Neurocomputing 363 (2019) 212–222

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Privacy-preserving lightweight face recognition

Yuancheng Li*, Yimeng Wang, Daoxing Li

School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China

ARTICLE INFO

Article history:
Received 22 April 2019
Revised 5 July 2019
Accepted 14 July 2019
Available online 19 July 2019

Communicated by Dr. Ran He

Keywords:
Differential privacy
Generative adversarial network (GAN)
Deep convolutional neural networks
Face recognition

ABSTRACT

Face recognition based on deep learning has become one of the mainstream identity authentication technologies. In recent years, many well-developed deep convolutional neural networks have emerged. But most of these network structures are very complicated, the training processes are very difficult and face recognition based on these network structures are too large to apply to mobile terminals. To solve this problem, we propose a lightweight face recognition algorithm: LightFace, which is based on depthwise separable convolution. In the process of training LightFace, the triplet loss algorithm is used to optimize model. However, LightFace requires a large amount of training data and the parameters of this model will record the users' data, which can be recovered by attackers. To address this problem, we propose an applicable approach to providing strong privacy guarantees for LightFace. In our approach, the generated data and additional noise are used to slightly disturb the original data distribution, so that the attackers cannot correctly predict the training data, which can improve the security of the model. Besides, in the process of training LightFace, ensemble learning increases the randomness of the original data distribution and enhances the robustness of the model. Within the differential privacy framework, we analyzed the privacy loss of the algorithm theoretically and conduct experiments on different face recognition datasets to demonstrate the effectiveness of our privacy preservation method. The experiment results show that LightFace with privacy preservation still has good recognition accuracy while protecting data privacy.

1. Introduction

With the rapid development of Internet technology, information security has been increasingly valued. Identification is one of the important methods in information security and has a very significant position in modern society. Face recognition is one of the mainstream technologies of identification technology. It has been widely used in many fields such as security, finance, and e-commerce. It is of great importance to ensure the security and privacy in face recognition process for expanding the face recognition application.

Recently, face recognition method based on deep learning has made breakthrough progress. Different from the traditional face recognition methods, the facial representation obtained by deep learning has important characteristics. For example, using convolutional neural network as feature extractor can transform the face images into a suitable internal representation. In recent years, many face recognitions have achieved good results by using deep learning method, such as Deep Face [32,35], Deep ID [30], DeepID2 [36], and VGG-Face [24], VGG-Face2 [5],HyperFace [26,9]. All the algorithms above are based on massive training data, allowing the deep learning algorithm to learn the invariable characteristics of illumination, expression, and angle. But these network structures are also complicated. To deploy face recognition systems on smart devices, Google proposed FaceNet [28] for mobile devices. At the same time, there are also many lightweight convolutional neural networks, such as: SqueezeNet [14], MobileNet [13], ShuffleNet [40], Xception [7], SE-Net [17], SK-Net [20] and so on.

Previous works of face recognition method with privacy production mainly focused on generating feature vectors from biometric face data [18,33]. But most face recognition methods nowadays are based on machine learning and we know that the machine learning systems are usually trained by a large amount of data. In the training process, some characteristic information of sensitive data is recorded and this sensitive data is stored in the form of model's parameters which had been considered private until Pre-Fredrikson et al. [12] proposed a reconstruction attack algorithm for face recognition model based on deep learning. By using Pre-Fredrikson's method, the attacker can learn sensitive information by accessing to the face recognition model, which leads to the disclosure of privacy. In consider of privacy leaks during machine learning, we proposed a privacy preservation method against this kind of attack.

* Corresponding author.
E-mail addresses: yuancheng@ncepu.cn (Y. Li), 1162227078@ncepu.edu.cn (Y. Wang), 1182227066@ncepu.edu.cn (D. Li).

https://doi.org/10.1016/j.neucom.2019.07.039
0925-2312/

There are many researches on privacy preservation in machine learning from different aspects. Smart uses secure multi-party computing to protect the privacy of non-trusted participants in the process of multi-party collaborative computing [8], with the purpose of protecting the computing process from revealing their private information. But in this article, we assume that only one party holds the sensitive data, and it is mainly concerned with the privacy leakage caused by the model output. The K-anonymous technology proposed in [31] protects the privacy of data by providing low-precision data through generalization and concealment techniques. However, K-anonymous technology is difficult to de-anonymize high-dimensional data due to theoretical limitations, so it is not suitable for our problem. Dwork proposed differential privacy to protect data security [11]. Differential privacy provides a very strict privacy preservation standard. Even if an attacker gains enough background knowledge, he cannot obtain the target data in the output.

Differential privacy is widely used to protect data security in many machine learning algorithms in recent years. Chen et al. [6] uses differential privacy to protect the data collected in the Mobile Crowd-Sensing, which ensures that the data can be analyzed without revealing the privacy of personal data. But most of the researches on the existing machine learning privacy preservation are studying the privacy preservation of the shallow machine learning model. Bassily et al. [2] uses differential privacy to minimize convex empirical risk. Song et al. [29] combines differential privacy with stochastic gradient descent algorithms and guarantees the performance of the algorithm by increasing the batch size. Duchi et al. [10] uses differential privacy to solve the statistical risk minimization problem, and establishes the upper and lower bounds of the convergence speed of the statistical estimation process to balance data protection and utility. There are also a few researches that use differential privacy in deep learning models. In [25], a self-adaptive Laplacian mechanism was used to combine differential privacy with deep neural networks, which enabled the model to achieve high efficiency and performance. Abadi et al. [1] proposes an improved stochastic gradient descent algorithm that satisfies differential privacy and use moment accountant to provide a more strict privacy loss boundary. Papernot et al. [22] combines differential privacy with the generative adversarial networks (GAN) and apply the proposed model to semi-supervised classification tasks. Good results have been obtained on MNIST and SVHN. Papernot et al. [23] also show their method can scale to learning tasks with large numbers of output classes and uncurated,imbalanced training data with errors.

There are also some related studies on quantitative privacy preservation. Mironov proposes a generalized definition of differential privacy based on R'enyi divergence using $\alpha(\lambda)$ to quantify privacy preservation [1]. The moment accountant proposed in [22] is also based on R'enyi differential privacy for privacy loss analysis. Dwork and Roth [11] propose centralized differential privacy to facilitate more accurate analysis of privacy losses. Bun and Steinke [4] propose an alternative form of centralized differential privacy based on the R'enyi divergence between the distributions of algorithm running results on adjacent data sets, providing more accurate privacy analysis and establishing a lower bound.

In this paper, we propose a lightweight face recognition algorithm LightFace, which is based on depthwise separable convolution and weight evaluation module. Compared with other face recognition models, LightFace has fewer parameters and less computational complexity. We also propose an applicable approach to providing strong privacy guarantees for LightFace. As we all known, the attacks can usually obtain private training data through the parameters of the model. Therefore, we use the generated data instead of the original data to train the model. In our approach, the generation model and additional noise is used to change the original data distribution to a certain extent, so that the attackers cannot correctly predict the training data, which can improve the security of the model. In the process of training the published model, ensemble learning increases the randomness of the original data distribution, and further protects the original data from being attacked. At the same time, this method improves the accuracy of the model and guarantees the performance of the model in the case of privacy preservation. The entire training process satisfies differential privacy, and the precise boundaries of privacy loss are given.

In summary, we make the following contributions in this paper:

1) We propose a lightweight face recognition algorithm named LightFace, which is based on depthwise separable convolutions and weight evaluation module. LightFace has lower computational complexity and is much smaller compared to VGG16 [24,14,13], it also gets higher accuracy than Mobilenet [13] and Sqeezenet [24,14].
2) We propose a privacy preservation strategy for face recognition models, which satisfies differential privacy. The privacy preservation strategy can protect the training data from being recovered while maintaining the accuracy of model.
3) We conduct experiments to evaluate the performance of the LightFace and the privacy preservation strategy. The results show that LightFace effectively reduces the amount of computation and the model size, and the strategy performs better than DPSGD [1] and PATE [22] on model privacy preservation.

