Enhanced Embedded AutoEncoders: An Attribute-Preserving Face De-identification Framework

Jianqi Liu, Zhiwei Zhao*, *Member, IEEE,* Pan Li, *Senior Member, IEEE,* Geyong Min, *Member, IEEE,* and Huiyong Li*

***Abstract*—Nowadays, face recognition technology has been dramatically boosted by the advances in deep learning and big data fields. However, this also poses grand challenges in protecting personal identity information in intelligent applications of the Internet of Things (IoT). Existing methods based on the K-Same algorithm have low effectiveness for protecting personal identity while preserving face attributes. In this paper, we propose an attribute-preserving face de-identification framework called Enhanced Embedded AutoEncoders to address this problem. Our framework consists of three parts: a Privacy Removal Network, a Feature Selection Network and a Privacy Evaluation Network. The main purpose of our framework is to ensure that the Privacy Removal Network is capable of discarding information involving identity privacy and retaining desired face attributes for certain prediction applications. In order to achieve this goal, the design of the Privacy Removal Network is crucial. Specifically, we employ two different autoencoders, one of which is embedded within the other. Extensive experimental results show that our framework outperforms existing methods by an average of 3.42%-26.22% in terms of data utility under comparable face de-identification performance, which indicates that the proposed framework can not only effectively retain face attributes but also protect personal identity well.**

***Index Terms*—deep learning, autoencoders, face attributes, face de-identification, privacy, Internet of Things (IoT).**

I. Introduction

The generation and acquisition of massive images and videos have been facilitated by the rapid development of network communication technology and the Internet of Things (IoT). Big data greatly promotes the development of machine learning technology. As a part of machine learning technology, deep learning plays an important role in many applications[1], [2], especially in computer vision, due to its capability of mining deep hidden features from vast data. In order to train deep learning models to achieve higher prediction accuracy, bulk data is needed. Many companies and organizations have been collecting data which contains user privacy information from the Internet. However, the invasion of privacy is becoming more and more serious. Uploading or sharing photos on social networking sites (such as Twitter, Facebook, etc.) probably exposes our privacy to others without our awareness. Moreover, advances in image recognition technology have accelerated the interest in Human Action Recognition. Our images and other identity-related information are taken by various IoT devices of the subway stations and the streets for analysis. It brings benefits for public safety but arises privacy concerns.

In recent years, due to frequent privacy leaks, a growing number of people are starting to pay attention to their privacy. Meanwhile, considerable researchers and companies have made a lot of efforts to protect personal privacy, particularly in face de-identification. Traditional methods, including blurring [3], black-box [4] and pixelation, are widely used to conceal personal identity information. For example, Google has applied blurring technology in Google Street View [5]. In order to retain the utility attributes while protecting personal identity information, the K-Same algorithm [6] which is based on the well-known K-Anonymity [7] has been proposed to protect individuals' privacy. The main idea of the K-Same algorithm is to select K most similar images by using Euclidean distances to generate an average image (a de-identified image). Except for the K-Same, the Model-based K-Same (K-Same-M) [8] and the K-Same-Select [9] algorithms are also guaranteed to make the face recognition rate lower than $1/K$. Unfortunately, methods based on the K-Same algorithm have low effectiveness for addressing personal identity protection issues. On the one hand, due to the curse of dimensionality [10], when a large number of data attributes are required, it is difficult to maintain good data utility while preserving data privacy. However, multiple data attributes are increasingly needed in deep learning-empowered IoT applications. On the

The work of Z. Zhao* was supported by the National Natural Science Foundation of China (No. 61972075 and No. 61972074), the National Key Research and Development Program of China (No. 2020YFE0200500), and the Natural Science Foundation of Sichuan Province (No. 2022NSFSC0885). The work of H. Li* was supported by the National Natural Science Foundation of China (No. 62231006). Zhiwei Zhao* and Huiyong Li* are the co-corresponding authors.

Jianqi Liu is with the School of Information and Communication Engineering, and is also with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China. E-mail: jianqi.liu@outlook.com.

Zhiwei Zhao* is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China. E-mail: zzw@uestc.edu.cn.

P. Li is with the Department of Electrical, Computer and Systems Engineering, Case Western Reserve University, USA. E-mail: lipan@case.edu.

G. Min is with the Department of Computer Science, University of Exeter, UK. E-mail: g.min@exeter.ac.uk.

Huiyong Li* are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China. E-mail: hyli@uestc.edu.cn.

An earlier version [17] of this paper was presented at The Second International Cognitive Cities Conference (IC3 2019) and was published in its Proceedings.

other hand, the proper K value is not easy to select. A higher K value can bring better privacy but will lead to a decrease in data utility.

With the development of deep learning networks, desired features can be automatically extracted from increasingly complex datasets. Compared with traditional multi-layer perceptron (MLP), convolutional neural networks (CNNs) work better for image recognition [11], [12], [13]. Inspired by these state-of-the-art CNNs, many researchers attempt to achieve privacy-preserving with deep learning networks [14], [15], [16], [17], [18], [19], [20]. Protecting the privacy of image data in deep learning applications is critically important. Recent studies show that autoencoder, which is based on CNNs, achieves good performance on image privacy protection. Mirjalili et al. propose Semi-adversarial networks based on autoencoders [21], [22], and their schemes are to prevent the extraction of personal gender information from transformed images. Malekzadeh et al. implement a privacy-preserving algorithm [23], replacement autoencoder, but the application scenarios are limited to protecting time-series sensory data. Another work which is also based on autoencoders aims to provide a privacy-preserving image data publishing method [24]. It is worth noting that these methods using autoencoder cannot be directly applied to protect personal identity. To ease the personal identity protection issues, Meden et al. [25] propose a method based on generative neural networks (GNNs) to generate de-identified face images. Furthermore, some recent methods [26], [27], [28], [29] exploit the generative adversarial networks (GANs) to anonymize face data. For face image synthesis, it is required that GNNs or GANs have been well-trained on high-quality face datasets. However, some of these face datasets are not publicly available, which limits the application of the proposed methods. Moreover, well-performed GANs need to go through a difficult training process.

For the purpose of obtaining better performance on both data utility and face de-identification, we propose the Enhanced Embedded AutoEncoders framework, which does not resort to synthesising new de-identified face images. Our framework aims to convert an original face image to an unrecognizable image (a de-identified image without a face) while retaining desired attributes. Compared with our previous work, in this paper, we introduce a Privacy Evaluation Network to ensure that the Privacy Removal Network is capable of removing identity-related information. The Feature Selection Network is also redesigned to extract multiple face attributes. We demonstrate that our framework is truly effective through extensive experiments. The main contributions of this paper are summarized as follows:

- We propose an attribute-preserving face de-identification framework, called Enhanced Embedded AutoEncoders, which can learn how to conceal personal identity information while preserving desired face attributes.
- We provide a new privacy-preserving face image data publishing method under the premise of protecting the privacy of face images. The trained Privacy Removal Network can produce de-identified images without a face.
- We evaluate the proposed framework with two real-world datasets. We prove that our framework has a comparable performance compared with existing face de-identification works aimed at generating new face images.

The rest of the paper is organized as follows. In Section II, related work on privacy protection methods for face images is introduced. In Section III, we present the proposed Enhanced Embedded AutoEncoders framework. The performance of our framework is tested against single-attribute prediction tasks and multi-attribute prediction tasks in Section IV. In Section V, we conclude our work.

II. Related work

Since the definition of privacy-preserving was proposed by T. Dalenius [30] in 1977, a number of researchers have studied it. For visual privacy protection, it is tough to conceal private information while keeping data utility. There is a detailed survey on visual privacy protection methods [31]. These methods can be divided into intervention, blind vision, secure processing, redaction and data hiding. In this section, we summarize approaches which focus on face de-identification.

Face de-identification based on naive methods. Prior works have addressed the problem of privacy-preserving in face recognition from different points of view. Blurring [3], black-box [4] and pixelation are common approaches. Blurring tries to obtain a de-identified image by applying Gaussian filters, and pixelation realizes identity protection by reducing the resolution of a face image. Instead of processing the entire image, black-box usually substitutes the local area (such as the eye area) of face images with a black rectangle. It is not easy for people to recognize someone in the image obtained by these naive de-identification methods. However, these methods fail to prevent manners like parrot recognition [6] from recognizing a person in the de-identified image, which will greatly reduce data utility when a high level of identity protection is required. Fig. 1 shows an example of these approaches.

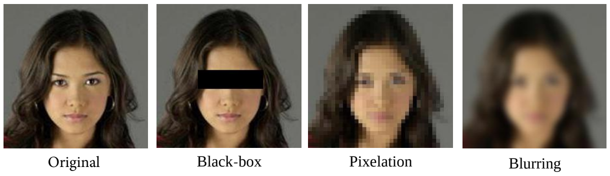

Fig. 1. Illustration of black-box, pixelation and blurring.

Face de-identification based on the K-Same algorithm. To hide the identity information and generate a new image which looks far different from the original image, the K-Same algorithm is proposed by Newton et al. [6], and it is also the most cited face de-identification scheme. With the K-Same algorithm, K initial face images are replaced by an identical synthetic image. It guarantees that the recognition rate is no more than $1/K$ but fails to guarantee the data utility of the generated image. In order to overcome this shortcoming, Gross et al. make a contribution to it. The proposed K-Same-Select [9] and the K-Same-M algorithm [8] successfully

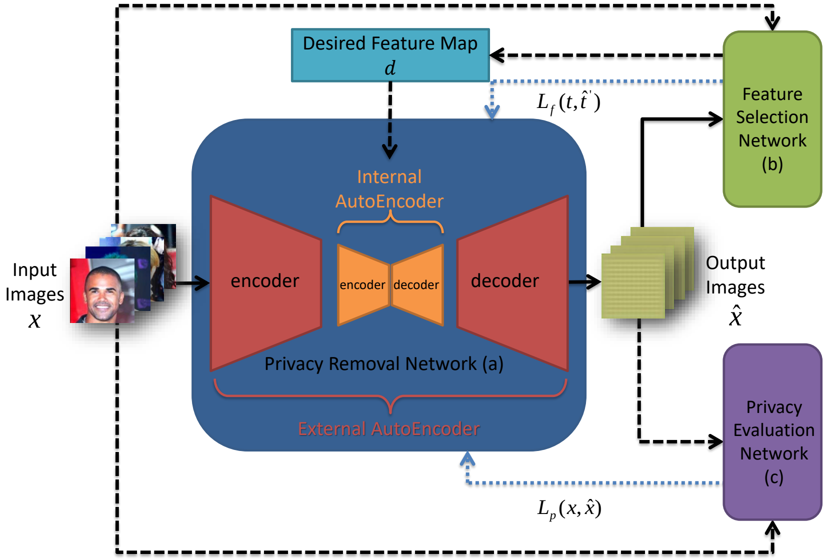

Fig. 2. The Structure of Enhanced Embedded AutoEncoders Framework. This framework consists of three components: (a) the Privacy Removal Network including two convolution autoencoders, (b) the Feature Selection Network and the Privacy Evaluation Network (c).

achieves the trade-off between privacy and utility. For the purpose of improving the quality of the generated image, other algorithms [32], [33] based on the K-Same are proposed. Nevertheless, the above algorithms have low effectiveness in keeping multiple face attributes and protecting personal identity, which we discussed in section I.

