Visual Information Hiding Based on Obfuscating Adversarial Perturbations

Zhigang Su  
Xidian University

Dawei Zhou  
Xidian University

Decheng Liu  
Xidian University

Nannan Wang *  
Xidian University

Zhen Wang  
Zhejiang Lab

Xinbo Gao  
Chongqing University of Posts and Telecommunications

Abstract

Growing leakage and misuse of visual information raise security and privacy concerns, which promotes the development of information protection. Existing adversarial perturbations-based methods mainly focus on the de-identification against deep learning models. However, the inherent visual information of the data has not been well protected. In this work, inspired by the Type-I adversarial attack, we propose an adversarial visual information hiding method to protect the visual privacy of data. Specifically, the method generates obfuscating adversarial perturbations to obscure the visual information of the data. Meanwhile, it maintains the hidden objectives to be correctly predicted by models. In addition, our method does not modify the parameters of the applied model, which makes it flexible for different scenarios. Experimental results on the recognition and classification tasks demonstrate that the proposed method can effectively hide visual information and hardly affect the performances of models. The code is available in the supplementary material.

1. Introduction

Deep neural networks (DNNs) have been widely applied in the computer vision [9, 16, 23]. However, the increasing leakage and misuse of visual information has raised serious concerns [18, 41]. A representative case is the security issue of data stored in the cloud environment [3, 17, 30]. Due to potential vulnerabilities in cyberspace [2], uploaded private images can be easily stolen and used maliciously [33]. Therefore, it is meaningful and urgent to explore effective strategies to protect visual information.

A classic strategy is visual information hiding, which mainly consists of two types [35]: homomorphic encryption (HE)-based methods [1, 27, 42] and perceptual encryption (PE)-based methods [6, 14, 34, 35, 37]. However, affected by the nonlinear activation functions in DNNs, HE-based methods are difficult to perform well on advanced DNNs [35]. Although PE-based methods are applicable to DNNs, existing methods typically require retraining with data in the encrypted domain to guarantee accuracy on encrypted data [35, 37]. This results in interferes with the performance on the original data and additional resource consumption (especially for large models).

To alleviate these negative effects, we expect to hide the visual information without making any modifications to DNNs. Namely, we hide sensitive visual information only by varying the input image. Previous researches have made some explorations in this regard. The transformation network-based methods [14], which try to protect the original image via a transformation function parameterized by a neural network, share the same philosophy. However, the method cannot easily recover the original image from the protected image for other purposes. They may suffer from adversarial vulnerability [8, 12] because the introduced neu-

*preprint. Corresponding to nnwang@xidian.edu.cn.
ral network may be destroyed by adversarial attacks.

In fact, the negative effects of adversarial attacks can be utilized positively to protect privacy. Some works have exploited adversarial attacks for de-identification [29, 31, 44]. These methods add imperceptible perturbations or non-suspicious patches generated by adversarial attacks to original images, hindering DNNs to extract effective features and recognize identities, thus protecting the identity privacy in the image. However, in this work, we focus on visual information hiding, which means that the protected image is completely different from the original image visually but can still be correctly predicted by DNNs (see Fig. 1). Fortunately, we observe an special type of adversarial attacks (called Type-I attack) [38] that is significantly different from the type used in previous methods. This type of attacks guides DNNs to make consistent predictions on two quite distinct samples.

Inspired by the Type-I adversarial attack, in this paper, we propose an Adversarial Visual Information Hiding (AVIH) method. The proposed method hides the visual information in images while preserving their functional (e.g., recognition and classification) features for DNNs. It can recover original images from protected images for their owners. Specifically, we reduce the visual correlation between the protected image and the original image while minimizing their distance in the feature space of DNNs, to generate the protected image. Meanwhile, we exploit a generative model pre-trained in a private training setting as the key model. Then optimize the protected image based on it so that the recovered image is similar to the original image. Furthermore, to break through the tough trade-off between the capability of privacy protection and the quality of restored image, we design the variance consistency loss to enhance privacy protection without compromising image recovery (see Sec. 3.3). Note that the protected image generated by our method can only be accurately recovered by the own key model, other models (even if the model architecture is the same) are difficult to recover well (see Sec. 4.3).