The rest of our paper is organized as follows: in Section 2 we introduce LightFace, a lightweight face recognition algorithm; in Section 3 we introduce a face recognition algorithm with privacy preservation; in Section 4 we analyze the loss of privacy of the newly proposed face recognition algorithm in Section 3; in Section 5 we conduct experiments and analyze experimental results; and in Section 6 we conclude our paper.

2. Lightweight face recognition model: LightFace

Deep convolution network has been widely used for task in vision [37–39] and LightFace also uses deep convolutional network. In this section, we first introduce the core layer of LightFace construction, namely depthwise separable convolution. Then, we introduce triple loss algorithm, which can achieve individual-level recognition accuracy. Finally, we introduce LightFace's network structure and training methods.

2.1. Depthwise separable convolution

The MobileNet [13] model is based on depthwise separable convolution. Depthwise separable convolution can decompose the standard convolution into depthwise convolution and pointwise convolution. Depthwise convolution applies each convolution kernel to each channel, and pointwise convolution is used to combine the output of channel convolutions. This decomposition can effectively reduce the amount of calculations and the size of the model. Fig. 1 illustrates how the standard convolution is decomposed.

Among them, there are M input feature maps and N output feature maps. The standard convolution uses N convolution kernels of size $D_k \times D_k$, and the size of convolution kernel for operation is $D_k \times D_k \times M$. Depthwise convolution uses M size convolution kernels for operation, and pointwise convolution uses N convolution kernels for operation. The pointwise convolution method is the same as the traditional convolution method, which uses 1×1 convolution kernel to make each feature map contain the information of each feature map in the previous layer.

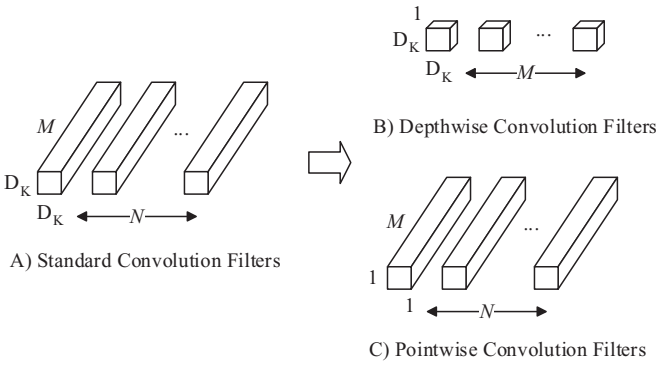


Fig. 1. Depthwise separable convolution.

Then, the amount of reduction calculation using depthwise separable convolution is shown in formula (1):

$$\frac{D_k \times D_k \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_k \times D_k \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_k^2} \tag{1}$$

$D_F \times D_F$ is the size of input feature maps. From the formula, it can be seen that the reduction of the parameters is mainly related to the size of the convolution kernel D_F.

Although depthwise separable convolution can reduce the amount of parameter, it can't change the number of input's channel. So when the number of channel is small (in the first few layers of the model), it can not effetely extract features in low-dimensional space.

2.2. Triplet loss

Triple loss is a fine-grained recognition algorithm, which is suitable for individual level recognition. Compared with the traditional classification method, it can further improve the accuracy of the training model.

The triple consists of three samples, one sample A randomly selected from the training data set, the positive sample P and negative sample N selected from dataset at random. Positive samples are of the same type with A and negative samples are of the different type. The goal of triple loss is to make the distance between the sample and the positive sample as small as possible, and the distance between the sample and the negative sample as large as possible. A threshold α is set so that the difference between the two distances is greater than the threshold. We record the characteristic expressions of the three samples as: $f(x_i^A)$, $f(x_i^P)$, $f(x_i^N)$. Then, we can express the above process as formula (2):

$$\left\| f\left(x_i^A\right) - f\left(x_i^P\right) \right\|_2^2 + \alpha < \left\| f\left(x_i^A\right) - f\left(x_i^N\right) \right\|_2^2 \forall \left(f\left(x_i^A\right), f\left(x_i^P\right), f\left(x_i^N\right) \right) \in T \tag{2}$$

Among them, T represents the set of possible triplets in the training set. The number of elements in the set is M.

From this, we can get the corresponding objective function, as shown in formula (3):

$$\sum_i^M \left[\left\| f\left(x_i^A\right) - f\left(x_i^P\right) \right\|_2^2 - \left\| f\left(x_i^A\right) - f\left(x_i^N\right) \right\|_2^2 + \alpha \right]_+ \tag{3}$$

As can be seen from the objective function, a loss occurs when the negative distance is less than the sum of positive distance and α, and the loss is zero when it is the opposite situation.

2.3. LightFace network structure and training

Inspired by MobileNet, LightNet is also based on depthwise separable convolution which is used to build light weight deep neural network. But LightFace uses specific structure to increase the accuracy and the specific structure is shown in Table 2. The LightFace model has a total of 28 layers. Its first layer is a standard convolution. Except for the last fully connected layer, all layers are followed by the Batch Normalization and ReLU nonlinear activation functions. The downsampling problem can be handled in the first layer convolution and depthwise convolution. Compared with MobileNet, LightFace adds weight evaluation process behind partial convolutional layers. The weight evaluation module can learn the importance of the channels, help depthwise convolution extract more features in low-dimensional space. The specific network structure is shown in Table 1, $D_F \times D_F$ represents the size of the input feature map, M represents the number of channels of the feature map. In the weight evaluation process, we perform the average pooling operation on the feature maps of the pointwise convolution outputs, and the information on the global receptive field can be obtained. The results of the average pooling layers are transmitted to the two-layer fully connected layer for channel weights learning. The two layers of fully connected layers are similar to traditional BP neural networks. In order to reduce the amount of parameters, we add parameter η to the first fully connected layer for dimensionality reduction, and the dimension of the second fully connected layer expands back to the original dimension. In our paper, we set $\eta = 8$. The new feature maps can be obtained by calibrating channels according to the weights through

Table 1
Weight evaluation structure.

Layer	Size-in	Size-out
Avg pool	$D_F \times D_F \times M$	$1 \times 1 \times M$
FC1	$1 \times 1 \times M$	$1 \times 1 \times \frac{M}{\eta}$
FC2	$1 \times 1 \times \frac{M}{\eta}$	$1 \times 1 \times M$

Table 2
LightFace network structure.

Layer	Size-in	Size-out	Kernel
Conv1	$224 \times 224 \times 3$	$112 \times 112 \times 32$	$3 \times 3 \times 3 \times 32$, 2
WE	$112 \times 112 \times 32$	$112 \times 112 \times 32$	
Conv2 dw	$112 \times 112 \times 32$	$112 \times 112 \times 32$	$3 \times 3 \times 32$, 1
Conv2	$112 \times 112 \times 32$	$112 \times 112 \times 64$	$1 \times 1 \times 32 \times 64$, 1
WE	$112 \times 112 \times 64$	$112 \times 112 \times 64$	
Conv3 dw	$112 \times 112 \times 64$	$56 \times 56 \times 64$	$3 \times 3 \times 64$, 2
Conv3	$56 \times 56 \times 64$	$56 \times 56 \times 128$	$1 \times 1 \times 64 \times 128$, 1
WE	$56 \times 56 \times 128$	$56 \times 56 \times 128$	
Conv4 dw	$56 \times 56 \times 128$	$56 \times 56 \times 128$	$3 \times 3 \times 128$, 1
Conv4	$56 \times 56 \times 128$	$56 \times 56 \times 128$	$1 \times 1 \times 128 \times 128$, 1
WE	$56 \times 56 \times 128$	$56 \times 56 \times 128$	
Conv5 dw	$56 \times 56 \times 128$	$28 \times 28 \times 128$	$3 \times 3 \times 128$, 2
Conv5	$28 \times 28 \times 128$	$28 \times 28 \times 256$	$1 \times 1 \times 128 \times 256$, 1
WE	$28 \times 28 \times 256$	$28 \times 28 \times 256$	
Conv6 dw	$28 \times 28 \times 256$	$28 \times 28 \times 256$	$3 \times 3 \times 256$, 1
Conv6	$28 \times 28 \times 256$	$28 \times 28 \times 256$	$1 \times 1 \times 256 \times 256$, 1
WE	$28 \times 28 \times 256$	$28 \times 28 \times 256$	
Conv7 dw	$28 \times 28 \times 256$	$14 \times 14 \times 256$	$3 \times 3 \times 256$, 2
Conv7	$14 \times 14 \times 256$	$14 \times 14 \times 512$	$1 \times 1 \times 256 \times 512$, 1
WE	$14 \times 14 \times 512$	$14 \times 14 \times 512$	
Conv8 dw			
Conv8	$14 \times 14 \times 512$	$14 \times 14 \times 512$	$3 \times 3 \times 512$, 1
–Conv12 dw	$14 \times 14 \times 512$	$14 \times 14 \times 512$	$1 \times 1 \times 512 \times 512$, 1
Conv12	$14 \times 14 \times 512$	$14 \times 14 \times 512$	
WE			
Conv13 dw	$14 \times 14 \times 512$	$7 \times 7 \times 512$	$3 \times 3 \times 512$, 2
Conv13	$7 \times 7 \times 512$	$7 \times 7 \times 1024$	$1 \times 1 \times 512 \times 1024$, 1
WE	$7 \times 7 \times 1024$	$7 \times 7 \times 1024$	
Conv14 dw	$7 \times 7 \times 1024$	$7 \times 7 \times 1024$	$3 \times 3 \times 1024$, 1
Conv14	$7 \times 7 \times 1024$	$7 \times 7 \times 1024$	$1 \times 1 \times 1024 \times 1024$, 1
Avg Pool	$7 \times 7 \times 1024$	$1 \times 1 \times 1024$	Pool 7×7, 1
FC	$1 \times 1 \times 1024$	$1 \times 1 \times 128$	1024×128, 1
L2	$1 \times 1 \times 128$	$1 \times 1 \times 128$	

