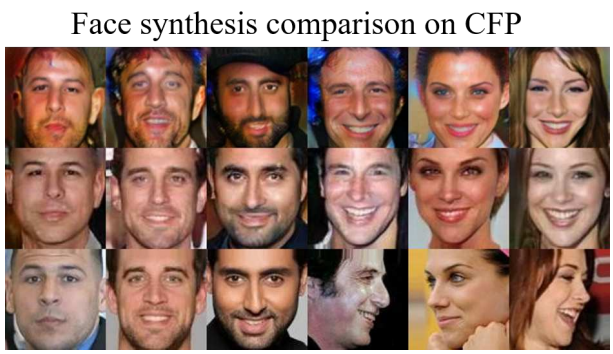

Figure 6. The bottom row gives the images from the dataset used as the inputs, the top row shows the faces made by DR-GAN and the middle row shows the face made by DR-WGAN. The left three are frontal-to-frontal, the right three are profile-to-frontal.

Table 2. Performance on CFP

Method	Frontal-Frontal	Frontal-Profile
Sengupta et al.[20]	96.40±0.69	84.91±1.82
Sankarana et al.[18]	96.93±0.61	89.17±2.35
Chen et al.[5]	98.67±0.36	91.97±1.70
DR-GAN[25]	97.13±0.62	90.82±0.28
Peng et al.[15]	98.67	93.76
Pal et al.[14]	98.11	91.70
DR-WGAN$_{\mathbf{C}}$	90.71±1.00	81.62±1.08
DR-WGAN$_{\mathbf{C},Aug}$	93.76±1.05	88.07±1.76
DR-WGAN$_{LN}$	98.07±1.01	91.67±1.03
DR-WGAN$_{LN,Aug}$	98.07±1.42	92.32±0.74
DR-WGAN$_{GN}$	98.64±0.97	92.87±1.07
DR-WGAN$_{GN,Aug}$	98.43±1.24	93.19±1.40

to-Profile performance, as the dataset before augmentation does not provide sufficient data with large orientations in yaw.

Upon the same training data as that reported in other works, i.e., CASIA WebFace and MPIE, we provide four versions in Table 2: the DR-WGAN$_{LN}$ is D with LN and G with BN; DR-WGAN$_{GN}$ is both G and D with GN;

Table 3. Performance comparison on IJB-A

Method	Verification		Identitfication	
	@FAR=.01	@FAR=.001	@Rank-1	@Rank-5
Wang et al.[26]	72.9±3.5	51.0±6.1	82.2±2.3	93.1±1.4
PAM [13]	73.3±1.8	55.2±3.2	77.1±1.6	88.7±0.9
DCNN [4]	78.7±4.3	-	85.2±1.8	93.7±1.0
DR-GAN [25]	77.4±2.7	53.9±4.3	85.5±1.5	94.7±1.1
Wu et al. [27]	98.7±0.1	93.9±0.9	97.7±0.3	99.0±0.1
DR-WGAN$_{GN}$	78.9±2.1	54.5±2.7	84.9±1.9	95.5±1.4
DR-WGAN$_{GN,Aug}$	80.4±2.2	57.9±3.8	87.6±1.0	96.3±0.9

the DR-WGAN$_{LN,Aug}$ and DR-WGAN$_{GN,Aug}$ are DR-WGAN$_{LN}$ and DR-WGAN$_{GN}$ trained on the data with augmented data added in. It can be seen that the DR-WGAN$_{LN}$ performs slightly better than DR-GAN, and it is slightly outperformed by the DR-WGAN$_{LN,Aug}$. This shows that although the DR-WGAN performs better than the DR-GAN in stabilizing the training, their performances are also affected by the training data. Because the MPIE offers a large set of data with extreme poses, the pose insufficiency of CASIA WebFace can be partially treated by adding in the MPIE to the training set. Nevertheless, the better training induced by the WGAN and the data augmentation by the nonlinear 3DMM are both verified to be able to improve the performance. In addition, the improvements made by the GN can also be verified by the performance for DR-WGAN$_{GN}$ and DR-WGAN$_{GN,Aug}$.

To demonstrate the contribution made by the data augmentation to face synthesis, Fig. 7 shows a comparison of the faces made by the DR-WGAN trained on the CASIA WebFace only and that trained with the augmented data added in. The faces with large yaw can hardly be generated by the DR-WGAN trained on the CASIA WebFace only, although the synthesized faces look well in preserving the identities. However, when using the DR-WGAN trained on the database with the augmented data added in, the poses can be well synthesized.

Table 3 shows the performance on the IJB-A. We only show the best performing DR-WGAN$_{GN}$ and DR-WGAN$_{GN,Aug}$, and compare with DR-GAN and other approaches. It again shows that the DR-WGAN$_{GN,Aug}$ outperforms all for its integration of the WGAN, the data augmentation and group normalization.

