applied sciences

MDPI

Article

CLEANIR: Controllable Attribute-Preserving Natural Identity Remover

Durkhyun Cho, Jin Han Lee and Il Hong Suh *

The Department of Electronics and Computer Engineering, Hanyang University, Seoul 04763, Korea; chodurkhyun@hanyang.ac.kr (D.C.); jinhanlee@hanyang.ac.kr (J.H.L.)

* Correspondence: ihsuh@hanyang.ac.kr

Received: 27 December 2019; Accepted: 30 January 2020; Published: 7 February 2020

Abstract: We live in an era of privacy concerns. As smart devices such as smartphones, service robots and surveillance cameras spread, preservation of our privacy becomes one of the major concerns in our daily life. Traditionally, the problem was resolved by simple approaches such as image masking or blurring. While these provide effective ways to remove identities from images, there are certain limitations when it comes to a matter of recognition from the processed images. For example, one may want to get ambient information from scenes even when privacy-related information such as facial appearance is removed or changed. To address the issue, our goal in this paper is not only to modify identity from faces but also keeps facial attributes such as color, pose and facial expression for further applications. We propose a novel face de-identification method based on a deep generative model in which we design the output vector from an encoder to be disentangled into two parts: identity-related part and the rest representing facial attributes. We show that by solely modifying the identity-related part from the latent vector, our method effectively modifies the facial identity to a completely new one while the other attributes that are loosely related to personal identity are preserved. To validate the proposed method, we provide results from experiments that measure two different aspects: effectiveness of personal identity modification and facial attribute preservation.

Keywords: privacy preserving; face de-identification; generative model; variational auto-encoder

1. Introduction

In recent years, cameras are becoming widespread. Surveillance cameras are deployed in almost every public places (e.g., airports, streets, and buildings) and even in some private spaces such as smart houses and vehicles. Smartphones equipped with high-performance cameras increase the convenience of taking pictures or recording daily events anywhere, anytime. Consequently, in our daily lives, there is a huge number of images and videos processed and even shared through online social networking services. Furthermore, recent advances in computer vision and artificial intelligence technologies have enabled many image-based applications. Due to high computational burden, however, these techniques tend to require images to be uploaded to high-capacity servers on public networks, resulting inevitable vulnerability to attacks. Also, these deep learning-based techniques demand large amounts of data to properly train, but their collection is plagued by privacy concerns.

In the meantime, for many applications, facial attributes such as expression, gender, pose and gaze play important role and there are many research works dedicated to extracting these attributes. Many early works on face de-identification are based on deteriorating of the image by blurring [1], pixelization, image segmentation [2], downsampling [3], deletion of part of the face, or cartoonizing [4]. While these methods can effectively remove identity-related information from images, they also get rid of such useful facial attributes which are loosely related to personal identity, making difficult to adopt in further applications.

(a) Original (b) Blurred (c) Masked (d) Result from Ren et al. [5] (e) Ours ($\mathcal{I}^{90}$) (f) Ours ($\mathcal{I}^{180}$)

Figure 1. Comparison of various face de-identification methods: (**a**) input image, (**b**) blurred image, (**c**) masked image, (**d**) de-identified image by a generative adversarial networks-based method [5], (**e**,**f**) de-identification results from the proposed method by rotating the disentangled identity-related latent vector 90 and 180 degrees, respectively. As it can be seen from the figures, the proposed method results in more natural facial images while the identity is effectively modified.

To address the issue, we propose a novel de-identification method based on a deep generative model that effectively modifies facial appearance while keeping the useful attributes. There are two main streams in deep generative models: variational auto-encoder (VAE)-based models and generative adversarial networks (GAN)-based models. GAN-based models are composed of two competing networks, a generator that outputs desired samples given random vectors and a discriminator that distinguishes between generated and real samples. Therefore, GAN-based models show impressive results while it is hard to converge, sometimes generating unnatural results. Also, in GAN-based methods, because the generator has a random vector as an input, it is difficult to design explicit relation from features to outputs. VAE-based models are usually constructed in an encoder–decoder structure as in Auto-Encoders. However, unlike Auto-Encoders that directly output latent vectors, the encoder in VAE-based methods outputs parameters for probabilistic distribution of latent feature space and use them to sample a latent vector from the distribution. Therefore, because VAE-based methods learn distribution from training samples, they tend to generate more natural samples. Also, the decoders in VAE-based methods take latent vectors from encoders, we can design explicit guidance between features and outputs inside the network.

Regarding preservation of facial attributes while modifying personal identity, some research works use additional attribute classifiers or facial landmarks detector [5–9]. These methods can successfully preserve some facial attributes. However, they only focus on the specific facial attributes and depend on additional algorithms such as facial landmark detector, facial expression estimator, or action detector.

Another approach is to use face swapping methods for face de-identification [10–13]. These methods produce more realistic, attribute-preserved outputs. However, the outputs are not images of new person. Rather, the result is a mapping from one to another which also exists in the training dataset. Therefore, it is hard to consider as a privacy-preserving technique.

In this work, we do not try to estimate facial attributes to preserve them. Instead, we aim to extract identity-related vector, and by modifying solely this vector we show that the proposed method can effectively change appearance of face while preserving rest of facial attributes that are loosely related to personal identity. Also, we show that with simple transformations we can control the amount of de-identification. As shown in Figure 1, the proposed method generates more natural faces than the other existing methods. Our method can be applied to smart devices such as service robots, surveillance cameras or smartphone applications for social networking service in which people does not want to be invaded their privacy. Because the applications should be able to analyze captured images to provide useful information, it is desirable only to remove privacy-sensitive information from images.

The contribution of this work can be summarized as follows:

- A network architecture that explicitly disentangle latent vector to parts of personal identity and facial attributes

- An end-to-end scheme that can effectively change appearance of faces while keeping important attributes
- Exhaustive experiments carried thoroughly to validate the proposed method

2. Related Work

2.1. Deep Generative Model

Variational Auto-Encoders (VAE) [14] and Generative Adversarial Networks (GAN) [15] are representative deep learning-based generative models that are able to tackle intractable probabilistic distribution and large datasets. Similar to Auto-Encoders (AE), VAEs are usually composed of two parts, an encoder and a decoder, in which encoders in VAEs are responsible for capturing the probabilistic distribution of latent features while encoders in AEs are designed to directly output latent features. Therefore, it is well known that VAEs are effective in modelling latent probability distributions. Several works [16–18] have shown how VAEs can be used to learn structured, disentangled and interpretable representations in the latent space. However, outputs from VAEs tend to be blurry. GAN and its variations [19–21] are the most popular generative network recently. They alternately train a generative model to create samples and a discriminative model to distinguish between real and fake samples. Compared to VAE-based models, GAN-based models generate high-quality and realistic images while it is harder to converge and output inconsistent samples in some cases. Furthermore, their inputs for the generative model are meaningless random noise, thus difficult to manipulate.

VAE and GAN have also been applied to perform conditional generation of samples. Based on Conditional Variational Auto-Encoders (CVAE) [22] or Conditional Generative Adversarial Networks (CGAN) [23], there are works performing interesting tasks [24–27]. Odena et al. [24] proposed an image synthesis model which conditionally generates samples for 1000 classes. Reed et al. [26] demonstrated to generate images from text descriptions. Yan et al. [25] showed a conditioned image generation from visual attributes using CVAE. Walker et al. [27] proposed a model to generate possible future trajectories conditioned on the present image. Most recently, for person re-identification from images, Zheng et al. [28] proposed a GAN-based architecture containing two distinct encoders resulting an appearance-related latent vector and a structure-related latent vector, respectively. With these latent vectors, similar to our work, they manipulate appearance and structure from input images to generate new pedestrian images. Additionally, in a work from Larsen et al. [29], the authors proposed the combination method of VAE and GAN. In [30], Conditional VAE-GAN for data augmentation and image inpainting is proposed. They show impressive results but also suffer from aforementioned problems in GAN.