AVIH can greatly improve the security and flexibility of image storage, which is extremely obvious in the protection of gallery sets for cloud metric learning. A typical application example is gallery set protection for cloud metric learning tasks. Take a face recognition system in a cloud environment as an example. According to the target face recognition model, the face database manager can generate protected images locally, or generate protected image in the cloud and save the key model locally. The protected image contains no visual information and can be used by the target model to extract features correctly. Then, these protected images can be stored in the gallery set of the face recognition system in the cloud for normal face recognition tasks. These protected images can be recovered using the key model when needed (such as maintaining the dataset or using face images for other tasks, etc.). The process is shown in Fig. 1.

Taking visual information hiding of gallery set images for cloud-deployed face recognition systems as the basic task, we conduct a comprehensive evaluation of our proposed method in terms of both effectiveness and security. In order to compare with existing methods suitable for information hiding tasks, we extend our method to classification tasks. Experimental results on multiple target models and datasets show that the proposed method is effective. In addition, to prove the effectiveness of our proposed loss, we conduct an ablation study about it to further present the advantages of our proposed method.

Our main contributions are as follows:

- Inspired by Type-I adversarial attacks, we propose a visual information hiding method AVIH. To alleviate the difficult trade-off between capability of information hiding and quality of recovered image, we design a variance consistency loss.

- Our proposed method has following properties: 1) The visual information in the image is clearly obfuscated; 2) Our method does not require retrain; 3) The protected image can be recovered by the own key model, but external models are difficult to recover.

- We validate the effectiveness of the proposed method on the face recognition task and the classification task. In addition, we conduct qualitative and quantitative ablation studies to show efficiency of the proposed loss.

## 2. Related Work

### 2.1. Visual Information Hiding

The visual information hiding of images is from the perspective of human vision. It is most directly manifested by the fact that the protected images are visually unrecognizable. HE-based methods which are more secure are mainly derived from cryptography. However, it is usually not suitable for nonlinear computations. Since most DNNs contain a large amount of nonlinear computation, HE-based methods are hardly applicable to state-of-the-art DNNs [35]. Therefore, this type of method is not discussed in this paper. For the PE-based methods, some works [6, 34, 35, 37] focus on finding an encrypted domain and training the model directly using the encrypted images. However, this has a significant impact on the accuracy of the model. To improve the accuracy of the classification model for protected images, Ito et al. [14] trained a transformation network to keep the classifier correctly classified while hiding visual information. However, the weakness of this method is that the protected image cannot be recovered.
Unlike the above work used in classification and segmentation tasks, we mainly focus on metric learning tasks represented by face recognition, which are more prone to privacy leakage. To compare with existing methods that can achieve visual information hiding, we also extend the AVIH method to classification tasks. Our proposed method can also compensate for the drawbacks of the above methods. It generates protected images for a specific model that already exists. It takes advantage of the vulnerability in the model itself to provide strong privacy protection to the image. Since our method does not modify the target model, it does not affect the accuracy on the original image.

2.2. Adversarial Attack

DNNs are very vulnerable to some adversarial examples [8,12]. There are many adversarial attack methods [4,8,28] to find adversarial examples efficiently. Tang et al. [38] divided the adversarial attacks into adversarial attack Type-I and adversarial attack Type-II based on the statistical Type I error and Type II error. We take an optimization perspective on adversarial attacks. Then the Type-II attack is to maximize the difference in the model output while ensuring that the difference with the original input samples is slight. Mathematically, it can be formulated as follows:

\[
\max_{x'} f(x) - f(x') \quad \text{s.t.} \quad \|x' - x\| < \epsilon,
\]

(1)

where \(x\) is the original sample, \(x'\) is the adversarial sample, and \(f(\cdot)\) is the model which is attacked. The Type-I attack, in contrast to the Type-II attack, looks for an input sample that differs the most possible from the original input sample, but makes the output of the model consistent. Mathematically, it can be formulated as follows:

\[
\max_{x'} \|x' - x\| \quad \text{s.t.} \quad f(x) = f(x').
\]

(2)

The Type-I attack on classification networks and generative adversarial networks was also implemented in the work of Tang et al. [38]. Then, Sun et al. [36] implemented a Type-I attack against variational autoencoder (VAE) [21]. In this work, we implement Type-I attacks on face recognition tasks and classification tasks. Inspired by these attacks, we propose the AVIH method.