Face de-identification based on deep learning networks. Thanks to the ability of deep learning to efficiently select features, the combination of deep learning and traditional methods like the K-Same algorithm has gradually become a trend. Moreover, most deep learning applications are interested in specific attributes of face images (such as facial expression, gender, etc.). Along this direction, the K-Same-Net method which combines GNNs and the K-Same algorithm is introduced by Meden et al. [25]. The aim of this work is to retain facial expressions of interest and use the K-Same to achieve privacy-preserving. Yan et al. [26] present a face de-identification method which exploits the K-Same and GANs to preserve face attributes. Instead of the K-Same, Wu et al. build a new framework named Privacy-Protective-GAN for face de-identification [27], which makes good use of the effectiveness of GANs to generate de-identified face images. Li et al. [28] present the AnonymousNet based on GANs for synthesizing photo-realistic images with fake identities. Lin et al. [29] propose the FPGAN method based on GANs for face de-identification and full-body de-identification. These methods are designed to synthesize new de-identified face images, ignoring the fact that it requires large amounts of high-quality face data and computation for training a good generative model. The Fawkes [19] and LowKey [20] are two poisoning attack methods which aim to generate perturbed face images to prevent unauthorized facial recognition systems from using user face images. However, if these poisoning attack methods are employed to protect identity privacy, the generated perturbed images have to remain effective against all existing and future face recognition models, which is hard to hold [34].

Although the attribute-preserving face de-identification problem has been widely studied, few of them focus on generating the unrecognizable image, especially in the deep learning area. Our framework fits in the deep learning-empowered IoT applications that require an unrecognizable image containing desired face attributes.

III. Enhanced Embedded AutoEncoders

In this section, we show our framework for personal identity protection, whose structure of it is illustrated in Fig. 2. The Privacy Removal Network (PRN) we designed consists of two autoencoders: an Internal AutoEncoder and an External AutoEncoder. It is worth noting that the architecture of these two autoencoders is different. The PRN aims to remove identity-related information and only to keep the interested face attributes. The Feature Selection Network (FSN) and the Privacy Evaluation Network (PEN) are flexible parts. By choosing appropriate FSN and PEN, the framework can be applied to different deep learning applications involving identity privacy. Consider the IoT scenario in which a retail store intends to know customers' sentiments about its products based on their facial expressions. We can employ CNNs which are trained to recognize facial expressions as the FSN, and a trained face recognition model can be chosen as the PEN. Then, our framework can be employed to solve privacy concerns in this scenario. The main challenge of our framework is to choose an appropriate manner to train the PRN. Inspired by adversarial training approaches used in [15], [21], [35], [36], the FSN and the PEN provide feedback to the PRN during its training process. Notice that the data flow represented by the dotted line only exists in the training period.

A detailed description of the Enhanced Embedded AutoEncoders framework is given in the following subsections.

A. Privacy Removal Network

Since the PRN is comprised of two different autoencoders, we first introduce the details of the autoencoder.

1) AutoEncoder: Autoencoder is a neural network comprised of multilayer structures which can usually be divided into two parts: encoder and decoder. In general, an autoencoder is designed to learn the representation of datasets, which can reduce the dimension of input data. Like the other tools for feature extraction and generation, the Principal Component Analysis (PCA) is widely used for data dimensionality reduction [37]. Nevertheless, autoencoder has better performance than PCA in some applications, especially for tasks with high-dimensional data [38], [39]. This is because the autoencoder has a multilayer and nonlinearity structure. In addition to reducing dimensionality, the autoencoder has the ability to copy the input features. Namely, it tries to make decoder output features to be approximately equal to the input. A detailed explanation of the traditional autoencoder model [40] that has only one hidden layer is given as follows.

Suppose we have some facial attributes analysis tasks in which a large face dataset $\mathcal{X}$ needs to be first encoded as latent representation vectors with the most important features of the input face images. Then the latent representations will be further used for specific prediction applications. To train an autoencoder for the first encoding task, let us first define the input face image of an autoencoder as $\boldsymbol{x}$, $\boldsymbol{x} = (x_1, x_2, x_3, ..., x_n) \in \mathcal{X} \subset \mathbb{R}^n$, and obtain a hidden representation vector (the encoded data) $\boldsymbol{r} = (r_1, r_2, r_3, ..., r_{n'}) \in \mathcal{R} \subset \mathbb{R}^{n'}$ through feeding $\boldsymbol{x}$ into the mapping function:

$$\boldsymbol{r} = f_{\boldsymbol{\theta}}(x) = g(\boldsymbol{wx} + \boldsymbol{b}) \tag{1}$$

where $\boldsymbol{\theta} = \{\boldsymbol{w}, \boldsymbol{b}\}$, g is an activation function and $\boldsymbol{w}$ denotes the weight matrix whose dimensions are $n' \times n$, and $\boldsymbol{b}$ is a bias vector. The generated latent representation $\boldsymbol{r}$ can be utilized to reconstruct an output face image corresponding to $\boldsymbol{x}$. The reconstruction function is defined as

$$\hat{\boldsymbol{x}} = z_{\boldsymbol{\theta}'}(\boldsymbol{r}) = g(\boldsymbol{w'x} + \boldsymbol{b'}) \tag{2}$$

where $\boldsymbol{\theta}^{'} = \{\boldsymbol{w'}, \boldsymbol{b'}\}$. Next, the generated latent representation $\boldsymbol{r}$ is inputted into the reconstruction function to acquire a reconstructed face image $\hat{\boldsymbol{x}}$ whose dimension is the same as $\boldsymbol{x}$.

Finally, in order to make the input face image and the reconstructed face image as similar as possible, we minimize the average reconstruction error $M(\boldsymbol{x}, \hat{\boldsymbol{x}})$ to reduce the gap between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$:

$$\boldsymbol{\theta}^*, \boldsymbol{\theta}^{'*} = \arg\min_{\boldsymbol{\theta}, \boldsymbol{\theta}'} \frac{1}{S} \sum_{i=1}^{S} M(\boldsymbol{x}^{(i)}, \hat{\boldsymbol{x}}^{(i)}) \tag{3}$$

where S is the size of training face samples, and M is usually the mean square error loss function, which measures the gap between the input face $\boldsymbol{x}$ and the reconstructed face $\hat{\boldsymbol{x}}$.

In fact, an autoencoder usually includes more than one hidden layer based on CNNs. Compared with fully connected (FC) layers, convolutional layers have the characteristics of parameter sharing and sparsity of connections. These two attributes make CNNs more efficient than FC neural networks. Hence, we choose to use the convolutional autoencoder in the PRN.

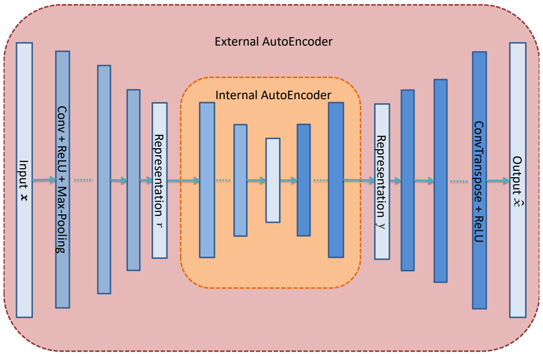

Fig. 3. The architecture of Embedded AutoEncoders. The input size of the External AutoEncoder is $3 \times 224 \times 224$, and the output size is consistent with the input size. For Internal AutoEncoder, the input size is also the same as the output size, which is $512 \times 14 \times 14$.

2) Embedded AutoEncoders: As depicted in Fig. 3, the Internal AutoEncoder is embedded in the External AutoEncoder. Let $\boldsymbol{x}$ be the input of the External AutoEncoder, the size of $\boldsymbol{x}$ is $3\times224\times224$. Let $\hat{\boldsymbol{x}}$ be the output of the External AutoEncoder, the size of $\hat{\boldsymbol{x}}$ is the same as the input. Similarly, let $\boldsymbol{r}$ and $\boldsymbol{y}$ be the input and output of the Internal AutoEncoder. The encoder part of the External AutoEncoder is composed of five convolutional layers. At first, the input image is passed through the external encoder part to get the representation vector $\boldsymbol{r}$ which is a $512 \times 14 \times 14$ dimensional feature map. Next, the representation vector $\boldsymbol{r}$ is fed into the encoder part of the Internal AutoEncoder as an input. Note that the encoder part of the Internal AutoEncoder consists of six convolutional layers, which is different from the structures of the External AutoEncoder. After passing through the decoder part with six convolutional layers of the internal autoencoder, the output vector $\boldsymbol{y}$ (representation $\boldsymbol{y}$), whose size is consistent with the input $\boldsymbol{r}$, is reconstructed. Finally, the output $\boldsymbol{y}$ is processed through the decoder part consisting of five convolutional layers of the External AutoEncoder to obtain the out image $\hat{\boldsymbol{x}}$ without sensitive information.