2.2. Face Swapping

Face swapping or face replacement is the task of transferring a face from source to target image. Early works of face swapping are based on 3D Morphable Model (3DMM) [31,32]. A drawback is that these methods only works properly when there is a large number of images of the target subject and the source subject because a 3D Model must be first built. Furthermore, estimation of 3D geometries along with different lighting conditions using 3DMM is still difficult.

As the result of the success of deep learning, many deep learning-based methods are emerging. In [33], the authors proposed new face swapping method as a style transfer task. They consider facial attributes and identity as a style. In [34], the authors proposed a method to work in more challenging conditions. They used convolutional neural network for blending technique. Region-Separative GAN (RS-GAN) [35] uses an approach that swaps in the latent space by disentangling the latent representations. FSNet [36] uses the latent space which separates identity and geometric components. Face Swapping GAN (FSGAN) [37] uses subject-agnostic method. In other words, their method does not need person-specific training.

Although some of the face swapping works have been proposed due to privacy concerns [10–13,38–40] and their techniques are similar to face de-identification, in the sense of privacy preservation, they do not adequately protect privacy of the person on the other side because this technique just transforms one face to the target face. Therefore, it is important to get explicit consent from owners of target facial images to use them for users' facial de-identification. For this reason, these methods are more suitable for recreation or entertainment purposes.

2.3. Face De-Identification

Earlier works on face de-identification had simply used blurring [1], downsampling, masking, or pixelation [41]. Although these methods had been easily applicable and removing privacy-sensitive information successfully, it had deleted other useful information. To solve this problem, k-Same family motivated by k-Anonymity [42] have been proposed. Vanilla k-Same method [43] created a new face by averaging k-closest faces of a gallery. It normally suffered from ghosting artifacts in the result images. k-Same-Select method [44] aimed at preserving facial attributes. To do that, this method partitions a gallery into mutually exclusive subsets. k-Same-M method [45] tried to avoid the undesirable artifacts due to misalignment. This method used Active Appearance Models (AAM) [46] for alignment. In [47,48], the authors also used the k-Same family method. These methods used the AAM and facial attribute classifiers to keep facial attributes. Problem of the k-Same family methods is lack of generalization. These methods need a large and various set of faces and simultaneously each subject should be only represented once in the set. In addition, a method cannot include all kind of the facial attribute classifiers, and the AAM also have a generalization problem.

Emerging approaches are using deep learning-based generative models [5–9]. These methods have produced higher quality images thanks to deep generative models. However, GAN-based methods [5,7–9] have sometimes generated awkward facial images and cannot manipulate the amount of de-identification. In addition, randomly generated facial images may result in looking similar to the original one. In [6], the authors proposed a method using VAE with GAN. This method can control de-identification by using conditional vector for identity. However, because this control vector is the one-hot encoded vector, the range of the de-identification is limited in the training set. In [8], the authors proposed a method to preserve facial pose by using facial landmark detector to generate a new random face with the same pose. However, due to the aforementioned limitation in GAN-based methods, this method also would not assure that the generated random face is different from the input face.

For preserving attributes, many of those methods have focused on one or two attributes explicitly and use additional classifier(s) to do this. In [5], the authors focused only on preserving action and use action detector. The method of [6] preserved facial expression and use facial expression estimator. The authors of [7] tried to preserve structural similarity index of image, i.e., luminance, contrast and structural differences. The method of [9] had a bit different perspective. The authors viewed identity as a combination of facial attributes. They used 40 classifiers to predict facial attributes and selected facial attributes to preserve based on their criterion for protecting privacy. Finally, based on preserved attributes, it generated a new face. Leaving the computational power for running 40 classifiers, this perspective cannot meet our objectives to preserve useful information of an original face.

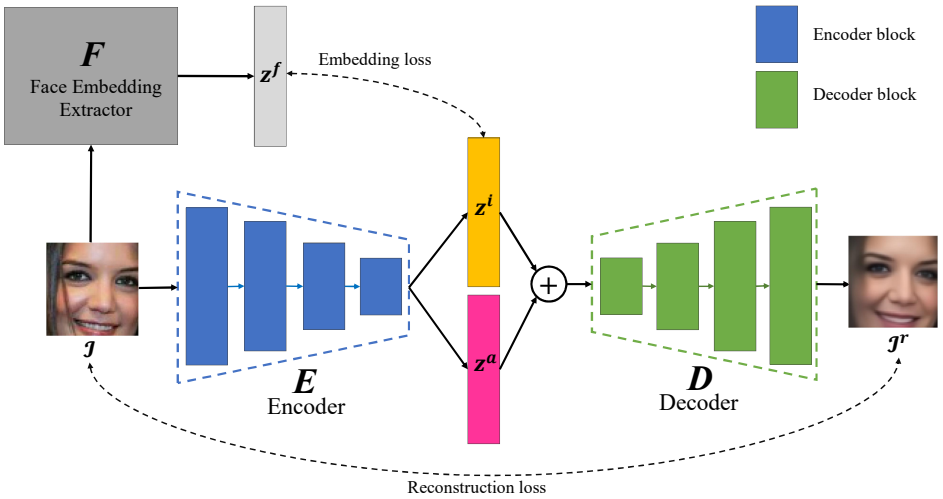
3. Proposed Method

The process of the proposed method differs in training and testing phases. We provide in detail the proposed network architecture, then describe the training and testing process in the following subsections.

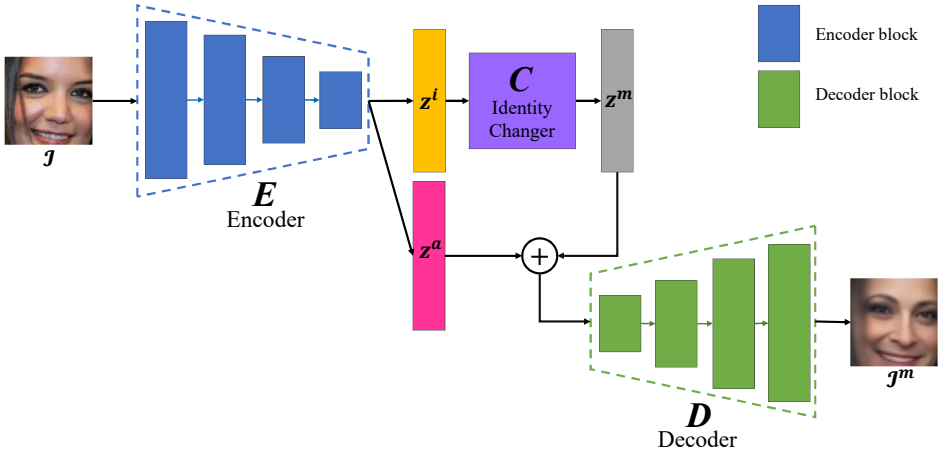
3.1. Network Architecture

As it can be seen from Figure 2, the proposed network adopts the VAE architecture with skip connections [49]. Using the VAE architecture, the proposed method can learn to have latent

space organized, enabling the encoded feature vector to be split into two parts: identity-related and attributes-related parts. Furthermore, benefited from skip connections, the network is able to generate new faces which do not exist in training images (i.e., not merely transforms one face to another which also exists in the training dataset) with high quality.



