Veil Privacy on Visual Data: Concealing Privacy for Humans, Unveiling for DNNs *(Supplementary Material)*

Shuchao Pang[1(✉)], Ruhao Ma[1], Bing Li[2(✉)], Yongbin Zhou[1], and Yazhou Yao[3]

[1] School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing, China {pangshuchao,maruhao,zhouyongbin}@njust.edu.cn
[2] School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China bing_li@uestc.edu.cn
[3] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China yazhou.yao@njust.edu.cn

A Simple Explorations

The core goal is to generate privacy-preserving surrogate data when dealing with the outsourcing of sensitive original data, i.e., its privacy information perceptible at the human visual level is well hidden, while the latent features recognized by models for training and prediction are almost unaffected. Moreover, the trained model on surrogate data, also called Veiled data in our paper, can effectively recognize the original data in practical applications.

Here, we also conduct some simple explorations for the comprehensive comparison with our proposed Veil Privacy on visual data, considering whether the existing processing techniques for images can be applied directly to this task, such as image noise, image blurring and image blending.

Fig. 1: Examples of image noise, image blurring and image blending

✉ Corresponding author. Shuchao Pang and Ruhao Ma contribute equally.

A.1 Several Basic Methods

Image Noise. Image noise [4] refers to unnecessary or redundant interference in images, where the presence of noise seriously affects the quality of images. Meanwhile, there are a lot of works to study denoising [3]. Therefore, we used it to protect the privacy of images for this task.

For this, we introduce three commonly used image noises, i.e., Gaussian noise, Salt-pepper noise and Speckle noise. Specifically, Gaussian noise means its probability density follows the Gaussian distribution. Salt-pepper noise appears as random white or black spots. Speckle noise comes from the phenomenon of speckle interference in optics, appearing as speckles. Fig. 1 shows some examples of image noise, where the second column is Gaussian noise with a variance of 0.01 and a mean of 0, the third column is salt-pepper noise with a proportion of 0.1 and the salt-pepper ratio of 0.5, and the fourth column is speckle noise with the variance of 0.1 and the mean of 0.

Image Blurring. Image blurring [8] often occurs in our real life. The quality of photography equipment and shooting conditions are the main reasons for this. Therefore, people always try to improve images through deblurring [7]. On the contrary, we use it to protect the privacy of images.

We consider the following three common methods. Mean filtering replaces a specific pixel value with the average value of surrounding pixels. Gaussian filtering replaces a specific pixel value with the weighted value of surrounding pixels according to the Gaussian distribution. Median filtering replaces a specific pixel value with the middle value of surrounding pixels. Fig. 1 shows some examples of image blurring. Among them, the fifth column is mean filtering, the sixth column is Gaussian filtering with $sigmaX = 5$ and $sigmaY = 5$ and the seventh column is median filtering. The convolutional kernel size of all filters is 5*5.

Image Blending. Image blending [5] means recombining multiple images to generate a new image. There are some works toward realistic high-resolution blending [6]. This technology is useful in many fields, nevertheless, we first try it to protect the privacy of images.

We just implement a simple method for image blending, which mixes two images in a certain proportion to produce a new image. Specifically, we designed three elementary mixing masks (noise mask, strip mask and face mask). Fig. 1 shows some examples of image blending, where the eighth column is the noise mask, the ninth column is the strip mask, and the tenth column is the face mask. The mixing ratio is equal to 0.5.

A.2 Preliminary Experiments

For the above basic methods, we conduct experiments on the CK+48 dataset. CK+48 is a facial expression recognition dataset, containing all grayscale images with 48 × 48 size. It includes 7 categories (i.e., anger, contempt, disgust, fear, happiness, sadness and surprise), with a total of 981 facial images. As the baseline model, A convolutional neural network (CNN) model is designed with three

Table 1: SSIM between X and different X_p. A, B and C are shown in Fig. 1.

	GaussianNoise	Salt-pepperNoise	SpeckleNoise	MeanBlur	GaussianBlur	MedianBlur	NoiseMask	StripMask	FaceMask
Image A	0.47325	0.56612	0.37347	0.77249	0.78213	0.81772	0.44500	0.58708	0.46378
Image B	0.51435	0.62239	0.43869	0.77023	0.77996	0.80607	0.45743	0.59061	0.43337
Image C	0.41650	0.57723	0.42712	0.80352	0.81178	0.86035	0.40075	0.56569	0.39352

Table 2: The accuracy for image noise, image blurring and image blending

	Baseline Model	GaussianNoise	Salt-pepperNoise	SpeckleNoise	MeanBlur	GaussianBlur	MedianBlur	NoiseMask	StripMask	FaceMask
Processed Data	/	98.94%	91.49%	94.68%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Original Data	100.00%	98.94%	95.74%	96.81%	94.68%	91.49%	95.74%	93.62%	93.62%	77.66%

convolutional layers and three fully connected layers. We use Rule as the activation function in the model and CrossEntropy as the loss function on the training phase with SGD optimizer. The baseline model trained with the original data X achieves 100.00% accuracy.