The key challenge of training the Embedded AutoEncoders is to train the Internal AutoEncoder. In order to achieve it, we first pre-train the External AutoEncoder by minimizing the loss function $L_e(\boldsymbol{x}, \hat{\boldsymbol{x}})$. This is to guarantee that the External AutoEncoder has the faculty to reconstruct the original images. After pre-training the External AutoEncoder, the Internal AutoEncoder is embedded in the External AutoEncoder for training. Instead of simply reconstructing the input images by minimizing the loss function $L_i(\boldsymbol{r}, \boldsymbol{y})$, the Internal AutoEncoder needs to learn how to drop sensitive features but retain desired features in output images. Towards that goal, the Internal AutoEncoder needs to generate $\boldsymbol{y}$ as close as possible to the desired feature map $\boldsymbol{d}$ coming from the FSN, which can be achieved by employing the loss function $L_i(\boldsymbol{d}, \boldsymbol{y})$. In so doing, the Internal AutoEncoder can reserve the desired features.

B. Feature Selection Network

The FSN is to extract the desired feature map that is denoted by a vector $\boldsymbol{d} \in \mathbb{R}^{512\times14\times14}$. For multi-attribute prediction, the correlation between face attributes should be considered. For example, the ”long hair” attribute and the ”female” attribute have a strong correlation. Thus, considering the correlation between attributes can improve learning performance. Existing works show that the performance of multi-attribute prediction can be improved by leveraging the attribute correlations [41], [42]. In order to utilize the attribute correlations, the two works employ the hard parameter sharing multi-task learning structure [43]. Specifically, the hard parameter sharing structure (Fig. 4) consists of the shared layers and the task-specific layers. For a face image, the shared layers can extract shared facial features, and the task-specific layers can further extract the attribute-specific features for each face attribute prediction task. Compared with the alternative structure which simply merges the attributes to form a long FC layer in the last layer to classify m face attributes (if a face sample has m labels, m neuron units are set in the last FC layer), our FSN adopts the hard parameter sharing multi-task learning structure which has multiple FC groups to extract more discriminative features for each face attribute. We take each face attribute as a binary classification task, and the number of neuron units in the last FC layer is 2. Thus, each binary classification task needs one FC group. If a face image has m attribute labels, we set m FC groups. We also experimentally compared our FSN structure with the merged long FC layer structure. We found that the FSN structure has better performance than the merged long FC layer structure, especially in terms of model convergence speed. Concretely, we take the {Gender, Smiling, Big_Nose, Eyeglasses} attributes prediction task as the comparison experiment for the FSN structure and the merged long FC layer structure (denoted as Merged-LFC). Notably, except for the last FC layer, the merged long FC layer structure has the same convolutional layer structure and FC layer structure as the FSN. Fig. 5 shows that the FSN structure achieves better performance. As shown in Fig. 6, our FSN contains six shared convolutional layers and m FC Groups. The size of the input image $\boldsymbol{x}$ is $3 \times 224 \times 224$, and both convolutional layers use modules consisting of Convolutional-BatchNorm-LeakyRelu-Max-pooling. After passing through each FC Group which consists of three FC layers, the corresponding vector $\hat{\boldsymbol{t}} \in \mathbb{R}^{1\times2}$ is produced. Here, $\hat{\boldsymbol{t}}$ is inputted into the softmax function to obtain a probability vector $\hat{\boldsymbol{t}}'$ whose sum is 1, and the desired feature map $\boldsymbol{d}$ comes from the output of the shared convolutional layers. In fact, the structures of FSN are flexible because different networks can be designed for different tasks. In this paper, the proposed networks are applied to face attribute prediction.

There are two main purposes for designing the FSN. On the one hand, we pre-train it by minimizing the loss function $L_f(\boldsymbol{t}, \hat{\boldsymbol{t}}')$ ($\boldsymbol{t}$ is the ground truth) so as to generate desired feature map well. During training, we use 0.5 probability for FC dropout layers to prevent overfitting. On the other hand, during the training of the Internal AutoEncoder, the input of the FSN is replaced with $\hat{\boldsymbol{x}}$ to provide feedback to the Internal AutoEncoder.

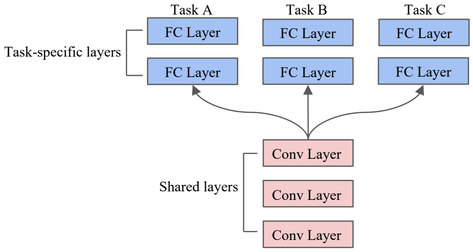

Fig. 4. Hard parameter sharing structure of multi-task learning.

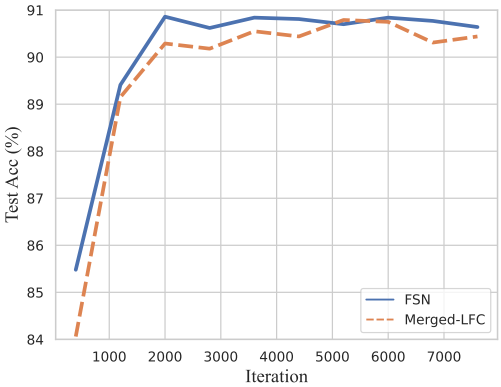

Fig. 5. Performance comparison of our FSN structure and the merged long FC layer structure.

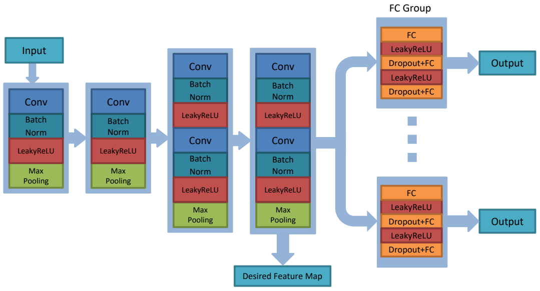

Fig. 6. Architecture of our FSN consisting of six shared convolutional layers and m FC Groups.

C. Privacy Evaluation Network

Since 2014, many successful face recognition techniques based on deep learning methods have been proposed [44]. In order to evaluate the privacy of de-identified images generated by Embedded AutoEncoders, we employ ArcFace [45] which is one of the state-of-the-art face recognition techniques as the PEN. As shown in Fig. 7, the general architecture of the ArcFace consists of a deep convolutional neural network (DCNN) model and one FC layer. In this work, we adapt the ResNet50 model as the DCNN. The input size of the DCNN is $3 \times 112 \times 112$, and we set the output feature dimension of the DCNN to 512 for better performance. We pre-train the ArcFace model with the ArcFace loss described in [45] and utilize the cosine similarity function $L_p(\boldsymbol{x}, \hat{\boldsymbol{x}})$ to verify if the two images represent the same person. When we train the Internal AutoEncoder, the cosine similarity function provides feedback to the Internal AutoEncoder. It is worth noting that we use the ArcFace model without the FC layer to compute the facial similarity.

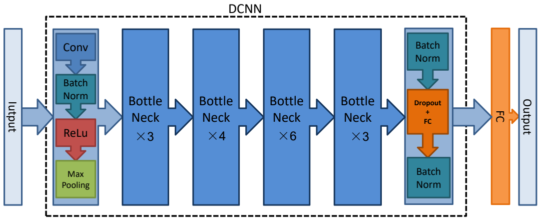

Fig. 7. Architecture of Privacy Evaluation Network based on ArcFace.

D. Loss Function

In this section, we clarify the details of the loss functions mentioned above. For each face attribute prediction task of the FSN, the corresponding loss function $L_f(\boldsymbol{t},\hat{\boldsymbol{t}}')$ is a cross-entropy function:

$$L_f(\boldsymbol{t},\hat{\boldsymbol{t}}') = -\frac{1}{S}\sum_{i=1}^{S} \boldsymbol{t}^{(i)} \log(\hat{\boldsymbol{t}}'^{(i)}) \tag{4}$$

which enables the output of the FSN, i.e., $\boldsymbol{d}$, to preserve desired features.

The cosine similarity function $L_p(\boldsymbol{x},\hat{\boldsymbol{x}})$ employed to the PEN is defined as

$$L_p(\boldsymbol{x},\hat{\boldsymbol{x}}) = \frac{\boldsymbol{x}\cdot\hat{\boldsymbol{x}}}{||\boldsymbol{x}||||\hat{\boldsymbol{x}}||} \tag{5}$$

In order to train the PRN, we first pre-train the External AutoEncoder of the PRN with the mean square error loss:

$$L_e(\boldsymbol{x},\hat{\boldsymbol{x}}) = \frac{1}{S}\sum_{i=1}^{S} ||\boldsymbol{x}^{(i)} - \hat{\boldsymbol{x}}^{(i)}||^2 \tag{6}$$

This ensures that the External AutoEncoder has the capability to reconstruct input images so that the latent representation of the External AutoEncoder can keep all the information of the input face images, including the identity-related privacy. After the pre-training, we change the mode of the External AutoEncoder from training mode to evaluation mode. Then, we embed the Internal AutoEncoder in the External AutoEncoder and train it using an adversarial training approach implemented by the multiple loss:

$$\begin{aligned} L_{multiple}(\boldsymbol{d},\boldsymbol{y},\boldsymbol{t},\hat{\boldsymbol{t}}',\boldsymbol{x},\hat{\boldsymbol{x}}) =& L_i(\boldsymbol{d},\boldsymbol{y}) + \alpha_1 \sum_{n=1}^{m} L_{f\_n}(\boldsymbol{t},\hat{\boldsymbol{t}}') \\ &+ \alpha_2 L_p(\boldsymbol{x},\hat{\boldsymbol{x}}) \end{aligned} \tag{7}$$

where α_1 and α_2 are the hyperparameters for the multiple loss. The second loss item comes from the FSN with m FC Groups, and the third loss item is from the PEN. The loss of FSN indicates whether the desired attribute features are retained by the Internal AutoEncoder, and the PEN's loss indicates whether the Internal AutoEncoder discards identity-related information. With the help of the multiple loss, the Internal AutoEncoder can learn to abandon identity-related features in the latent representation of the External AutoEncoder and only retain desired attribute features. Notably, the FSN and the PEN are set to evaluation mode when they participate in the training of the Internal AutoEncoder. In this work, we set $\alpha_1 = \alpha_2 = 1$ to weight the two feedback equally.

