Journal of Information Security and Applications 93 (2025) 104149

Contents lists available at ScienceDirect

Journal of Information Security and Applications

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

IPMN: Invertible privacy-preserving mask network with intellectual property protection

Yang Yang [a,b], Xiangjie Huang [a], Han Fang [c,*], Weiming Zhang [d]

[a] Anhui University, Hefei, 230039, China
[b] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, 2300881, China
[c] National University of Singapore, Singapore
[d] University of Science and Technology of China, Hefei, 230027, China

ARTICLE INFO

Keywords:
Facial privacy
Intellectual property
Invertible mask network

ABSTRACT

Facial information is widely used in security fields like identity authentication. But the large number of facial images online makes them vulnerable to unauthorized capture, posing privacy and security risks. Existing face privacy protection methods aim to mitigate these risks. However, many of these methods lack reversibility, making it impossible to restore the original face when needed. Additionally, they often neglect model intellectual property (IP) protection, leaving methods vulnerable to unauthorized stealing. Therefore, to address the shortcomings of existing face privacy protection methods in IP protection, this paper proposes an invertible privacy protection mask network with IP protection. The proposed method consists of two main parts: facial privacy protection and IP protection. For facial privacy protection, the mask generator replaces facial features with other faces and generates the mask, which is then embedded with the watermark to generate the watermarked mask. This watermarked mask conceals the original face by the putting on mask network, and the original face can be restored by the putting off mask network. For IP protection, the watermark extractor network is a key component that can extract the watermark from images of the sender, receiver and attacker to verify the method's IP. Experimental results show that the proposed method has good effects in both privacy protection and IP protection, providing double security for face privacy protection.

1. Introduction

The recognition and importance of data have significantly increased due to the explosive growth of global data. Consequently, there has been a proliferation of data-related services and applications, leading to a heavy reliance on data in people's lives. Particularly, individuals are increasingly enthusiastic about uploading their daily photos containing facial images to online platforms for storage, sharing purposes, or utilizing facial information for tasks such as identity authentication. But the ease with which unauthorized individuals can misuse captured or published facial images poses a significant threat to personal privacy and security. ClearView AI reportedly collected personal photos without user consent to train the facial recognition system. This case demonstrates that publicizing facial images poses a serious privacy risk for users.

Consequently, a growing number of researchers are devoting their endeavors to improving the safeguarding of users' facial privacy. Currently, the main emphasis in facial privacy protection revolves around two fundamental techniques: face de-recognition [1–5] and face replacement [6–11]. Face de-recognition techniques involve introducing subtle and imperceptible modifications to facial images with the intention of deceiving unauthorized facial recognition systems. However, these methods are fragile when attackers adopt some defense methods, such as image filtering. A safer safeguard strategy is face replacement, which involves substituting the original face with either a synthetic or anonymized face. However, in today's social era, users need to recover the original facial information for authorized users (e.g., family members and friends) when they need to show the real face that shows the real state of the individual. Therefore, how to restore the original facial information for authorized users has become one of the challenges of current face replacement technology.

To address the aforementioned concerns, Yang et al. introduced an invertible mask network (IMN) as a means of protecting face privacy [12]. The IMN method effectively safeguards the user's original face and allows authorized users to restore it without any loss, achieving state-of-the-art performance. However, IMN's outstanding performance gives rise to IP infringement concerns when a seller distributes the model to multiple buyers. Since the requirements of different buyers

* Corresponding author.
E-mail address: fanghan@nus.edu.sg (H. Fang).

https://doi.org/10.1016/j.jisa.2025.104149

Available online 1 July 2025

may vary, the seller has the flexibility to customize the model for each buyer, resulting in multiple tailored copies of the model. However, safeguarding the intellectual property rights of each buyer's customized models becomes challenging in multi-buyer scenarios. Furthermore, we have observed that attackers can exploit deep learning models by stealing them and imitating their facial privacy protection capabilities. Specifically, attackers can generate extensive sets of input–output training pairs based on a trained black-box model and then train their own personal model using supervised learning techniques by treating the output of the target model as genuine labels. This poses a potential risk to the intellectual property rights of models acquired by buyers. Consequently, we are exploring methods to protect users' facial privacy while also preserving the intellectual property of these models.

Recently, there have been notable advancements in deep model IP protection. Some works aim to incorporate watermarks into the network weights [13,14] or predictions [15–18] while maintaining the performance of the original model. Specifically, in [13,14], a weight regularizer is introduced to the objective loss function to ensure that the learned weights adhere to a specific distribution. On the other hand, prediction watermarking techniques such as those proposed in [15–17] involve incorporating a set of special image triggers during training so that the learned network can classify them into predefined labels. As for future directions, there is an increasing focus on tailoring protection methods for domain-specific task models within deep learning model watermarking with an aim to enhance security. However, it should be noted that currently no dedicated watermarking method exists specifically for face protection methods like IMN.

Drawing inspiration from existing deep learning model watermarking frameworks and analyzing the characteristics of the IMN, we propose an invertible privacy-preserving mask network with intellectual property protection (IPMN). This network consists of two main components: Facial privacy protection and IP protection. In terms of facial privacy protection, our primary objective is to ensure that a user's facial information is protected while allowing authorized users to restore the original image. To protect intellectual property, we employ a spatial invisible watermark mechanism that verifies the IP from both the sender's and receiver's perspectives. The key component of this framework is the IP verifier, which can extract watermarks from various sources including the sender's watermarked masked face, the receiver's recovered watermarked mask, and the simulated mask generated by the stealing model. The convergence of these verifiable IPs serves as the core purpose for validating our model's IP. The experimental results prove the effectiveness of the proposed method. It not only preserves reversibility in the original facial data and enables lossless recovery by authorized users but also effectively verifies the IP associated with our facial privacy protection method.

The main contributions of this paper are as follows:

- To safeguard users' facial privacy while also providing robust protection for the IP of various buyer's models, this paper proposes an invertible privacy-preserving mask network with intellectual property protection (IPMN).
- This paper integrates a spatial invisible watermarking mechanism with facial privacy protection. Instead of directly embedding the watermark, we incorporate the watermark in an intermediate step of face privacy protection. This method allows the watermark to be extracted from images originating from various sources, thereby providing comprehensive protection for the model's intellectual property.
- Extensive experiments confirm the effectiveness of the proposed framework in achieving facial privacy protection and reversible recovery, offering comprehensive and resilient protection for the model's intellectual property.

2. Related work

2.1. Face replacement

With the rapid evolution of the internet, human connections have grown closer, and individuals are more inclined to share their personal photos online. However, this can inadvertently leak private information, particularly facial information. Therefore, researchers have been exploring face replacement methods as an alternative method to preserving facial privacy. The core of this method is to replace the face of the original identity in the image with another identity face thus preventing others from being illegally recognized. Existing face replacement methods can be broadly categorized into Structural Prior-Guided Models [6,19,20], Reconstruction-Based Models [7,21], StyleGAN-Based Models [8,9,22–25] and Diffusion Models [10,11].

Diffusion models have gained attention in recent years for their improved controllability and fidelity. Researchers have started exploring the possibility of diffusion models for face replacement tasks [10,11]. By leveraging the capabilities of diffusion models, these face replacement methods offer enhanced control, scalability, and fidelity compared to earlier techniques. However, these face replacement methods do not reversibly recover the original face information. Recently, Yang et al. first utilized the invertible neural network (INN) to build an invertible mask network (IMN) [12]. This method not only provides robust face privacy protection but also allows authorized individuals to restore the original face without any loss, thereby ensuring the security and confidentiality of facial information. However, in the IMN framework IP security attributes are not taken into account.

The method centered around face replacement is geared toward safeguarding the privacy of a user's facial features through the manipulation of their characteristics. However, these methods predominantly focus on shielding the user's facial privacy, overlooking the critical aspect of safeguarding the model's intellectual property rights. It is important to recognize that these methods do not adequately address the potential for malicious attackers to exploit the target model's functionality, thus infringing upon its intellectual property rights through the act of model theft.

2.2. Deep watermarking

The swift evolution of the Internet and multimedia technologies has substantially simplified the distribution and replication of multimedia content. However, these technological strides have inadvertently intensified the jeopardy faced by multimedia content copyright. In response to these challenges, deep watermarking has emerged as an evident solution to mitigate these concerns. The fundamental concept entails embedding a watermark into the cover image using an embedding algorithm. Subsequently, the watermark can be extracted using an extraction algorithm to establish ownership and provide evidence of copyright [26].

Zhu et al. were the first to propose an end-to-end learning architecture for robust watermarking, named HiDDeN [27]. This method aims to ensure robustness by augmenting the marked image with a differential noise layer. To be robust against non-differential JPEG distortion, Jia et al. proposed the method named MBRS, which switches randomly between simulated differential JPEG and real JPEG during small batch training. Xiang et al. proposed a generative adversarial framework based on embedded digital watermarking to mislead DNN models to output wrong classification results [28]. Furthermore, owing to the imperceptibility and extractability of watermarks, researchers are now exploring the utilization of deep watermark technology to safeguard the intellectual property rights of developers of machine learning models. Zhang et al. were the first to propose a model watermarking framework for protecting image processing models [18]. This method offers robust protection for the IP of image processing models, establishing a significant advancement in protecting their ownership.

To address the watermark removal problem, Wang et al. proposed an adversarial visible watermarking scheme to improve the resistance of visible watermarks to watermark removal methods [29]. In parallel, Cao et al. introduced the channel attention mechanism into the DCT domain and proposed a generalized screen-shooting robust image watermarking framework [30]. However, this method is not robust to stealing attacks.