3. Methodology

The objective of our proposed method is to learn an protected image \(x'\) which satisfies: 1) \(x'\) is completely different from the original image \(x\), 2) for the target model the output \(f_t(x')\) is the same as the original image \(f_t(x)\), and 3) \(x'\) can be recovered as the original image \(x\) by the key model.

Mathematically,

From \(x \in \mathcal{X}\) Generate \(x' = \mathcal{A}(x)\)

s.t. \(\|x' - x\| > \epsilon\)

\(f_t(x') = f_t(x), \text{key}(x') = x.\)  \hspace{1cm} (3)

The pipeline of our method is shown in Fig. 2.

3.1. Image Visual Information Hiding

Exploiting the adversarial vulnerability of DNNs, we can perform Type-I attack on DNN model to get images that are visually completely different from the original image but with extremely similar features. Suppose that \(d(\cdot)\) measures the difference between the outputs of the model. For different tasks, it behaves as different functions. Then for a particular target model \(f_t\), the difference between the output of the adversarial sample and the original sample is

\[
\mathcal{L}_t(x', x) = d(f_t(x), f_t(x')).
\]

(4)

We define \(\mathcal{L}_d\) as

\[
\mathcal{L}_d(x', x) = \|x - x'\|_2^2,
\]

(5)

then the loss of the Type-I attack on the target model is formulated as

\[
\mathcal{L}_t(x', x) = \mathcal{L}_t(x', x) - \lambda \cdot \mathcal{L}_d(x', x),
\]

(6)

where \(\lambda\) is a positive hyperparameter which balances image level differences and output level differences. Then with a certain number of iterations \(K\), we can optimize \(\mathcal{L}_r\) by the following operations to get the final adversarial sample.

\[
g_{k+1} = \alpha \cdot g_k + \frac{\nabla \mathcal{L}(x'_{k}, x)}{\|\nabla \mathcal{L}(x'_{k}, x)\|_2^2},
\]

(7)

\[
x'_{k+1} = x'_{k} - \beta \cdot g_{k+1},
\]

(8)

where

\[
g_0 = \frac{\nabla \mathcal{L}(x'_{0}, x)}{\|\nabla \mathcal{L}(x'_{0}, x)\|_2^2}.
\]

(9)

At the first iteration, \(x'_{0}\) can be randomly initialized, or it can be made \(x'_{0} = x\). The former provides an easier way to get an adversarial example with greater visual differences. In comparison, the latter provides a faster way to get an adversarial example that meets the criteria.

3.2. Image Recovery

The image obtained by the loss function of Eq. (6) can have the effect of visual information hiding and can keep its function for the target model. However, the protected image obtained is not recoverable. To achieve the goal of Eq. (3), we attach a recovery module which can recover the generated information hidden image. We refer to the generated images with perturbations as protection images. This perturbation has the function of protecting the visual information of the image.

To get the protected images, we first train the generative model \(G\), which can generate the same images as the input. In this work, we use a generative model based on the
pix2pix framework [13]. Then we perform Type-I attacks on both the target model and the generative model. We define $L_g$ as follows:

$$L_g(x', x) = \|x - G(x')\|^2. \quad (10)$$

The loss $L_g$ can help to keep the protected images recoverable by the key model we have chosen. Thus, the loss becomes as follows:

$$L_e(x', x) = L_r(x', x) + \mu \cdot L_g(x', x), \quad (11)$$

where $\mu$ is a hyperparameter that balances the protection quality and restore quality. With Eq. (7) and Eq. (8), we can optimize $L_e$ to obtain the protected image $x'$, which can satisfy the objective of Eq. (3).

### 3.3. Variance Consistency Loss

The protected image obtained by the objective function of Eq. (6) satisfies the requirements of Eq. (3), but its protection quality is not high. The obtained protected images have the problem of the difficult trade-off between protection quality and recovery quality. That is, if we want to obtain an image that is difficult to be cracked successfully, the quality of the image recovered by the key model will be poor. Another problem is that the obtained protected image, although differing greatly from the original in color, exists obvious visual information that the original image has, which has a negative impact on visual information protection. To solve the above problem, we propose a variance consistency loss. It improves the quality of protection by limiting the differences between each part of the image to make the protected image visually more confusing.