IV. Experiments

In this section, we first introduce three face datasets used in our experiments. Next, we present the details of two face attribute prediction experiments (single-attribute prediction and multi-attribute prediction) conducted to investigate the effectiveness of the proposed framework. Finally, we evaluate our framework from three aspects: data utility, face de-identification and visualization. Experimental results show that the proposed framework performs well on both data utility and face de-identification. Except for performance evaluation experiments, we also conduct ablation experiments to explore whether the same performance can be achieved with one simple autoencoder structure. We demonstrate that the embedded autoencoder structure is critical and indispensable.

A. Datasets

There are three popular face datasets used in this work: the CelebFaces Attributes (CelebA) dataset [46], the Labeled Faces in the Wild (LFW) dataset [47] and the CASIA-WebFace [48] dataset. The CelebA dataset is a large-scale face attribute dataset containing 202,599 face images of 10,177 individuals, in which each face image has 40 binary facial attribute labels, such as Gender (Male (yes/no)), Big_Nose (yes/no), Smiling (yes/no), Eyeglasses (yes/no), etc. Similarly, the LFW dataset is a public face dataset which contains 13,233 face images belonging to 5,749 identities. In particular, each face image in the LFW dataset has more than 40 facial attribute labels. The CASIA-WebFace, a publicly available face dataset, has 494,414 face images of 10,575 people. For the data preprocessing, we use MTCNN [49] to detect face areas and then crop them to 224 × 224.

In order to investigate the performance of the proposed framework in depth, we conduct two face attribute prediction experiments with the above three face datasets. Specifically, for comparison with our previous work, we continue to choose the CelebA dataset for single-attribute prediction experiments. In the meanwhile, to further validate the effectiveness of our framework, we carry out multi-attribute prediction experiments on the LFW dataset whose scale is smaller than the CelebA dataset. The goal of the multi-attribute prediction is to achieve multiple face attributes preservation and face de-identification. Achieving the multi-attribute prediction on the smaller dataset is more challenging due to the difficulty of learning general features on small samples and maintaining generalization. Notably, to train the PEN, we employ another face dataset: the CASIA-WebFace dataset, because using the trained face recognition model to recognize never-before-seen face images is more in line with actual face recognition scenarios.

TABLE I
THE DATASET FOR SINGLE-ATTRIBUTE PREDICTION.

	Male		Smiling		Big_Nose		Young	
	yes	no	yes	no	yes	no	yes	no
CelebA-train	3,609	4,391	3,958	4,042	1,912	6,088	5,802	2,198
CelebA-test	876	1,124	966	1,034	444	1,556	1,473	527

TABLE II
THE DATASET FOR MULTI-ATTRIBUTE PREDICTION.

	GS\GSB\GSBE
LFW-train	4,578
LFW-test	1,143

TABLE III
THE NUMBER OF POSITIVE (YES) AND NEGATIVE (NO) SAMPLES OF THE GENDER, SMILING, BIG_NOSE AND EYEGLASSES ATTRIBUTES OF THE SELECTED LFW DATASETS.

	Male		Smiling		Big_Nose		Eyeglasses	
	yes	no	yes	no	yes	no	yes	no
LFW-train	3,434	1,144	2,007	2,571	3,102	1,476	704	3,874
LFW-test	834	309	471	672	752	391	159	984

B. Experiment Setup

The key challenge of our proposed framework is to train the PRN. For the purpose of obtaining trained PRN (T-PRN) for these experiments, we first train the FSN and the PEN. Then, the PRN is trained with the Pre-trained FSN (P-FSN) and the Pre-trained PEN (P-PEN). The details of the training steps are depicted in Procedure 1, Procedure 2 and Procedure 3. During the FSN and the PRN training, we utilize mini-batch SGD and set the initial learning rate to 1e-3. For the PEN training, we resize the input image to 112 × 112 to fit the ArcFace. In addition to applying mini-batch SGD, the initial learning rate is set to 1e-1. The final test accuracy of face recognition on the LFW dataset is 98.5%. In this work, all networks are implemented with the PyTorch framework [50].

Procedure 1: FSN Training Phase

```
Input  : attribute training dataset, attribute number $m$.
Output: P-FSN: Pre-trained FSN.
FSN ← the FSN having $m$ FC Groups
A_D ← attribute training dataset (e.g. {Gender, Smiling} attribute dataset)
P-FSN ← train FSN on A_D
```

Procedure 2: PEN Training Phase

```
Input  : whole CASIA-WebFace dataset.
Output: P-PEN: Pre-trained PEN.
PEN ← the ArcFace network
CW_D ← whole CASIA-WebFace dataset
P-PEN ← train PEN on CW_D
```

Procedure 3: PRN Training Phase

```
Input  : whole CelebA dataset, attribute training dataset, P-FSN, P-PEN.
Output: T-PRN: Trained PRN.
E-AE ← the External AutoEncoder
I-AE ← the Internal AutoEncoder
CA_D ← whole CelebA dataset
A_D ← attribute training dataset (e.g. {Gender, Smiling} attribute dataset)
P-E-AE ← train E-AE on CA_D
PRN ← embed I-AE in P-E-AE
T-PRN ← for PRN, only train I-AE with P-FSN and P-PEN on A_D with the multiple loss $L_{multiple}(\boldsymbol{d}, \boldsymbol{y}, \boldsymbol{t}, \hat{\boldsymbol{t}}', \boldsymbol{x}, \hat{\boldsymbol{x}})$
```

The detailed experimental settings of single-attribute prediction and multi-attribute prediction are as follows.

1) single-attribute prediction: For single-attribute prediction, we compare the proposed framework with our previous work which focuses on preserving single face attribute. To compare the performance between the two frameworks, we select Gender, Smiling, Big_Nose and Young attributes for single-attribute prediction experiments. We randomly choose 10,000 identities from the CelebA dataset. For each identity, only one face image is randomly selected. As shown in Table I, we divide the selected 10,000 face images into two datasets: the CelebA-train dataset and the CelebA-test dataset. Specifically, the CelebA-train dataset contains 8,000 face images, and the CelebA-test dataset has 2,000 face images. We count the number of positive (yes) and negative (no) samples for these four attributes, respectively. For example, in the CelebA-train dataset: there are 3,609 males, 3,958 people are smiling, 1,912 people have Big_Noses, and 5,802 people are young people. Because the ratio of positive and negative samples of the Eyeglasses attribute of the CelebA-train dataset is almost 1:13, we substitute the Eyeglasses attribute with the Young attribute.

For this proposed framework, we first train four FSNs for the Gender-prediction, Smiling-prediction, Big_Nose-prediction and Young-prediction on the CelebA-train dataset, respectively. We denote the four trained FSNs as P-FSN-G, P-FSN-S, P-FSN-B and P-FSN-Y. Next, we train the PRN-G\S\B\Y with the corresponding P-FSN-G\S\B\Y and P-PEN on the CelebA-train dataset. At last, the T-PRN-G\S\B\Y for Gender\Smiling\Big_Nose\Young prediction is obtained.

In our previous work, the architecture of the Feature Selection Network is different from the FSN utilized in this work. Thus, we denote the previous work's Feature Selection Network as PFSN. The Privacy Removal Network employed in our previous work is also denoted as PPRN. Similarly, in order to obtain the T-PPRN-G\S\B\Y for Gender\Smiling\Big_Nose\Young prediction, we first train four PFSNs for the Gender, Smiling-prediction, Big_Nose-prediction and Young-prediction on the CelebA-train dataset, respectively. The four trained PFSNs are denoted as P-PFSN-G, P-PFSN-S, P-PFSN-B and P-PFSN-Y. Then, we train the

TABLE IV

COMPARISON OF PREDICTION ACCURACY (%) RESULTS ON THE ORIGINAL CELEBA-TEST DATASET FOR OUR PREVIOUS WORK AND OUR FRAMEWORK.

	Previous work				**Our framework**			
	P-PFSN-G	P-PFSN-S	P-PFSN-B	P-PFSN-Y	P-FSN-G	P-FSN-S	P-FSN-B	P-FSN-Y
CelebA-test	94.75	90.90	82.40	84.85	95.60	92.05	83.10	85.65

PPRN-G\S\B\Y with the corresponding P-PFSN-G\S\B\Y on the CelebA-train dataset.

2) multi-attribute prediction: For multi-attribute prediction, we compare the proposed framework with three methods: K-Same-M, K-Same-Net and LowKey. There are mainly two different operations when we implement K-Same-Net. Firstly, we replaced the Radboud Faces Database (RaFD) consisting of high-quality face images with eight different facial expressions with the CelebA dataset, because RaFD turned down our application. Hence, the implemented K-Same-Net is to predict multiple face attributes instead of different facial expressions. Secondly, since the GNN in the K-Same-Net is utilized to generate face images with facial expressions, we choose StarGAN [51] to generate face images with different face attributes in our implementation, StarGAN has the same ability to generate high-quality face images. We use 161,544 samples of 8,000 people from the CelebA dataset for training StartGAN.

With the LFW dataset, we randomly select 5,721 identities from the LFW dataset, in which each identity has one face image. We set up three different face attribute sets for multi-attribute prediction. These attribute sets are

- GS: {Gender, Smiling}.
- GSB: {Gender, Smiling, Big_Nose}.
- GSBE: {Gender, Smiling, Big_Nose, Eyeglasses}.

As shown in Table II, The GS attribute dataset, GSB attribute dataset and GSBE attribute dataset are the same selected LFW dataset. There are 4,578 images belonging to the LFW-train dataset, while the LFW-test dataset has 1,143 images. Table III reports the number of positive (yes) and negative (no) samples for the Gender, Smiling, Big_Nose and Eyeglasses attributes. Then, three FSN (FSN-GS, FSN-GSB and FSN-GSBE) are trained on the LFW-train dataset and tested on the LFW-test dataset, respectively. To reduce overfitting, we take some data augmentation methods (e.g., RandomHorizontalFlip) in PyTorch.

When the P-FSN-GS, P-FSN-GSB and P-FSN-GSBE are ready, we can get the T-PRN-GS, T-PRN-GSB and T-PRN-GSBE in the same process (introduced in the single-attribute prediction part), respectively.

C. Data Utility

In this subsection, we present the performance of our framework on face attribute preservation.