Table 3
Resource per layer type.

Type	Multi-adds (%)	Parameters (%)
Standard convolution	1.7964	0.0379
Depthwise convolution	8.6451	5.8887
Pointwise convolution	89.4210	57.6349
Weight evaluation	0.1156	30.6748
Fully connected	0.0217	5.7634

learning. Weight evaluation module can make the feature map fuse global information, especially for the first few layers in the network structure. The feature maps of first few layers the have small receptive fields and cannot obtain global information. The last average pooling layer reduces the dimension of the feature map to 1×1 dimension before the fully connected layer. The LightFace model removes the softmax layer used for classification in the traditional network. It calculates L2-norm of the output of the full-connection layer and normalizes L2-norm to obtain feature representations. Here, we set the value of the L2-norm to 1, then all the features of the image will be mapped to a hypersphere. Based on this feature, we use the triple loss to optimize features. According to the distance of points in the feature space, we can indicate whether the two pictures are of the same kind.

The LightFace model is trained in Tensorflow and uses an asynchronous gradient descent algorithm to update the parameters. We set the threshold to 0.2. Unlike training large models, we use less regularization and data augmentation techniques because small models are not easy to overfit. In the process of training, we take an online approach to speed up convergence. We select all positive image pairs in the mini-batch and the most difficult negative image pair. In order to avoid being trapped in the local optimum during the training process, we choose an image pair that satisfies the distance between the positive sample P and the sample A less than the distance between the negative sample N and the sample A.

The LightFace model is based on depthwise separable convolution and has fewer Mult-Adds computation. However, this is not enough for increasing the speed of network computing. We must ensure that these operations can be implemented efficiently. According to the principle of depthwise separable convolution, the calculation amount of the depthwise convolution layer is $D_k \times D_k \times M \times D_F \times D_F$, and the calculation amount of the pointwise convolution layer is $1 \times 1 \times M \times N \times D_F \times D_F$. N is larger than $D_k \times D_k$, which is 3×3 in LightFace. So, the main calculation amount in the model is in the pointwise convolution operation. Usually, highly optimized general-purpose matrix multiplication GEMM functions are used to implement convolution operations, and 1×1 convolution operations can be directly implemented by GEMM without reordering in memory. Therefore, using GEMM can achieve high-efficiency operations in the network. As shown in Table 3, about 95% of LightFace's calculation time is spent in pointwise convolution operations, however, the weight evaluation module only adds 0.11% of the computation, with little impact on the speed of the network. The overall parameter quantity is much lower than most other models. In LightFace, most of the parameters are in the point convolutional layer, and the second is in the Weight Evaluation module. The increase of parameters in the Weight Evaluation module is worthwhile, which can improve the network recognition accuracy. The performance will be demonstrated in experiments in Section 5.

3. Face recognition with privacy preservation

In this section, we describe the face recognition algorithm with privacy preservation in detail. We will introduce how to use data to train models and generate labels with privacy preservation. Then we will introduce how to train the released model with high accuracy and will ensure that it does not reveal privacy.

3.1. Differential privacy

Privacy refers to information that individuals, organizations, and other entities do not want to be known by others. For example, personal salary, medical records, etc. Although a variety of methods based on k-anonymity and partitioning privacy preservation frameworks have emerged, the differential privacy [11] protection technology is still recognized as a relatively strict and robust protection model. This protection method ensures the insertion or deletion of a record in a dataset does not affect the output results. In addition, the protection model does not care about the background knowledge of the attacker. Even if the attacker has mastered all the records except for a certain record, the privacy of the unknown record cannot be disclosed. The formal definition of differential privacy is as follows.

Definition 1. There is a random algorithm M, domain D, and range R satisfy (ε, δ)-differential privacy. Then, for any two input neighboring datasets d, $d' \in D$, and output subsets $S \subseteq R$, satisfies inequality (4):

$$Pr[M(d) \in S] \leq e^{\varepsilon} Pr\left[M\left(d'\right) \in S\right] + \delta \tag{4}$$

Where, the probability $Pr[\cdot]$ is controlled by the randomness of the algorithm M and represents the risk of privacy leakage. The parameter ε indicates the degree of privacy preservation, and with ε is smaller, the degree of privacy preservation is higher.

From Definition 1, it can be seen that differential privacy technology limits the impact of any record on the output of the algorithm. This definition is to ensure that the algorithm satisfies the ε– differential privacy from a theoretical perspective. In practical applications, we can implement differential privacy preservation by adding noise mechanisms.

3.2. Bayesian generative adversarial network

Generative adversarial networks have emerged as a powerful framework for learning generative models of arbitrarily complex data distribution [15,21]. The Bayesian GAN proposed by Saatchi [27] adopts a practical Bayesian formula to generate anti-network training, which can conduct unsupervised and semi-supervised learning. By deploying an expressive posterior mechanism to the parameters in the generator, Bayesian GAN can avoid pattern collisions and produce deterministic and diverse candidates. In some existing benchmarks, such as SVHN, CelebA and CIFAR-10, Bayesian GAN can provide the best semi-supervised learning quantification results, far more than DCGAN, Wasserstein GANs and other models.

In Bayesian GAN, the iterative sampling of conditional posterior probabilities shown in Eqs. (5) and (6) can be used to infer the weight vectors θ_g and posterior probabilities of θ_d of generator and recognizer:

$$p(\theta_g|z, \theta_d) \propto \left(\prod_{i=1}^{n_g} \sum_{y=1}^{K} D\left(g\left(z^{(i)}; \theta_g\right) = y; \theta_d\right)\right) p(\theta_g|\alpha_g) \tag{5}$$

$$p(\theta_d|z, X, Y_s, \theta_g) \propto \prod_{i=1}^{n_g} \sum_{y=1}^{K} D\left(x^{(i)} = y; \theta_d\right) \times \prod_{i=1}^{n_g} D\left(G\left(z^{(i)}; \theta_g\right) = 0; \theta_d\right)$$
$$\times \prod_{i=1}^{n_g} \left(D\left(x_s^{(i)} = y_s^{(i)}; \theta_d\right)\right) p(\theta_d|\alpha_d) \tag{6}$$

Where, K represents the number of classes, N_s represents the number of samples with labels, z represents white noise, n_g and n_d