To our knowledge, deep watermarking techniques have made significant progress in various domains, including image and video watermarking. The current research primarily concentrates on aspects such as the watermark's uniqueness, robustness, and universality. Nevertheless, the existing watermarking techniques often overlook the verification problem in the case of images from multiple sources. Therefore, drawing inspiration from existing watermarking technology and combining it with the IMN to propose an invertible privacy-preserving mask network with intellectual property protection(IPMN). Our method can extract the watermark from images of various sources, thereby offering comprehensive IP protection to our method.

3. Method

3.1. Threat model

With the rapid advancements in artificial intelligence technology, an increasing number of facial privacy protection models are being deployed in the cloud to offer effective privacy protection services to users through API interfaces. However, it has come to our attention that there are malicious attackers who attempt to carry out model-stealing attacks on these target models, aiming to steal the function of these target models. In a model stealing attack, the attacker targets the black box model of the victim and obtains a lot of input/output pairs of the victim's model through multiple queries. Leveraging this acquired data, the attacker builds a stealing model that is very similar to the functionality of the victim's model. The attacker utilizes this substitute model to provide services and generate revenue. However, this unethical behavior results in significant infringement of the IP owned by the model creator.

Therefore, as the owner of a facial privacy protection model, it is crucial to consider the following points:

1. **Visual Quality**: Ensure that the face privacy protection model maintains good visual quality. The result should appear natural and visually appealing, without any noticeable artifacts or distortions.
2. **Privacy Restoration**: When authorized, it is necessary to consider restoring the privacy of the original face. Users should be able to revert the result to its original state, ensuring privacy restoration when required.
3. **Intellectual Property Protection**: Safeguard the intellectual property rights of the model. To ensure comprehensive protection of the model's intellectual property rights, it is crucial to verify them from the viewpoints of the sender, receiver, and potential attackers.

Therefore, based on the above three needs, we propose an invertible privacy-preserving mask network with intellectual property protection(IPMN). As shown in Fig. 1, we will describe each from the perspective of the Sender, the Receiver, and the IP Verifier.

From the perspective of the Sender, the user needs to replace the protected face with another identity to protect the original facial privacy. Therefore, the mask generator is first used for face replacement. Subsequently, the Watermark embedding network embeds the watermark into the mask. Finally, the Putting on mask network puts the watermarked mask on the protected face, and the watermarked masked face is obtained.

From the Receiver's perspective, when required, the process involves restoring the watermarked masked face. This is achieved through the Putting off mask network, resulting in the recovery of both the recovered protected face and the recovered watermarked mask.

From the perspective of the IP Verifier, their role is pivotal in safeguarding the IP of both the Sender and Receiver models. To verify the IP, the correct watermark should be extracted from the watermarked masked face, the recovered watermarked mask, and the simulated mask by the Watermark extractor network.

Table 1 presents the notations used in this paper.

Table 1
Summary of notations in the paper.

Notation	Description
x_p	The protected face
x_r	The face used for replacing
x_m	The mask
w	The watermark
x_{wm}	The mask with watermark
x_{wmf}	The watermarked masked face
x'_{wm}	The recovered watermarked mask
x'_p	The recovered protected face
x_{sm}	The simulated mask generated by Stealing model
x_{clean}	The blank watermark image
MG	The Mask Generator
H	The Watermark embedding network
R	The Watermark extractor network
D	The Discriminator network
SM	The Stealing model
$\mathbf{M}$	The lost matrix
$\mathbf{N}$	The random matrix

3.2. Sender and receiver

For both the Sender and the Receiver, how to realize the naturalness of the mask and the recoverability of the original protected face are their primary concerns. In this section, we will elaborate on our method from both the Sender and Receiver perspectives. The following section describes the specific network models.

3.2.1. Mask generator

In order to attain facial privacy protection while maintaining high visual quality, we use the Mask Generator MG to replace the protected face based on the replaced face. Different from IMN, our method utilizes DiffFace [10] as the Mask Generator for face replacement. As shown in Fig. 1, DiffFace [10] is composed of training ID Conditional DDPM, sampling with facial guidance, and target-preserving blending. During the training process, the ID Conditional DDPM is trained to generate face images that possess the desired identity. The facial guidance enables the model to transfer the identity from the replaced face while preserving the protected face's attributes. To achieve desirable face exchange results, a target-preserving blending strategy is employed, which ensures the preservation of the target image's background while exchanging the faces. Compared to previous GAN-based methods, DiffFace [10] demonstrates superior advantages in terms of training stability, high fidelity, and controllability.

Hence, within the Mask Generator, we employ DiffFace [10] to substitute the protected face x_p with the replaced face x_r. We summarize the process of replacement as follows:

$$x_m = MG\left(x_p\ ,\ x_r\right) \tag{1}$$

It is important to note that various other face replacement techniques also perform effectively within our framework, which shows that our framework has strong generalization capabilities.

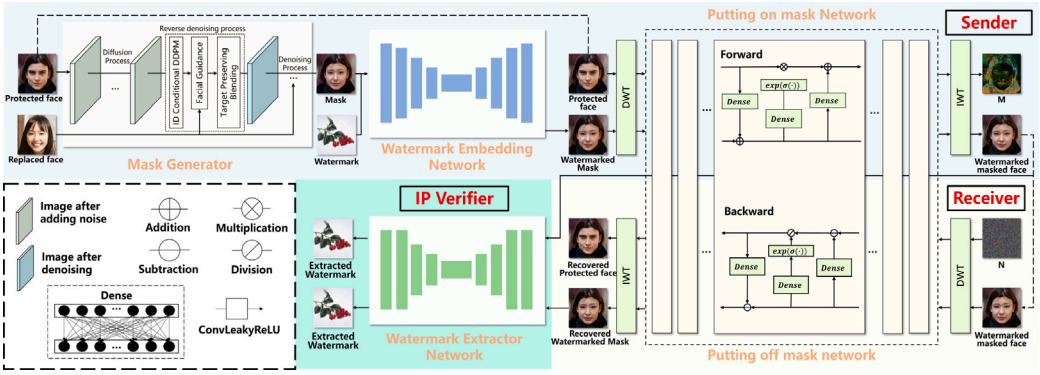

Fig. 1. Framework of invertible privacy-preserving mask network with intellectual property protection.

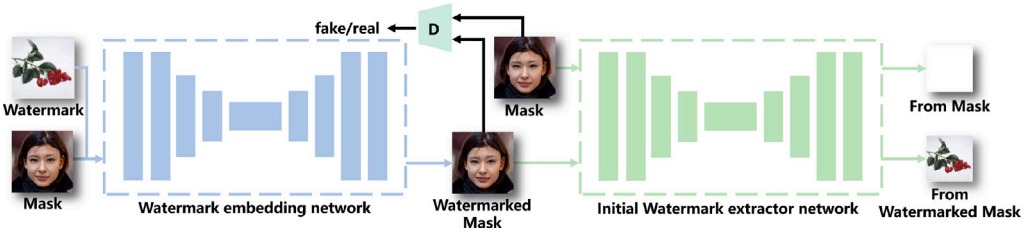

Fig. 2. Training strategy of Watermark embedding network.

3.2.2. Watermark embedding network

To effectively validate the IP of our method, we employ the watermark as a validation tool. We adopt a strategy that combines the spatially invisible watermarking mechanism with facial privacy protection. During this process, we encountered a decision-making challenge regarding selecting a suitable carrier for the watermark. We observed that embedding the watermark directly onto the protected face resulted in a degradation of visual quality and compromised the reversibility of the protection. Therefore, we chose to use the mask as the carrier for the embedded watermark and use the Watermark embedding network to generate the watermarked mask. Then, we embed the protected face as a secret message into the watermarked mask and obtain the watermarked masked face. The resulting watermarked masked face can be extracted watermark by the Watermark extractor network. This effectively verifies the IP of the Sender's model while preserving the visual integrity of the protected face, as perceived by the receiver.

The watermark used in this context differs from traditional watermarks in several vital aspects. Firstly, the watermark can be extracted from various sources, including the watermarked masked face, the recovered watermarked mask, and the simulated mask, all achieved through the Watermark extractor network. This multisource extraction capability offers robust safeguards for the intellectual property of the facial privacy preservation model. Furthermore, to ensure intellectual property protection for different model copies, our method can opt for distinct watermarked images to embed, which may incorporate specific buyer-related information to substantiate IP rights.

During training, to embed the watermark into the mask and subsequently extract it, we utilize the Watermark embedding network H and the Watermark extractor network R, training them jointly. The initial training strategy is illustrated in Fig. 2. Initially, the watermark w and the mask x_m are used as inputs to the Watermark embedding network H to obtain the watermarked mask x_{wm}. We require the x_{wm} to be as visually consistent as possible with the x_m. To enhance the image quality of the watermarked mask x_{wm}, we incorporate the Discriminator network D after the Watermark embedding network H, to minimize the domain disparity between the x_m and the x_{wm}. Considering that UNet [31] performs well in tasks where the output has the same attributes as the input, we have chosen to adopt UNet as the network structure for our Watermark embedding network H. We summarize the embedding process as follows:

$$x_{wm} = H\left(x_m\ ,\ w\right) \tag{2}$$

We provide a detailed explanation of the Watermark extractor network R in Section 3.3.