1) single-attribute prediction: Table IV reports the prediction accuracy of the P-PFSN-G\S\B\Y and the P-FSN-G\S\B\Y on the original CelebA-test. Our framework outperforms our previous work in these four attribute prediction tasks on the CelebA-test dataset, which is credited to the redesigned FSN architecture. Concretely, in the Gender attribute prediction task, our framework achieves 95.60% accuracy compared to 94.75% of our previous work and 92.05% accuracy compared to 90.90% of our previous work in the Smiling attribute prediction task. In the Big_Nose attribute prediction task, our framework achieves 83.10% accuracy compared to 82.40% of our previous work and 85.65% accuracy compared to 84.85% of our previous work in the Young attribute prediction task.

TABLE V

PREDICTION ACCURACY (%) RESULTS ON DE-IDENTIFIED CELEBA-TEST DATASETS GENERATED BY OUR PREVIOUS WORK.

	Previous work			
	D-T-G-1	D-T-S-1	D-T-B-1	D-T-Y-1
P-PFSN-G	92.10	-	-	-
P-PFSN-S	-	85.56	-	-
P-PFSN-B	-	-	79.39	-
P-PFSN-Y	-	-	-	82.83
Ave-Pred-Acc	84.97			

TABLE VI

PREDICTION ACCURACY (%) RESULTS ON DE-IDENTIFIED CELEBA-TEST DATASETS GENERATED BY OUR FRAMEWORK.

	Our framework			
	D-T-G-2	D-T-S-2	D-T-B-2	D-T-Y-2
P-FSN-G	94.65	-	-	-
P-FSN-S	-	90.70	-	-
P-FSN-B	-	-	83.05	-
P-FSN-Y	-	-	-	85.15
Ave-Pred-Acc	88.39			

To demonstrate that our proposed framework preserves the desired attribute more efficiently compared to our previous work, we conducted the following exploration experiments. Firstly, for our previous work, we generate four de-identified test datasets corresponding to the original CelebA-test dataset with the T-PPRN-G\S\B\Y, respectively. We denote these four de-identified test datasets as D-T-G-1, D-T-S-1, D-T-B-1 and D-T-Y-1. For our framework, we also generate four de-identified test datasets corresponding to the original CelebA-test dataset with the T-PRN-G\S\B\Y, respectively. These four de-identified test datasets are denoted as D-T-G-2, D-T-S-2, D-T-B-2 and D-T-Y-2. Next, we conduct prediction experiments on these eight de-identified test datasets. The prediction accuracy of the P-PFSN-G\S\B\Y and the P-FSN-G\S\B\Y on the corresponding de-identified test datasets are reported in Table V and Table VI, respectively. Our framework achieves higher prediction accuracy on the de-identified test datasets. Specifically, our framework achieves 94.65%, 90.70%, 83.05% and 85.15% on D-T-G-2, D-T-S-2, D-T-B-2 and D-T-Y-2,

TABLE VII

COMPARISON OF MULTI-ATTRIBUTE PREDICTION ACCURACY (%) RESULTS FOR K-SAME-M, K-SAME-NET, LOWKEY AND OUR FRAMEWORK.

		K-Same-M	**K-Same-Net**			**LowKey**	**Our framework**		
		KSM-D-T	KSN-D-T-GS	KSN-D-T-GSB	KSN-D-T-GSBE	L-D-T	D-T-GS	D-T-GSB	D-T-GSBE
P-FSN-GS	K= 2	59.06	87.31	-	-	90.82	**91.29**	-	-
	K= 5	60.10	88.98	-	-				
	K= 9	62.12	86.57	-	-				
	K= 14	61.42	87.97	-	-				
	K= 20	61.91	87.31	-	-				
P-FSN-GSB	K= 2	61.15	-	84.28	-	88.58	-	**89.18**	-
	K= 5	61.53	-	85.16	-				
	K= 9	63.40	-	85.48	-				
	K= 14	63.81	-	83.32	-				
	K= 20	64.28	-	84.25	-				
P-FSN-GSBE	K= 2	67.02	-	-	87.14	89.85	-	-	**90.53**
	K= 5	67.87	-	-	87.86				
	K= 9	69.31	-	-	88.76				
	K= 14	69.14	-	-	86.81				
	K= 20	69.58	-	-	85.67				
Ave-Pred-Acc		64.11	86.46			89.75	**90.33**		

while our previous work achieves 92.10%, 85.56%, 79.39% and 82.83% on D-T-G-1, D-T-S-1, D-T-B-1 and D-T-Y-1. Compared with our previous work, the prediction accuracy of the Gender, Smiling, Big_Nose and Young attributes increased by 2.55%, 5.14%, 3.66% and 2.32%, which shows that our Enhanced Embedded AutoEncoders framework can retain the desired attribute more efficiently.

2) multi-attribute prediction: We test the three multi-attribute predictors (P-FSN-GS, P-FSN-GSB and P-FSN-GSBE) on the original LFW-test dataset (Table II), of which the results are 91.47%, 89.62% and 90.99%. The test results show that the three networks perform well on the original test dataset.

In order to present the proposed framework has better performance, we compare our framework with K-Same-M, K-Same-Net and the state-of-the-art poisoning attack method: LowKey. At first, we utilize K-Same-M, K-Same-Net, LowKey and our framework to generate the de-identified test datasets from the original LFW-test dataset. Because K-Same-M tries to keep all the attributes of the face, the three generated de-identified datasets (KSM-D-T-GS, KSM-D-T-GSB and KSM-D-T-GSBE) with the specific K value are the same dataset, and we denote it as KSM-D-T. Different from K-Same-M, the K-Same-Net method aims to keep desired face attributes. Thus the three de-identified datasets (KSN-D-T-GS, KSN-D-T-GSB and KSN-D-T-GSBE) with the specific K value generated by K-Same-Net are different. In order to show the performance of K-Same-M and K-Same-Net under different K values, we generate the corresponding de-identified datasets using five K values which are 2, 5, 9, 14 and 20. Therefore, for K-Same-M, we have generated five de-identified KSM-D-T. For K-Same-Net, we have generated five KSN-D-T-GS, five KSN-D-T-GSB and five KSN-D-T-GSBE. For the LowKey, we generate the corresponding de-identified datasets through their web tool (can be found at lowkey.umiacs.umd.edu). Notice that we choose its strongest protection mode (strong (100%)) to generate corresponding perturbed face images. The generated de-identified dataset is denoted as L-D-T. Another three different de-identified datasets (D-T-GS, D-T-GSB, D-T-GSBE) are generated by our T-PRN-GS, T-PRN-GSB and T-PRN-GSBE. Then we conduct prediction experiments, observing how the multi-attribute predictors perform on these de-identified datasets. Specifically, for each K value, we run the three multi-attribute predictors on the KSM-D-T related to K. The three multi-attribute predictors are also tested on the K-related KSN-D-T-GS, KSN-D-T-GSB and KSN-D-T-GSBE which are generated by K-same-Net, respectively. Since our framework is not affected by K, we only need to run the three multi-attribute predictors on our three de-identified datasets, respectively.

We show the comparison results in Table VII. The accuracy in this table is the average prediction value of the corresponding face attribute sets. From the Table, we can observe that for K-Same-Net, as K increases, the multi-attribute prediction accuracy will first increase to the peak and then start to decrease. This result was also reported by the authors of K-Same-Net. The prediction accuracies of K-Same-M are between 59.06% and 69.58%, and the results belonging to K-Same-Net are between 83.32% and 88.98%. Compared with K-Same-M and K-Same-Net, LowKey achieves better prediction accuracies with 90.82%, 88.58%, and 89.85% for GS, GSB and GSBE attribute sets. The three prediction accuracies of our framework are 91.29%, 89.18% and 90.53%, which are higher than all results of K-Same-M, K-Same-Net and LowKey. These results indicate that our framework outperforms the three comparison methods. Besides, compared with the prediction results on the original LFW-test dataset, our framework has not much performance loss, which loses 0.18%, 0.44% and 0.46% precision for the GS, GSB and GSBE attribute sets. It means that the proposed framework can keep most of the desired features for face multi-attribute. This can be explained by the fact that during the training phase, the introduced PEN forces the PRN to be more concentrated on the desired face attribute instead of other identity-related

TABLE VIII
COMPARISON OF RANK-1 RECOGNITION RATE (%) RESULTS FOR OUR PREVIOUS WORK AND OUR FRAMEWORK.

	Previous work				Our framework			
	D-T-G-1	D-T-S-1	D-T-B-1	D-T-Y-1	D-T-G-2	D-T-S-2	D-T-B-2	D-T-Y-2
P-PEN	0.10	0.10	0.30	0.10	0	0	0.10	0
VGG-Face	0.10	0.20	0.20	0.10	0	0	0.10	0.10
MagFace	0.10	0.20	0.10	0.10	0.10	0.10	0.10	0
Ave-Rank-1-RR	0.14				0.05			

TABLE IX
COMPARISON OF RANK-1 RECOGNITION RATE (%) RESULTS FOR K-SAME-M, K-SAME-NET, LOWKEY AND OUR FRAMEWORK.

		K-Same-M	*K*-Same-Net			LowKey	Our framework		
		KSM-D-T	KSN-D-T-GS	KSN-D-T-GSB	KSN-D-T-GSBE	L-D-T	D-T-GS	D-T-GSB	D-T-GSBE
P-PEN	$K = 2$	0.17	0	0.09	0.17	6.12	0	0	0.09
	$K = 9$	0.17	0.09	0	0.26				
	$K = 20$	0.17	0.17	0.09	0.09				
VGG-Face	$K = 2$	0.09	0	0	0.09	95.89	0.09	0	0
	$K = 9$	0	0.09	0.09	0.09				
	$K = 20$	0.26	0	0	0.17				
MagFace	$K = 2$	0.35	0.09	0.09	0	67.63	0.09	0	0.09
	$K = 9$	0	0.17	0.35	0.26				
	$K = 20$	0.09	0.17	0	0.17				
Ave-Rank-1-RR		0.14	0.10			56.55	**0.04**		

features.

At last, the above experimental results of single-attribute and multi-attribute prediction suggest that our trained PRN can be used to publish de-identified datasets.