(a) Training process



(b) Testing process

Figure 2. Overview of the proposed method: (**a**) an overview of the training process and (**b**) an overview of the testing process, where m is degrees of identity modification.

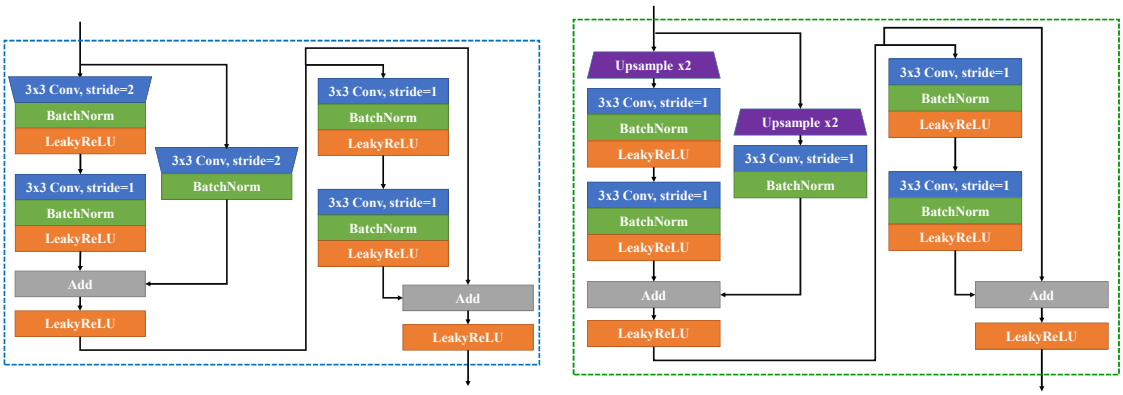
The encoder network contains four blocks using skip connections as shown in Figure 3a. In this work, we use facial images in a shape of $64 \times 64 \times 3$ as input for the encoder and the output latent vector z is 1024-dimensional. With z, we treat the first 512 dimensions as an identity-related vector z^i and the rest 512 dimensions as an attributes-related vector z^a. As depicted in Figure 3b, the decoder network also contains four blocks using skip connections. z is directly input to the decoder network and the shape of the output is the same as the input image, $64 \times 64 \times 3$. Detailed architecture of the proposed network is described in Table 1.

Table 1. Architecture of the proposed network. Where LReLU is leaky ReLU, BN is batch normalization, FC is fully connected layer, '3x3 Conv' is a convolution of which filter size is 3 by 3, and, '3x3 AvgPool' is an average pooling of which filter size is 3 by 3.

Name	Operations	Input	Output Size
Enc0	7x7 Conv(stride=2)-BN-LReLU	Input image	$32 \times 32 \times 64$
EncBlock1_1	3x3 Conv(stride=2)-BN-LReLU- 3x3 Conv(stride=1)-BN-LReLU	Enc0	$16 \times 16 \times 64$
EncBlock1_2	3x3 Conv(stride=2)-BN	Enc0	$16 \times 16 \times 64$
EncBlock1_3	Add-LReLU	EncBlock1_1, EncBlock1_2	$16 \times 16 \times 64$
EncBlock1_4	{ 3x3 Conv(stride=1)-BN-LReLU } x2	EncBlock1_3	$16 \times 16 \times 64$
EncBlock1_5	Add-LReLU	EncBlock1_3, EncBlock1_4	$16 \times 16 \times 64$
EncBlock2_1	3x3 Conv(stride=2)-BN-LReLU- 3x3 Conv(stride=1)-BN-LReLU	EncBlock1_5	$8 \times 8 \times 128$
EncBlock2_2	3x3 Conv(stride=2)-BN	EncBlock1_5	$8 \times 8 \times 128$
EncBlock2_3	Add-LReLU	EncBlock2_1, EncBlock2_2	$8 \times 8 \times 128$
EncBlock2_4	{ 3x3 Conv(stride=1)-BN-LReLU } x2	EncBlock2_3	$8 \times 8 \times 128$
EncBlock2_5	Add-LReLU	EncBlock2_3, EncBlock2_4	$8 \times 8 \times 128$
EncBlock3_1	3x3 Conv(stride=2)-BN-LReLU- 3x3 Conv(stride=1)-BN-LReLU	EncBlock2_5	$4 \times 4 \times 192$
EncBlock3_2	3x3 Conv(stride=2)-BN	EncBlock2_5	$4 \times 4 \times 192$
EncBlock3_3	Add-LReLU	EncBlock3_1, EncBlock3_2	$4 \times 4 \times 192$
EncBlock3_4	{ 3x3 Conv(stride=1)-BN-LReLU } x2	EncBlock3_3	$4 \times 4 \times 192$
EncBlock3_5	Add-LReLU	EncBlock3_3, EncBlock3_4	$4 \times 4 \times 192$
EncBlock4_1	3x3 Conv(stride=2)-BN-LReLU- 3x3 Conv(stride=1)-BN-LReLU	EncBlock3_5	$2 \times 2 \times 256$
EncBlock4_2	3x3 Conv(stride=2)-BN	EncBlock3_5	$2 \times 2 \times 256$
EncBlock4_3	Add-LReLU	EncBlock4_1, EncBlock4_2	$2 \times 2 \times 256$
EncBlock4_4	{ 3x3 Conv(stride=1)-BN-LReLU } x2	EncBlock4_3	$2 \times 2 \times 256$
EncBlock4_5	Add-LReLU	EncBlock4_3, EncBlock4_4	$2 \times 2 \times 256$
EncL	3x3 AvgPool-FC-LReLU-BN	EncBlock4_5	1×1024
Dec0	FC-Reshape-LReLU	EncL	$4 \times 4 \times 512$
DecBlock1_1	Upsample- { 3x3 Conv(stride=1)-BN-LReLU } x2	Dec0	$8 \times 8 \times 256$
DecBlock1_2	Upsample-3x3 Conv(stride=1)-BN	Dec0	$8 \times 8 \times 256$
DecBlock1_3	Add-LReLU	DecBlock1_1, DecBlock1_2	$8 \times 8 \times 256$
DecBlock1_4	{ 3x3 Conv(stride=1)-BN-LReLU } x2	DecBlock1_3	$8 \times 8 \times 256$
DecBlock1_5	Add-LReLU	DecBlock1_3, DecBlock1_4	$8 \times 8 \times 256$
DecBlock2_1	Upsample- { 3x3 Conv(stride=1)-BN-LReLU } x2	DecBlock1_5	$16 \times 16 \times 128$
DecBlock2_2	Upsample-3x3 Conv(stride=1)-BN	DecBlock1_5	$16 \times 16 \times 128$
DecBlock2_3	Add-LReLU	DecBlock2_1, DecBlock2_2	$16 \times 16 \times 128$
DecBlock2_4	{ 3x3 Conv(stride=1)-BN-LReLU } x2	DecBlock2_3	$16 \times 16 \times 128$
DecBlock2_5	Add-LReLU	DecBlock2_3, DecBlock2_4	$16 \times 16 \times 128$

Table 1. *Cont.*

Name	Operations	Input	Output Size
DecBlock3_1	Upsample- { 3x3 Conv(stride=1)-BN-LReLU } x2	DecBlock2_5	32 × 32 × 64
DecBlock3_2	Upsample-3x3 Conv(stride=1)-BN	DecBlock2_5	32 × 32 × 64
DecBlock3_3	Add-LReLU	DecBlock3_1, DecBlock3_2	32 × 32 × 64
DecBlock3_4	{ 3x3 Conv(stride=1)-BN-LReLU } x2	DecBlock3_3	32 × 32 × 64
DecBlock3_5	Add-LReLU	DecBlock3_3, DecBlock3_4	32 × 32 × 64
DecBlock4_1	Upsample- { 3x3 Conv(stride=1)-BN-LReLU } x2	DecBlock3_5	64 × 64 × 32
DecBlock4_2	Upsample-3x3 Conv(stride=1)-BN	DecBlock3_5	64 × 64 × 32
DecBlock4_3	Add-LReLU	DecBlock4_1, DecBlock4_2	64 × 64 × 32
DecBlock4_4	{ 3x3 Conv(stride=1)-BN-LReLU } x2	DecBlock4_3	64 × 64 × 32
DecBlock4_5	Add-LReLU	DecBlock4_3, DecBlock4_4	64 × 64 × 32
DecL	LReLU-3x3 Conv(stride=1)	DecBlock4_5	64 × 64 × 3



(a) Architecture of an encoder (E) block. (b) Architecture of a decoder (D) block.