3.2.3. Putting on mask network

In order to ensure the reversible recovery of the facial information of the protected face, after generating the watermarked mask, the next step is to use the Putting on mask network to place the watermarked mask on the protected face and get the watermarked masked face.

Like IMN [12], we utilize the Invertible Neural Network (INN) to achieve protection and high-quality reversible recovery of the protected face. The INN is made up of an encoder and a decoder, where the encoder and the decoder are composed of K embedding blocks and recovery blocks, respectively. It is worth noting that the recovery blocks share the same network parameters as the embedding blocks. Thus the inputs during the encoding process of the INN can be restored almost without loss during the decoding. Therefore, in this paper, we refer to IMN [12] and use the Putting on mask network to put on the watermarked mask and use the Putting off mask network to achieve reversible recovery of the protected face, as illustrated in Fig. 1. The uniqueness of this method is that we use the watermarked mask with a hidden watermark as a carrier and embed the protected face in it to generate the watermarked masked face. This watermarked masked face can then be utilized as a source for verifying the IP of the Sender.

In the Putting on mask network, the protected face x_p is embedded into the watermarked mask x_{wm}. At first, a pair of the x_p and the x_{wm} are accepted as inputs. The x_p and x_{wm} are then separated into low-frequency wavelet subbands and high-frequency wavelet subbands by the Discrete Wavelet Transform (DWT), which are then fed into the invertible embedding module. The invertible embedding module consists of K invertible embedding blocks having the same architecture connected in series, as shown in Fig. 1. For the ith invertible embedding block, the formula is as follows,

$$x_{wm}^{i+1} = x_{wm}^{i} + Dense\left(x_p^i\right) \tag{3}$$

$$x_p^{i+1} = x_p^i \odot \exp\left(\alpha\left(Dense\left(x_{wm}^{i+1}\right)\right)\right) + Dense\left(x_{wm}^{i+1}\right) \tag{4}$$

where α is a sigmoid function multiplied by a constant factor served as a clamp, $\odot$ indicates the dot product operation and *Dense* indicates the dense block, which is proven to ensure good representation ability in hinet [32]. In the Kth invertible embedding block, the watermarked masked face $x_{textitwmf}$ and the loss matrix $\mathbf{M}$ are obtained by doing the Inverse Wavelet Transform (IWT) on the outputs. The process of Putting on the mask network is summarized below:

$$\left(x_{wmf}, \mathbf{M}\right) = f\left(x_p, x_{wm}\right) \tag{5}$$

3.2.4. Putting off mask network

When the Receiver receives the watermarked masked face x_{wmf}, they have the option to utilize the Putting off mask network to generate both the recovered protected face x'_p and the recovered watermarked mask x'_{wm}, as shown in Fig. 1. We summarize the Putting off mask network as follows:

$$\left(x'_p, x'_{wm}\right) = f\left(x_{wmf}, \mathbf{N}\right) \tag{6}$$

Similar to the Putting on mask network, the watermarked masked face x_{wmf} and the random matrix $\mathbf{N}$ need to be DWT processed first in the Putting off mask network. To ensure reversibility, we need to ensure that the number of input and output channels are consistent. We use a random matrix $\mathbf{N}$ generated by sampling from a Gaussian distribution in place of the lost matrix $\mathbf{M}$. Then, the results of DWT are sent to the invertible recovery module. Like the invertible embedding module, K invertible recovery blocks of the same structure are connected in series to form the invertible recovery module, as shown in Fig. 1. For the ith recovery block in the putting off the mask, the formula is as follows,

$$\mathbf{N}^{i+1} = \left(\mathbf{N}^i - Dense\left(x_{wmf}^i\right)\right) \odot \exp\left(-\alpha\left(Dense\left(x_{wmf}^i\right)\right)\right) \tag{7}$$

$$x_{wmf}^{i+1} = x_{wmf}^i - Dense\left(\mathbf{N}^{i+1}\right) \tag{8}$$

Similar to Putting on the mask network, in the Kth invertible recovery block, IWT is done on $\mathbf{N}^K$ and x_{wmf}^K, and the recovered protected face x'_p and the recovered watermarked mask x'_{wm} are obtained. Finally, we do the IWT with outputs $\mathbf{N}^K$ and x_{wmf}^K, then obtain the recovered protected face x'_p and the recovered watermarked mask x'_{wm}.

3.3. IP verifier

For the IP verifier, verifying the model's IP is very important. The IP Verifier should consider not only how to verify IP from the sender and receiver, but also how to verify IP when the model is subject to a stealing attack. Attackers can train the stealing model to mimic the target model's behavior in a teacher-student manner. If a malicious attacker employs the stealing model to copy the functionality of the buyer's model, the IP of the buyer is severely jeopardized. Given these potential challenges, it becomes a formidable task for IP verifiers to ensure IP verification of the model effectively.

To address this issue, this paper uses the spatially invisible watermark as a verification tool to protect model intellectual property rights. We initially conducted joint training of the Watermark embedding network and the Watermark extractor network. Subsequently, we enhanced the training of the Watermark extractor network to ensure that it could accurately extract the watermark from the watermarked masked face of the Sender, the recovered watermarked mask of the Receiver and the simulated mask generated by the stealing model. This method serves to strengthen the security and verifiability of the model's IP in various applications.

3.3.1. Watermark extractor network

To ensure that the Watermark embedding network has initial watermark extraction capability, we use the Watermark embedding network H and the Watermark extractor network R respectively and train them jointly, as shown in Fig. 2.

Firstly, the Watermark embedding network H generates the watermarked mask by embedding the watermark into the mask. Then, we utilize the watermarked mask and the mask, respectively, as inputs to the Watermark extractor network, which is trained simultaneously with the Watermark embedding network. To prevent overfitting of the Watermark extractor network, where it outputs the target watermark regardless of whether the input image contains the watermark or not, the Watermark extractor network is guided to output a blank image when the input is the mask. Conversely, when the input is the watermarked mask, the Watermark extractor network is encouraged to extract the correct watermark.

Since the Watermark extractor network requires layer separation of the input to determine the watermark, we use CEILNet [33] as the Watermark extractor network R. Specifically, CEILNet consists of two cascaded sub-networks: the edge prediction network E-CNN and the image reconstruction network I-CNN. In addition, to improve the learning ability, nine residual blocks are inserted between the E-CNN and I-CNN.

To comprehensively safeguard the IP of the model, we also consider the problem of how to protect the IP of the model when an attacker uses the stealing model to mimic our model's facial privacy protection feature.

In the role of an attacker, the objective is to employ the stealing model to mimic the face replacement functionality of the black-box model and generate the simulated mask x_{sm} as the output of the stealing model, as shown in Fig. 3. The attacker's only available access is the trained black-box model. Consequently, the attacker can choose the protected face x_p and the replaced face x_r from the individual's dataset $D_s(X)$ as input samples, and obtain the watermarked masked face x_{wmf} as the output through the trained black-box model. Subsequently, the attacker can choose an appropriate stealing model SM and utilize x_p, x_r, and x_{wmf} as input–output pairs to train it. In this process, the stealing model takes the x_p and the x_r as inputs and generates the x_{sm}. Simultaneously, the objective is to minimize the distance $\mathcal{L}$ between the x_{sm} and the x_{wmf}. We summarize the process of stealing as follows:

$$\begin{cases} x_{sm} = SM\left(x_p, x_r\right) \\ \mathcal{L}\left(x_{sm}, x_{wmf}\right) \to 0 \end{cases} \tag{9}$$

Since the attacker aims to minimize the distance between x_{wmf} and x_{sm}, the x_{sm} from the trained stealing model should also include the w within the image [18]. However, in preliminary experiments, we found that during the initial training stage, the Watermark extractor network failed to extract the watermark from the simulated mask. That is because during this initial training phase, the Watermark extractor network can only observe the clean watermarked mask, not the image generated by the stealing model, which may contain some noises to affect the function of the Watermark extractor network. As a result, it becomes necessary to fine-tune the Watermark extractor network to improve its capability to extract the watermark.

3.3.2. Enhance watermark extractor network

As described in Section 3.3.1, due to potential noise in the simulated mask, the Watermark extractor network trained in the initial training phase cannot extract the watermark from the simulated mask. Hence, to enhance the extraction capability of the Watermark extractor network, we utilize the stealing model to emulate the attacker's behavior, and then we incorporate the simulated mask into the training dataset. Additionally, to augment the number of clean image samples in the dataset, we include the protected face and the recovered protected face in the training dataset, as shown in Fig. 4.

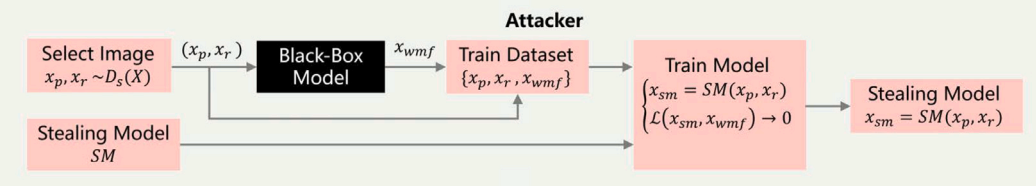

Fig. 3. The process of the attacker steals the attack.

