Abstract

Reliable facial expression recognition plays a critical role in human-machine interactions. However, most of the facial expression analysis methodologies proposed to date pay little or no attention to the protection of a user’s privacy. In this paper, we propose a Privacy-Preserving Representation-Learning Variational Generative Adversarial Network (PPRL-VGAN) to learn an image representation that is explicitly disentangled from the identity information. At the same time, this representation is discriminative from the standpoint of facial expression recognition and generative as it allows expression-equivalent face image synthesis. We evaluate the proposed model on two public datasets under various threat scenarios. Quantitative and qualitative results demonstrate that our approach strikes a balance between the preservation of privacy and data utility. We further demonstrate that our model can be effectively applied to other tasks such as expression morphing and image completion.

1. Introduction

The recent proliferation of sensors in living spaces is propelling the development of “smart” rooms that can sense and interact with occupants to deliver a number of benefits such as improvements in energy efficiency, health outcomes, and productivity [11]. Automatic facial expression recognition is an important component of human-machine interaction. To date, a wide variety of methods have been proposed to accomplish this, however they typically rely on high-resolution images and ignore the visual privacy [24] of users. Growing privacy concerns will prove to be a major deterrent in the widespread adoption of camera-equipped smart rooms and the attainment of their concomitant benefits. Therefore, reliable and accurate privacy-preserving methodologies for facial expression recognition are needed.

One approach to increase visual privacy is to reduce identity traits within a face image via modification or redaction methods such as pixelization or blurring. However, this will also reduce the visual quality of the modified image and an algorithm’s ability to accurately recognize the facial expression from it. Another extreme approach is to withhold releasing the face image altogether and only release an estimate of the facial expression. While this approach guarantees visual privacy, it provides no visual utility. In order to strike a balance between privacy and data utility, we propose a third radically different approach: seamlessly replace the user-identity in an image without significantly degrading its visual quality or the ability to accurately infer facial expression. We leverage variational generative-adversarial networks (VGANs) to learn an identity-invariant representation of an image while enabling the synthesis of a utility-equivalent, realistic version of this image with a different identity (Fig. 1). We call this framework Privacy-Preserving Representation-Learning Variational Generative Adversarial Network (PPRL-VGAN). Beyond its application to privacy-preserving visual analytics, our approach could also be used to generate realistic avatars for animation and gaming.

Our proposed framework combines the generative power of two models: the Variational Auto-Encoder (VAE) [18] and the Generative Adversarial Network (GAN) [13].
VAE consists of two networks: the encoder, which maps a data sample to a latent representation, and the decoder, which maps this representation back to data space. VAE networks are trained by minimizing a cost function that encourages learning a latent representation which leads to realistic data synthesis while ensuring sufficient diversity in the synthesized data. Like a VAE, a GAN also consists of two networks: a generator network \( G \) which aims to synthesize realistic data from a random noise input vector and a discriminator network \( D \) which aims to differentiate between real and synthetic data. GANs are trained via a game between \( G \) and \( D \) in which \( G \) aims to fool \( D \) into believing that the data samples synthesized by it are realistic, and \( D \) which aims to accurately distinguish between real and “fake” samples. In this work, we combine VAEs with GANs by replacing the generator in a conventional GAN, which uses random noise as input, with a VAE encoder-decoder pair, which takes a real image as an input and outputs a synthesized image. As shown in Fig. 2, the encoder learns a mapping from a face image \( I \) to a latent representation \( f(I) \). The representation is subsequently fed into the decoder to synthesize a face image with some target identity (specified by identity code \( c \)) but with the same facial expression as the input image. The discriminator includes multiple classifiers that are trained to (i) distinguish real face images from synthesized ones, (ii) recognize the identity of the person in a face image and (iii) recognize the expression in a face image. During training, feedback signals from \( D \) guide \( G \) to create realistic expression-preserving face images. In addition, as the identity of the synthesized images is determined by the identity code \( c \), the network will learn to disentangle the identity-related information from the latent representation.

This paper makes the following contributions:
1. We propose a framework for learning an identity-invariant representation for a face image. This representation is discriminative for facial expression recognition and generative for expression-preserving, identity-altered face image synthesis.
2. We thoroughly evaluate our approach under three threat scenarios to demonstrate that our method strikes a balance between privacy and data utility.
3. We demonstrate that our model can synthesize new face images with or without an input image, and illustrate how our model can also be applied to other image processing tasks such as expression morphing and image completion.