D. Face De-identification

In this subsection, we investigate the performance of the proposed framework on face de-identification. It is difficult for people to recognize subjects in de-identified images. However, face recognition models may infer identity information from de-identified images. In this work, we use the rank one (Rank-1) recognition rate as the criterion to evaluate whether the proposed framework can protect identity information well. Therefore, the lower the Rank-1 rate, the better the de-identified performance of our framework. In the meanwhile, to obtain the Rank-1 rate, we employ three face recognition models, including the P-PEN, the VGG-Face model and the MagFace model. The VGG-Face model introduced by Parkhi et al. [52] contains thirteen convolutional layers and three FC layers, and its weight parameters are publicly available. When we implement VGG-Face model, we discard the last three FC layers and convert the output of the last convolutional layer to a one-dimensional vector. At last, the input size of the implemented VGG-Face model is $3 \times 224 \times 224$, and the output size is $1 \times 25,088$. The MagFace [53] proposed by Meng et al. is a recent state-of-the-art face recognition model. A publicly available MagFace model is provided by the authors, which adopts IResNet100 as its backbone model and trains on the MS1MV2 dataset [45].

1) single-attribute prediction: For single-attribute prediction, we compare the performance of our previous work with this work on Face De-identification. Specifically, the original CelebA-test dataset forms the gallery set. In the meantime, the eight de-identified test datasets (D-T-G-1\2, D-T-S-1\2, D-T-B-1\2 and D-T-Y-1\2) are served as eight probe sets, respectively. After we have the gallery set and eight probe sets, we conduct identification experiments with the gallery set and each probe set, and we compute the corresponding Rank-1 rate. Besides, we computed the average Rank-1 recognition rate (Ave-Rank-1-RR) of the recognition rates obtained by the three face recognition models for our previous work and this work, respectively. The comparison results are reported in Table VIII. For our previous work, the Rank-1 recognition rates are between 0.1% and 0.3%, and the Ave-Rank-1-RR is 0.14%. For our framework, the Rank-1 recognition rate does not exceed 0.1%, and the Ave-Rank-1-RR is only 0.05%. In order to more intuitively illustrate the advantages of our framework, we convert the Rank-1 recognition rate to the corresponding number of successfully recognized identities. Because the gallery set has 2,000 identities, our previous work has up to 6 identity protection failures, but the number of failures of our framework is at most 2. Thus, the Rank-1 rate results of our framework outperform our previous work. Besides, it proves that with the help of the introduced PEN, the PRN can remove identity-related features more effectively.

2) multi-attribute prediction: We make a face de-identification performance comparison with K-Same-M, K-Same-Net and LowKey. Similarly, the original LFW-test dataset serves as the gallery set, and the de-identified datasets serve as the probe sets. The de-identified datasets are generated in the Subsection IV-C. Only the de-identified datasets with $K = 2$, $K = 9$ and $K = 20$ are selected for K-Same-M and K-Same-Net. In order to obtain the corresponding Rank-1 rate,

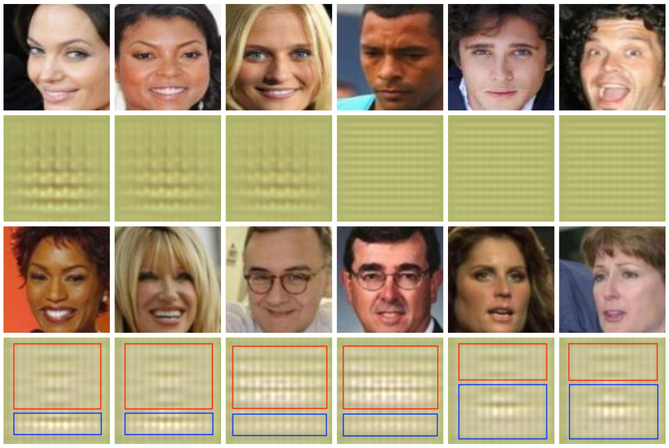

Fig. 8. De-identified images for the Gender attribute prediction and the GSBE attribute prediction. The first row shows the original inputs, and the second row shows the corresponding de-identified outputs coming from the T-PRN-G. The third row is the original inputs, and the fourth row presents the corresponding de-identified outputs of the T-PRN-GSBE.

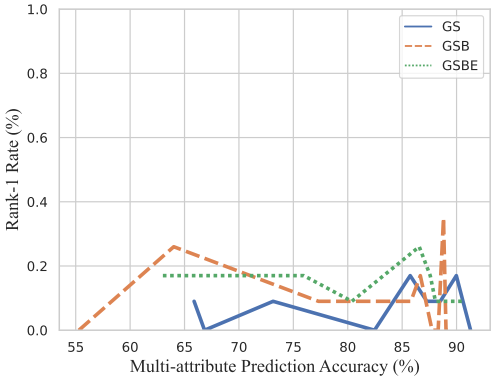

Fig. 9. The test result curves of GS, GSB and GSBE prediction during the corresponding PRN training phase.

we also conduct identification experiments with the gallery set and each probe set. The evaluation results are reported in Table IX. The Ave-Rank-1-RR of K-Same-M, K-Same-Net, LowKey and our framework is 0.14%, 0.10%, 56.55% and 0.04%. Since LowKey is designed to generate perturbed face images while maintaining image quality, perturbed face images are still at risk of being recognized by face recognition models, such as the three face recognition models that we used, resulting in the high Ave-Rank-1-RR. The Rank-1 results obtained by our framework are all lower than K-Same-M, K-Same-Net and LowKey. Fig. 9 shows the test result curves of GS, GSB and GSBE during the corresponding PRN training phase. The first result is tested at iteration 400. The multi-attribute prediction accuracies of GS, GSB and GSBE are 65.88%, 55.29% and 63.01%, and the Rank-1 rates of GS, GSB and GSBE are 0.09%, 0 and 0.17%. As the number of iterations increases, the multi-attribute prediction accuracies gradually increase, and the Rank-1 rates always remain below 0.35%, which indicates that the de-identified images generated by our framework hardly contain identity-related information. Furthermore, the low Rank-1 rates prove that the added PEN can effectively ensure that the PRN has good performance on face de-identification. At the last iteration, the prediction accuracies and the Rank-1 rates of GS, GSB and GSBE reach the reported results in Table VII and IX, respectively. Based on the above results, we conclude that the proposed Enhanced Embedded AutoEncoders framework is a competitive framework for protecting identity information.

E. Visualization

Visualization has the ability to give us a direct insight into whether our framework can protect personal identity. As shown in Fig. 8, for single-attribute prediction, we choose gender attribute to show the effectiveness of face attribute preservation and face de-identification. The first row shows six images of female and male faces, respectively. The corresponding de-identified images are demonstrated in the second row. It is easy to find that the de-identified male and female images have distinctly different characteristics. For example, the de-identified female images have the same texture structure, which looks like water ripples.

For multi-attribute prediction, we choose the most challenging attribute set, GSBE, to show the effectiveness of our framework. In the third row, we put faces with the same GSBE attributes together in pairs, and the corresponding de-identified images are demonstrated in the fourth row. Specifically, the GSBE attributes of the first and the second female face images are {Male (no), Smiling (yes), Big_Nose (yes), Eyeglasses (no)}. The third and the fourth male images have {Male (yes), Smiling (yes), Big_Nose (yes), Eyeglasses (yes)} attributes. The GSBE attributes belonging to the fifth and sixth female face images are {Male (no), Smiling (no), Big_Nose (no), Eyeglasses (no)}. We circle two different regions with red and blue boxes in the de-identified images. It is worth noting that face images with the same GSBE attributes obtained identical de-identified images. Take the first and the second female face images as an example, their corresponding de-identified images are identical due to the same texture structure. Concretely, focusing on the most obvious texture structure, there are four short black blocks in the two red regions, and both the two blue regions have one long black block and one long white block.

The above de-identified images for single-attribute prediction and multi-attribute prediction prove that our framework successfully removes identity information and retains desired features.

F. Performance Analysis

The goal of our proposed framework is to improve the data utility while keeping low face recognition rates. Therefore, we focus on achieving comparable face de-identification performance while obtaining better data utility than existing methods. We compared our framework with three existing works (K-Same-M, K-same-Net and Lowkey). Table IX shows that K-Same-M and K-same-Net have impressive face de-identification performance. This is because a) for K-Same-M, it selects K most similar images by using Euclidean distances to generate an average image and replaces the K

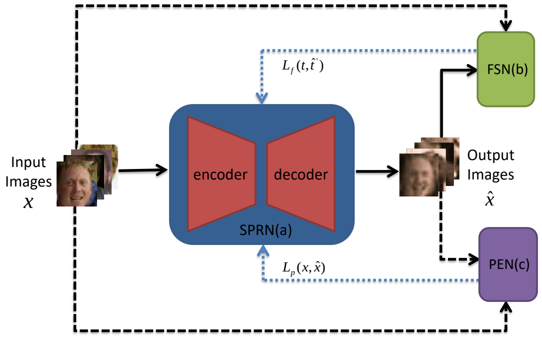

Fig. 10. The Structure of the SPRN.

face images with the averaged face image, which makes the face recognition rate lower than $1/K$; b) for K-Same-Net, it uses the GAN to generate new faces that are completely different from the source faces and replaces the source face with the generated new face. Different from the two methods, our framework converts an original face image to an unrecognizable image (a de-identified image without a face), resulting in lower Rank-1 recognition rates than K-Same-M and K-same-Net and Lowkey. Compared to K-Same-M and K-same-Net, our framework achieves comparable face de-identification performance and significantly outperforms Lowkey.

Under the comparable face de-identification performance, our framework improves the data utility. Specifically, as shown in Table VII, we also calculated the average prediction accuracy (Ave-Pred-Acc) of the attribute prediction results obtained by the three multi-attribute predictors for the three comparison methods and our framework, respectively. The Ave-Pred-Acc of K-Same-M, K-Same-Net, LowKey and our framework is 64.11%, 86.46%, 89.75% and 90.33%. Compared with K-Same-M and K-Same-Net, our framework improves the data utility by 26.22% and 3.87% on average, respectively. Although our framework improves the data utility by an average of 0.58% compared to Lowkey, the rank-1 recognition result of Lowkey is as high as 56.55%. Compared with our previous work (Table V and Table VI), the Ave-Pred-Acc on the de-identified test datasets generated by our framework is 88.39%, and the Ave-Pred-Acc of our previous work on the de-identified test datasets is 84.97%. Our proposed framework improves data utility by 3.42% on average. Thus, under the comparable face de-identification performance, our proposed framework improves the data utility by an average of 3.42%-26.22% over competing methods. From the performance of the face de-identification and data utility, we claim that our proposed method can not only effectively retain face attributes but also protect personal identity well.