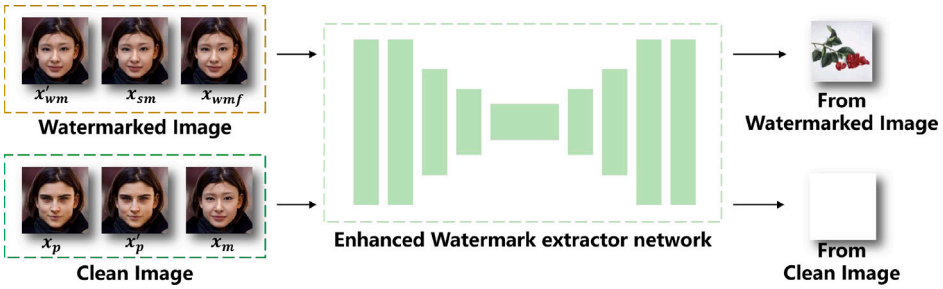

Fig. 4. The enhancement of the Watermark extractor network.

First, we use a stealing model network to imitate the attacker's behavior. This involves training the stealing model using pairs consisting of the protected face, the replaced face, and the watermarked masked face. Afterward, we incorporate the simulated mask into the training dataset, which consists of the watermarked masked face and the recovered watermarked mask, and perform fine-tuning on the Watermark extractor network to enhance its watermark extraction capability.

By adopting this method to augment the training of the Watermark extractor network, the Watermark extractor network gains the capability to successfully extract the watermark from the watermarked masked face, the simulated mask, and the recovered watermarked mask. As a result, it effectively verifies the IP of the model, offering protection and security.

3.4. Training strategy

Our method is divided into three training phases, which are the initial training phase, the reversible mask training phase and the enhanced training phase.

During the initial training phase, we conduct joint pretraining of the Watermark embedding network H and the Watermark extractor network R. In this phase, the Watermark embedding network H is able to embed the watermark w into the mask x_m and generate the watermarked mask x_{wm}, while the Watermark extractor network R is able to extract the watermark from the watermarked mask x_{wm}.

During the reversible mask training phase, we trained two networks: the Putting on mask network and the Putting off mask network. The Putting on mask network uses the watermarked mask x_{wm} and the protected face x_p as inputs to obtain the watermarked masked face x_{wmf} and the lost matrix **M**. Then, the Putting off mask network obtains the recovered protected face x'_p and the recovered watermarked mask x'_{wm} based on the watermarked masked face x_{wmf} and the random matrix **N**.

During the enhanced training phase, the Watermark extractor network was trained jointly in the initial training stage because it lacked the ability to extract the watermark from the simulated mask. Specifically, we choose a stealing model SM to mimic the attacker's behavior, which is trained with the simple ℓ_2 loss. Subsequently, we proceed with additional fine-tuning of the Watermark extractor network R using a mixed dataset. This fine-tuning process further enhanced the ability of the Watermark extractor network, making it to extract the watermark from various scenarios.

Below, we describe each training stage and the corresponding formula in detail.

3.4.1. The initial training phase

In the initial training phase, we first generate the mask x_m using the Mask Generator. To enable watermark embedding and extraction, we conduct a joint pre-training of the Watermark embedding network H and the Watermark extractor network R. The loss function of the initial training phase consists of two parts: the watermark embedding loss ℓ_{wemd} and the watermark extracting loss ℓ_{wext}:

$$\ell_{IP} = \lambda_{wemd}\ell_{wemd} + \lambda_{wext}\ell_{wext} \tag{10}$$

Watermark embedding loss: To ensure the visual quality of the generated watermarked mask, the watermark embedding loss ℓ_{wemd} consists of three different types of visual consistency losses:

$$\ell_{wemd} = \lambda_{bs}\ell_{bs} + \lambda_{vgg}\ell_{vgg} + \lambda_{adv}\ell_{adv} \tag{11}$$

where ℓ_{bs} is the basic ℓ_2-norm, ℓ_{vgg} is the perceptual loss and ℓ_{adv} is the adversarial loss.

The ℓ_{bs} is the pixel value difference between the mask x_m and the watermarked mask x_{wm} :

$$\ell_{bs} = \|x_{wm} - x_m\|_2 \tag{12}$$

The perceptual loss ℓ_{vgg} is defined as the difference in VGG feature between mask x_m and watermarked mask x_{wm}:

$$\ell_{vgg} = \|VGG_k(x_{wm}) - VGG_k(x_m)\|_2 \tag{13}$$

where $VGG_k()$ denotes the features extracted at layer k.

To enhance the ability of the Watermark embedding network H to effectively embed the watermark w, making it challenging for the discriminator D to differentiate between the watermarked mask x_{wm} and the original mask x_m, we define the adversarial loss as follows:

$$\ell_{adv} = \mathbb{E}_{x_m} log\left(D(x_m)\right) + \mathbb{E}_{x_{wm}} log\left(1 - D(x_{wm})\right) \tag{14}$$

Watermark extracting loss: The goal of the Watermark extractor network R is to extract the target watermark from the watermarked mask x_{wm} and the blank image from the mask x_m. Therefore, the watermark extracting loss ℓ_{wext} is defined as follows:

$$\ell_{wext} = \lambda_{wm}\ell_{wm} + \lambda_{clean}\ell_{clean} + \lambda_{cst}\ell_{cst} \tag{15}$$

The reconstruction loss ℓ_{wm} is defined as the difference between the output of the Watermark extractor network R when given the watermarked mask x_{wm} as input and the actual watermark w. The reconstruction loss ℓ_{wm} is defined as:

$$\ell_{wm} = \|R(x_{wm}) - w\|_2 \tag{16}$$

The clean loss ℓ_{clean} is defined as the difference between the output of the Watermark extractor network R when given the mask x_m as input and the constant blank image x_{clean}. The clean loss ℓ_{clean} is defined as:

$$\ell_{clean} = \|R\left(x_m\right) - x_{clean}\|_2 \tag{17}$$

where x_{clean} is the constant blank image.

Moreover, the watermark extracted by the Watermark extractor network R from different watermarked images is supposed to be consistent. Therefore, we select different watermarked mask images as inputs to the Watermark extractor network R and require the extracted watermarks to be as consistent as possible. The consistent loss ℓ_{cst} is defined as:

$$\ell_{cst} = \|R\left(x_{wm}^{i}\right) - R\left(x_{wm}^{j}\right)\|_2 \tag{18}$$

where x_{wm}^{i} and x_{wm}^{j} are samples in the watermarked mask x_{wm}.

3.4.2. The reversible mask training phase

In the initial training phase, the Watermark embedding network H embeds the watermark w into the mask x_m to generate the watermarked mask x_{wm}. To ensure facial privacy protection, in the reversible mask training stage, we utilize both the watermarked mask and the protected face as inputs and we train the Putting on mask network and the Putting off mask network to generate the watermarked masked face x_{wmf}, the recovered protected face x'_p and the recovered watermarked mask x'_{wm}. This process requires that the protected face x_p be closely similar to the recovered protected face x'_p in terms of visual appearance and pixel-level details. Additionally, the watermarked mask x_{wm} should closely resemble the recovered watermarked mask x'_{wm}. To achieve these objectives, we employ three distinct loss functions. The loss function of facial privacy protection ℓ_{fp} is a weighted sum of the recover loss $\ell_{recover}$, the guide loss ℓ_{guide} and the image distortion loss ℓ_{id}:

$$\ell_{fp} = \lambda_1 \ell_{recovery} + \lambda_2 \ell_{guide} + \lambda_3 \ell_{id} \tag{19}$$

Recovery loss: The main objective of the recovery process is to extract the protected face from the watermarked masked face x_{wmf}. Thus, the recovered protected face x'_p and the protected face x_p should be as identical as possible. Therefore, we define the recovery loss as:

$$\ell_{recovery} = \|x_p - x'_p\|_2 \tag{20}$$

Guide loss: In order to verify the IP of the model from the receiver, the Watermark extractor network R is employed to extract the watermark from the receiver's recovered watermarked mask x'_{wm}. Hence, it is essential for the watermarked mask x_{wm} and the recovered watermarked mask x'_{wm} to exhibit as many similarities as possible at the pixel level. So, we define the guide loss as follows:

$$\ell_{guide} = \|x_{wm} - x'_{wm}\|_2 \tag{21}$$

Image distortion loss: In facial privacy protection, the sender sends the watermarked masked face x_{wmf} to the receiver. To ensure the visibility of the watermarked masked face x_{wmf}, it needs to be as similar as possible to the watermarked mask x_{wm}. Therefore, the image distortion loss is defined as:

$$\ell_{id} = \|x_{wm} - x_{wmf}\|_2 \tag{22}$$

3.4.3. The enhanced training phase

With the initial training phase, the training set contains only clean watermarked images, so the Watermark extractor network R cannot observe the images produced by the stealing model. The images generated by the stealing model might include some noise, which can potentially impair the watermark extraction capability of the Watermark extractor network R. Therefore we added an enhanced training phase to improve the extraction ability of the Watermark extractor network R. Specifically, one stealing model SM is trained with the simple ℓ_2 loss by default. Then we mix the simulated mask x_{sm} into a dataset containing images of the mask x_m, the watermarked masked face x_{wmf} and the recovered watermarked mask x'_{wm}, as shown in Fig. 4.