2. Related Work

Privacy-Preserving Visual Analytics: There is a growing body of research on methods to perform various visual analysis tasks from data in a manner that does not disclose subject’s identity. According to how privacy is protected, the literature can be broadly classified as reversible and irreversible approaches [5].

Reversible methods include scrambling and encryption [12, 32, 35, 36] that permit exact data recovery, but are also prone to exposing the original data to possible hacks. In particular, methods for recognizing facial expression directly in the encrypted domain have been proposed [3, 28]. However, these methods rely upon public-key homomorphic cryptosystems, such as Paillier [25], which are known to be computationally heavy due to their use of large encryption and decryption keys. In order to relieve the computational burden, lightweight algorithms based on randomization techniques have been proposed in [29]. Although methods proposed in [3, 28, 29] perform well for facial expression recognition in the encrypted domain, no tests have been conducted to ascertain whether the identity information is indeed removed in the encrypted domain. It is unclear whether a classifier that is trained on encrypted-domain images will fail to recognize the identity of a person from the encrypted image.

Irreversible methods include image processing and filtering techniques [7, 8, 11, 15, 19, 26, 31]. However, it has been shown that simple filtering methods do not fool identity-recognition algorithms if they are trained using images that have the same distortion as the test images [23]. A face de-identification method was proposed in [16] wherein several face images with appearance attributes similar to the target image are fused by minimizing a cost function promoting attribute preservation and de-identification. A recent line of irreversible methods makes use of adversarial networks [6, 27, 30]. In [6], the focus is on full-body de-identification without an additional utility criterion such as accuracy of facial expression. Their methodology also relies upon a segmentation algorithm to accurately extract the silhouette of the person to be de-identified. Moreover, the synthesized images are blurry. While [30] uses adversarial networks to jointly optimize privacy and utility objectives, it focuses on the relatively simple task of detecting and removing a QR code embedded in an image. Moreover, the synthesized images are poor-quality renderings of the input image. The approach in [27] is similar in spirit to [30] but the output is not required to look realistic. Our approach differs from these methods in that we use a VAE within a GAN in order to explicitly learn an identity-invariant facial expression representation with the explicit goal of expression-preserving identity replacement in the synthesized output image which is required to look realistic. As we show, our learned representation is not only discriminative for expression recognition, but also robust to both human and algorithm-based privacy attacks. Our framework can also be used for other tasks such as expression morphing.

Disentangled Representation Learning: A number of models have been proposed in the literature to learn a so-
called “disentangled representation”. In early work, a bilinear model was proposed to separate content and style for face and text images [33]. An autoencoder (AE) augmented with simple regularization terms during training was proposed in [9] and demonstrated to discover and explicitly learn various latent factors of variation. Methods proposed in [17, 21] use VAEs in a semi-supervised manner. Their models disentangle label information from the latent representation by providing additional labels as input to the decoder. However, methods based on AE/VAE tend to produce blurry images due to the pixel-wise reconstruction error used in the loss function. Our model may be viewed as replacing the image reconstruction error with an adversarial loss to improve the visual quality of synthesized images. Recently, a two-stage pipeline was proposed [20] to learn disentangled image representations of background, foreground, and pose to generate novel person images. However, this method requires a pre-processing step to estimate a coarse pose mask of the input image.

Among works on disentangled representation learning, perhaps the closest to ours are those in [22, 34]. The approach proposed in [22] addresses the problem of disentanglement by combining a deep convolutional VAE with a form of adversarial training. It can disentangle the latent factors of variation within a labeled dataset, and separate them into complementary codes. However, it has not been tested on a real-world dataset. Our approach is different from that in [22] as we completely discard the VAE’s reconstruction error in the objective function. Instead, we employ the adversarial loss from a GAN for high-quality image synthesis and improved representation learning. In [34], a disentangled representation-learning GAN was proposed for pose-invariant face recognition. The proposed model is a fusion of an AE and a GAN. It explicitly disentangles the identity representation from pose variation by passing a pose code to the decoder during training. The major difference between this model and ours is that in PPRL-VGAN we use a VAE instead of an AE which permits learning a probability distribution over the latent space. This enables our model to synthesize new images without an input image; all we need to do is generate a latent vector from the prior distribution and pass it to the decoder along with an identity code.