Figure 3. Architectures of blocks of the encoder and the decoder.

3.2. Training Process

An overview of the training process is shown in Figure 2a. The encoder network E maps a facial image $\mathcal{I}$ to z which consists of identity-related vector z^i and attribute-related vector z^a. Then, the decoder network D generates a reconstruction image $\mathcal{I}^r$ from z. As is common in encoder–decoder architectures based on VAE, we also use binary cross entropy (BCE) loss to measure reconstruction error as well as Kullback–Leibler (KL) loss for regularization. Since in an image, pixel intensity values follow a conditional probability distribution, we assume that the values can be interpreted as probabilities for pixels being on/off after the values are scaled to [0, 1]. Therefore, the BCE loss can be adopted to our formulation. The BCE loss function results in the minimum loss when the value of a pixel on the input image $\mathcal{I}_j(x,y)$ and the value of the corresponding pixel on the reconstructed image $\mathcal{I}_j^r(x,y)$ are the same. We define the BCE loss $\mathcal{L}_r$ and the KL loss $\mathcal{L}_{kl}$ as follows,

$$\mathcal{L}_r = \sum_{j=1}^{N} \left(- \sum_{x=1}^{W} \sum_{y=1}^{H} \mathcal{I}_j(x,y) log(\mathcal{I}_j^r(x,y)) \right) \tag{1}$$

where N is the number of samples, W and H are width and height of the image, respectively.

$$\mathcal{L}_{kl} = \sum_{j=1}^{N} KL(q(z|\mathcal{I}_j) \| \mathcal{N}(0, I)) \tag{2}$$

where q is the encoder network.

Our key idea in this work is to design the latent feature vector resulting from the encoder to have disentangled into identity-related part and the rest facial attribute-related part, enabling effective identity modification by solely manipulating the identity-related part from the latent vector. To train the network to result such disentanglement, we present an embedding loss using an external facial embedding extractor F. Since, we employ the F which is pre-trained for face recognition and verification, we assume that it is well trained to provide plenty distinctive features for facial identity. Therefore, by transforming a point on the identity space defined by F, we expect that identities of given facial images can be transformed with ease by producing new facial images. In this work, we use a Keras implementation [50] of FaceNet [51] as the face embedding extractor F. The network architecture is based on the Inception-Resnet-v1 [52] and the model was trained on VGGFace2 dataset [53] using a triplet loss.

To make z^i resulted from the proposed network get closer to the output of F, z^f, we design the embedding loss function using cosine distance as follows:

$$\mathcal{L}_e = \sum_{j=1}^{N} \left(1 - \frac{z_j^f \cdot z_j^i}{\|z_j^f\|_2 \|z_j^i\|_2} \right). \tag{3}$$

For a sample j, the cosine distance between z_j^f and z_j^i ranges in [0, 2], and the loss is the sum of the distances of N samples in a batch.

Finally, our training loss is defined as the sum of the loss functions with control parameters λ_r, λ_{kl} and λ_e:

$$\mathcal{L} = \lambda_r \mathcal{L}_r + \lambda_{kl} \mathcal{L}_{kl} + \lambda_e \mathcal{L}_e, \tag{4}$$

where by the embedding loss $\mathcal{L}_e$, the latent space related to facial identities is trained while non-identity latent space is also be learned due to the reconstruction loss $\mathcal{L}_r$ because it demands the rest information to reconstruct the input image properly.

3.3. Testing Process

An overview of the testing process is shown in Figure 2b. In the testing, it does not require the face embedding extractor F, but an identity changer C to transform z^i to z^m which is a new identity-related vector. For the transformation, we first L2-normalize the identity-related vector z^i. Then, we adopt the well-known Gram–Schmidt process to rotate z^i 90 degrees with following equation,

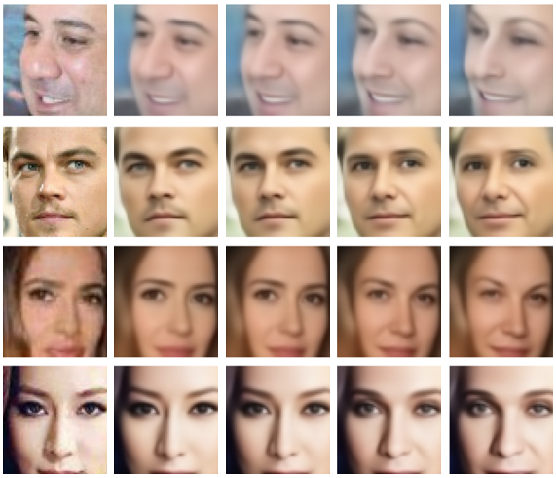
$$z^{90} = \frac{z^r - proj_{z^i}(z^r)}{||z^r - proj_{z^i}(z^r)||}, \tag{5}$$

where $z^r \in R^{512}$ is a random vector which determines the rotational axis and $proj_b(a)$ denotes a function which projects the vector a onto b. Finally, with z^0 (*i.e.*, z^i) and z^{90}, we can get a modified identity-related vector z^m for arbitrary degrees with following equation:

$$z^m = z^0 \cos m + z^{90} \sin m. \tag{6}$$

With the concatenated vector of z^a and z^m, the decoder generates a facial de-identified image as shown in Figure 2b. Therefore, by modifying z^i we can change the identity from the input image. In particular, the proposed method can generate new image of a person who does not exist because a condition vector of the decoder z^m is not an one-hot encoded vector but the face embedding that

it contains rich information of a face as aforementioned. Figure 4 shows the facial de-identification examples from the proposed method in which $\mathcal{I}^{\{0,60,120,180\}}$ denote transformation results rotated on a hyper-plane with 0, 60, 120 and 180 degrees, respectively. As it can be seen from the figure, the proposed network gradually modifies identities from given facial images as the rotation degree increases.



(a) $\mathcal{I}$ (b) $\mathcal{I}^0$ (c) $\mathcal{I}^{60}$ (d) $\mathcal{I}^{120}$ (e) $\mathcal{I}^{180}$

Figure 4. Facial de-identification examples from the proposed method. From left to right: input facial images and transformation results rotated on a hyper-plane by $\{0, 60, 120, 180\}$ degrees, respectively.

4. Experiments

To validate the effectiveness of the proposed method, we conduct two types of experiments. The first experiment is to show how well our method can remove identity from a facial image while the second one is to confirm how well it preserves facial attributes.