Therefore, in order to better ensure that R can extract the watermark, the clean loss ℓ_{clean}, the consistent loss ℓ_{cst} and reconstruction loss ℓ_{wm} of the Watermark extractor network R will be changed to enhance its extracting ability. The enhanced watermark extracting loss ℓ'_{wext} of the Watermark extractor network for the enhanced training phase is:

$$\ell'_{wext} = \lambda'_{wm}\ell'_{wm} + \lambda'_{clean}\ell'_{clean} + \lambda'_{cst}\ell'_{cst} \tag{23}$$

where the enhanced clean loss ℓ'_{clean}, the enhanced reconstruction loss ℓ'_{wm}, and the enhanced consistent loss ℓ'_{cst} are respectively:

$$\ell'_{wm} = \|R\left(x'_{wm}\right) - w\|_2 + \|R\left(x_{sm}\right) - w\|_2 + \|R\left(x_{wmf}\right) - w\|_2 \tag{24}$$

$$\ell'_{clean} = \|R\left(x_m\right) - x_{clean}\|_2 + \|R\left(x_p\right) - x_{clean}\|_2 + \|R\left(x'_p\right) - x_{clean}\|_2 \tag{25}$$

$$\ell'_{cst} = \|R\left(x\right) - R\left(y\right)\|_2 \tag{26}$$

where x, y are samples in the mixed dataset, which consists of x'_{wm}, x_{wmf}, x_{sm}.

4. Experiments

The experiments are tested on Ubuntu 22.04.3 and conducted using two NVIDIA RTX A6000 GPUs within the PyTorch framework. As with most face replacement methods, the experiments were performed on a publicly available dataset called CelebA, which is a large-scale face attribute dataset.

Training parameter settings. In the initial training phase, the number of batch sizes for the Watermark embedding network and the Watermark extractor network is set to 8. The initial learning rate is set to 0.0001, and if the loss does not decrease within 5 calendar elements, the learning rate will decrease by 0.2. The epoch is set as 500.

In the reversible mask training phase, the number of batch sizes for the Putting on/off mask network is set to 32 and the number of invertible embedding/recovery blocks K is set to 16. The epoch parameters are set as 8000. The initial learning rate is set as 1e-5 and the weight decay is set as 1000.

In the enhanced training phase, the number of epochs for the stealing model is set as 300 and the learning rate is set as 0.0001. For the Watermark extractor network R, R is trained with the learning rate of 0.0001 and the epoch is set as 500.

Evaluation metrics. In the facial privacy protection part experiment, there are three evaluation metrics used to measuring the visual quality between images, which are PSNR, SSIM [34], and LPIPS [35]. Larger values of PSNR, larger values of SSIM and smaller values of LPIPS indicate better visual quality of the image. In the IP-protected part of the experiment, we use the classic normalized correlation (NC) to measure the effect of invisible watermarks. To determine whether the watermark was successfully extracted, we employ the NC metric. This metric is employed to measure the similarity between the extracted watermark and the original watermark. If the NC value of the watermark exceeds 0.95, the watermark extraction is considered successful. On this basis, the success rate of watermark extraction (SR_E) is further defined as the proportion of the watermark image in which the hidden watermark is successfully extracted. In order to measure the effectiveness of the method in a complex network environment (i.e., preventing recognition by a face detection system), we define the success rate of facial protection (SR_P) as the ratio of processed images that cannot be correctly recognized by the face recognition system.

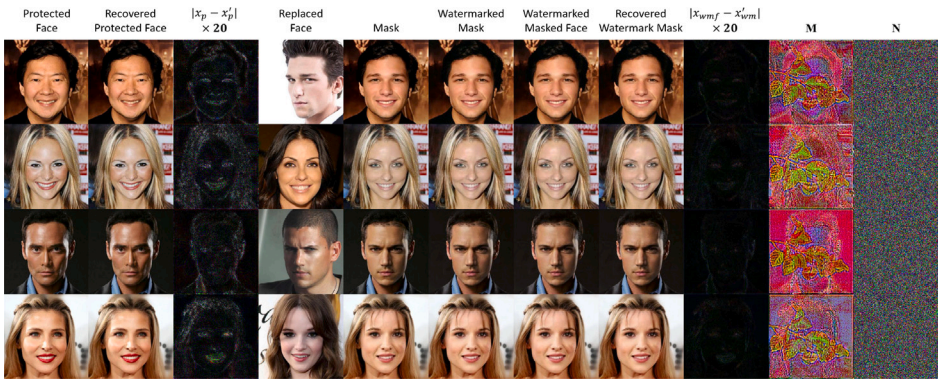

Fig. 5. Four groups of subjective experimental results of the facial privacy protection.

Table 2
Objective results corresponding to facial privacy protection.

Image Pairs	PSNR	SSIM	LPIPS
(x_p, x'_p)	41.05	0.982	0.005262
(x_{wm}, x_{wmf})	53.65	0.998	0.000092
(x_{wm}, x'_{wm})	55.97	0.998	0.000070

4.1. Subjective and objective results

Visual effects are important indicators for measuring the effect of facial privacy protection and IP protection. In this section, we will introduce the visual effects of facial privacy protection and IP protection from the perspectives of subjective visual effects and objective parameters. Section 4.1.1 will provide a detailed introduction to the experimental results of facial privacy protection. Following that, in Section 4.1.2, we will elaborate on the experimental results of IP protection.

4.1.1. Subjective and objective results of facial privacy protection

As an invertible face privacy protection method, visibility and invertible recovery are two important reference indexes. Therefore, we conduct a series of experiments to discuss the subjective visualization effect and objective performance of facial privacy protection.

As shown in Fig. 5, there are four groups of subjective experimental results. The experimental results demonstrate that the images obtained through our method appear remarkably clear and natural, exhibiting excellent visibility. It can be found that the difference between the watermarked mask and the watermarked masked face is almost imperceptible. This signifies that the watermarked masked face, generated by the watermarked mask putting on the protected face, exhibits excellent visual quality. Furthermore, the distinction between the protected face and the recovered protected face is nearly imperceptible. As a result, our method can flawlessly restore the protected face, resulting in the recovered protected face.

In addition to Fig. 5, we have used PSNR, SSIM, and LPIPS metrics to measure the objective results of the experiment. Table 2 clearly indicates the strong performance of our method across the three image pairs(protected face x_p/recovered protected face x'_p, watermarked mask x_{wm}/watermarked masked face x_{wmf}, and watermarked mask x_{wm}/recovered watermarked mask x'_{wm}). Specifically, for the x_p/x'_p pairs, the PSNR, SSIM, and LPIPS can reach 41.05 dB, 0.982, and 0.005262 respectively. For the x_{wm}/x_{wmf} pairs, the PSNR, SSIM, and LPIPS can reach 53.65 dB, 0.998, and 0.000092 respectively. Simultaneously, to ensure that the recovered watermarked mask image retains watermark information for verifying the model's IP, we impose a requirement for a high degree of similarity between the x_{wm} and the x'_{wm}. It can be found from Table 2 that the PSNR, SSIM, and LPIPS can reach 55.97 dB, 0.998, and 0.000070 for the x_{wm}/x'_{wm} pairs. The results show that these image pairs are almost identical.

4.1.2. Subjective and objective results of IP protection

In IP protection, to ensure the efficacy of our framework in verifying the model's IP, the Watermark extractor network should be capable of extracting the watermark from the watermarked masked face on the sender's side, from the recovered watermarked mask on the receiver's side, as well as from the simulated mask to resist the stealing attack.

In this experiment, we give subjective and objective results of the method when embedding different watermarks. A colorful flower image and an Anhui University logo image are used as example watermark images. As depicted in Fig. 6, the Watermark extractor network can accurately extract watermarks from various sources. Specifically, it can be observed that when the input image, such as the protected face or the mask, does not contain the embedded watermark, the Watermark extractor network generates the blank image as the output. Conversely, when the input image, such as the watermarked masked face, the recovered watermarked mask or the simulated mask, contains the embedded watermark, the Watermark extractor network accurately generates the correct watermark as the output. The experimental results validate that our method can extract watermarks from images originating from diverse sources. Consequently, our method exhibits comprehensive IP protection for the model, ensuring its security and ownership.

In addition to Fig. 6, we present objective data from IP protection experiments to demonstrate the effectiveness of our method in ensuring IP protection. In this part, we utilize the classic normalized correlation (NC) and the success rate of watermark extraction (SR_E) to measure the effect of invisible watermarks. In Table 3, we test the NC and SR_E of the watermark extracted from the watermarked masked face x_{wmf}, the recovered watermarked mask x'_{wm}, and the simulated mask x_{sm} respectively. Specifically, the Watermark extractor network R successfully extracts the watermark from images originating from diverse sources, achieving an average NC value surpassing 0.99 (SR_E=100%). This objective data from IP protection experiments unequivocally demonstrates the effectiveness of our method in accurately extracting watermarks from images of various sources.

4.2. Ablation study

4.2.1. Effectiveness of enhanced training

In order to enable the Watermark extractor network R to validate the IP of our method in a multi-directional way, we enhance its extraction capability by training it in the Enhanced training phase. Therefore, to verify that the enhanced training acts as an enhancement to the

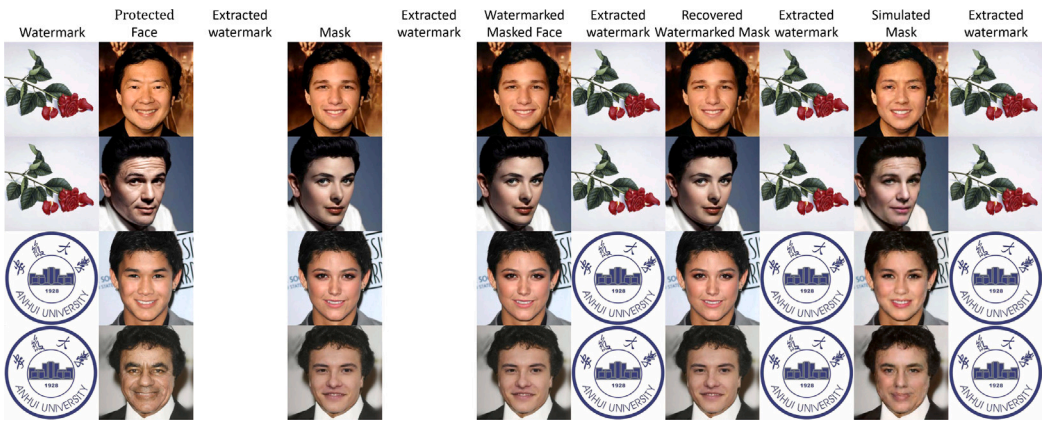

Fig. 6. Six examples of embedding different image watermarks into the mask with the proposed method.