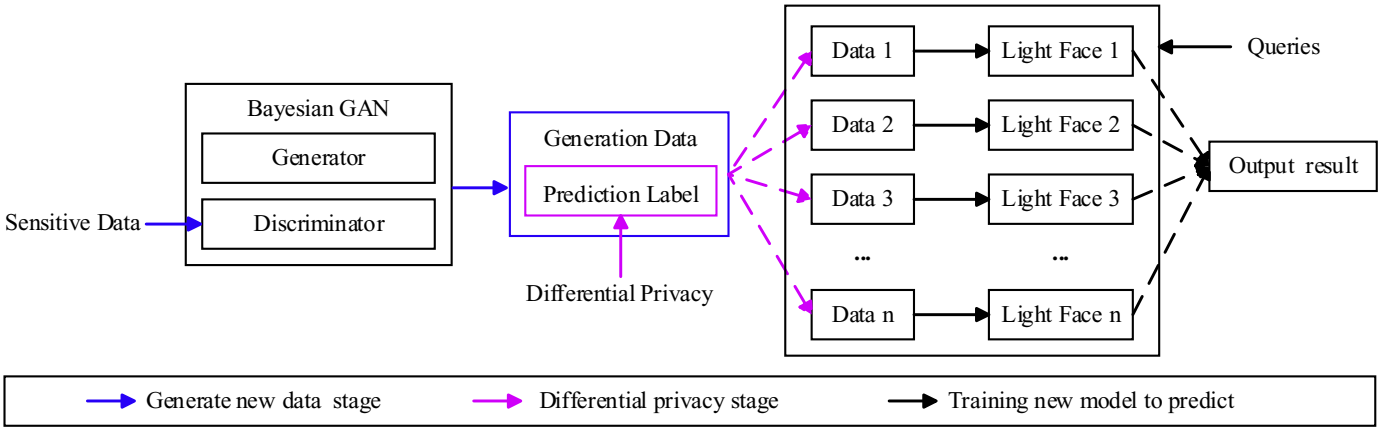


Fig. 2. Overview of the approach: (1) Bayesian GAN is trained to generate data by using sensitive data; (2) Differential privacy is used to disturb the prediction label; (3) generation data and noised label are used to train the ensemble model to provide the queries.

respectively represent the number of small batch training samples of the generator and recognizer, θ_g and θ_d represent the weight vector of the generator and recognizer, $p(\theta_g|\alpha_g)$ and $p(\theta_d|\alpha_d)$ represent the prior probability of the parameters of generator and recognizer under the hyper-parameters α_g and α_d, $D(x^{(i)} = y^{(i)}; \theta_d)$ represents the probability that the sample $x^{(i)}$ belongs to the tag $y^{(i)}$, and 0 represents the class tag of the generator-generated sample.

The formula (7) can be used to calculate the predicted distribution of the output y_* when the input is X_*:

$$p(y_*|X_*, D) = \int p(y_*|X_*, \theta_d)p(\theta_d|D)d\theta_d \approx \frac{1}{T}\sum_{k=1}^{T} p\left(y_*|X_*, \theta_d^{(k)}\right), \theta_d^{(k)} \sim p(\theta_d|D) \tag{7}$$

In Bayesian GAN, dynamic gradient Hamilton Monte Carlo is used to marginalize the weights in the generation network and recognition network. This method of obtaining results is very straightforward and achieves good performance without any intervention.

3.3. Face recognition algorithm with differential privacy

In the researches of deep learning privacy preservation, Papernot et al. [22] proposed a general machine learning strategy Private Aggregation of Teacher Ensembles (PATE). PATE provides privacy preservation for training data that meets differential privacy. Besides, PATE is independent from learning algorithms and is applicable to multiple deep learning models. PATE is also a relatively complicated learning strategy, and the network structure of the existing face recognition model is relatively complex. If PATE is combined with the face recognition algorithm, the scale and calculation amount of the model will be very huge. In this article, we use the lightweight face recognition algorithm proposed in Section 2 to propose a lightweight face protection algorithm with privacy preservation. The specific algorithm is shown in Fig. 2.

In our proposed learning strategy, to protect users' data privacy, Bayesian GAN is first trained using sensitive data, and generate new data which is not sensitive. Then, the generative data is fed into the discriminator of the generated model, and we can get the prediction probability of each data. The generative data with the prediction probability labels is sent to the subsequent data process.

In the privacy preservation stage, we want to add uncertainty to the probability labels of generative data to protect privacy. In our experiment, we add noise to the labels which satisfies the Laplacian distribution. The dimension of label is $k + 1$, and the last dimension represents the probability of the fake data. We only take the pre-k dimensional of labels, P_i represents the probability of the data belong to the i_{th} class, and then we add the noise on P_i as shown in Eq. (8). In Eq. (8), γ represents the privacy preservation parameter, $Lap(\frac{1}{\gamma})$ represents the Laplacian distribution of the location of 0, the scale of $\frac{1}{\gamma}$, P_i' represents the probability value after adding noise. The parameter γ indicates the strength of the privacy preservation we can provide. The larger is γ, the stronger the privacy preservation is provided. At the same time, the accuracy of the label will decrease. Therefore, it is important to set the right size of the noise. Finally, we normalize P_i' to obtain P_i'', and generate the final noisy label, as shown in Eq. (9).

$$P_i' = P_i + Lap\left(\frac{1}{\gamma}\right) \tag{8}$$

$$P_i'' = \frac{P_i'}{\sum_{i=1}^{k} P_i'} \tag{9}$$

We use the data with noisy labels to train the published model. To improve the accuracy of the model, we use ensemble learning approach to train models for external access. During training, we first draw bootstrap samples of size m from the data with noisy labels, and then use bootstrap samples to train one LightFace model. We use the same strategy to train other $n-1$ LightFace models, and the time complexity for training the ensemble model is O(nm) which is linear complexity. During testing, we aggregate the output of the sub models as the output of the input data, as shown in Eq. (10). In Eq. (10), $f(\vec{x})$ represents the final output of the ensemble model, $\vec{x}$ represents the input data, $n_j(\vec{x})$ represents the number of results $\vec{x}$ belong to the class j. This voting mechanism is very similar to random forests, and we take majority principle to get the result.

$$f(\vec{x}) = arg\max_j \left\{n_j(\vec{x})\right\} \tag{10}$$

In our proposed privacy policy, the privacy data is only used to train the GAN, and is not used to train the released model. Therefore, even if the parameters of the external model are published, the adversary cannot obtain sensitive data through the reconstruction attack. At the same time, the data for training published model is non-sensitive data generated by the GAN model, and adding noise to the label can disturb the reconstructed attack, so that the reconstructive attack cannot recover clear images of sensitive face images. In order to maintain the accuracy of the model, we also use ensemble learning to train the final released model, which can ensure that the model can maintain a high accuracy when implementing privacy preservation.

4. Privacy analysis

In this section, we analyze privacy loss in face recognition model with privacy-protected. We use moment accountant to track the privacy loss of the model when training the published model and provide a very accurate privacy loss boundary.

4.1. Moment accountant

Moment accounting is a method proposed by Abadi et al. [1] to accurately calculate privacy loss. First, according to the definition of differential privacy, we can get a method for calculating privacy loss, as shown in formula (11).

Definition 2. Assuming *aux* represents an auxiliary input. For an output $o \in R$, privacy loss can be defined as:

$$c\left(o; M, aux, d, d'\right) \triangleq \log \frac{Pr[M(aux, d) = o]}{Pr[M(aux, d') = o]} \tag{11}$$

Definition 3. Assuming M: $D \to R$ is a random algorithm, and d and d' are adjacent datasets. Assuming *aux* represents an auxiliary input, then time accounting is defined as:

$$\alpha_M(\lambda) \triangleq \max_{aux,d,d'} \alpha_M\left(\lambda; aux, d, d'\right) \tag{12}$$

Where, $\alpha_M(\lambda; aux, d, d') \triangleq \log E[\exp(\lambda C(M, aux, d, d'))]$ is the time generation function of the random variable of privacy loss.

At the same time, the following important properties have also been demonstrated in [1, 22].

Theorem 1.