G. Ablation Study

In the following ablation experiments, we use one simple autoencoder as the PRN (we denote it as SPRN) to explore whether the trained SPRN is capable of preserving desired face attributes while removing identity-related information. Fig. 10 shows the SPRN structure. Remarkably, there are two differences between our framework and the SPRN. Firstly, the SPRN only contains one simple autoencoder, the External AutoEncoder. Secondly, the training of the SPRN no longer requires the participation of the desired feature map because the desired feature map is only utilized for training the Internal AutoEncoder. The multiple loss function for training the SPRN is

$$\begin{aligned} L_{multiple'}(\boldsymbol{t},\hat{\boldsymbol{t}}',\boldsymbol{x},\hat{\boldsymbol{x}}) =& L_e(\boldsymbol{x},\hat{\boldsymbol{x}}) + \alpha_1\sum_{n=1}^{m} L_{f\_n}(\boldsymbol{t},\hat{\boldsymbol{t}}') \\ &+ \alpha_2 L_p(\boldsymbol{x},\hat{\boldsymbol{x}}) \end{aligned} \tag{8}$$

The FSN and the PEN provide feedback during the training process of the SPRN, which is the same as the training of the PRN. We keep $\alpha_1 = \alpha_2 = 1$.

In order to evaluate the performance of the SPRN, we conduct the single-attribute prediction and the multi-attribution prediction experiments. For comparison with our framework, we select the same attributes utilized in the single-attribute\multi-attribution prediction experiments in the previous section, and the CelebA dataset and the LFW dataset keep the previous settings. For the purpose of obtaining the T-SPRN-G\S\B\Y, we train the SPRN-G\S\B\Y with the corresponding P-FSN-G\S\B\Y and P-PEN on the CelebA-train dataset. Similarly, the SPRN-GS\GSB\GSBE is trained with the corresponding P-FSN-GS\GSB\GSBE and P-PEN on the LFW-train dataset. It is worth noting that the P-PEN, P-FSN-G\S\B\Y and P-FSN-GS\GSB\GSBE are obtained in the single-attribute and multi-attribution prediction experiments in the previous section, respectively. We also evaluate the SPRN from three perspectives: data utility, face de-identification and visualization.

1) data utility: We first compare the data utility between our framework and the SPRN. For single-attribute prediction tasks, the de-identified datasets of the Gender\Smiling\Big_Nose\Young attribute prediction are produced by the corresponding T-SPRN-G\S\B\Y, respectively. We denote the four de-identified datasets as SPRN-D-T-G, SPRN-D-T-S, SPRN-D-T-B and SPRN-D-T-Y, respectively. For the multi-attribution prediction task, the three de-identified datasets of the GS\GSB\GSBE attribute prediction are generated by the T-SPRN-GS\GSB\GSBE, respectively. We denote them as SPRN-D-T-GS, SPRN-D-T-GSB and SPRN-D-T-GSBE. Then, the SPRN-D-T-G\S\B\Y is tested by the corresponding P-FSN-G\S\B\Y, and the SPRN-D-T-GS\GSB\GSBE is tested by the corresponding P-FSN-GS\GSB\GSBE. The test results are shown in Table X. In the single-attribute prediction experiments, compared with the data utility results of our framework, the SRN suffers from 0.95%, 2.05%, 1.25% and 0.75% performance degradation, respectively. The Ave-Pred-Acc of the SPR is 87.14% compared to 88.39% of our framework. In the multi-attribution prediction experiments, the SPRN loses 0.91%, 1.05% and 0.79% prediction accuracy, respectively. The Ave-Pred-Acc of the SPR is 89.42% compared to 90.33% of our framework. Thus, these results show that SPRN achieves comparable data utility to our method.

2) face de-identification: In this face de-identification investigation experiment, the original CelebA-test dataset serves

TABLE X

PREDICTION ACCURACY (%) RESULTS ON DE-IDENTIFIED DATASETS GENERATED BY THE TRAINED SPRN. AS FOR P\Q, P DENOTES THE RESULT OF SPRN, AND Q DENOTES THE RESULT OF OUR FRAMEWORK REPORTED IN TABLE VI AND VII

	Single-attribute prediction				**Multi-attribute prediction**		
	SPRN-D-T-G	SPRN-D-T-S	SPRN-D-T-B	SPRN-D-T-Y	SPRN-D-T-GS	SPRN-D-T-GSB	SPRN-D-T-GSBE
P-FSN-G	93.70\94.65	-	-	-	-	-	-
P-FSN-S	-	88.65\90.70	-	-	-	-	-
P-FSN-B	-	-	81.80\83.05	-	-	-	-
P-FSN-Y	-	-	-	84.40\85.15	-	-	-
P-FSN-GS	-	-	-	-	90.38\91.29	-	-
P-FSN-GSB	-	-	-	-	-	88.13\89.18	-
P-FSN-GSBE	-	-	-	-	-	-	89.74\90.53
Ave-Pred-Acc	87.14\88.39				89.42\90.33		

TABLE XI

RANK-1 RECOGNITION RATE (%) RESULTS OF THE TRAINED SPRN. AS FOR P\Q, P DENOTES THE RESULT OF THE SPRN, AND Q DENOTES THE RESULT OF OUR FRAMEWORK REPORTED IN TABLE VIII AND IX.

	Single-attribute prediction				**Multi-attribute prediction**		
	SPRN-D-T-G	SPRN-D-T-S	SPRN-D-T-B	SPRN-D-T-Y	SPRN-D-T-GS	SPRN-D-T-GSB	SPRN-D-T-GSBE
P-PEN	93.45\0	92.25\0	98.05\0.10	99.95\0	97.11\0	40.94\0	11.29\0.09
VGG-Face	69.25\0	68.95\0	81.00\0.10	99.50\0.10	70.60\0.09	14.35\0	7.79\0
MagFace	94.60\0.10	93.70\0.10	98.30\0.10	100.00\0	93.88\0.09	60.98\0	49.69\0.09
Ave-Rank-1-RR	90.75\0.05				49.63\0.04		

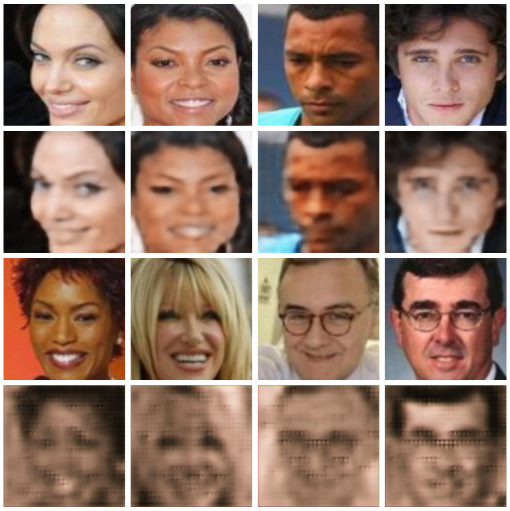

Fig. 11. De-identified images for the Gender attribute prediction and the GSBE attribute prediction. The first row shows the original inputs, and the second row shows the corresponding de-identified outputs coming from the T-SPRN-G. The third row is the original inputs, and the fourth row presents the corresponding de-identified outputs of T-SPRN-GSBE.

as the gallery set and the SPRN-D-T-G\S\B\Y serves as the probe sets. Similarly, the original LFW-test dataset serves as the gallery set and the SPRN-D-T-GS\GSB\GSBE serves as the probe sets. In order to obtain corresponding Rank-1 recognition rates for each probe set, we use the P-PEN, the VGG-Face and the MagFace face recognition model to recognize a subject included in the gallery set based on the probe set, respectively. The Rank-1 recognition results are reported in Table XI. For single-attribute prediction, the lowest Rank-1 recognition rate of SPRN is 68.95%, and the Rank-1 recognition rate obtained by the MagFace face recognition model on SPRN-D-T-Y is up to 100%. We report the Ave-Rank-1-RR of the single-attribute prediction experiments of the SPRN, which is 90.75% compared to 0.05% of our framework. In the meanwhile, for multi-attribution prediction experiments, the Ave-Rank-1-RR of the SPRN is 49.63% compared to 0.04% of our framework. Thus, the performance of the SPRN on face de-recognition drops severely, which indicates that the SPRN has a great risk of identity information leakage.

3) visualization: We finally investigate the performance of the modified framework from the visualization perspective. For comparison with our framework, we choose the Gender attribute prediction and GSBE attribute prediction to show the effect of the modified framework. As shown in Fig. 11, the first and the third rows present the original face images, and we select the same identities which are used in Fig. 8. The second row shows the de-identified images of the Gender attribute prediction, and the fourth row is the de-identified images of the GSBE attribute prediction. It can be observed that the desired face attributes have remained, but the original faces are mostly recovered, especially for the Gender attribute prediction task. This can explain why the SPRN can preserve data utility but fail to protect personal identities. Moreover, from the visual perspective, it is easy to find that the SPRN cannot protect personal identity.

Through the above data utility, face de-identification and visualization results, we claim that one single autoencoder cannot meet the requirement of retaining the desired face attributes while discarding information involving identity privacy. Our Embedded AutoEncoders structure is critical and indispensable.

V. Conclusion

This paper focuses on the attribute-preserving face de-identification problem. We have proposed an attribute-preserving face de-identification framework, Enhanced Embedded AutoEncoders, which aims to protect personal identity while preserving desired face attributes. Experiments on single-attribute and multi-attribute prediction further demonstrate that the Enhanced Embedded AutoEncoders framework has good performance on both face attribute preservation and personal identity protection.

References

[1] K. Khalil, O. Eldash, A. Kumar, and M. Bayoumi, “Machine learning-based approach for hardware faults prediction,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 11, pp. 3880–3892, 2020.