Table 3
Quantitative results of the IP protection. (Use the Flower image and the Anhui University logo as watermarks, respectively).

Task	w' from x_{wmf}		w' from x'_{wm}		w' from x_{sm}	
	NC	SR_E	NC	SR_E	NC	SR_E
Flower	0.99	100%	0.99	100%	0.99	100%
Anhui	0.99	100%	0.99	100%	0.99	100%

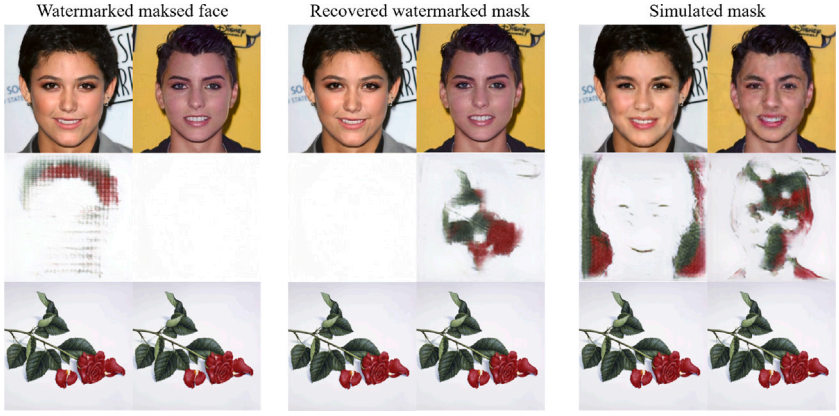

Fig. 7. Comparison subjective results without (second row) and with (third row) the Enhanced training phase. The second and third rows are the extracted watermarks without and with the Enhanced training phase.

Table 4
Comparison results of the SR_E with and without the Enhanced training phase.

Image	x_{wmf}	x'_{wm}	x_{sm}
With the Initial training phase	0%	0%	0%
With the Enhanced training phase	100%	100%	100%

extraction ability of the Watermark extractor network R, we compare subjective results with and without the Enhanced training phase.

Fig. 7 illustrates that the Watermark extractor network without the Enhanced training phase exhibits some sensitivity to the watermark in the watermarked masked face, recovered watermarked mask and simulated mask. However, due to limited extraction capabilities, it fails to completely extract the watermark from the image. In contrast, the Watermark extractor network in the Enhanced Training stage successfully extracts the watermark from the watermarked masked face, recovered watermarked mask and simulated mask.

In addition to the subjective comparison of watermarks extracted by the Watermark extractor network R in the initial and enhanced training stages, we evaluate the success rate of watermark extraction SR_E in Table 4. In this experiment, the Watermark extractor network R from both training stages is used to extract the watermark from the watermarked masked face x_{wmf}, recovered watermarked mask x'_{wm}, and simulated mask x_{sm}. As shown in Table 4, the Watermark extractor network R in the initial training stage fails to extract the watermark from these images($SR_E = 0\%$), meaning it cannot verify the intellectual property from the sender, receiver, or attacker. However, after fine-tuning in the enhanced training stage, the Watermark extractor network R is able to extract the watermark from all three image types($SR_E = 100\%$). Therefore, the fine-tuning of the Watermark extractor network R in the enhanced training stage effectively improves the watermark extraction ability of the Watermark extractor network R, providing multi-faceted IP protection for the proposed method.

Table 5
Objective results of different watermark embedding positions.

Position	(x_m, x_{wmf})			(x_p, x'_p)		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
On x_m	37.33	0.969	0.022964	41.05	0.982	0.005262
On x_{mf}	21.03	0.405	0.738303	24.24	0.706	0.430941

Table 6
The SR_E of different watermark embedding positions.

Method	x_{wmf}	x'_{wm}	x_{sm}
On x_m	100%	100%	100%
On x_{mf}	100%	0%	100%

4.2.2. Effectiveness of the watermark embedding position

To ensure comprehensive IP protection for the proposed method and the reversible recovery of the watermarked masked face, we selected different position images for watermark embedding to evaluate the influence of different watermark embedding positions on the experimental outcomes.

In this experiment, we try to embed the watermark at two different positions to assess the influence of these positions on IP protection and invertibility. Firstly, the watermark can be embedded into the mask x_m; Secondly, the watermark also can be embedded into the masked face (named as x_{mf}) which is generated by the Putting on mask network. The objective results of different watermark embedding positions are shown in Table 5. In this part, we employ the PSNR, SSIM, and LPIPS to evaluate the visual quality of x_m/x_{wmf} image pairs and x_p/x'_p image pairs. Additionally, Table 6 presents the SR_E for different watermark embedding positions, aiming to assess the influence of these positions on IP protection.

From the observations provided in Table 5, it can be observed that embedding the watermark into the masked face x_{mf} may have a negative impact on both the reversible recovery of the x'_p and the visual quality of the watermarked masked face x_{wmf}. Specifically, when the watermark is embedded into the masked face x_{mf}, the resulting PSNR and SSIM values between the x_m and the x_{wmf} are significantly lower at 16.3 dB, 0.564 and the resulting LPIPS values are significantly higher at 0.715339, respectively, compared to embedding the watermark into the mask x_m. At the same time, the resulting PSNR and SSIM values between the x_p and the x'_p are significantly lower at 16.81 dB, 0.276 and the resulting LPIPS values are significantly higher at 0.425679, respectively, compared to embedding the watermark into the mask x_m. We analyze that the irreversible damage to the reversibility of the INN structure occurs when choosing to embed the watermark into the masked face x_{mf}.

In addition to Table 5, which demonstrates the impact of different watermark embedding positions on visual quality, Table 6 also illustrates the effect of these positions on IP protection. We utilize the Watermark extractor network to test the SR_E at different watermark embedding positions. From Table 6, it can be observed that when the watermark is embedded in the x_{mf}, the Watermark extractor network is unable to extract the watermark from the x'_{wm}. This can be attributed to two reasons. Firstly, embedding the watermark in the x_{mf} significantly affects the reversible structure of the INN. Additionally, the x'_{wm} should closely resemble the x_m, which does not contain embedded watermark information. Consequently, the Watermark extractor network fails to extract the watermark from the x'_{wm} when embedding the watermark into the x_{mf}. Considering these factors, this paper opts to embed the watermark into the x_m, thereby obtaining the x_{wm}.

4.3. Comparative experiments with previous methods

In this section, we conduct experiments to compare the reversibility, facial privacy-preserving and IP protection of the proposed method with related methods. Section 4.3.1 evaluates reversibility by comparing subjective visual results and objective experimental results with the methods of You et al. [36] and Yang et al. [12]. You et al.'s method [36] presents a reversible face privacy protection technique that safeguards face privacy by employing reversible mosaic transformations. Yang et al.'s method [12] presents the invertible mask network (IMN), which uses the mask to hide real face privacy and reversibly recover real face information. In Section 4.3.2, we compare the proposed method with the classic deep learning-based watermarking method called HiDDeN [27] in terms of IP protection experiments. And in Section 4.3.3, we compare the subjective visual experimental results with You et al.'s method [36], Yang et al.'s method [12] and Shan et al.'s method [37] to evaluate the face privacy protection capabilities of different methods. Shan et al.'s method [37] adds adversarial samples to the image to hinder the correct recognition of the face recognition system.

4.3.1. Comparative experimental results of reversibility

For reversible face privacy protection methods, ensuring the high-quality restoration of original face information is crucial. Therefore, this section aims to provide a comparative analysis, both subjectively and objectively, between the proposed method, Yang et al.'s method [12] and You et al.'s method [36].

Fig. 8 presents a subjective visual comparison of the proposed method with other method in the reversible restoration experiment. As shown in Fig. 8, all three methods successfully restore the original face. However, the recovered face in You et al.'s method [36] appears noticeably blurrier compared to the proposed method and Yang et al.'s method [12].

Furthermore, an objective comparison of the reversible recovery performance between the proposed method and other methods is shown in Table 7. We use the average of PSNR, SSIM and LPIPS between the protected face and the recovered face to measure the reversible recovery performance of different methods. As shown in Table 7, Yang et al.'s method [12] exhibits the best recovery performance, followed by the proposed method. This discrepancy in performance can be attributed to the enhanced constraints between the watermarked mask and the recovered watermarked mask in the training process of the Putting on/off mask network, which aims to protect the model's IP but may slightly affect the restoration of the original face. Indeed, it is important to highlight that despite the impact of the enhanced constraints on the restoration process, the PSNR, SSIM, and LPIPS values are 41.05 dB, 0.982 and 0.005262 between the original face and the restored face. Hence, the proposed method exhibits a high level of recovery performance, ensuring the successful restoration of the original face.