1. *Composability: Assuming algorithm M consists of a series of adaptive algorithms* $M_1, \ldots, M_k$, *where* $M_i : \prod_{j=1}^{i-1} R_j \times D \to R_i$. *For any output sequence* $o_1, \ldots, o_{k-1}$ *and any* λ:

$$\alpha_M\left(\lambda; d, d'\right) = \sum_{i=1}^{k} \alpha_{M_i}\left(\lambda; o_1, \ldots, o_{i-1}, d, d'\right) \tag{13}$$

2. *Tail bound: For any* $\varepsilon > 0$, *the algorithm M for formula (12) satisfies* (ε, δ)*– differential privacy.*

$$\delta = \min_{\lambda} \exp(\alpha_M(\lambda) - \lambda\varepsilon) \tag{14}$$

Another important formula for the privacy calculation of the aggregated model is as follows:

Theorem 2. *Assuming two adjacent datasets d and d′, the corresponding number of votes* n_j *for each class will differ by a maximum of one. Assuming M is an algorithm for recording* $\arg\max_j\{n_j(\vec{x}) + Lap(\frac{1}{\gamma})\}$, *then M satisfy* $(2\gamma,\ 0)$ *– differential privacy. And, for any l, aux, d and d′,*

$$\alpha\left(l; aux, d, d'\right) \le 2\gamma^2 l(l+1) \tag{15}$$

4.2. Privacy analysis of face recognition model

We track the loss of privacy during training by using moment accountant. At each step of the training, we generate a tag with privacy preservation, which satisfies $(2\gamma,\ 0)$– differential privacy. After T steps, the algorithm will satisfy $(4T\gamma^2 + 2\gamma\sqrt{2T\ln\frac{1}{\delta}}, \delta)$ – differential privacy [22]. This is quite large. To solve this problem, Papernot et al. [22] proposed a method to provide more rigorous boundaries for privacy loss. When the number of LightFaces is very large, the result of voting is the same as minority obeys most principle, which requires a small loss of privacy. The following theorem is proved in [22].

Table 4
Mult-Adds and parameters comparison to other models.

Model	Million mult-adds	Million parameters
LightFace	**572**	**4.7**
MobileNet	568	4.2
FaceNet	1600	7.5
GoogleNet	1550	6.8
VGG 16	15,300	138
ResNet-50	4100	25.5

Theorem 3. *Assuming M Satisfaction* $(2\gamma,\ 0)$ *– differential privacy, satisfies* $q \ge Pr[M(d) \ne o^*]$ *for arbitrary output* o^*. *Assuming l,* $\gamma \ge 0$ *and* $q < \frac{e^{2\gamma}-1}{e^{4\gamma}-1}$, *then, for any aux, d and d′, M satisfy:*

$$\alpha\left(l; aux, d, d'\right) \le \log\left((1-q)\left(\frac{1-q}{1-e^{2\gamma}q}\right)^l + q\exp\left(2\gamma l\right)\right) \tag{16}$$

Theorem 4. *Assume n represents label fraction vector of the data set d, for all j, satisfies* $n_{j^*} \ge n_j$. *So,*

$$Pr[M(d) \ne j^*] \le \sum_{j \ne j^*} \frac{2 + \gamma\left(n_{j^*} - n_j\right)}{4\exp\left(\gamma\left(n_{j^*} - n_j\right)\right)} \tag{17}$$

Based on these properties, we can provide upper limit q for a specific fractional vector n to constrain specific moments. Then, we calculate these moments for some of λ, and we can get a privacy margin smaller than the moment accountant.

5. Experimental results and analysis

In this section, we first discuss the effects of lightweight face recognition model and compare the results with other models. Meanwhile, we evaluate the accuracy of the proposed face recognition algorithm with privacy preservation.

5.1. Reduce model volume and computational density

We compare the proposed LightFace model with other models that show good performance on face recognition. As shown in Table 4, the computation and the model size of FaceNet and GoogleNet are about 1.5 times as large as LightFace. The computation and model size of VGG 16 are about 30 times larger than LightFace. MobileNet has the least amount of Multi-Adds and parameters. Compared with Mobilenet, LightFace is slightly increased, and LightFace satisfies the standard of lightweight model.

In order to prove the generalization performance of the model, we choose to test the model performance on different datasets, such as LFW [19], YTF [34] and SFC [3] datasets. The pictures of the data in SFC have more changes in quality, illumination and character expression than LFW and YTF. The pictures of LFW and YTF are mostly taken by professional cameras. The SFC dataset includes 4.4 million labeled faces from 4030 people each with 800–1200 faces, the LFW dataset consists of 13,323 web photos of 5749 celebrities, the YTF dataset collects 3425 YouTube videos of 1595 subjects, we extracted 5 frames in each video and made up the dataset in our experiment.

We performed a unified processing of the images in the dataset. We generated a tight bounding box around each face by running a face detector on each image. The cropped face images were adjusted to the input size of the corresponding network. The resolution of the input image in the experiment was 224×224. In our experiment, the model is trained on 4 GPUs. The loss function of the model is defined as triplet loss. Data enhancement uses three strategies: random cropping, horizontal symmetry, and rotation angle $[-30, 30]$. The face feature dimension is set to 128. During the process of network parameter updating, the weight decay is set to

Table 5
verfication rate comparison to other models on different datasets.

Model	LFW	YTF	SFC
LightFace	**84.7% ± 1.9**	**82.7% ± 2.0**	**84.0% ± 1.7**
MobileNet	81.9% ± 1.2	80.4% ± 1.6	81.6% ± 1.4
FaceNet	84.5% ± 1.6	82.7% ± 1.8	83.8% ± 1.4
GoogleNet	81.7% ± 1.6	80.2% ± 1.9	81.4% ± 1.5
VGG 16	82.8% ± 1.9	81.1% ± 2.2	82.1% ± 1.8
ResNet-50	84.1% ± 1.7	82.4% ± 2.0	83.5% ± 1.5

Table 6
the Evaluation of the WE module on different dataset.

Model	LFW	YTF	SFC
LightFace	**84.7% ± 1.9**	**82.7% ± 2.0**	**84.0% ± 1.7**
LightFace (without WE)	81.7% ± 1.6	80.6% ± 1.8	81.8% ± 1.4

Table 7
The evaluation of the Depthwise separable convolutions on LFW dataset.

Model	Million mult-adds	Million parameters	Performance accuracy
LightFace	572	4.7	**84.7% ± 1.9**
LightFace (with standard convolution)	770	28.3	85.9% ± 2.0

1e−5, the momentum is set to 0.9, the initial learning rate is set to 0.001, and the batch_size is set to 64. This paper defines the verfication rate as the probability of the same type of face image are judged to be the same class, and the false accept rate (FAR) is the probability that the face images of different identities are judged to be the same class. The similarity for each pair of faces is evaluated by the L2-distance. The smaller is the L2-distance, the higher similarity is between the two face images. We trained the LightFace, MobileNet, FaceNet, GoogleNet and VGG 16 models using datasets above separately, and obtained the model accuracy at FAR of 0.0001. The experimental results are shown in Table 5.

We can learn from Table 5 that compared with GoogleNet and VGG16, LightFace not only reduces the amount of Multi-Adds and parameters, but also increases the recognition accuracy by 2.5–3% and 1.6–1.9%, respectively. As we all known, FaceNet has great performance on face recognition problem. LightFace has a 0.2% improvement at model accuracy than FaceNet on the LFW and SFC datasets with fewer parameters and Multi-Adds. LightFace also has a 2.3–2.8% higher accuracy than the lightweight network MobileNet. As above, LightFace achieves the highest accuracy on their different datasets.

To quantify the impact of WE module in the model, the performance of the model is tested on different data sets. LightFace is compared with the model from which the WE module was removed, and the experimental results are shown in Table 6. Under different test datasets, the WE module increase the model verification rate by 2.1–3.0%.

In order to quantify the impact of the depthwise separable convolution in the model, the performance of the model is tested on the LFW. Depthwise separable convolutions in LightFace are replaced with the traditional convolution layer, which is compared with original model. The experimental results are shown in Table 7. Depthwise separable convolution greatly reduces the amount of calculations and parameters of the model in the case of slightly decreasing the verification rate of the model.