For the input image, we divide the image in each channel (R, G, B) into $N$ blocks respectively, i.e., $\{b_1, b_2, \ldots, b_N\}_c$, where $c \in \{R, G, B\}$, $b_n \in \mathbb{R}^{h \times w}$ and $h, w$ denote the height and width of the block. The pixels of the blocks are allowed to have overlapping parts between them. Let $p_{i,j}^{b_n} \in [0, 1]$ denote the normalized pixel value at $(i, j)$ in block $b_n$.

Then, we calculate the sum of each block:

$$S_n = \sum_{i=1}^h \sum_{j=1}^w p_{i,j}^{b_n}. \quad (12)$$

We use $S$ denote the set of sum of the blocks for each channel, i.e., $S = \{S_1, S_2, \ldots, S_N\}_c$. In our practice, we convolve the image with a convolution kernel of size $h \times w$ to obtain blocks. Then, we calculate the variance of $S_c$ and obtain $\sigma^2 = \text{var}(S_c)$. Finally, we define the variance consistency loss as:

$$L_v(x') = \sigma^2_R + \sigma^2_G + \sigma^2_B, \quad (13)$$

where $\sigma_R$, $\sigma_G$ and $\sigma_B$ denote the variances for R, G, B channels, respectively. By minimizing $L_v$, we can get a protected image with a more uniform distribution of pixel values. Proof-of-concept experiments (see Sec. 4.4) show that $L_v$ can eliminate the visual semantics in the protected images which are similar to the original images, and can help obtain protected images with high quality of protection and recovery.

Based on the variance consistency loss, the loss function in Eq. (6) is modified as:

$$L_e(x', x) = L_t(x', x) + \lambda \cdot L_v(x'), \quad (14)$$

and the Eq. (11) is reformulated as:

$$L_{AVIH}(x', x) = L_t(x', x) + \lambda \cdot L_v(x') + \mu \cdot L_g(x', x), \quad (15)$$

where $\mu$ are hyperparameters. This is our proposed AVIH framework for visual information hiding. The algorithm is summarized in Algorithm 1.

### 3.4. AVIH Method for Specific Tasks

Our method can provide effective protection for the stored image and can be applied to a wide range of tasks. In this work, we take the face recognition and the classification as examples to illustrate the ability of our method.

---

**Figure 2.** Overview of adversarial attack-based visual information hiding (AVIH) method. Taking the face recognition as an example, given a target model and a pre-trained key model, we protect the original image $x$ and obtain the protected image $x'$.
For the face recognition task, we want the features extracted by the target face recognition model from the protected image and the original image to be as same as possible. Thus, we reify \( L_i \) as follows:
\[
L_i(x', x) = \|f_i(x) - f_i(x')\|^2_2.
\] (16)

For the classification task, compared with directly minimizing the mean square error (MSE) of \( f_i(x) \) and \( f_i(x') \), we find that maximizing \( f_i^c(x') \) where \( c \) is the prediction class of the target model on the original image can more effectively reduce the impact of visual information hiding on the accuracy. We thus first obtain the logit output of the original image \( f_i(x) \) and convert it to one-hot format \( \delta(f_i(x)) \). Then, we reify \( L_i \) as follows:
\[
L_i(x', x) = -\delta(f_i(x)) \cdot f_i(x').
\] (17)

4. Experiments

In this section, we first verify the effectiveness of the AVIH method for face recognition tasks. Then, taking the face recognition system as an example, the security of the AVIH method and the effectiveness of the variance consistency loss are explored. Finally, we verify the effectiveness of the AVIH method for classification tasks and compare it with other methods.

4.1. Experimental Settings

Dataset and target models. Our experiments were mainly evaluated based on the Labeled Face in the Wild (LFW) [11] dataset. We choose four face recognition models: ArcFace [5], CosFace [40], SphereFace [25] used in [44] and AdaFace [20] to fully evaluate the performance of our method. Among them, the input size of ArcFace and AdaFace [20] is \( 112 \times 112 \), and the input size of CosFace and SphereFace is \( 112 \times 96 \). Therefore, we use the MTCNN [45] to align and crop the face images to the input size of the corresponding face recognition model first.