As Fig. 1 shows, we generate the processed data X_p from the original data X through the above basic methods. SSIM between X and different X_p is shown in Tab. 1, which demonstrates that there is still high visual similarity between the original images X and the processed images X_p. In other words, these basic methods fail to protect or hide sensitive visual privacy information, i.e., faces. Then we train models with different training sets X_p respectively and their results are shown in Tab. 2. On the testing set, the GaussianNoise model achieves 98.94% accuracy on X_p and 98.94% accuracy on X. The Salt-pepperNoise model achieves 91.49% accuracy on X_p and 95.74% accuracy on X. The SpeckleNoise model achieves 94.68% accuracy on X_p and 96.81% accuracy on X. The MeanBlur model achieves 100.00% accuracy on X_p and 94.68% accuracy on X. The GaussianBlur model achieves 100.00% accuracy on X_p and 91.49% accuracy on X. The MedianBlur model achieves 100.00% accuracy on X_p and 95.74% accuracy on X. The NoiseMask model achieves 100.00% accuracy on X_p and 93.62% accuracy on X. The StripMask model achieves 100.00% accuracy on X_p and 93.62% accuracy on X. The FaceMask model achieves 100.00% accuracy on X_p and 77.66% accuracy on X. Obviously, these basic methods can retain the same features for face recognition on the processed data to some extent. Overall, none of these methods can effectively achieve practical privacy protection for visual data.

As we can see, the key to all these methods is how to balance hiding the visual privacy information and preserving the potential features. However, these simple explorations do not solve the problem, which urgently requires us to design a new methodology for addressing the practical problem, especially, for current customized AI security services.

B More Security Evaluation

Model Inversion Attack (MIA). We explore whether the original data X are possible to be recovered through MIA on the veiled model $\tilde{F}(\cdot)$ due to similar

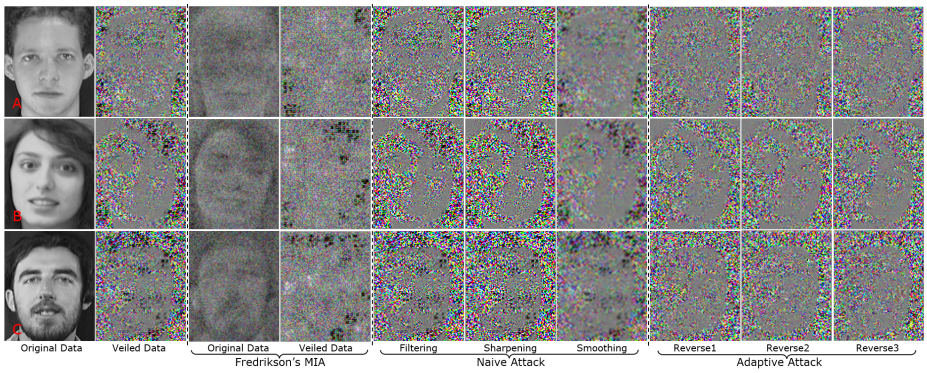

Fig. 2: Examples for Fredrikson's MIA, naive attack and adaptive attack

latent features with the veiled data $\tilde{X}$. We employ Fredrikson's MIA [1] to attack $\tilde{F}(\cdot)$ with label $y_i \in Y$. In addition, we perform MIA on the baseline model $F(\cdot)$ for comparison. As demonstrated in Fig. 2, the reconstructed data resemble $\tilde{X}$, but they are dramatically different from X. The result demonstrates that MIA on $\tilde{F}(\cdot)$ is ineffective.

Naive Attack. Attackers who lack knowledge of the Veil Privacy framework, including random pixel flipping $RPF(\cdot)$ and gradient iteration algorithm $GIA(\cdot)$, attempt to recover the original data X from the veiled data $\tilde{X}$. We apply common denoising methods, including filtering, sharpening and smoothing. As shown in Fig. 2, the naive attack is useless.

Adaptive Attack. With the knowledge of the random pixel flipping $RPF(\cdot)$ with the flipping ratio p and gradient iteration algorithm $GIA(\cdot)$ with the veiled model $\tilde{F}(\cdot)$, adaptive attackers aim to generate the veiled data X from the veiled data $\tilde{X}$ contrarily. As shown in Fig. 2, the adaptive attack is invalid.

C Time Complexity

Compared to high computational costs from dataset distillation and encrypted computing, our framework has lower time complexity and our computational cost of generating the veiled data is acceptable. All veiled data generation was implemented using one NVIDIA-Geforce-RTX-3090-24GB machine. The complexity and computational costs for the process of generating veiled data on various datasets are listed in Tab. 3.

For example, dataset distillation has a great computational cost for the distillation process, as [2] stated compute limitations: "Such costs mean that, using a single V100 GPU with 16GB of RAM, we were (i) only able to sample shallow kernels; (ii) for convolutional kernels, limited to small support sets and small target batch sizes; (iii) unable to use pooling if learning more than just a few images. Scaling up KIP to deeper, more expensive architectures, achievable using multi-device training, will be the subject of future exploration".