4.1. Experimental Setup

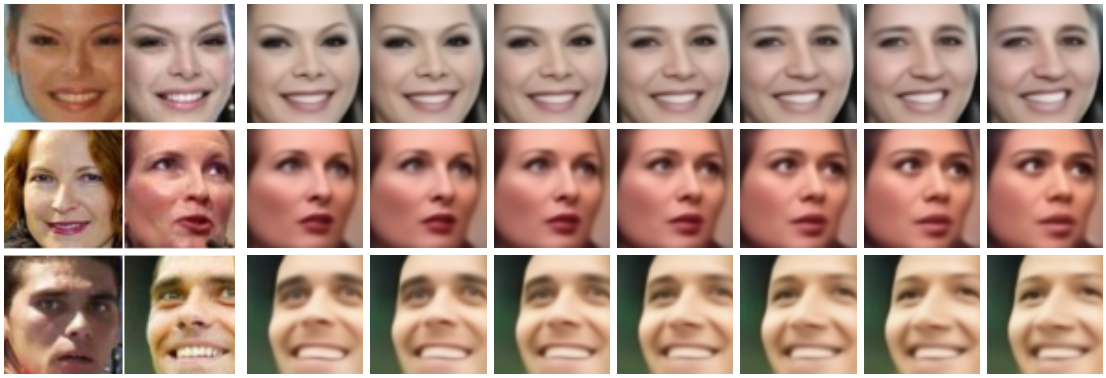
We implement the proposed system using TensorFlow and conducted all of the experiments in this work with a workstation equipped with four Nvidia GeForce RTX 2080 ti GPUs, Intel(R) Core(TM) i9-9900X CPU (3.50 GHz), 128 GB of RAM. For face part detection from given images, we employ a method from King et al. [54].

To train the proposed model, we use VGGFace2 [53], one of the major large-scale datasets for face recognition. The images in the dataset have significant variations in pose, age, illumination, ethnicity and profession, amounting 3.31 million images from 9131 identities. Since the official test split by Cao et al. [53] (167,559 images from 500 identities) contains evenly sampled images from the whole dataset, we use the split as our training set in this work. We train our network with Adam [55] with $\beta_1 = 0.9, \beta_2 = 0.999$, starting learning rate 0.0001 with a time-based decay of 10^{-6}, for 300 epochs with batch size of 32 taking about 30 h on our system.

4.2. Evaluation on De-Identification

To validate facial de-identification performance of the proposed method, we adopt the state-of-the-art face verification method FaceNet [51] which provides a similarity distance given two facial images. Given two images of the same person, we measure how the proposed method can separate them as the transformation degree in M varies. As the test set, we use a widely adopted public benchmark dataset, Labeled Faces in the Wild Deep Funneled dataset (LFW) [56–58] which provides matched and mismatched facial image pairs of 1680 people. From the dataset, for each of 3000 matched pairs, we de-identify an image from a pair using the proposed method, then compute a similarity distance between the de-identified image and the other in the pair. Finally, if the resulting

distance is lower than a threshold (η), we count the sample as a matching pair. Example images for evaluation on de-identification are shown in Figure 5.



(a) Matched pairs (b) $\mathcal{I}^0$ (c) $\mathcal{I}^{30}$ (d) $\mathcal{I}^{60}$ (e) $\mathcal{I}^{90}$ (f) $\mathcal{I}^{120}$ (g) $\mathcal{I}^{150}$ (h) $\mathcal{I}^{180}$

Figure 5. Examples for evaluation on de-identification. (**a**) shows matched pairs from LFW [58], (**b**–**h**) show de-identified images of the right image on each matched pair rotated by {0, 30, 60, 90, 120, 150, 180} degrees, respectively.

Table 2 provides the result in which we compute the matching rate for 3000 pairs varying the threshold and transformation degree. For a comparison purpose, we also compute the matching rate for original image pairs (i.e., no images are transformed in the pairs) and reconstruction (i.e., one of the images in the pair is transformed with 0 degrees) pairs. Therefore, with the matching rate in the case of the original image pairs, we can see the performance of the face verification algorithm we used, while with the results on the reconstruction pairs, we can see the effect of degradation caused by the decoder network. As we can see from the table, the FaceNet algorithm performs well on the original and reconstruction pairs. However, after applying the proposed method, FaceNet cannot identify the same person. Interestingly, $\mathcal{I}^{180}$ transformation results the best even high thresholds as we expect.

Table 2. Quantitative de-identification results using 3000 matched facial image pairs from LFW dataset [58]. From left to right, η: cosine similarity distance, $\mathcal{I}$: matching rate using original pairs given η, $\mathcal{I}^m$: matching rate in which one of the images in the testing pairs is rotated by m degrees.

η	Matching Rate							
	$\mathcal{I}$	$\mathcal{I}^0$	$\mathcal{I}^{30}$	$\mathcal{I}^{60}$	$\mathcal{I}^{90}$	$\mathcal{I}^{120}$	$\mathcal{I}^{150}$	$\mathcal{I}^{180}$
0.1	0.017	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
0.2	0.215	0.014	0.009	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
0.3	0.524	0.109	0.085	0.014	**0.000**	**0.000**	**0.000**	**0.000**
0.4	0.763	0.342	0.290	0.097	0.002	**0.000**	**0.000**	**0.000**
0.5	0.905	0.608	0.565	0.309	0.019	0.001	**0.000**	**0.000**
0.6	0.963	0.797	0.767	0.585	0.081	0.001	**0.000**	**0.000**
0.7	0.984	0.913	0.900	0.788	0.224	0.011	0.002	**0.001**
0.8	0.990	0.964	0.959	0.916	0.452	0.050	0.007	**0.005**
0.9	0.994	0.986	0.986	0.972	0.704	0.131	0.030	**0.019**
1.0	0.998	0.995	0.995	0.991	0.880	0.280	0.093	**0.061**

Table 2. *Cont.*

η	Matching Rate							
	$\mathcal{I}$	$\mathcal{I}^0$	$\mathcal{I}^{30}$	$\mathcal{I}^{60}$	$\mathcal{I}^{90}$	$\mathcal{I}^{120}$	$\mathcal{I}^{150}$	$\mathcal{I}^{180}$
1.1	0.998	0.998	0.998	0.997	0.965	0.507	0.219	**0.161**
1.2	0.999	0.999	1.000	1.000	0.990	0.731	0.409	**0.343**
1.3	1.000	1.000	1.000	1.000	0.999	0.899	0.657	**0.577**
1.4	1.000	1.000	1.000	1.000	1.000	0.978	0.843	**0.784**
1.5	1.000	1.000	1.000	1.000	1.000	0.995	0.964	**0.935**
1.6	1.000	1.000	1.000	1.000	1.000	1.000	0.996	**0.992**
1.7	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**

4.3. Evaluation on Preserving Facial Attributes

To evaluate the performance of facial attributes preservation while de-identification of the proposed method, we conduct both of qualitative and quantitative experiments in this subsection.

4.3.1. Qualitative Analysis

Our goal in this work is to de-identify facial images while preserving facial attributes such as pose, color, gender, expression as much as possible. To confirm the performance qualitatively, we apply our method on three different datasets: VGG2Face, LFW, and Japanese Female Facial Expression dataset (JAFFE) [59]. The results are shown in Figures 6–8. As we can see from the figures, the proposed method effectively preserves non-identity-related attributes while identity changes.