3. Background Material

3.1. Variational Autoencoder Network

A VAE network consists of two neural networks: an encoder network (\(Enc\)) and a decoder network (\(Dec\)). The encoder is a randomized mapping of a data sample \(x\) to a latent representation \(z\) while the decoder is a randomized mapping \(z\) from a latent representation back to data space:

\[
z \sim Enc(x) = q(z|x) \tag{1}
\]

\[
\hat{x} \sim Dec(z) = p(x|z) \tag{2}
\]

In practice, these randomized mappings are implemented via deterministic maps (given by the neural networks) with additional inputs which provide the source of randomness. For example, it is common to set \(z = \mu_x + A_x w\) where the vector \(\mu_x\) and the square matrix \(A_x\) are the outputs of a neural network with input \(x\), and \(w \sim \mathcal{N}(0, I)\), a standard multivariate Gaussian, is the source of randomness. Then, \(q(z|x) = \mathcal{N}(\mu_x, A_xA_x^T)\). VAE networks are trained by minimizing a cost function which is additive over all training data samples. The cost function for a single data sample \(x\) is given by

\[
L_{VAE} = -E_{z \sim q(z|x)}[\log p(x|z)] + KL(q(z|x)||p(z)) \tag{3}
\]

where \(KL\) is the Kullback-Leibler divergence and \(p(z)\), the marginal distribution of the latent representation, is typically taken to be \(\mathcal{N}(0, I)\). The first term encourages the decoder to assign higher probability to the observed data samples \(x\). In practice, the expectation in the first term is replaced by an empirical average across a small batch of independent and identically distributed \(z\) for a given \(x\). The \(KL\) term encourages the encoder \(q(z|x)\) to be close to a target \(p(z)\) which has sufficient spread (diversity) in the latent space. The \(KL\) term has a closed analytic form since both its arguments are Gaussian [18]. The total cost across all data samples is typically minimized via mini-batch gradient descent.

3.2. Generative Adversarial Network

A standard GAN consists of a generator neural network \(G\) and a discriminator neural network \(D\) that are trained by making them compete in a two-player min-max game. The discriminator network \(D\) adjusts its weights so as to reliably distinguish real data samples \(x \sim p_d(x)\) from fake data samples \(G(z)\) generated by \(G\). The discriminator network \(D\) adjusts its weights to fool \(G\). The discriminator \(D\) assigns probability \(D(x) \in [0, 1]\) to the event that \(x\) is a “real” training data sample and the probability \(1 - D(x)\) to the event that \(x\) is a “fake” sample synthesized by the generator. The two networks are trained iteratively using a loss function given by

\[
L_{GAN}(G, D) = E_{x \sim p_d(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{4}
\]

with \(G\) aiming to minimize \(L_{GAN}(G, D)\) and \(D\) aiming to maximize it. In practice, the expectations are replaced by empirical averages over a mini-batch of samples and the loss function is alternately minimized and maximized from one mini-batch to the next as in mini-batch gradient descent.
Given a face image $I$ with an identity label $y_{id} = 1, \ldots, N_{id}$ and an expression label $y_e = 1, \ldots, N_e$, where $N_{id}$ and $N_e$ are the numbers of distinct subjects and facial expressions, respectively, the proposed model has two objectives: 1) to learn an identity-invariant face image representation $f(I)$ for facial expression recognition, and 2) to synthesize a realistic face image $\hat{I}$ with the same facial expression as in $I$ and target identity specified by a one-hot encoded identity code $c \in \{0, 1\}^{N_{id}}$.

**Discriminator:** Different from the discriminator network in a conventional GAN, the discriminator $D = (D^1, D^2, D^3)$ in PPRL-VGAN is a multi-task classifier consisting of three separate neural networks (Fig. 2): 1) the $D^1$ network classifies an input face image $I$ as real or synthetic, 2) the $D^2$ network estimates the identity of the person in the input face image, and 3) the $D^3$ network classifies the facial expression in the input face image. The weights of the networks in $D$ are trained to classify real face image inputs $I$ as real and accurately recognize the person’s identity and the facial expression. They are also trained to classify synthetic image inputs $\hat{I}$ as fake. This is accomplished by adjusting the network weights to maximize the following discriminator cost function:

$$
\mathcal{L}_D(G, D) = \lambda_1^D \left\{ E_{I \sim \text{data}(I),c \sim \text{p}(c)}[\log D^1(I)] + E_{I \sim \text{data}(I),c \sim \text{p}(c)}[\log(1 - D^1(G(I, c)))] \right\} + \\
E_{(I,y) \sim \text{data}(I,y)}[\lambda_2^D \log D^2_{y_{id}}(I) + \lambda_3^D \log D^3_{y_e}(I)] + \lambda_4^D \text{KL}(q(f(I) || p(f(I))))
$$

where $D^2_{y_{id}}$ and $D^3_{y_e}$ are the predicted probabilities of the $i$th class for identity and facial expression, respectively. The tuning parameter $\lambda_1^D, \lambda_2^D, \lambda_3^D$ and $\lambda_4^D$ control the relative importance between image quality, identity recognition, and expression recognition objectives.

**Generator:** In contrast to the generator in a conventional GAN which directly maps a “noise” vector to a synthesized image, the generator $G$ in a PPRL-VGAN maps a real input image $I$ with identity $y_{id}$ and expression $y_e$ to a synthesized output image $\hat{I} = G(I, c)$ with a target identity $y'(c)$ and the same expression $y_e$. This is accomplished via a VAE-like encoder-decoder structure. Specifically, the encoder aims to learn an image representation $f(I)$ from $I$ via a randomized mapping $f(I) \sim q(f(I) || I)$ parameterized by the weights of the encoder neural network. Similarly to a VAE, the cost function for training the generator includes $KL$ divergence between a prior distribution on the latent space $p(f(I)) \sim N(0, I)$ and the conditional distribution $q(f(I) || I)$. Training attempts to minimize this $KL$ term. The generator cost function also includes a term that encourages the decoder to learn to synthesize a face image $\hat{I} \sim p(I)f(I, c)$ that can fool $D$ into classifying it as a real face image having the same facial expression $y_e$ as the input image $I$, but with a target identity $y'(c)$ determined by $c$. Specifically, the generator network weights are adjusted during training to minimize the following generator cost function:

$$
\mathcal{L}_G(G, D) = E_{I(y) \sim \text{data}(I,y),c \sim \text{p}(c)}[\lambda_1^G \log \left(1 - D^1(G(I, c))\right) + \\
\lambda_2^G \log \left(1 - D^2_{y_{id}}(G(I, c))\right) + \lambda_3^G \log \left(1 - D^3_{y_e}(G(I, c))\right)] + \\
\lambda_4^G \text{KL}(q(f(I) || p(f(I))))
$$

where $\lambda_1^G, \lambda_2^G, \lambda_3^G$ and $\lambda_4^G$ are tuning parameters of the loss functions for $D^1, D^2, D^3$ and $KL$ divergence respectively. A key difference compared to the cost in Eq. 3 is that first term (reconstruction error) in Eq. 3 has been replaced with a perceptual loss term for the discriminator $D^1$ in Eq. 6.

Training alternates between maximizing Eq. 5 with respect to the weights of the networks in $D$ and minimizing Eq. 6 with respect to the weights of the networks in $G$. As the target identity code $c$ ranges over all $N_{id}$ distinct subjects, $N_{id}$ synthetic images $\hat{I}$ are produced for each training or test image $I$. As in the training of VAEs and GANs, the expectations are approximated by empirical averages computed from a mini-batch of training examples. Over successive training epochs, $G$ learns to fit the true data distribution and create a realistic face image that can fool $D^1$ having the same facial expression as the input image, which can be...
common to optimize the discriminator more frequently than $\lambda$. We found that the following values:

$\lambda = 0.25$, $\lambda^D_2 = 0.5$, $\lambda^D_3 = 0.25$ for discriminator training and $\lambda^G_1 = 0.108$, $\lambda^G_2 = 0.6$, $\lambda^G_3 = 0.29$, $\lambda^G_4 = 0.002$ for generator training work well. In conventional GANs, it is common to optimize the discriminator more frequently than the generator. However, we update the generator twice as frequently as the discriminator in training because the class labels used in PPRL-VGAN provide additional labeled data that help the discriminator training. The source code, additional implementation details and more experimental results are available on our project website [1].

5.3. Threat Scenarios

We evaluate privacy-preserving performance of the proposed PPRL-VGAN under three threat scenarios.