Key models. We choose the pix2pix framework [13] to train our key model. It has the same structure as the generator used in CycleGAN [46]. To pre-train the key model, we first randomly selected 1,287 images from the CelebA [26] dataset as the training set. Then we set the input and output of the model to the same size, set the batch size to 1, and train 4 epochs. Finally, the trained generator is used directly as the key model. Unless otherwise stated, we use only one key model to protect the entire dataset.

Evaluation Metrics. To evaluate the effectiveness of our method in a more realistic way and inspired by the evaluation method in MegaFace [19], we modified the evaluation method of LFW. We randomly selected 12 persons from the LFW dataset as the probe set. Each of them contains more than 12 facial images, for a total of 355 images. The other 1287 images, we use as gallery set. In the testing phase, we take one face image of a person in the probe set and put it into the gallery set. Then use the remaining images of this person as the test set. Next, we use the above-divided dataset to test the accuracy of the face recognition model. In this way, we put each image of each person in the probe set into the gallery set in turn to measure the total accuracy. This metric can well demonstrate the impact of our method on face recognition models in practical applications.

| Models       | Original | AVIH-ONE | AVIH-ALL |
|--------------|----------|----------|----------|
| AdaFace      | 98.6     | 98.6     | 98.6     |
| ArcFace      | 96.5     | 96.5     | 96.5     |
| CosFace      | 89.4     | 89.2     | 89.3     |
| SphereFace   | 80.3     | 80.0     | 80.6     |

Table 2. Accuracy (percentage) of face recognition models for original image and different protected image on LFW. The results of AVIH-ONE and AVIH-ALL are expected to be close to the results of Original.
We use the structural similarity (SSIM) [43] to quantitatively measure the quality of the images. In this work, SSIM$_e$ represents the average of the SSIM values between the protected image $x'$ and the original image $x$, and SSIM$_d$ represents the average of the SSIM values between $x$ and the recovered image $G(x')$.

### 4.2. Effectiveness of Face Privacy Protection

**Effectiveness and Impact on Target Model.** To the best of our knowledge, our method is the first (PE)-based methods for face recognition, and there is no closely related method yet to compare the performance of protected images. Therefore, we conduct qualitative and quantitative evaluations of protected images. High visual metric scores and very slight impact on accuracy demonstrate the effectiveness of our method.

We use the AVIH method to protect the original face image $x$ in the probe set to test its effectiveness on face visual information hiding. Then, we used the obtained protected image $x'$ as the input of the key model $G(\cdot)$. Finally get the recovered image $G(x')$. The results are shown in Fig. 3. From Fig. 3, it is evident that the protected image generated by our method is significantly different from the original image. Since no useful visual information is obtained from the protected image at all, it has the effect of visual information hiding for the original image. In Tab. 1, the average SSIM value between the protected image and the original image for each target model is less than 0.04, while the cosine similarity between their features is higher than 0.99. Therefore, the protected image generated by AVIH can completely replace the original image while hiding the visual information.

To evaluate the impact of our proposed AVIH method on the accuracy of the face recognition models, we first generate a protected face image in the probe set before putting it into the gallery set. Then put the protected image into the gallery set instead of the original image. We tested the accuracy of the face recognition models in this way. The result AVIH-ONE is shown in Tab. 2. Compared with the original accuracy of the models, it can be concluded that the impact of the AVIH method on the accuracy of the models is very slight, which can achieve the same accuracy as the ArcFace model.

Considering the real situation, all the images stored in the gallery set are also protected images. So we replaced all the images in the gallery set with protected images and then retested the accuracy of the models, as AVIH-ALL in Tab. 2. In this case, our method also has a very slightly impact on the accuracy of the models and has a beneficial effect on the accuracy of the SphereFace model.

We test the average of SSIM values, as shown in Tab. 1. Combined with Fig. 3 shows that the recovered image has good quality and can recover most of the visual information of the original image. Since the recovered images are generated by the generative model, a better trained generative model can achieve better recovery quality. We also explore the time taken by AVIH to protect image in the Sec. A.1 of
the supplementary material.

**Randomness of the dataset for training the key model.** In this work, we use *CelebA*, which is also a face dataset, to train the key model. However, in real scenarios, a large amount of face images is often difficult to obtain due to the privacy of each individual involved. So we changed *CelebA* to *COCO* [24] containing various objects to test the effectiveness of the newly key model used to protect the image. We choose the test set of *COCO* as the train set, and train 6 epochs on the key model. The results are shown in Tab. 3. It can be concluded that both the impact of protected images on the model accuracy and the quality of recovered images are very slightly different from the key model trained with *CelebA*. Therefore, we can use different types of datasets to train the key model, not just limited to face images.