(a) $\mathcal{I}$ (b) $\mathcal{I}^0$ (c) $\mathcal{I}^{90}$ (d) $\mathcal{I}^{180}$ (e) $\mathcal{I}$ (f) $\mathcal{I}^0$ (g) $\mathcal{I}^{90}$ (h) $\mathcal{I}^{180}$

Figure 6. Results from the VGGFace2 dataset. (**a**,**e**) show original input images, (**b**,**f**) show reconstructed images (i.e. rotated with 0 degrees), (**c**,**g**) show de-identified faces by rotating 90 degrees, and (**d**,**h**) show de-identified faces by rotating 180 degrees.

Figure 7. Results from the Labeled Faces in the Wild (LFW) dataset. (**a**,**e**) show original input images, (**b**,**f**) show reconstructed images (i.e. rotated with 0 degrees), (**c**,**g**) show de-identified faces by rotating 90 degrees, and (**d**,**h**) show de-identified faces by rotating 180 degrees.

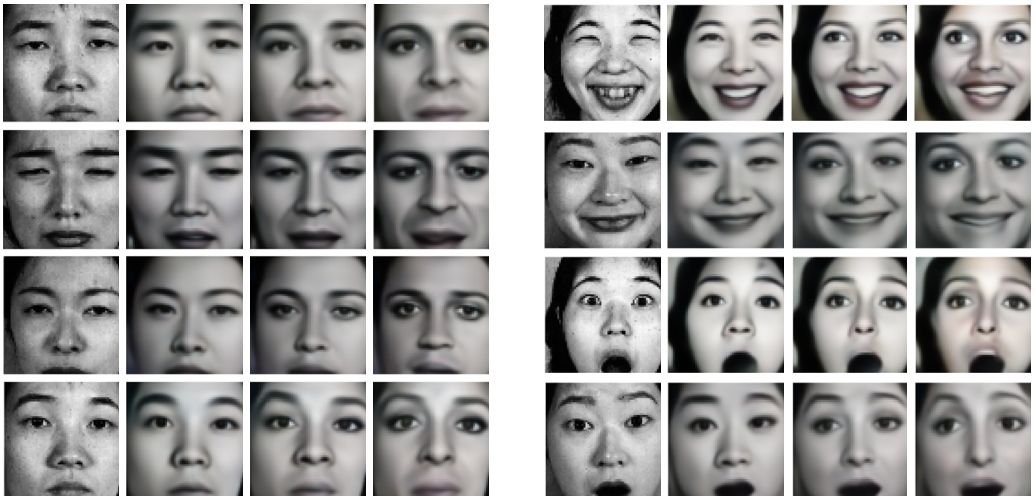


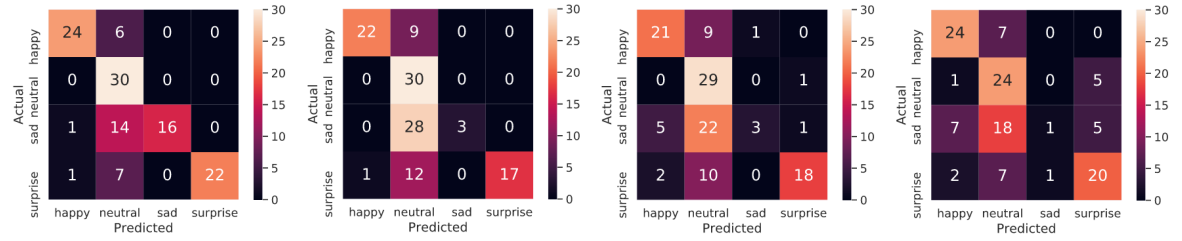
Figure 8. Results from the Japanese Female Facial Expression (JAFFE) dataset. (**a**,**e**) show original input images, (**b**,**f**) show reconstructed images (i.e. rotated with 0 degrees), (**c**,**g**) show de-identified faces by rotating 90 degrees, and (**d**,**h**) show de-identified faces by rotating 180 degrees.

4.3.2. Quantitative Analysis

To quantitatively analyze the facial attribute preservation performance of our work, we adopt a facial expression recognition algorithm, Microsoft Azure face API [60], and compute confusion matrices with the ground truth labels for four types of image sets: original, $\mathcal{I}^0$, $\mathcal{I}^{90}$ and $\mathcal{I}^{180}$. For this experiment, we choose Japanese Female Facial Expression (JAFFE) dataset [59], which contains 213 images of 7 facial expressions (i.e., angry, disgust, fear, happy, neutral, sad and surprise) by 10 Japanese female models. Among the facial expressions, in this experiment, we use only four of them (i.e., happy, neutral, sad and surprise) showing high accuracy from the adopted facial expression algorithm. Figure 9 provides the results. In the case of using original images shows the best accuracy, the transformed results processed by the proposed method also provide comparable accuracy except only from the case of 'sad'. We analyze this as an effect of the degradation of details by our method. Since the proposed network architecture is based on VAE, it bounds to the generative power of

stochastic sampling methods although it enables capturing meaningful features from the input domain, thus such disentanglement of latent feature space we benefit from in this work.

We also provide F-measures which are harmonic means of the precision and recall calculated from the confusion matrices. Averages of F-measures for original images, $\mathcal{I}^0$, $\mathcal{I}^{90}$ and $\mathcal{I}^{180}$ are 0.448, 0.338, 0.333 and 0.324, respectively. Since gaps between de-identification results are minimal, we can confirm that the proposed method can preserve facial expression while identity removes but the degree of preservation is bounded to the generative power of the VAE-based encoder we adopt in this work.

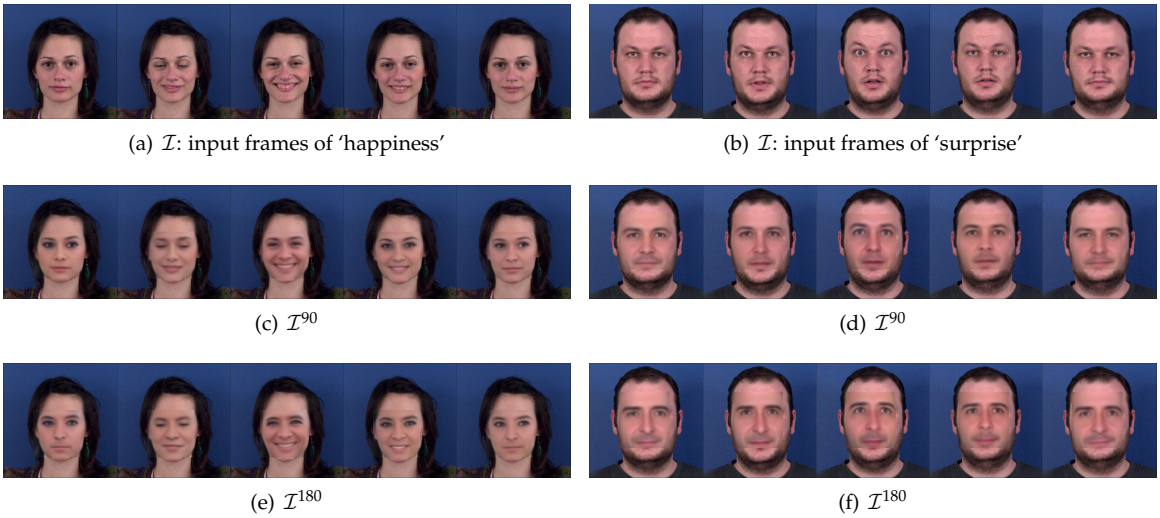


(a) $\mathcal{I}$ (b) $\mathcal{I}^0$ (c) $\mathcal{I}^{90}$ (d) $\mathcal{I}^{180}$

Figure 9. Confusion matrices of facial expression recognition using MS Azure face API: (**a**) result of original images, (**b**) result of reconstruction images (i.e., rotated with 0 degrees), (**c,d**) results of de-identification images by rotating 90 and 180 degrees, respectively.