To verify the performance of the model on mobile devices, we conducted a series of experiments on the Nexus 6p. The operating system of the device is LineageOS 15.1 (Android 8.1.0), the CPU is Snapdragon 810 (Cortex-A57 2.0 GHz × 4 + Cortex-A53 1.55 GHz × 4), and the RAM size is 3 G. In the mini-caffe framework [16], LightFace, MobileNet and SqueezeNet inference run 23 times respectively in a loop, the results of the first three times were removed, and the average of the running time was calculated. VGG16 run 9 times in a loop, the first result is removed, and the average of the time is calculated. In order to prevent the temperature of the machine from affecting the experimental results, the machine will sleep for 1 min before each benchmark run. The results of the experiment are shown in Table 8. Compared with the lightweight models SqueezeNet and MobileNet, LightFace is slightly lower, however, it is much quicker than VGG 16. Therefore, LightFace is suitabl e for running on the mobile devices.

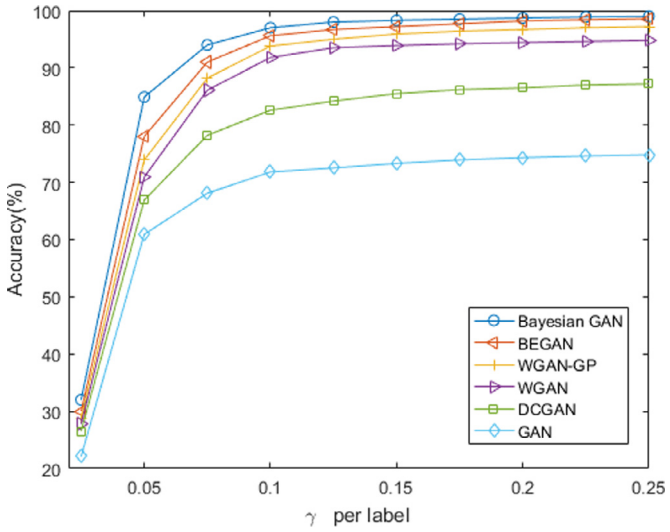


Fig. 3. Accuracy of the publish model under different GAN and varying γ value per label.

5.2. Generate dataset for training

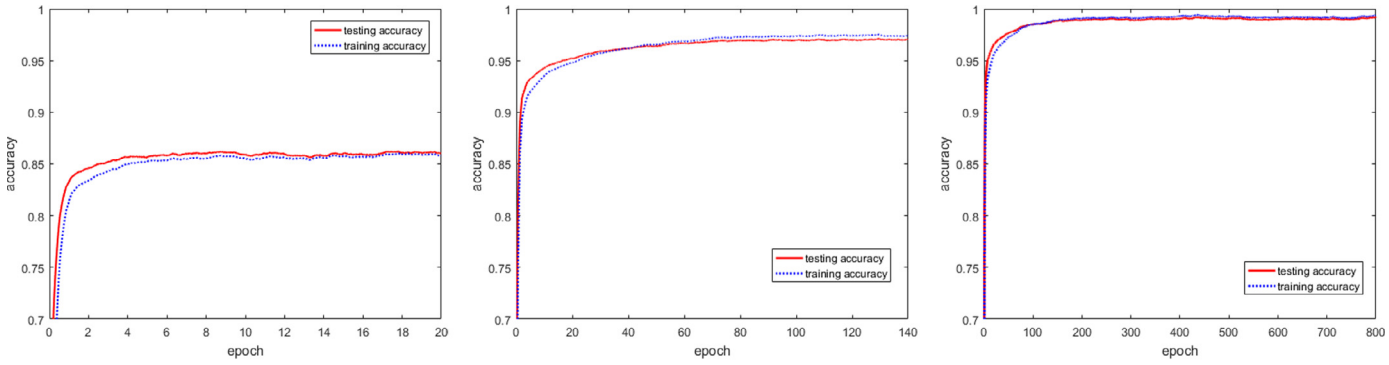
In our proposed privacy policy, we generate a dataset that is same distributed with sensitive data through generative adversarial network. Then, we use the generative data to train the released model. GAN has problems including unstable training, gradient disappearance, and collapse mode. To solve these problems, researchers have developed many variants of GAN, such as DCGAN, WGAN, WGAN-GP, BEGAN and Bayesian GAN. We use Bayesian GAN to generate training data in our proposed privacy policy. It deploys an expressive posterior to the parameters in the generator to effectively avoid mode-collapse, produces measurable and diverse candidate samples, and provides the best semi-supervised learning quantification results on some existing benchmark tests. We evaluate GAN and five variants of GAN in our problem, include Bayesian GAN. The experiments conduct under different size of noise, and the model accuracy are shown in Fig. 3.

We trained the model under the tensorflow framework and optimized model with asynchronous gradient descent algorithm. We performed experiments on the SFC dataset, trained each type of GAN to generate 500 face images for each sample, and cropped the face region for each image by detecting face key points. In the experiment, we set $n = 50$, and γ varies between 0.01 and 0.25.

From Fig. 3, we can see that using Bayesian GAN to generate training data achieves higher accuracy than DCGAN, WGAN and GAN, with varying γ. When the noise size is relatively small, the Bayesian GAN performs better than WGAN-GP, and nearly same with BEGAN. With the size of noise increasing, the Bayesian GAN performs better than BEGAN. Therefore, the data generated by Bayesian GAN is more diverse, more realistic, and closer to the distribution of real samples.

Table 8
Speed and performance comparison on mobile device.

Model	Time (ms)	CPU usage (%)	Power consumption (mW)	Performance per watt (1000 fps/w)	Memory usage (M)
LightFace	290.2	99.1	4438	0.937	54
SqueezeNet	125.9	98.6	4174	1.903	19
MobileNet	279.5	99	4358	0.821	52
VGG16	3418.3	99.5	4298	0.068	707



(1) Large noise (2) Medium noise (3) Small noise

Fig. 4. Results on the accuracy for different noise levels on the SFC dataset.

5.3. Generate labels with privacy preservation

In our proposed privacy strategy, we add noise to the data labels generated by the Bayesian GAN network and train the model with the generative data. We select γ of 0.05, 0.1, and 0.2, which stands for large noise, medium noise, and small noise, respectively. Under different size of noise, we evaluate the accuracy of model and the results as shown in Fig. 4.

From Fig. 4, we learn that as the noise intensity increases, the accuracy of the label decreases and the accuracy of the model decreases. The noise size not only effects the accuracy of the model, but also matters the strength of privacy preservation. It is essential to set the right size of noise. In each plot, we demonstrate the evolution of the model's training accuracy and test accuracy as epoch increases. We have 86%, 97%, and 99% accuracy for large noise, medium noise, and small noise, respectively. In order to protect data privacy while ensuring that the model still has high accuracy, we chose γ of 0.15 in the subsequent experiments.

Although using the generative data to train LightFace protects data privacy, the accuracy of the model also decreases a lot. So, to improve the accuracy of the model, we use an ensemble learning strategy to train the published model. To show our model has good generalization capabilities across multiple datasets. We generate the same amount of data with noisy labels on the LFW and YTF datasets through the same strategy and use the generated data to train the released model. We extract n data from the dataset for training n models with bootstrap sampling, and adopt the majority principle to aggregate the output of the n models to get the final result. Under different n, we obtain the accuracy of the model, the results are shown in Fig. 5.

From Fig. 5, with the increase of n, the accuracy of the model increases, and ecentually converges to a stable value after $n = 70$. The n is the number of LightFaces we use in ensemble learning and it represent the size of the whole model. To balance the size and accuracy of the model, we finally set $n = 50$

In order to prove effectiveness of strategy, we compare our model with the other two state-of-the-art privacy preservation strategies: DPSGD [1] and PATE [22] respectively. We train the model in experiments with the optimal parameters provided in the paper. In order to prove that our privacy preservation strategy is not only applicable to specific dataset, we conducted comparative experiments on three datasets. The experimental results are shown in Table 9.