**Attack scenario I**: This is a simple scenario in which the attacker has access to the unaltered training set $(I_{\text{train}}, y^{id}_{\text{train}})$. However, the attacker’s test set consists of all images in the original test set after they have been passed through the trained PPRL-VGAN network. Thus, the attacker never gets to see the original test image $I_{\text{test}}$ but only its privacy-protected version $\hat{I}_{\text{test}}$. Also, the test set for the attacker contains all $N_{id}$ distinct privacy-protected versions $\hat{I}_{\text{test}}$ of each $I_{\text{test}}$ corresponding to $N_{id}$ distinct values of the identity code $c$.

**Attack scenario II**: This is a more challenging scenario (from the perspective of protecting privacy) where the attacker has access to the privacy-protected training images $\hat{I}_{\text{train}}$ and knows their underlying ground-truth identities $y^{id}_{\text{train}}$. Therefore, the attacker can train an identifier on training images that have the same type of identity-protecting transformation as the test images. If the proposed privacy-preserving transformation is weak and the identifier has sufficient learning capacity, it may be possible for a trained identifier to correctly predict the underlying ground-truth identity even from a privacy-protected test image. Similarly to scenario I, there are $N_{id}$ images for each training and testing image.

**Attack scenario III**: In this scenario, the attacker gets access to the encoder network and can obtain the latent representation $f(I)$ for any image $I$. Then, if the produced latent representation is not void of identity traits, the attacker can train an identifier using $(f(I_{\text{train}}), y^{id}_{\text{train}})$ and apply it to $f(I_{\text{test}})$ for identification. Although more challenging than scenario II, because the attacker can access the “more pristine” $f$, there are fewer training and test samples available since the identity code $c$ does not enter into the picture and thus there is no $N_{id}$-fold dataset expansion. Moreover whereas $\hat{I}$ resembles a real image, $f$ needs not (and typically does not).

In terms of utility, we train a dedicated facial expression classifier in each scenario with the available format of training data and the corresponding ground-truth expression labels. Then, we apply this classifier to test data and measure the facial expression recognition performance.
summarizes the performance of the proposed approach on the FERG and MUG datasets under a privacy-unconstrained scenario (training and testing sets are both unaltered), under a random-guessing attack and under the three attack scenarios described earlier. In each scenario, the identification and facial expression are estimated separately by different neural network classifiers.

**Table 1:** Architecture of PPRL-VGAN. ↓ and ↑ represent down- and upsampling operations, respectively. $D^1$, $D^2$ and $D^3$ share the weights of all convolutional layers and of the first fully-connected layer.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Encoder</th>
<th>Decoder</th>
<th>Discriminator</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$5 \times 5 \times 32$ conv. ↓ BNorm, LeakyReLU</td>
<td>2048 FC layers ↓ Reshape → $4 \times 4 \times 128$, LeakyReLU</td>
<td>$5 \times 5 \times 32$ conv. BNorm, LeakyReLU</td>
</tr>
<tr>
<td>2</td>
<td>$5 \times 5 \times 64$ conv. ↓ BNorm, LeakyReLU</td>
<td>$5 \times 5 \times 256$ deconv. ↑ BNorm, LeakyReLU</td>
<td>$5 \times 5 \times 64$ conv. BNorm, LeakyReLU</td>
</tr>
<tr>
<td>3</td>
<td>$5 \times 5 \times 128$ conv. ↓ BNorm, LeakyReLU</td>
<td>$5 \times 5 \times 128$ deconv. ↑ BNorm, LeakyReLU</td>
<td>$5 \times 5 \times 128$ conv. BNorm, LeakyReLU</td>
</tr>
<tr>
<td>4</td>
<td>$5 \times 5 \times 256$ conv. ↓ BNorm, LeakyReLU</td>
<td>$5 \times 5 \times 64$ deconv. ↑ BNorm, LeakyReLU</td>
<td>$5 \times 5 \times 256$ conv. BNorm, LeakyReLU</td>
</tr>
<tr>
<td>5</td>
<td>128 fully-connected (FC), Linear</td>
<td>$5 \times 5 \times 3$ deconv, tanh</td>
<td>256 fully-connected, LeakyReLU</td>
</tr>
<tr>
<td>6</td>
<td>28, 28, 5</td>
<td>256, 256, 256</td>
<td>$D^1$: 1 FC, $D^2$: Nid FC, $D^3$: N, FC</td>
</tr>
</tbody>
</table>

**5.4. Privacy Preservation versus Data Utility**

We first conduct a detailed evaluation of the proposed framework with respect to privacy preservation and data utility. We use correct classification rate (CCR) in person identification to measure how much privacy is preserved (the lower the CCR, the better) and also in facial expression recognition to measure the utility of data (the higher the CCR, the better). Table 2 summarizes the performance of the proposed approach on the FERG and MUG datasets under a privacy-unconstrained scenario (training and testing sets are both unaltered), under a random-guessing attack and under the three attack scenarios described earlier. In each scenario, the identification and facial expression are estimated separately by different neural network classifiers.