**Real World Face Privacy Protection.** We randomly selected real-world face photos taken with phones and then protected them to test the performance of the AVIH method for using real-world scenarios. Part of results are shown in Fig. 4. In realistic scenarios, the cosine similarity of the features between the protected image generated by the AVIH method and the original image is still higher than 0.99. The recovered image can still be recovered well.

### 4.3. Security Analysis

**Key Model Randomness Analysis.** To test the randomness of the key model, we first trained 16 key models consecutively using the same settings but with different initialization values. Different initialization values may result in significantly different trained key models. Then, we choose one of them in turn as the private key model to participate in the protection of the image, and use the others as the external key model to try to recover the protected image. From the results as shown in Fig. 5, it can be concluded that the protected image obtained by one private key model cannot be recovered by other external key models, which demonstrates our proposed method can greatly improves the security of the face visual information. In addition, it is convenient that we can obtain mutually independent keys by simply changing the initialization without changing factors such as the structure of the key model and the training dataset. We suppose that one reason for this phenomenon is due to the instability of GAN training, where different initialization values lead to different locally optimal solutions. Another reason could be that our method tends to find the boundary points of the input field corresponding to the output error allowed by the key model. However, the instability of GAN training causes this boundary to change significantly when the initialization value is changed.

**Responding to a Possible Attack.** For the key model, even if the key model structure and its training settings are leaked, the protection is still difficult to break if the initialization point of the key training model is not leaked. As for the image, if pairs of original and protected images are leaked in large quantities, then an attacker can directly use these image to train a generative model to directly map the protected domain to the original domain. Ito et al. [15] exploited this method to evaluate the robustness of protection. So, we protect all the face images in the gallery set using the same key model, producing pairs of protected and original images. Then we use these paired images as a training set, and train a model using the same structure as the attack model. Using such a model, we attempt to recover the protected images in the probe set. The results are shown in Fig. 6. When the training set images and the protected image to be cracked are protected from the same key model, the attack model can recover the protected image. However, when they are protected with different key models, the
trained model cannot crack the protected image. Therefore, we can increase the frequency of changing key models to effectively defend against such attacks.

The protected images obtained by our method also have good statistical properties. The correlation analysis and histogram analysis are shown in Sec. A.2 of the supplementary material. We also verify that the protected image generated by a target face recognition model cannot be recognized by other models with the same function.

4.4. Ablation Study

We compare the distance loss $L_d$ with the variance consistency loss $L_v$ to verify the effectiveness of $L_v$. For each loss we choose a set of suitable weights $\lambda$ separately to represent its trade-off between the protection quality and recovery quality. Its results are shown in Fig. 7. The protected image generated using $L_d$ has a smaller SSIM value compared to the protected image using $L_v$, but a clear outline can be seen. So the visual information of the original image still exists. If the protected image is to be made free of such visual information, there is a significant loss in the quality of the recovered image. In contrast, using $L_v$, it is possible to generate protected images with both no residual visual information and high recovery quality. We also use another key to recover the protected image to further investigate the effect of the two different losses on the protection quality. From Fig. 7, it can be concluded that the protected image using $L_v$ is more difficult to be recovered by similar key models with different initialization while ensuring the quality of the recovered image.

4.5. Privacy protection for classification models

We trained ResNet50 [10], VGGNet19 [32] using the train set of CIFAR-10 [22] and implemented AVIH method for the images of the test set. The results are shown in Fig. 8. We also compared two existing methods suitable for visual information hiding and the results are shown in Tab. 4, where the ITP [14] do not recover protected images. Since the LIE [37] uses protected images to train the model, the prediction accuracy on the original data is low. We show the figure of protection results of the two compared methods and their SSIM values in Sec. A.3 of the supplementary material. Our method has minimal impact on the accuracy of the models, and the average SSIM value of the recovered images can reach above 0.9 for both models after our tests.
5. Conclusion

In this paper, we propose a visual information hiding method AVIH based on Type-I attack. We evaluate our method by image protection of face recognition system in cloud. Experiments show that the AVIH method can protect images while preserving their functionality for the target model. We also propose a variance consistency loss to solve the problem of the difficult trade-off between protection quality and recovery quality. Finally, we use the AVIH method in classification tasks with satisfactory results.