4.4. Qualitative Analysis on Videos

Finally, we present experimental results of the proposed method on videos (Supplementary Videos S1). In this experiment, we use Multimedia Understanding Group (MUG) facial expression dataset [61], which consists of image sequences of 86 subjects performing various facial expressions. The proposed method is applied on those image sequences frame by frame to see if the modified identities retain in a sequence which is preferable for various applications. As we can see in Figure 10, with the same m which is the control parameter for the de-identification, the modified identities tend to retain in the sequence as we intended in this work. However, as it also can be seen from the results, there are some limitations. The result has discontinuity on boundaries of facial parts (which have been processed by the proposed method) and loses small details such as moles from input faces.



(a) $\mathcal{I}$: input frames of 'happiness' (b) $\mathcal{I}$: input frames of 'surprise'

(c) $\mathcal{I}^{90}$ (d) $\mathcal{I}^{90}$

(e) $\mathcal{I}^{180}$ (f) $\mathcal{I}^{180}$

Figure 10. Results from the MUG dataset [61]: (**a,b**) results of original images, (**c,d**) results of de-identification images by rotating 90 degrees, (**e,f**) results of de-identification images by rotating 180 degrees.

5. Conclusions

In this work, we proposed a novel facial de-identification method for privacy preservation. Our method is aimed at not only removing identity-related information from input facial images but also preserving the rest facial attributes that are useful for further applications. The proposed method disentangles an identity-related vector and a facial attributes-related vector from a facial image and then we efficiently transform the identity-related vector to change the identity of the input image to a completely new identity which have not seen in the training. Through various experiments, we have shown that the proposed method can effectively change the identity from input facial images while preserving the rest attributes as we designed. However, we also have seen that the output of the proposed method is suffered from degradation when compared to real images and discontinuity on facial boundaries. Therefore, we will extend our method to construct with an adversarial architecture while having manipulated latent space to overcome the degraded quality and discontinuity on facial boundaries of the resulting de-identified images.

Supplementary Materials: The following are available online at http://www.mdpi.com/2076-3417/10/3/1120/s1, Video S1: Results from the MUG dataset [61].

Author Contributions: Conceptualization, D.C., J.H.L., I.H.S.; methodology, D.C.; software, D.C.; validation, D.C., J.H.L., I.H.S.; formal analysis, D.C., J.H.L.; investigation, D.C., J.H.L.; resources, J.H.L., I.H.S.; data curation, D.C.; writing—original draft preparation, D.C.; writing—review and editing, D.C., J.H.L.; visualization, D.C.; supervision, I.H.S.; project administration, J.H.L., I.H.S.; funding acquisition, J.H.L., I.H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Technology Innovation Program (10080638) funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea), as well as by Institute for Information and communications Technology Promotion (IITP) grant funded by MSIT (No. 2018-0-00622).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Neustaedter, C.; Greenberg, S. Balancing privacy and awareness in home media spaces. Workshop on Ubicomp Communities: Privacy as Boundary Negotiation. In Proceedings of the Conjunction with the 5th International Conference ON Ubiquitous Computing (UBICOMP), Seattle, WA, USA, 12–15 October 2003.
2. Butler, D.J.; Huang, J.; Roesner, F.; Cakmak, M. The Privacy-Utility Tradeoff for Remotely Teleoperated Robots. In Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, HRI '15, Portland, OR, USA, 2–5 March 2015; ACM: New York, NY, USA, 2015; pp. 27–34. doi:10.1145/2696454.2696484. [CrossRef]
3. Ryoo, M.; Rothrock, B.; Fleming, C.; Yang, H.J. Privacy-Preserving Human Activity Recognition from Extreme Low Resolution. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
4. Brooks, A.L. Subject Anonymisation in Video Reporting. Is Animation an option? In Proceedings of the 9th International Conference on Disability, Virtual Reality and Associated Technologies, Laval, France, 10–12 September 2012; Sharkey, P. M., Klinger, E., Eds.; University of Reading: Reading, UK, 2012; pp. 431–433.
5. Ren, Z.; Lee, Y.J.; Ryoo, M.S. Learning to Anonymize Faces for Privacy Preserving Action Detection. *arXiv* **2018**, arXiv:1803.11556.
6. Chen, J.; Konrad, J.; Ishwar, P. VGAN-Based Image Representation Learning for Privacy-Preserving Facial Expression Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Salt Lake City, UT, USA, 18–22 June 2018.
7. Wu, Y.; Yang, F.; Xu, Y.; Ling, H. Privacy-Protective-GAN for Privacy Preserving Face De-Identification. *J. Comput. Sci. Technol.* **2019**, *34*, 47–60. doi:10.1007/s11390-019-1898-8. [CrossRef]
8. Hukkelås, H.; Mester, R.; Lindseth, F. DeepPrivacy: A Generative Adversarial Network for Face Anonymization. *arXiv* **2019**, arXiv:cs.CV/1909.04538.
9. Li, T.; Lin, L. AnonymousNet: Natural Face De-Identification with Measurable Privacy. *arXiv* **2019**, arXiv:cs.CV/1904.12620.

10. Bitouk, D.; Kumar, N.; Dhillon, S.; Belhumeur, P.; Nayar, S.K. Face Swapping: Automatically Replacing Faces in Photographs. In *ACM SIGGRAPH 2008 Papers*; SIGGRAPH '08; ACM: New York, NY, USA, 2008; pp. 39:1–39:8. doi:10.1145/1399504.1360638. [CrossRef]
11. Vlasic, D.; Brand, M.; Pfister, H.; Popović, J. Face Transfer with Multilinear Models. In *ACM SIGGRAPH 2005 Papers*; SIGGRAPH '05; ACM: New York, NY, USA, 2005; pp. 426–433. doi:10.1145/1186822.1073209. [CrossRef]
12. Dale, K.; Sunkavalli, K.; Johnson, M.K.; Vlasic, D.; Matusik, W.; Pfister, H. Video Face Replacement. In Proceedings of the 2011 SIGGRAPH Asia Conference, SA '11, Hong Kong, China, 11–16 December 2011; ACM: New York, NY, USA; pp. 130:1–130:10. doi:10.1145/2024156.2024164. [CrossRef]
13. Korshunova, I.; Shi, W.; Dambre, J.; Theis, L. Fast Face-Swap Using Convolutional Neural Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
14. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
15. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
16. de Bem, R.; Ghosh, A.; Ajanthan, T.; Miksik, O.; Siddharth, N.; Torr, P. A semi-supervised deep generative model for human body analysis. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
17. Kingma, D.P.; Mohamed, S.; Rezende, D.J.; Welling, M. Semi-supervised learning with deep generative models. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 3581–3589.
18. Siddharth, N.; Paige, B.; Van de Meent, J.W.; Desmaison, A.; Goodman, N.; Kohli, P.; Wood, F.; Torr, P. Learning disentangled representations with semi-supervised deep generative models. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5925–5935.
19. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of wasserstein gans. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5767–5777.
20. Mroueh, Y.; Sercu, T.; Goel, V. McGan: Mean and covariance feature matching GAN. In Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, Sydney, Australia, 6–11 August 2017; pp. 2527–2535.
21. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv* **2017**, arXiv:1710.10196.
22. Sohn, K.; Lee, H.; Yan, X. Learning structured output representation using deep conditional generative models. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 3483–3491.
23. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
24. Odena, A.; Olah, C.; Shlens, J. Conditional Image Synthesis with Auxiliary Classifier GANs. In Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLE. org, Sydney, Australia, 6–11 August 2017; pp. 2642–2651.
25. Yan, X.; Yang, J.; Sohn, K.; Lee, H. Attribute2Image: Conditional Image Generation from Visual Attributes. In *Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2016; pp. 776–791.
26. Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; Lee, H. Generative Adversarial Text to Image Synthesis. In Proceedings of the 33rd International Conference on International Conference on Machine Learning, ICML'16, New York, NY, USA, 19–24 June 2016; Volume 48, pp. 1060–1069.
27. Walker, J.; Doersch, C.; Gupta, A.; Hebert, M. An Uncertain Future: Forecasting from Static Images Using Variational Autoencoders. In *Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2016; pp. 835–851.
28. Zheng, Z.; Yang, X.; Yu, Z.; Zheng, L.; Yang, Y.; Kautz, J. Joint Discriminative and Generative Learning for Person Re-Identification. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019. doi:10.1109/cvpr.2019.00224. [CrossRef]