[2] R. Cioffi, M. Travaglioni, G. Piscitelli, A. Petrillo, and F. De Felice, “Artificial intelligence and machine learning applications in smart production: Progress, trends, and directions,” *Sustainability*, vol. 12, no. 2, p. 492, 2020.

[3] M. Boyle, C. Edwards, and S. Greenberg, “The effects of filtered video on awareness and privacy,” in *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. ACM, 2000, pp. 1–10.

[4] S. Ribaric and N. Pavesic, “An overview of face de-identification in still images and videos,” in *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, vol. 4. IEEE, 2015, pp. 1–6.

[5] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent, “Large-scale privacy protection in google street view.” in *ICCV*, 2009, pp. 2373–2380.

[6] E. M. Newton, L. Sweeney, and B. Malin, “Preserving privacy by de-identifying face images,” *IEEE transactions on Knowledge and Data Engineering*, vol. 17, no. 2, pp. 232–243, 2005.

[7] L. Sweeney, “k-anonymity: A model for protecting privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[8] R. Gross, L. Sweeney, F. De la Torre, and S. Baker, “Model-based face de-identification,” in *null*. IEEE, 2006, p. 161.

[9] R. Gross, E. Airoldi, B. Malin, and L. Sweeney, “Integrating utility into face de-identification,” in *International Workshop on Privacy Enhancing Technologies*. Springer, 2005, pp. 227–242.

[10] C. C. Aggarwal, “On k-anonymity and the curse of dimensionality,” in *VLDB*, vol. 5, 2005, pp. 901–909.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[14] S. A. Osia, A. Taheri, A. S. Shamsabadi, K. Katevas, H. Haddadi, and H. R. Rabiee, “Deep private-feature extraction,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 1, pp. 54–66, 2018.

[15] X. Ding, H. Fang, Z. Zhang, K.-K. R. Choo, and H. Jin, “Privacy-preserving feature extraction via adversarial training,” *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[16] S. A. Osia, A. S. Shamsabadi, S. Sajadmanesh, A. Taheri, K. Katevas, H. R. Rabiee, N. D. Lane, and H. Haddadi, “A hybrid deep learning architecture for privacy-preserving mobile analytics,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4505–4518, 2020.

[17] J. Liu, J. Liu, P. Li, and Z. Kuang, “Embedded autoencoders: A novel framework for face de-identification,” in *International Cognitive Cities Conference*. Springer, 2019, pp. 154–163, doi: 10.1007/978-981-15-6113-9_17.

[18] B. K. Beaulieu-Jones, Z. S. Wu, C. Williams, R. Lee, S. P. Bhavnani, J. B. Byrd, and C. S. Greene, “Privacy-preserving generative deep neural networks support clinical data sharing,” *Circulation: Cardiovascular Quality and Outcomes*, vol. 12, no. 7, p. e005122, 2019.

[19] S. Shan, E. Wenger, J. Zhang, H. Li, H. Zheng, and B. Y. Zhao, “Fawkes: Protecting privacy against unauthorized deep learning models,” in *29th USENIX security symposium (USENIX Security 20)*, 2020, pp. 1589–1604.

[20] V. Cherepanova, M. Goldblum, H. Foley, S. Duan, J. P. Dickerson, G. Taylor, and T. Goldstein, “Lowkey: Leveraging adversarial attacks to protect social media users from facial recognition,” in *International Conference on Learning Representations*, 2021.

[21] V. Mirjalili, S. Raschka, A. Namboodiri, and A. Ross, “Semi-adversarial networks: Convolutional autoencoders for imparting privacy to face images,” in *2018 International Conference on Biometrics (ICB)*. IEEE, 2018, pp. 82–89.

[22] V. Mirjalili, S. Raschka, and A. Ross, “Flowsan: privacy-enhancing semi-adversarial networks to confound arbitrary face-based gender classifiers,” *IEEE Access*, vol. 7, pp. 99 735–99 745, 2019.

[23] M. Malekzadeh, R. G. Clegg, and H. Haddadi, “Replacement autoencoder: A privacy-preserving algorithm for sensory data analysis,” in *Internet-of-Things Design and Implementation (IoTDI), 2018 IEEE/ACM Third International Conference on*. IEEE, 2018, pp. 165–176.

[24] T. Kim and J. Yang, “Selective feature anonymization for privacy-preserving image data publishing,” *Electronics*, vol. 9, no. 5, p. 874, 2020.

[25] B. Meden, Ž. Emeršič, V. Štruc, and P. Peer, “k-same-net: k-anonymity with generative deep neural networks for face deidentification,” *Entropy*, vol. 20, no. 1, p. 60, 2018.

[26] B. Yan, M. Pei, and Z. Nie, “Attributes preserving face de-identification.” in *ICCV Workshops*, 2019, pp. 1217–1221.

[27] Y. Wu, F. Yang, Y. Xu, and H. Ling, “Privacy-protective-gan for privacy preserving face de-identification,” *Journal of Computer Science and Technology*, vol. 34, no. 1, pp. 47–60, 2019.

[28] T. Li and L. Lin, “Anonymousnet: Natural face de-identification with measurable privacy,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[29] J. Lin, Y. Li, and G. Yang, “Fpgan: Face de-identification method with generative adversarial networks for social robots,” *Neural Networks*, 2020.

[30] T. Dalenius, “Towards a methodology for statistical disclosure control,” *statistik Tidskrift*, vol. 15, no. 429-444, pp. 2–1, 1977.

[31] J. R. Padilla-López, A. A. Chaaraoui, and F. Flórez-Revuelta, “Visual privacy protection methods: A survey,” *Expert Systems with Applications*, vol. 42, no. 9, pp. 4177–4195, 2015.

[32] L. Meng, Z. Sun, A. Ariyaeeinia, and K. L. Bennett, “Retaining expressions on de-identified faces,” in *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*. IEEE, 2014, pp. 1252–1257.

[33] L. Meng and Z. Sun, “Face de-identification with perfect privacy protection,” in *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*. IEEE, 2014, pp. 1234–1239.

[34] E. Radiya-Dixit and F. Tramèr, “Data poisoning won’t save you from facial recognition,” *arXiv preprint arXiv:2106.14851*, 2021.

[35] Y. Wu, F. Yang, and H. Ling, “Privacy-protective-gan for face de-identification,” *arXiv preprint arXiv:1806.08906*, 2018.

[36] S. Liu, J. Du, A. Shrivastava, and L. Zhong, “Privacy adversarial network: representation learning for mobile data privacy,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 4, pp. 1–18, 2019.

[37] K. Khalil, O. Eldash, A. Kumar, and M. Bayoumi, “Intelligent fault-prediction assisted self-healing for embryonic hardware,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 4, pp. 852–866, 2020.

[38] K. Siwek and S. Osowski, “Autoencoder versus pca in face recognition,” in *2017 18th International Conference on Computational Problems of Electrical Engineering (CPEE)*, Sep. 2017, pp. 1–4.

[39] J. Almotiri, K. Elleithy, and A. Elleithy, “Comparison of autoencoder and principal component analysis followed by neural network for e-learning using handwritten recognition,” in *Systems, Applications and Technology Conference (LISAT), 2017 IEEE Long Island*. IEEE, 2017, pp. 1–5.

[40] Y. Bengio *et al.*, “Learning deep architectures for ai,” *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[41] D. V. Sang, L. T. B. Cuong, and V. Van Thieu, “Multi-task learning for smile detection, emotion recognition and gender classification,” in *Proceedings of the Eighth International Symposium on Information and Communication Technology*, 2017, pp. 340–347.

[42] H. Han, A. K. Jain, F. Wang, S. Shan, and X. Chen, "Heterogeneous face attribute estimation: A deep multi-task learning approach," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 11, pp. 2597–2609, 2017.

[43] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[44] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, "Deep face recognition: A survey," in *2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*. IEEE, pp. 471–478.

[45] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[46] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[47] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[48] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[49] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems*, 2019, pp. 8026–8037.

[51] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.

[52] O. M. Parkhi, A. Vedaldi, A. Zisserman *et al.*, "Deep face recognition." in *BMVC*, vol. 1, no. 3, 2015, p. 6.

[53] Q. Meng, S. Zhao, Z. Huang, and F. Zhou, "Magface: A universal representation for face recognition and quality assessment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 225–14 234.

Jianqi Liu is currently pursuing the PhD degree with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China. He received the BS degree in School of Communication and Information Engineering from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2017. His research interests mainly include machine learning, edge computing, and security and privacy.

Zhiwei Zhao (Member, IEEE) is a Professor at the School of Computer Science and Engineering in University of Electronic Science and Technology of China (UESTC). He received his PhD degree at the College of Computer Science, Zhejiang University and the BS degree from Xi'an Jiaotong University. His research interests include edge computing and IoT systems, heterogeneous wireless networks, and protocol design. He is a member of ACM, CCF and IEEE.

Pan Li (Senior Member, IEEE) is currently an Associate Professor with the Department of Electrical, Computer and Systems Engineering, Case Western Reserve University. He received the B.E. degree in Electrical Engineering from Huazhong University of Science and Technology, Wuhan, China, in June 2005, and the Ph.D. degree in Electrical and Computer Engineering from University of Florida, Gainesville, Florida, in August 2009, respectively. His research interests include machine learning, security and privacy, and network science and economics. He received the NSF CAREER Award in 2012, and has served as an editor for many prestigious IEEE journals.

Geyong Min (Member, IEEE) is a Professor of High Performance Computing and Networking in the Department of Computer Science within the College of Engineering, Mathematics and Physical Sciences at the University of Exeter, United Kingdom. He received the PhD degree in Computing Science from the University of Glasgow, United Kingdom, in 2003, and the B.Sc. degree in Computer Science from Huazhong University of Science and Technology, China, in 1995. His research interests include Computer Networks, Wireless Communications, Parallel and Distributed Computing, Ubiquitous Computing, Multimedia Systems, Modelling and Performance Engineering.

Huiyong Li received the BS, MS, and PhD degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1997, 2000, and 2009, respectively. He is a Professor with the School of Information and Communication Engineering, UESTC. His research interests include adaptive signal processing, space-time filter, anti-interference technology, and multiple input multiple output (MIMO) signal processing.