4.3.2. Comparative experimental results of IP protection

To evaluate the effectiveness of the proposed method in IP protection, we compare it with the classic deep learning-based watermarking method called HiDDeN [27]. To ensure a fair comparison, we retrain the HiDDeN model using the same training dataset and calculated the average of PSNR, SSIM, LPIPS, and SR_E. It is important to note that the original HiDDeN model is used for digital watermarking, which is

Fig. 8. Comparison of subjective visual results of reversibility.

Table 7
Objective comparison of reversible recovery effects with other methods.

Method	You *et al.* [36]	Yang *et al.* [12]	Ours
PSNR	36.67	52.02	41.05
SSIM	0.988	0.997	0.982
LPIPS	0.0229	0.0008	0.005262

Table 8
Comparative experimental results of facial privacy protection.

Method	x_m/x_{wmf}			x_p/x'_p		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
HiDDeN [27]	37.01	0.950	0.043513	41.03	0.982	0.005458
Proposed method	37.33	0.969	0.022964	41.05	0.982	0.005262

Table 9
Comparative experimental results of the SR_E in IP protection.

Method	x_{wmf}	x'_{wm}	x_{sm}
HiDDeN [27]	0%	0%	0%
Our method	100%	100%	100%

inconsistent with the requirement of IP protection in this paper. To enable the network to perform facial privacy protection effectively, we have fused HiDDeN with our putting on/off mask network to create a combined architecture and then retrain the network.

We have provided objective data on visual quality in Table 8. Additionally, Table 9 compares the SR_E for these methods. According to Table 8, it is evident that the visual quality of HiDDeN [27] closely resembles the method proposed. However, Table 9 reveals that when using HiDDeN [27] to embed the watermark, it is unable to extract the watermark from the x_{wmf}, x'_{wm}, and x_{sm}. This limitation arises because HiDDeN [27] selects the mask as the embedding target for the watermark. As a result, HiDDeN [27] can only successfully extract the watermark from the mask but cannot extract watermarks from images originating from other sources. In contrast, the design of the proposed method enables the extraction of the watermark from the x_{wmf}, x'_{wm}, and x_{sm}. As a result, the proposed method offers comprehensive protection of IP of the protection method.

4.3.3. Comparative experimental results of facial privacy protection

For face privacy protection technology, it is essential to ensure that processed images retain natural while remaining undetectable by unauthorized face recognition systems. This not only enhances visual quality and user experience but also reduces the risk of attracting unwanted attention from potential attackers due to image anomalies.

In this experiment, we compare the subjective visual results of the proposed method with those of You et al. [36], Yang et al. [12], and Shan et al. [37]. As shown in Fig. 9, You et al.'s method [36] has poor visibility and obvious distortion. While Shan et al.'s method [37] preserves some facial features, it introduces significant artifacts, negatively impacting image quality. In contrast, both Yang et al.'s method [12] and the proposed method generate visually natural and aesthetically pleasing images.

In summary, the proposed method achieves effective face privacy protection by substituting original facial information with alternative facial features. As a result, the processed images maintain a high level of realism and visual appeal while ensuring privacy protection.

4.4. Robustness

As a reversible face privacy protection method with intellectual property protection, robustness is also an important performance. In this section, the robustness experiment is divided into two parts: the robustness of the stealing model attack and the robustness of facial privacy protection.

4.4.1. Robustness of stealing model attack

In addition to the primary objective of facial privacy protection, comprehensive protection of the model's IP is also one of the important features of the method. However, given the continuous evolution of attack methods, the facial privacy protection function of the proposed method may become vulnerable to various model-stealing attacks. Since it is not feasible to ascertain the specific model-stealing technique utilized by the attacker, we simulate this scenario by employing numerous stealing models. We use four different network structures as the stealing network: a convolutional network consisting of several convolutional layers ("CNet"), a residue network autoencoder with 9 and 16 blocks ("Res9", "Res16"), and the UNet. The SR_E of resisting the attack from stealing models is shown in Table 10. As shown in Table 10, it is evident that the proposed method exhibits robustness of resilience against different model-stealing attacks. Specifically, when employing Cnet, Res9, Res16, and Unet for stealing attacks individually, the SR_E of the proposed method are 74%, 92%, 93%, and 100%, respectively. Experimental results show that our method can effectively extract watermarks from the results generated by the steganographic

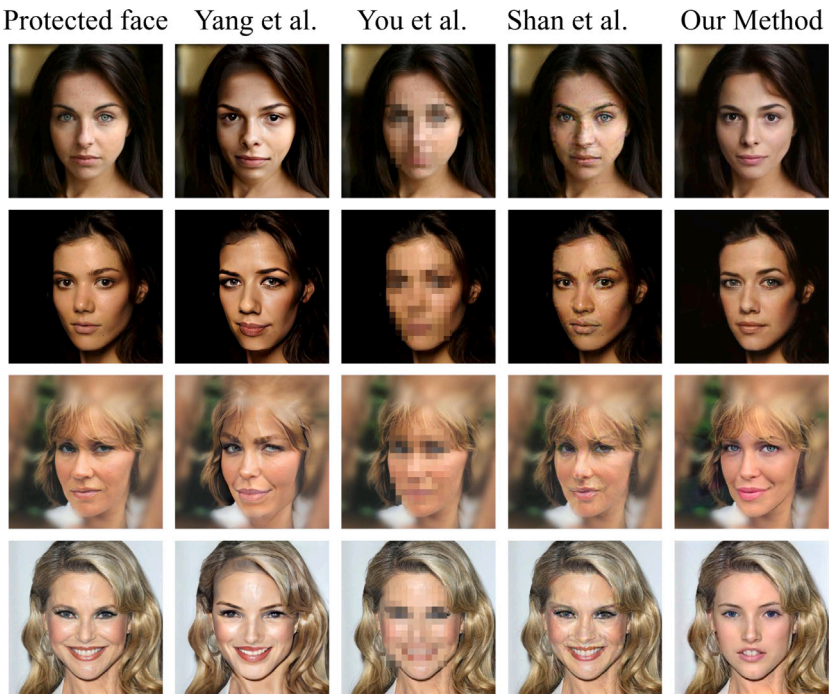

Fig. 9. Comparison of subjective visual results of face privacy protection.

Table 10
The SR_E of resisting the attack from stealing models.

Model	Cnet	Res9	Res16	Unet
SR_E	74%	92%	93%	100%

model in most cases. However, we have observed that in certain cases, the Watermark extractor network cannot extract the watermark from some of the results generated by the stealing model, such as Cnet, Res9, and Res16. We speculate that due to the lack of learning ability of these stealing networks, these stealing models cannot completely imitate the method proposed in this paper, and the simulated mask generated by the stealing model contains incomplete watermark information. Hence, the Watermark extractor network is unable to extract the watermark from certain results generated by the stealing network due to the incomplete learning of the invisible watermark embedded within the watermark sample. These experimental results serve as further validation of the effectiveness of our method in defending against attacks from different stealing models.

4.4.2. Robustness of facial privacy protection

To effectively safeguard face privacy and prevent recognition by face detection systems in complex network environments, it is important to ensure the robustness of processed images for face privacy protection. We utilize the success rate of facial protection (SR_P) as a metric to gauge the effectiveness of the proposed method in preserving face privacy. The SR_P indicates the percentage of processed images that cannot be correctly recognized by the face detection system.

In this experiment, we conduct a comparison of the SR_P of our method against Yang et al.'s method [12], You et al.'s method [36] and Shan et al.'s method [37] in complex environments. We initially randomly selected 1000 images as protected faces from the CelebA dataset. And use Yang et al.'s method [12], You et al.'s method [36], Shan et al.'s method [37] and the proposed method to process them to obtain the processed face. Subsequently, we choose three common types of attacks to apply to the processed faces: Gaussian noise, Gaussian blur, and JPEG compression. Gaussian noise is a type of random noise that follows a Gaussian distribution and is commonly used to simulate noise interference in real-world environments. In this experiment, the standard deviation of Gaussian noise is set to 0.001. Gaussian blur is a technique typically used to reduce image noise. In this experiment, the kernel size for Gaussian blur is set to 7. JPEG compression is a lossy compression technique that introduces some degree of image distortion. In this experiment, the JPEG compression quality is set to 30%. In the end, we measure the face recognition rate by utilizing Baidu Intelligent Cloud's face recognition tool to identify all the processed face images. Baidu Intelligent Cloud's face recognition service is built upon advanced deep learning and artificial intelligence technology. It possesses remarkable capabilities in face detection and recognition, allowing it to accurately identify faces even in intricate environments.

The results are listed in Table 11. As shown in Table 11, Shan et al.'s method [37] is divided into low-level (low) and high-level (high) face privacy protection and the success rate of face protection (SR_P) for low-level protection is 0% while for high-level protection, it is 72.8%. However, the robustness of Shan et al.'s method [37] is significantly impacted under simulated image attacks. This is because the Shan et al.'s method [37] relies on incorporating adversarial samples into the protected images to mislead face recognition classifiers, and these samples can be affected by simulated noise. In addition, Yang et al.'s method [12], You et al.'s method [36] and the proposed method effectively safeguard faces when tested on the attacked image using Baidu Intelligent Cloud's face recognition tool (SR_P = 100%). This indicates that these methods are resilient to Gaussian blur, Gaussian noise, and JPEG compression, respectively. You et al.'s method [36] damages the integrity of facial information by mosaic processing to facial privacy, rendering the processed image unrecognizable by the face recognition tool. Meanwhile, both Yang et al.'s method [12] and the proposed method rely on face replacement to preserve facial privacy. By replacing the facial features of a protected identity with those of another identity, the processed image cannot be accurately identified by the face recognition tool. It is noteworthy that

Table 11
The SR_P of the proposed method with other methods.