**Table 2:** Person identification and facial expression recognition performance in different scenarios on FERG and MUG datasets.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Identification (FERG)</th>
<th>Identification (MUG)</th>
<th>Expression Recognition (FERG)</th>
<th>Expression Recognition (MUG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privacy Unconstrained</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>87.90%</td>
</tr>
<tr>
<td>Random Guess</td>
<td>16.67%</td>
<td>12.50%</td>
<td>14.29%</td>
<td>14.29%</td>
</tr>
<tr>
<td>Attack Scenario I</td>
<td>17.01%</td>
<td>12.80%</td>
<td>93.02%</td>
<td>82.33%</td>
</tr>
<tr>
<td>Attack Scenario II</td>
<td>29.30%</td>
<td>22.08%</td>
<td>95.00%</td>
<td>85.14%</td>
</tr>
<tr>
<td>Attack Scenario III</td>
<td>22.42%</td>
<td>20.62%</td>
<td>100.00%</td>
<td>87.58%</td>
</tr>
</tbody>
</table>

For attack scenario I, we train an identifier using the original training set ($I_{train}$, $y_{train}$) and apply it to privacy-protected test images $\hat{I}_{test}$. The identifier has the same structure as $D^2$ (Fig. 2). We first observe that the identification CCRs are 17.01% for FERG and 12.80% for MUG. Both are close to a random guess (16.67% for FERG since there are 6 characters and 12.50% for MUG since we selected 8 subjects). However, the same classifier applied to the privacy-unconstrained test images results in 100% identification performance on both datasets. Such a huge performance gap confirms the proposed model effectively protects users’ privacy when the attacker has no information about the applied privacy-preserving transformation. For utility evaluation, we train a dedicated facial expression classifier, with the same structure as $D^3$, using ($I_{train}$, $y_{train}$) pairs and test it on $I_{test}$ images. The resulting expression recognition accuracies are 93.02% for FERG and 82.33% for MUG. These results are close to those achieved in the privacy-unconstrained scenario, which indicates that the synthesized images look realistic and retain the expression of the input images.

In attack scenario II, we use the privacy protected training data $\hat{I}_{train}$ and the corresponding ground-truth identity labels to train an identity recognizer and the ground-truth expressions to train a facial expression classifier (having the same architectures as in scenario I). We first observe that the identification accuracy in scenario II is about 11% higher than that of a random guess for both datasets, which suggests that some identity-related information is leaked into the synthesized images, but this is still much lower than in the privacy-unconstrained scenario. With respect to facial expression recognition, the performance in scenario II is consistently better than that in scenario I. This is likely because the number of training samples in scenario II is $N_{id}$ times that in scenario I, which benefits the training of the facial expression classifier.

In attack scenario III, we assume the attacker can access the latent representations of the training and probe images. We simulate this attack scenario by training an identifier using $(f(I_{train}), y_{train})$ and test it on $f(I_{test})$. However, as $f(I)$ is a 1-D vector, the 2-D ConvNet classifiers we used before are not suitable. We have experimented with 3 classifiers for $f(I)$, namely a Support Vector Machine (SVM), a customized 1-D ConvNet and a customized Artificial Neural Network (ANN). The customized ANN (3 hidden layers, each with 256 nodes) performed best in terms of identification and expression recognition accuracy. Therefore, only results for the customized ANN classifier are reported. As shown in Table 2, the identification performance is reduced in comparison with scenario II. However, the expression recognition performance in scenario III is the best among the three attack scenarios. Effectively, this suggests that the learned image representation $f(I)$ contains crucial facial expression information, but is largely disentangled from the identity information.