Our work provides a new perspective on image visual information protection, which has beneficial implications for adversarial learning communities and private protection communities. The AVIH method requires complete target model information. While this is feasible in most image storage situations, it limits the applicability of the AVIH method to a wider range of image protection scenarios. This is also a problem we will solve in our future work.

References

[1] Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, et al. Privacy-preserving deep learning via additively homomorphic encryption. **IEEE Transactions on Information Forensics and Security**, 13(5):1333–1345, 2017.

[2] L Arockiam and S Monikandan. Efficient cloud storage confidentiality to ensure data security. In 2014 International Conference on Computer Communication and Informatics, pages 1–5. IEEE, 2014.

[3] Deyan Chen and Hong Zhao. Data security and privacy protection issues in cloud computing. In 2012 International Conference on Computer Science and Electronics Engineering, volume 1, pages 647–651. IEEE, 2012.

[4] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In International conference on machine learning, pages 2206–2216. PMLR, 2020.

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4690–4699, 2019.

[6] Yi Ding, Guozheng Wu, Dajiang Chen, Ning Zhang, Linpeng Gong, Mingsheng Cao, and Zhiguang Qin. Deepedn: a deep-learning-based image encryption and decryption network for internet of medical things. **IEEE Internet of Things Journal**, 8(3):1504–1518, 2020.

[7] Hossein Movafegh Ghaediri, Ali Nodhei, and Rasul Enayatifar. An overview of encryption algorithms in color images. **Signal Processing**, 164:163–185, 2019.

[8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[11] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In Workshop on faces in Real-Life Images: detection, alignment, and recognition, 2008.

[12] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. Advances in neural information processing systems, 32, 2019.

[13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1125–1134, 2017.

[14] Hiroki Ito, Yuma Kinoshita, Maungmaung Aprilpyone, and Hitoshi Kiya. Image to perturbation: An image transformation network for generating visually protected images for privacy-preserving deep neural networks. **IEEE Access**, 9:64629–64638, 2021.

[15] Hiroki Ito, Yuma Kinoshita, and Hitoshi Kiya. Image transformation network for privacy-preserving deep neural networks and its security evaluation. In 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE), pages 822–825. IEEE, 2020.

[16] He Kaiming, Gkioxari Georgia, Dollar Piotr, and Girshick Ross. Mask r-cnn. **IEEE Transactions on Pattern Analysis & Machine Intelligence**, PP:1–1, 2017.

[17] Seny Kamara and Kristin Lauter. Cryptographic cloud storage. In International Conference on Financial Cryptography and Data Security, pages 136–149. Springer, 2010.

[18] Manjit Kaur and Vijay Kumar. A comprehensive review on image encryption techniques. **Archives of Computational Methods in Engineering**, 27(1):15–43, 2020.

[19] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4873–4882, 2016.

[20] Minchul Kim, Anil K Jain, and Xiaoming Liu. Adaface: Quality adaptive margin for face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 18750–18759, 2022.

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[23] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, 86(11):2278–2324, 1998.
A. Supplementary Material

A.1. Time Spent on Protection

The GPU model used for all our experiments is NVIDIA TITAN Xp. Since our method is an iterative method, it takes a certain amount of time to protect images one by one. However, we can increase the protection efficiency by increasing the batch size. As shown in Fig. 9, we tested the average time it takes to protect an image under different batch sizes, from which we can conclude that adjusting the batch size can greatly improve the efficiency of protection. Moreover, the number of iterations also greatly affects the time required for protection. We tested the effect of different iterations on the time required for protection and the quality of protected images as shown in Tab. 5 and Fig. 10.
Figure 9. The time required to protect an image by the AVIH method and the effect of batchsize on protection time. A represents the average time required to protect a batch, and B represents the average time required to protect the corresponding number of samples when the batch size is 1. We protected ten batches and averaged the time spent.