5. Conclusion

To address the issues of disentangled facial representation learning and better handle cross-pose face recognition and synthesis, we improve the state-of-the-art DR-GAN with three ingredients: embedding the Wasserstein loss to the discriminator, augmenting the training data by the nonlinear 3DMM and incorporation of the group normalization, and propose the DR-WGAN. Experiments show that the DR-WGAN framework is competitive to state-of-the-

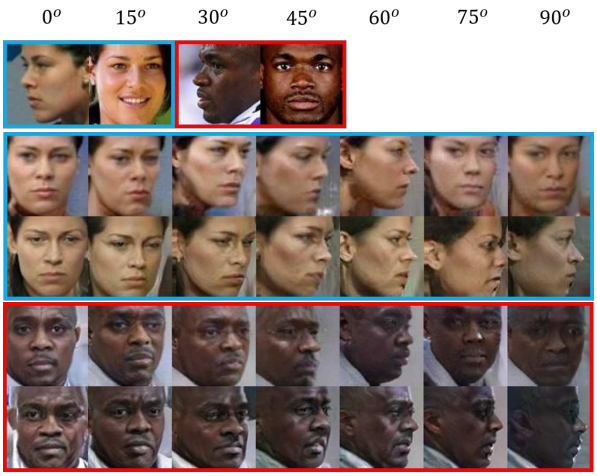

Figure 7. Face synthesis on pose augmented data. Given a frontal and a profile images of two subjects in the top row as input, the second and fourth rows show the faces synthesized by the DR-WGAN trained on CASIA WebFace only; the third and fifth rows show that faces synthesized by the DR-WGAN trained on the augmented CASIA WebFace. The augmentation is undertaken by the nonlinear 3DMM.

art approaches for handling cross-pose face recognition and face synthesis.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv*, 2017.

[2] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). *ICCV*, 2017.

[3] Jui-Shan Chan, Gee-Sern Jison Hsu, Hung-Cheng Shie, and Yan-Xiang Chen. Face recognition by facial attribute assisted network. *ICIP*, 2017.

[4] Jun-Cheng Chen, Vishal M Patel, and Rama Chellappa. Unconstrained face verification using deep cnn features. *WACV*, 2016.

[5] Jun-Cheng Chen, Jingxiao Zheng, Vishal M Patel, and Rama Chellappa. Fisher vector encoded deep convolutional features for unconstrained face verification. *ICIP*, 2016.

[6] Ugur Demir and Gozde Unal. Patch-based image inpainting with generative adversarial networks. *arXiv preprint arXiv:1803.07422*, 2018.

[7] Babak Esmaeili, Hongyi Huang, Byron C Wallace, and Jan-Willem van de Meent. Structured representations for reviews: Aspect-based variational hidden factor models. *arXiv preprint arXiv:1812.05035*, 2018.

[8] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *IVC*, 2010.

[9] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *NIPS*, 2017.

[10] Tal Hassner, Shai Harel, Eran Paz, and Roee Enbar. Effective face frontalization in unconstrained images. *CVPR*, 2015.

[11] Brendan F Klare, Ben Klein, Emma Taborsky, Austin Blanton, Jordan Cheney, Kristen Allen, Patrick Grother, Alan Mah, and Anil K Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. *CVPR*, 2015.

[12] Xiaoming Liu Luan Tran. Nonlinear 3D face morphable model github. `https://github.com/tranluan/Nonlinear_Face_3DMM`.

[13] Iacopo Masi, Stephen Rawls, Gérard Medioni, and Prem Natarajan. Pose-aware face recognition in the wild. *CVPR*, 2016.

[14] Dipan Pal, Chandrasekhar Bhagavatula, Yutong Zheng, Ran Tao, and Marios Savvides. Is pose really solved? a frontalization study on off-angle face matching. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2058–2067. IEEE, 2019.

[15] Xi Peng, Xiang Yu, Kihyuk Sohn, Dimitris N Metaxas, and Manmohan Chandraker. Reconstruction-based disentanglement for pose-invariant face recognition. *ICCV*, 2017.

[16] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[17] Christos Sagonas, Yannis Panagakis, Stefanos Zafeiriou, and Maja Pantic. Robust statistical face frontalization. *ICCV*, 2015.

[18] Swami Sankaranarayanan, Azadeh Alavi, Carlos Castillo, and Rama Chellappa. Triplet probabilistic embedding for face verification and clustering. *arXiv*, 2016.

[19] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *CVPR*, 2015.

[20] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. *WACV*, 2016.

[21] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv*, 2015.

[22] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. *NIPS*, 2014.

[23] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. *CVPR*, 2014.

[24] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *In Proceeding of IEEE Computer Vision and Pattern Recognition*, Salt Lake City, UT, June 2018.

[25] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. *CVPR*, 2017.

[26] Dayong Wang, Charles Otto, and Anil K Jain. Face search at scale. *TPAMI*, 2017.

[27] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 13(11):2884–2896, 2018.

[28] Yuxin Wu and Kaiming He. Group normalization. *arXiv*, 2018.

[29] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv*, 2014.

[30] Junho Yim, Heechul Jung, ByungIn Yoo, Changkyu Choi, Dusik Park, and Junmo Kim. Rotating your face using multi-task deep neural network. *CVPR*, 2015.

[31] Jian Zhao, Lin Xiong, Panasonic Karlekar Jayashree, Jianshu Li, Fang Zhao, Zhecan Wang, Panasonic Sugiri Pranata, Panasonic Shengmei Shen, Shuicheng Yan, and Jiashi Feng. Dual-agent gans for photorealistic and identity preserving profile face synthesis. In *Advances in Neural Information Processing Systems*, pages 66–76, 2017.

[32] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. Face alignment across large poses: A 3d solution. *CVPR*, 2016.

[33] Zhenyao Zhu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Multi-view perceptron: a deep model for learning face identity and view representations. *NIPS*, 2014.