29. Larsen, A.B.L.; Sønderby, S.K.; Larochelle, H.; Winther, O. Autoencoding beyond pixels using a learned similarity metric. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1558–1566.
30. Bao, J.; Chen, D.; Wen, F.; Li, H.; Hua, G. CVAE-GAN: Fine-grained image generation through asymmetric training. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017, pp. 2745–2754.
31. Blanz, V.; Romdhani, S.; Vetter, T. Face identification across different poses and illuminations with a 3d morphable model. In Proceedings of the Fifth IEEE International Conference on Automatic Face Gesture Recognition, Washington, DC, USA, 20–21 May 2002; pp. 202–207.
32. Blanz, V.; Vetter, T. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 1063–1074. [CrossRef]
33. Korshunova, I.; Shi, W.; Dambre, J.; Theis, L. Fast Face-swap Using Convolutional Neural Networks. *arXiv* **2016**, arXiv:1611.09577 .
34. Nirkin, Y.; Masi, I.; Tran Tuan, A.; Hassner, T.; Medioni, G. On Face Segmentation, Face Swapping, and Face Perception. In Proceedings of the 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), Xi'an, China, 15–19 May 2018. doi:10.1109/fg.2018.00024. [CrossRef]
35. Natsume, R.; Yatagawa, T.; Morishima, S. RSGAN: Face Swapping and Editing using Face and Hair Representation in Latent Spaces. In *ACM SIGGRAPH 2018 Posters on—SIGGRAPH '18*; Association for Computing Machinery: New York, NY, USA, 2018. doi:10.1145/3230744.3230818. [CrossRef]
36. Natsume, R.; Yatagawa, T.; Morishima, S. FSNet: An Identity-Aware Generative Model for Image-Based Face Swapping. In *Computer Vision—ACCV 2018*; Jawahar, C., Li, H., Mori, G., Schindler, K., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 117–132.
37. Nirkin, Y.; Keller, Y.; Hassner, T. FSGAN: Subject Agnostic Face Swapping and Reenactment. *arXiv* **2019**, arXiv:cs.CV/1908.05932.
38. Blanz, V.; Scherbaum, K.; Vetter, T.; Seidel, H.P. Exchanging faces in images. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2004; Volume 23, pp. 669–676.
39. Lin, Y.; Wang, S.; Lin, Q.; Tang, F. Face swapping under large pose variations: A 3D model based approach. In Proceedings of the 2012 IEEE International Conference on Multimedia and Expo, Melbourne, VIC, Australia, 9–13 July 2012; pp. 333–338.
40. Mosaddegh, S.; Simon, L.; Jurie, F. Photorealistic Face De-Identification by Aggregating Donors' Face Components. *Computer Vision—ACCV 2014*; Cremers, D., Reid, I., Saito, H., Yang, M.H., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 159–174.
41. Ribaric, S.; Ariyaeeinia, A.; Pavesic, N. De-identification for privacy protection in multimedia content: A survey. *Signal Process. Image Commun.* **2016**, *47*, 131–151. [CrossRef]
42. Sweeney, L. k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2002**, *10*, 557–570. [CrossRef]
43. Newton, E.M.; Sweeney, L.; Malin, B. Preserving privacy by de-identifying face images. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 232–243. doi:10.1109/TKDE.2005.32. [CrossRef]
44. Gross, R.; Airoldi, E.; Malin, B.; Sweeney, L. Integrating Utility into Face De-identification. In *Privacy Enhancing Technologies*; Danezis, G., Martin, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 227–242.
45. Gross, R.; Sweeney, L.; de la Torre, F.; Baker, S. Model-Based Face De-Identification. In Proceeding of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), New York, NY, USA, 17–22 June 2006; p. 161. doi:10.1109/CVPRW.2006.125. [CrossRef]
46. Cootes, T.F.; Edwards, G.J.; Taylor, C.J. Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 681–685. doi:10.1109/34.927467. [CrossRef]
47. Du, L.; Yi, M.; Blasch, E.; Ling, H. GARP-face: Balancing privacy protection and utility preservation in face de-identification. In Proceedings of the IEEE International Joint Conference on Biometrics, Clearwater, FL, USA, 29 September–2 October 2014; pp. 1–8. doi:10.1109/BTAS.2014.6996249. [CrossRef]
48. Jourabloo, A.; Yin, X.; Liu, X. Attribute preserved face de-identification. In Proceedings of the 2015 International Conference on Biometrics (ICB), Phuket, Thailand, 19–22 May 2015; pp. 278–285. doi:10.1109/ICB.2015.7139096. [CrossRef]

49. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.
50. Fausto Morales. Keras-facenet. 2019. Available online: https://github.com/faustomorales/keras-facenet (accessed on 18 November 2019).
51. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 815–823. doi:10.1109/CVPR.2015.7298682. [CrossRef]
52. Szegedy, C.; Ioffe, S.; Vanhoucke, V. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv* **2016**, arXiv:1602.07261.
53. Cao, Q.; Shen, L.; Xie, W.; Parkhi, O.M.; Zisserman, A. VGGFace2: A dataset for recognising faces across pose and age. In Proceedings of the International Conference on Automatic Face and Gesture Recognition, Xi'an, China, 15–19 May 2018.
54. King, D.E. Max-margin object detection. *arXiv* **2015**, arXiv:1502.00046.
55. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
56. Huang, G.B.; Mattar, M.; Lee, H.; Learned-Miller, E. Learning to Align from Scratch. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–8 December 2012.
57. Huang, G.B.; Ramesh, M.; Berg, T.; Learned-Miller, E. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*; Technical Report 07-49; University of Massachusetts: Amherst, MA, USA, 2007.
58. Learned-Miller, G.B.H.E. *Labeled Faces in the Wild: Updates and New Reporting Procedures*; Technical Report UM-CS-2014-003; University of Massachusetts: Amherst, MA, USA, 2014.
59. Lyons, M.; Akamatsu, S.; Kamachi, M.; Gyoba, J. Coding facial expressions with Gabor wavelets. In Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 14–16 April 1998; pp. 200–205. doi:10.1109/AFGR.1998.670949. [CrossRef]
60. Microsoft. Azure Cognitive Services Face API. 2019. Available online: https://azure.microsoft.com/en-us/services/cognitive-services/face (accessed on 18 November 2019).
61. Aifanti, N.; Papachristou, C.; Delopoulos, A. The MUG facial expression database. In Proceedings of the 11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10, Desenzano del Garda, Italy, 12–14 April 2010; pp. 1–4.