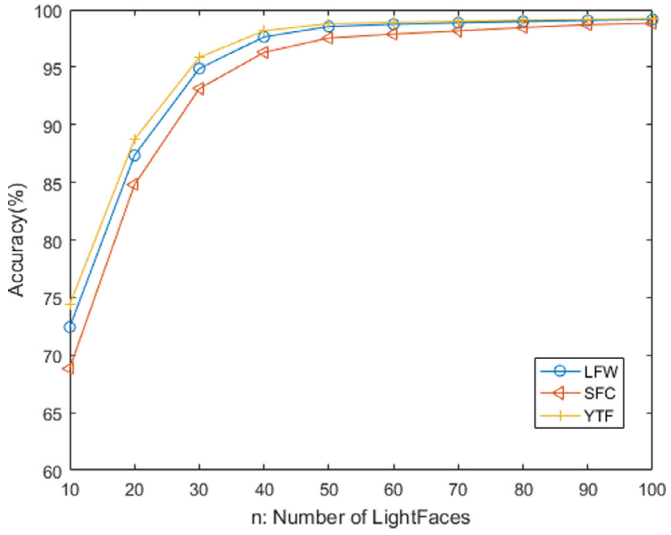


Fig. 5. Accuracy of the publish model with different number of LightFaces.

Table 9
Different privacy preservation method in different dataset.

Dataset	DPSGD (%)	PATE (%)	**Our method** (%)
LFW	97.5	98.0	**98.5**
YTF	95.1	95.7	**96.2**
SFC	96.4	97.2	**97.8**

As can be seen from Table 6, our privacy preservation strategy achieves the highest accuracy on all datasets. Our algorithm is about 1% higher accuracy than DPSGD on each of the datasets, and about 0.5% higher than PATE. Besides, compared with PATE, the size of our model is smaller.

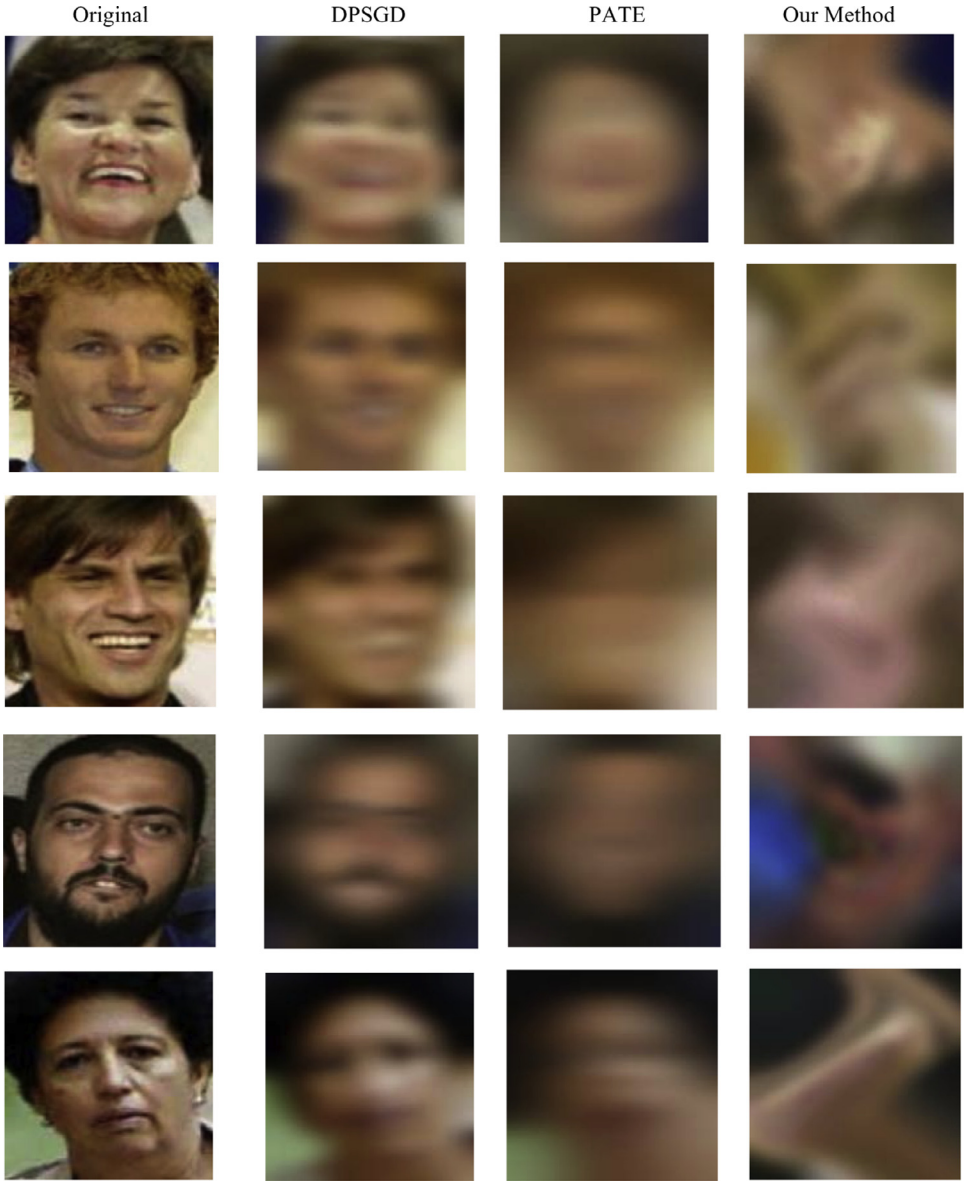


Fig. 6. The result of reconstruction attacks under three privacy policies. The first column is the original pictures, the second column is results through DPSGD privacy preservation algorithm, the third column is results through PATE privacy preservation strategy, and the last column is results through our privacy preservation method.

We perform reconstruction attacks on the models trained with the three different strategies, and select some images from the results of the attack, as shown in Fig. 6.

From the results shown in Fig. 6, we can intuitively learn that the model trained using our proposed privacy preservation strategy perform better under reconstruction attacks. Compared with the other two methods, the recovered face images from our published model are more blurred. We perform face recognition experiments on the images recovered under the three privacy preservation strategies, and analysis weather the recovered images can be recognized. Under the DPSDG algorithm, the accuracy of face recognition generated by the attack is 73%, PATE is 59%, and our algorithm is 52%. It can be seen that our privacy preservation strategy is superior to the other two privacy preservation algorithms.

In Table 10, we show different accuracies of the corresponding models under different level of the differential privacy. Compared with models trained without using noisy data, the accuracy of our published models is only reduced by 0.34%. When the probability of failure is fixed at 10^{-5}, it produces a strict privacy boundary $\varepsilon = 2.05$. Therefore, we prove that while protecting data privacy, our published model still gets a high recognition accuracy.

Table 10
Utility and privacy of the published model on SFC dataset.

ε	δ	Non-private baseline (%)	Our publish model accuracy (%)
2.05	10^{-5}	99.48	99.14
4.15	10^{-5}	99.48	99.14
8.13	10^{-5}	99.48	99.31

6. Conclusion

To protect the privacy of face recognition, we propose a face recognition algorithm with privacy preservation based on LightFace. LightFace is based on depthwise separable convolution and weight evaluation module. The experiment results show that: (1) we module increase the model verification rate. (2) Depthwise separable convolution greatly reduces the amount of calculations and

parameters of the model in the case of slightly decreasing the verification rate of the model. (3) The combination of these two methods make LightFace achieves high accuracy while maintaining small size. We also use Bayesian GAN to generate training data with noisy label which provide a formal bound on users' privacy loss and use ensemble learning method to improve the accuracy of the model. The experiment results show that: (1) although using the generative data to train LightFace protects data privacy, the accuracy of the model also decreases a lot. So, we set $\gamma = 0.15$ to make a trade-off between accuracy and privacy. (2) Although using ensemble learning method improve the accuracy of the model, the size of the model also increase dramatically. So we set $n = 50$ to make a trade-off between accuracy and model size.

The key advantage of our face recognition algorithm is that it achieves high recognition accuracy while protecting data privacy. However, the ensemble learning method make our model difficult to train and that's why our model is hard to apply in the actual scenario. So, our future work focus on the parallelization of the model in order to reduce the training time.

Declaration of interests

None.

Acknowledgments

This work is supported in part by the State Grid Corporation of China under "Research and Application of Key Technologies for Open Source Software Security Monitoring" (SGFJXT00YJJS1800074).