Method		Without attack	Gaussian noise	Gaussian blur	JPEG compression	IP protection
You *et al.* [36]		100%	100%	100%	100%	✗
Yang *et al.* [12]		100%	100%	100%	100%	✗
Shan *et al.* [37]	low	0%	20%	6.8%	0.4%	✗
	high	72.8%	82.2%	83.2%	71.2%	✗
Proposed Method		100%	100%	100%	100%	✓

the method proposed not only safeguards facial privacy but also offers comprehensive IP protection for the model, a feature not found in other methods.

5. Conclusion

With the rapid growth of online social networking platforms, users' facial privacy is being seriously threatened. In order to ensure that people's facial privacy is not disclosed and restored with high fidelity and, at the same time, ensure that the IP of the model owner is not infringed, this paper proposed an invertible privacy-preserving mask network with intellectual property protection (IPMN). IPMN consists of two main components: facial privacy protection and intellectual property protection. In the privacy protection process, the mask generator replaces the facial features of the protected face according to the replaced face and generates the mask for embedding the watermark. After embedding the watermark and generating the watermarked mask using the Watermark embedding network, the protected face is hidden by putting on mask network, and the protected face is restored with high quality by putting off mask network when necessary. In the IP protection process, the watermark extractor network plays a key role in verifying the intellectual property. It accurately extracts the watermark from images shared by senders, receivers, or potential attackers, ensuring multi-level IP security, and provides a reversible face privacy protection scheme with double-layer security. Experiments demonstrate that the proposed method effectively protects the IP of the method while the user face privacy is protected.

CRediT authorship contribution statement

Yang Yang: Methodology, Conceptualization. **Xiangjie Huang:** Writing – original draft, Visualization, Investigation, Data curation. **Han Fang:** Writing – review & editing. **Weiming Zhang:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62272003, in part by the Innovation Program for Quantum Science and Technology under Grant 2021ZD0302300, and in part by the Science and Technology Major Project of Anhui Province under Grant 202423s06050001.

Data availability

Data will be made available on request.

References

[1] Oh SJ, Benenson R, Fritz M, Schiele B. Faceless person recognition: Privacy implications in social media. In: Computer vision–ECCV 2016: 14th European conference, amsterdam, the netherlands, October 11-14, 2016, proceedings, part III 14. Springer; 2016, p. 19–35.
[2] Ribaric S, Ariyaeeinia A, Pavesic N. De-identification for privacy protection in multimedia content: A survey. Signal Process, Image Commun 2016;47:131–51.
[3] Zhang Y, Zhao R, Zhang Y, Lan R, Chai X. High-efficiency and visual-usability image encryption based on thumbnail preserving and chaotic system. J King Saud Univ-Comput Inf Sci 2022;34(6):2993–3010.
[4] Yuan L, Chen W, Pu X, Zhang Y, Li H, Zhang Y, et al. PRO-face C: Privacy-preserving recognition of obfuscated face via feature compensation. IEEE Trans Inf Forensics Secur 2024.
[5] Ryu G, Park H, Choi D. Adversarial attacks by attaching noise markers on the face against deep face recognition. J Inf Secur Appl 2021;60:102874.
[6] Wang Y, Chen X, Zhu J, Chu W, Tai Y, Wang C, et al. Hififace: 3d shape and semantic prior guided high fidelity face swapping. 2021, arXiv preprint arXiv:2106.09965.
[7] Chen R, Chen X, Ni B, Ge Y. Simswap: An efficient framework for high fidelity face swapping. In: Proceedings of the 28th ACM international conference on multimedia. 2020, p. 2003–11.
[8] Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 4401–10.
[9] Xu Z, Zhou H, Hong Z, Liu Z, Liu J, Guo Z, et al. Styleswap: Style-based generator empowers robust face swapping. In: European conference on computer vision. Springer; 2022, p. 661–77.
[10] Kim K, Kim Y, Cho S, Seo J, Nam J, Lee K, et al. Diffface: Diffusion-based face swapping with facial guidance. 2022, arXiv preprint arXiv:2212.13344.
[11] Zhao W, Rao Y, Shi W, Liu Z, Zhou J, Lu J. DiffSwap: High-fidelity and controllable face swapping via 3D-aware masked diffusion. 2023, CVPR.
[12] Yang Y, Huang Y, Shi M, Chen K, Zhang W. Invertible mask network for face privacy preservation. Inform Sci 2023;629:566–79.
[13] Uchida Y, Nagai Y, Sakazawa S, Satoh S. Embedding watermarks into deep neural networks. In: Proceedings of the 2017 ACM on international conference on multimedia retrieval. 2017, p. 269–77.
[14] Chen H, Rouhani BD, Fu C, Zhao J, Koushanfar F. Deepmarks: A secure fingerprinting framework for digital rights management of deep learning models. In: Proceedings of the 2019 on international conference on multimedia retrieval. 2019, p. 105–13.
[15] Adi Y, Baum C, Cisse M, Pinkas B, Keshet J. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In: 27th USeNIX security symposium (USeNIX security 18). 2018, p. 1615–31.
[16] Zhang J, Gu Z, Jang J, Wu H, Stoecklin MP, Huang H, et al. Protecting intellectual property of deep neural networks with watermarking. In: Proceedings of the 2018 on Asia conference on computer and communications security. 2018, p. 159–72.
[17] Le Merrer E, Perez P, Trédan G. Adversarial frontier stitching for remote neural network watermarking. Neural Comput Appl 2020;32:9233–44.
[18] Zhang J, Chen D, Liao J, Fang H, Zhang W, Zhou W, et al. Model watermarking for image processing networks. In: Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 07, 2020, p. 12805–12.
[19] Blanz V, Vetter T. A morphable model for the synthesis of 3D faces. In: Seminal graphics papers: pushing the boundaries, vol. 2, 2023, p. 157–64.
[20] Nirkin Y, Keller Y, Hassner T. Fsgan: Subject agnostic face swapping and reenactment. In: Proceedings of the IEEE/CVF international conference on computer vision. 2019, p. 7184–93.
[21] Li L, Bao J, Yang H, Chen D, Wen F. Faceshifter: Towards high fidelity and occlusion aware face swapping. 2019, arXiv preprint arXiv:1912.13457.
[22] Karras T, Laine S, Aittala M, Hellsten J, Lehtinen J, Aila T. Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020, p. 8110–9.
[23] Xu Y, Deng B, Wang J, Jing Y, Pan J, He S. High-resolution face swapping via latent semantics disentanglement. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, p. 7642–51.
[24] Gao G, Huang H, Fu C, Li Z, He R. Information bottleneck disentanglement for identity swapping. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, p. 3404–13.

[25] Zhu Y, Li Q, Wang J, Xu C-Z, Sun Z. One shot face swapping on megapixels. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, p. 4834–44.
[26] Awasthi D, Tiwari A, Khare P, Srivastava VK. A comprehensive review on optimization-based image watermarking techniques for copyright protection. Expert Syst Appl 2023;122830.
[27] Zhu J, Kaplan R, Johnson J, Fei-Fei L. Hidden: Hiding data with deep networks. In: Proceedings of the European conference on computer vision. 2018, p. 657–72.
[28] Xiang Y, Li T, Ren W, He J, Zhu T, Choo K-KR. AdvEWM: Generating image adversarial examples by embedding digital watermarks. J Inf Secur Appl 2024;80:103662.
[29] Wang J, Huang W, Zhang J, Luo X, Ma B. Adversarial watermark: A robust and reliable watermark against removal. J Inf Secur Appl 2024;82:103750.
[30] Cao F, Guo D, Wang T, Yao H, Li J, Qin C. Universal screen-shooting robust image watermarking with channel-attention in DCT domain. Expert Syst Appl 2024;238:122062.
[31] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention–mICCAI 2015: 18th international conference, munich, Germany, October 5-9, 2015, proceedings, part III 18. Springer; 2015, p. 234–41.
[32] Jing J, Deng X, Xu M, Wang J, Guan Z. Hinet: Deep image hiding by invertible network. In: Proceedings of the IEEE/CVF international conference on computer vision. 2021, p. 4733–42.
[33] Fan Q, Yang J, Hua G, Chen B, Wipf D. A generic deep architecture for single image reflection removal and image smoothing. In: Proceedings of the IEEE international conference on computer vision. 2017, p. 3238–47.
[34] Hore A, Ziou D. Image quality metrics: PSNR vs. SSIM. In: 2010 20th international conference on pattern recognition. IEEE; 2010, p. 2366–9.
[35] Zhang R, Isola P, Efros AA, Shechtman E, Wang O. The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 586–95.
[36] You Z, Li S, Qian Z, Zhang X. Reversible privacy-preserving recognition. In: 2021 IEEE international conference on multimedia and expo. IEEE; 2021, p. 1–6.
[37] Shan S, Wenger E, Zhang J, Li H, Zheng H, Zhao BY. Fawkes: Protecting privacy against unauthorized deep learning models. In: 29th USeNIX security symposium (USeNIX security 20). 2020, p. 1589–604.