Table 3: Time complexity of generating the veiled data for different datasets

	#Dimensionality	#Class	#Instance	Backbone	Time Complexity	
					Single(min)	All(min)
BioID	1*384*286	20	1,488	VGG16	0.116	173
ORL	1*92*112	11	110	LeNet-5	0.018	2
LFW	3*250*250	5,749	13,233	VGG16	0.124	1,646
CelebA	3*178*218	10,177	202,599	VGG16	0.045	9223
MNIST	1*28*28	10	60,000	AlexNet	0.012	735
CIFAR10	3*32*32	10	60,000	AlexNet	0.018	1,098
CIFAR100	3*32*32	100	60,000	VIT	0.235	1412

D Standard Baselines

As a pioneering work for this practical problem, we establish our own standard baselines for comparison and give standard evaluation criteria, i.e., SSIM(x, $\tilde{x}$), Acc(x) and Acc($\tilde{x}$) of the model trained with $\tilde{x}$.

As Table 4 states, we create the baselines based on *Veil Privacy* and compare it with classical explorations (details seen in *Supplementary Material A*).

Table 4: Own baselines for comparison

Dataset	Backbone	Standard	GaussianNoise($\mu = 0.1, \sigma = 0.1$)			MeanBlur(kernel size=(16,16))			FaceMask(mixing ratio=0.3)			**Veil Privacy**		
		Acc(x)	Acc($\tilde{x}$)	Acc(x)	SSIM	Acc($\tilde{x}$)	Acc(x)	SSIM	Acc($\tilde{x}$)	Acc(x)	SSIM	Acc($\tilde{x}$)	Acc(x)	SSIM
BioID	VGG16	99.34%	83.38%	39.74%	0.17990	83.38%	30.46%	0.66238	82.05%	25.17%	0.51812	**98.68%**	**98.01%**	**0.06462**
ORL	LeNet-5	96.97%	78.79%	27.28%	0.12766	87.88%	25.76%	0.47998	81.82%	21.21%	0.28563	**96.97%**	**96.97%**	**0.04729**
LFW	VGG16	88.89%	77.78%	34.07%	0.17794	70.37%	26.67%	0.61123	74.07%	19.26%	0.41019	**88.89%**	**85.19%**	**0.04625**
CelebA	VGG16	87.96%	61.29%	11.11%	0.19600	45.16%	14.81%	0.57244	48.39%	14.66%	0.42356	**87.09%**	**83.87%**	**0.03566**
MNIST	AlexNet	99.13%	88.38%	38.71%	0.47249	86.84%	10.09%	0.27217	78.55%	26.54%	0.22891	**98.58%**	**96.60%**	**0.01247**
CIFAR10	AlexNet	85.99%	44.20%	13.19%	0.25602	45.57%	15.57%	0.22510	52.70%	26.69%	0.23262	**84.09%**	**81.73%**	**0.01106**
CIFAR100	VIT	87.35%	19.90%	13.33%	0.23844	27.52%	12.74%	0.23417	33.31%	10.72%	0.22715	**83.22%**	**81.86%**	**0.02271**
RSS	ResNet18	88.50%	80.15%	23.06%	0.21053	72.23%	20.26%	0.55126	50.67%	17.06%	0.46221	**86.62%**	**85.36%**	**0.03314**

References

1. Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. pp. 1322–1333 (2015)
2. Nguyen, T., Chen, Z., Lee, J.: Dataset meta-learning from kernel ridge-regression. arXiv preprint arXiv:2011.00050 (2020)
3. Ramesh, G., Logeshwaran, J., Gowri, J., Mathew, A.: The management and reduction of digital noise in video image processing by using transmission based noise elimination scheme. ICTACT Journal on Image & Video Processing **13**(1) (2022)
4. Rank, K., Lendl, M., Unbehauen, R.: Estimation of image noise variance. IEE Proceedings-Vision, Image and Signal Processing **146**(2), 80–84 (1999)
5. Rankov, V., Locke, R.J., Edens, R.J., Barber, P.R., Vojnovic, B.: An algorithm for image stitching and blending. In: Three-dimensional and multidimensional microscopy: image acquisition and processing XII. vol. 5701, pp. 190–199. SPIE (2005)

6. Wu, H., Zheng, S., Zhang, J., Huang, K.: Gp-gan: Towards realistic high-resolution image blending. In: Proceedings of the 27th ACM international conference on multimedia. pp. 2487–2495 (2019)
7. Zhang, K., Luo, W., Zhong, Y., Ma, L., Stenger, B., Liu, W., Li, H.: Deblurring by realistic blurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2737–2746 (2020)
8. Zhang, S., Shen, X., Lin, Z., Měch, R., Costeira, J.P., Moura, J.M.: Learning to understand image blur. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6586–6595 (2018)