**Identity Replacement/Expression Transfer:** In addition to producing an identity-invariant image representation, PPRL-VGAN can be applied to an input face image of any identity to synthesize a realistic, expression-equivalent output face image of a target identity specified by the latent code $c$ (see Fig. 3). This may also be equivalently viewed as “transferring” an expression from one face to another. Unlike in a standard GAN, the synthesized image contains a lot of detail about the target identity due to the incorporation of the identifier $D^2$ and the expression classifier $D^3$. 

1688
5.5. Image Synthesis

Face Image Synthesis without Input Image: Once trained, our model can also synthesize face images without using an input image. This is due to the constraint we impose on the encoder which forces the distribution of the latent representation to follow a prior distribution (in our experiments: $f(I) \sim \mathcal{N}(0, I)$). To generate a new face image, we simply sample a latent vector from the prior distribution and concatenate it with an identity code. Then, we feed the concatenated vector into the decoder for image generation. As shown in Fig. 4, the synthesized images are realistic and the identities are consistent with the identity code. While the current model is incapable of controlling the facial expression of a generated image when no input image is given, we believe the synthesized images are useful for other applications, e.g., augmenting the original dataset.

Face Image Synthesis for Left-Out Expression: In order to further evaluate the generative capacity of PPRL-VGAN, we conducted experiments where we intentionally left out all samples of a specific facial expression $e$ from subject $i$ in training (images of expression $e$ from other subjects are still used) and then synthesized the left-out expression for subject $i$ after the model had been trained. This was done by feeding the generator $G$ an image with expression $e$ from subject $j, j \neq i$, and an identity code $c_i$ with $i$th entry equal to 1 and all other entries 0.

Figure 5 shows examples of left-out expression synthesis. While artifacts are clearly visible, the synthesized images capture the essential traits of a left-out expression, thus validating the generative capacity of PPRL-VGAN.

Expression Morphing: Facial expression morphing is a challenging problem because a human face is highly non-rigid and significantly deforms across expressions. Most methods perform face morphing in image space. Here, we leverage the latent representation and apply linear interpolation in latent space. Let $I_1, I_2$ be a pair of source images with different expressions for subject $i$ and $f(I_1), f(I_2)$ their corresponding latent representations. First, we linearly
interpolate $f(I_1)$ and $f(I_2)$ in the latent space to obtain a series of new representations $f(I_{\text{interp}})$ as follows:

$$f(I_{\text{interp}}) = (1 - \alpha)f(I_1) + \alpha f(I_2), \quad \alpha \in [0, 1] \quad (7)$$

Then, we feed $f(I_{\text{interp}})$ and identity code $c_i$ into the decoder to synthesize images. Figure 6 shows two examples of expression morphing. We can see that in both cases, the facial expression changes gradually from left to right. These smooth semantic changes indicate the model is able to capture salient expression characteristics in $f(I)$.

**Image completion:** PPRL-VGAN can be also applied to an image completion task. We tested two different masks (Fig. 7): one covering the eyebrows, eyes and nose, and the other covering the mouth (each mask occupies $\sim 7\%$ of the image). To complete the missing content of a query image $I_q$ of subject $j$, we first pass $I_q$ to the encoder to produce a latent representation $f(I_q)$. Then, we feed $f(I_q)$ and $c_i$ to the decoder for synthesizing a new image

$$I' \sim \text{Dec}(f(I_q), c_i).$$

Finally, we replace the missing pixel values of $I_q$ with values from corresponding locations in $I'$. Examples of both successful and unsuccessful image completions are shown in Fig. 7. Figure 7a shows examples for which our model was able to accurately estimate the missing image content. This demonstrates that our model learns correlations between different facial features, for example that opening the mouth is likely to appear jointly with raising eyebrows. However, our model occasionally fails (Fig. 7b). One possible reason for this is that some critical facial features (e.g., lowered eyebrows and narrowed eyes in the angry expression) are missing. A distortion may also occur when a face in the synthesized images is not accurately aligned with the one in the query image.

**6. Conclusion**

We presented a PPRL-VGAN for privacy-preserving facial expression recognition and face image synthesis. We proposed a novel architecture combining a VAE and a GAN to create an identity-invariant representation of a face image that also permits synthesis of an expression-preserving and realistic version. Experimental results on two public facial expression datasets demonstrate that our approach strikes a balance between privacy preservation and data utility. In addition, the proposed model can support a variety of applications like expression morphing and image completion. Generalizing the proposed framework to handle input images from unseen persons is part of our ongoing research.
References