Table 5. The effect of different iterations on protection time and protected image quality. Time represents the average time required to protect a batch, SSIM represents the average SSIM between the recovered image and the original image, and COS represents the average cosine similarity between the original image feature and the protected image features. We tested 10 samples and averaged these metrics.

| Iterations | 100  | 200  | 400  | 600  | 800  |
|------------|------|------|------|------|------|
| Time (s)   | 5.651| 11.251| 22.703| 32.703| 43.053|
| SSIM       | 0.721| 0.829| 0.886| 0.895| 0.912|
| COS        | 0.973| 0.983| 0.995| 0.997| 0.997|

We can conclude that, within a certain range, we can speed up the protection time with little impact on protection quality by reducing the number of iterations. We can also spend more time increasing the protection quality by increasing the number of iterations.

A.2. Security Analysis

A.2.1 Key Model Space Analysis:

In our proposed method, we choose the generative model as the key, which has a large key space. Take the key model used in our experiments as an example, it has a size of 11.383M. Such a large key space can be disastrous for those who use exhaustive attacks.

A.2.2 Histogram Analysis and Correlation Analysis:

Plain images have a strong correlation between two adjacent pixels in the horizontal and vertical directions [7], and protection methods with good properties often need to break this correlation [39]. Therefore, we performed a correlation analysis of our proposed method, and its results are shown in Fig. 11. Compared to the strong correlation of the original image, the protected image we obtained greatly reduces the correlation between adjacent pixels. We also performed histogram analysis on the original and protected images, and the results are shown in Fig. 12. From it, it can be concluded that the histogram statistical properties of the protected image and the original image are completely different, and the protected image more closely resembles a gaussian distribution. So the protected images obtained by the AVIH method have well statistical characteristics.

A.2.3 Identifiability of protected images:

We use one face recognition model as the target model to generate protected images, and then use another face recognition model to identify the results obtained in Tab. 6. It can be concluded that the protected image obtained by a target model cannot be used normally by other face recognition models. Storing such protected images in a cloud environment can greatly improve security.

A.3. Privacy Protection for Classification Tasks

A.3.1 Detailed experimental setup:

When implementing the AVIH method on the classification task, we adjust the batch size to 10 and initialize the protected image as the original image. We set the number of iterations to 600.

A.3.2 The Quality of Recovered Images:

We show the protection results of LIE [37] and ITP [14] in Fig. 14. We tested the average SSIM values of the AVIH method for the protected and recovered images of the test set in CIFAR-10 [22]. The results are shown in Tab. 7. We compare our results with LIE [37] and ITP [14], where LIE recovers the original image, but the protection quality is weaker and has a significant impact on the model accuracy. ITP mitigates the impact of the protected image on the model accuracy, but the protected image becomes unrecoverable. Our method ensures a strong protection strength while the impact on the model accuracy is very slight.
Figure 10. Protected and recovered images generated by different iterations. Each pair of images includes the protected image and recovered image, respectively. The number of iterations they correspond to is marked below the image.

Figure 11. Results of correlation analysis. We randomly select 3000 pairs of adjacent pixel points for the original and protected images, respectively, and calculate their correlation coefficients in horizontal, vertical and diagonal directions.

Figure 12. Results of histogram analysis of some original and protected images. The original image is on the left and its corresponding protected image is on the right.
Table 6. Accuracy (percentage) of predicting protected images using a model different from the target face recognition model. The same name represents the same model.

| Target model | Testmodel | AdaFace | ArcFace | CosFace | SphereFace |
|--------------|-----------|---------|---------|---------|------------|
| AdaFace      | 98.6      | 0       | -       | -       | -          |
| ArcFace      | 0         | 96.5    | -       | -       | -          |
| CosFace      | -         | -       | 89.4    | 0       |            |
| SphereFce    | -         | -       | 0       | 80.3    |            |

Table 7. Impact of privacy protection methods on the accuracy (percentage) of classification models on CIFAR-10.

| Method       | Model     | SSIM_e | SSIM_d |
|--------------|-----------|--------|--------|
| LIE [37]     | VGGNet19  | 0.178  | 1.000  |
|              | ResNet50  | 0.178  | 1.000  |
| ITP [14]     | VGGNet19  | 0.068  | -      |
|              | ResNet50  | 0.073  | -      |
| AVIH(our)    | VGGNet19  | 0.171  | 0.900  |
|              | ResNet50  | 0.198  | 0.923  |