Appendix

Table A.1
Bayesian gan parameter values.

Parameter	Value
Dim of z for generator	100
Num of gen features	128
Num of disc features	192
Minibatch size	32
NN weight prior std	1
Number of discrimitor weight samples	1
Number of MCMC NN weight samples per z	1
Number of supervised data samples	128
Train_iter	500
Learning rate	0.005
Learning rate decay	3

Table A.2
Lightface parameter values.

Parameter	Value
Minibatch size	64
Threshold γ	8
Threshold α	0.2
Learning rate	0.05
L2-norm	1

References

[1] M. Abadi, et al., Deep learning with differential privacy, in: Proceedings of the ACM Sigsac Conference on Computer & Communications Security, 2016.
[2] R. Bassily, A. Smith, A. Thakurta, Differentially private empirical risk minimization: efficient algorithms and tight error bounds, Comput. Sci. (2014) 464–473.
[3] B.C. Becker, E.G. Ortiz, Evaluating open-universe face identification on the web, in: Proceedings of the Computer Vision & Pattern Recognition Workshops, 2013.
[4] M. Bun, T. Steinke, Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. eprint arXiv:1605.02065, 2016: p. arXiv:1605.02065.
[5] Q. Cao, et al., VGGFace2: A dataset for recognising faces across pose and age. eprint arXiv:1710.08092, 2017: p. arXiv:1710.08092.
[6] J. Chen, et al., Correlated differential privacy protection for mobile crowdsensing, IEEE Trans. Big Data PP (99) (2017) 1-1.
[7] F. Chollet, Xception: Deep Learning with Depthwise Separable Convolutions. eprint arXiv:1610.02357, 2016: p. arXiv:1610.02357.
[8] S. Coretti, Secure multi-party computation, Inf. Secur. Commun. Privacy (2014).
[9] L. Du, H. Hu, Face Recognition Using Simultaneous Discriminative Feature and Adaptive Weight Learning Based on Group Sparse Representation, IEEE Signal Processing Letters 26 (3) (2019) 390–394.
[10] J.C. Duchi, M.I. Jordan, M.J. Wainwright, Privacy aware learning, J. ACM 61 (6) (2014) 1–57.
[11] C. Dwork, A. Roth, The algorithmic foundations of differential privacy, Found. Trends Theor. Comput. Sci. 9 (3–4) (2014) 211–407.
[12] M. Fredrikson, S. Jha, T. Ristenpart, Model inversion attacks that exploit confidence information and basic countermeasures, in: Proceedings of the ACM Sigsac Conference on Computer & Communications Security, 2015.
[13] A.G. Howard, et al., MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. eprint arXiv:1704.04861, 2017: p. arXiv:1704.04861.
[14] F.N. Iandola, et al., SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. eprint arXiv:1602.07360, 2016: p. arXiv:1602.07360.
[15] Y.Z. Ji, H.J. Zhang, Q.M.J. Wu, Saliency detection via conditional adversarial image-to-image network, Neurocomputing 316 (2018) 357–368.
[16] Y. Jia, et al., Caffe: Convolutional Architecture for Fast Feature Embedding. eprint arXiv:1408.5093, 2014: p. arXiv:1408.5093.
[17] J. Hu, et al., Squeeze-and-Excitation Networks. eprint arXiv:1709.01507, 2017: p. arXiv:1709.01507.
[18] T.A.M. Kevenaar, et al., Face recognition with renewable and privacy preserving binary templates, in: Proceedings of the Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID'05), 2005.
[19] E. Learned-Miller, et al., Labeled Faces in the Wild: A Survey, Springer, 2016.
[20] X. Li, et al., Selective Kernel Networks. eprint arXiv:1903.06586, 2019: p. arXiv:1903.06586.
[21] L.L. Liu, et al., Toward AI fashion design: an Attribute-GAN model for clothing match, Neurocomputing 341 (2019) 156–167.
[22] N. Papernot, et al., Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. eprint arXiv:1610.05755, 2016: p. arXiv:1610.05755.
[23] N. Papernot, et al., Scalable Private Learning with PATE. eprint arXiv:1802.08908, 2018: p. arXiv:1802.08908.
[24] O.M. Parkhi, A. V., A. Zisserman, Deep face recognition, in: Proceedings of the British Machine Vision Conference, 41, 2015, pp. 1–41.12.
[25] N. Phan, et al., Adaptive Laplace Mechanism: Differential Privacy Preservation in Deep Learning. eprint arXiv:1709.05750, 2017: p. arXiv:1709.05750.
[26] R. Ranjan, V.M. Patel, R. Chellappa, HyperFace: a deep multi-task learning framework for face Detection, landmark Localization, pose Estimation, and gender recognition, IEEE Trans. Pattern Anal. Mach. Intell. 41 (1) (2019) 121–135.
[27] Y. Saatchi, A.G. Wilson, Bayesian GAN. eprint arXiv:1705.09558, 2017: p. arXiv:1705.09558.
[28] F. Schroff, D. Kalenichenko, J. Philbin, FaceNet: a unified embedding for face recognition and clustering, in: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[29] S. Song, K. Chaudhuri, A.D. Sarwate, Stochastic gradient descent with differentially private updates, in: Proceedings of the Global Conference on Signal & Information Processing, 2014.
[30] Y. Sun, X. Wang, X. Tang, Deeply learned face representations are sparse, selective, and robust, in: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[31] L. Sweeney, k-anonymity: a model for protecting privacy, Int. J. Uncert. Fuzzin. Knowl. Based Syst. 10 (5) (2012) 557–570.
[32] Y. Taigman, et al., DeepFace: closing the gap to human-level performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, 2014.
[33] Y. Wang, D. Hatzinakos, Face recognition with enhanced privacy protection, in: Proceedings of the IEEE International Conference on Acoustics, 2009.
[34] L. Wolf, T. Hassner, I. Maoz, Face recognition in unconstrained videos with matched background similarity, in: Proceedings of the Computer Vision & Pattern Recognition, 2011.
[35] Z.Y. Yang, et al., P-2: privacy-preserving communication and precise reward architecture for V2G networks in smart grid, IEEE Trans. Smart Grid 2 (4) (2011) 697–706.
[36] Y. Sun, X. Wang, X. Tang, Deep Learning Face Representation by Joint Identification-Verification. eprint arXiv:1406.4773, 2014: p. arXiv:1406.4773.
[37] N. Zeng, et al., Image-based quantitative analysis of gold immunochromatographic strip via cellular neural network approach, IEEE Trans. Med. Imag. 33 (5) (2014) 1129–1136.
[38] N.Y. Zeng, et al., Deep belief networks for quantitative analysis of a gold immunochromatographic strip, Cognit. Comput. 8 (4) (2016) 684–692.
[39] N.Y. Zeng, et al., Facial expression recognition via learning deep sparse autoencoders, Neurocomputing 273 (2018) 643–649.
[40] X. Zhang, et al., ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. eprint arXiv:1707.01083, 2017: p. arXiv:1707.01083

Yuancheng Li received the Ph.D. degree from University of Science and Technology of China, Hefei, China, in 2003. From 2004 to 2005, he was a postdoctoral research fellow in the Digital Media Lab, Beihang University, Beijing, China. Since 2005, he has been with the North China Electric Power University, where he is a professor and the Dean of the Institute of Smart Grid and Information Security. From 2009 to 2010, he was a postdoctoral research fellow in the Cyber Security Lab, college of information science and technology of Pennsylvania State University, Pennsylvania, USA.

Yimeng Wang was born in 1994 in Shenyang, Liaoning Province, China. She received B.S. degree in information security from North China Electric Power University, Beijing, in 2016. Since 2016, she is a M.S. candidate in information security at North China Electric Power University. Her research interests include machine learning and artificial intelligence, big data analysis and privacy protection in deep learning.

Daoxing Li was born in 1995 in Beijing, China. He received B.S. degree in Information and Computing Science from North China Electric Power University, Beijing, in 2018. Since 2018, he is a M.S. candidate in information security at North China Electric Power University. His research interests include adversarial machine learning and security of artificial intelligence.