

Unauthorized AI cannot Recognize Me: Reversible Adversarial Example

Jiayang Liu^{1,3}, Weiming Zhang¹, Kazuto Fukuchi^{2,3}, Youhei Akimoto^{2,3}, Jun Sakuma^{2,3}

¹ University of Science and Technology of China

² University of Tsukuba

³ RIKEN AIP

Abstract

In this study, we propose a new methodology to control how user's data is recognized and used by AI via exploiting the properties of adversarial examples. For this purpose, we propose reversible adversarial example (RAE), a new type of adversarial example. A remarkable feature of RAE is that the image can be correctly recognized and used by the AI model specified by the user because the authorized AI can recover the original image from the RAE exactly by eliminating adversarial perturbation. On the other hand, other unauthorized AI models cannot recognize it correctly because it functions as an adversarial example. Moreover, RAE can be considered as one type of encryption to computer vision since reversibility guarantees the decryption. To realize RAE, we combine three technologies, adversarial example, reversible data hiding for exact recovery of adversarial perturbation, and encryption for selective control of AIs who can remove adversarial perturbation. Experimental results show that the proposed method can achieve comparable attack ability with the corresponding adversarial attack method and similar visual quality with the original image, including white-box attacks and black-box attacks.

Introduction

Online services, including social networking services that utilize artificial intelligence trained with users' data, make our lives more convenient. For example, Facebook uses facial recognition technology to determine whether users are in the photo image data being uploaded and suggest tags indicating the user identities. Facial recognition technology, including face authentication, is convenient and impactful, but at the same time, it has many ethical issues. There have been concerns pointed out about privacy issues. For example, it would automatically identify an social network software (SNS) user from a selfie uploaded to her anonymous SNS account whose identity is not disclosed. These concerns are not limited to facial images. User's gender, race, and other sensitive attributes, including health conditions and diseases, could be estimated from voice data automatically and unintentionally (Oh et al. 2019). There is no widespread agreement among users on how to handle privacy issues related to data collected from users and the results of recognizing these data through AI (e.g., identities and other sensitive attributes). Technologies that allow users to control how AI

recognizes their data would be helpful from the perspective of enhancing privacy protection.

In (Gafni, Wolf, and Taigman 2019), techniques were proposed to eliminate features in face images of movies that allow identifying the person from face movies without making a significant change in its appearance. These techniques could, for example, allow users to leave anonymous, natural-looking video messages in public places, which would not be recognized by face recognition unintentionally. Furthermore, they can prevent users from being tracked for their activities by facial recognition or from being identified from videos uploaded to social networking sites. Adversarial examples have long been studied as an attack technique to manipulate the identification results of an image without changing its appearance, which can be designed not only for face images (Sharif et al. 2016; Dong et al. 2019; Deb, Zhang, and Jain 2019), but also for any objects (Xu et al. 2020; Eykholt et al. 2018; Cao et al. 2019). Adversarial example is used initially to make the classification performance of the classifier robust (Xie et al. 2020; Miyato et al. 2018), while several studies proposed to use adversarial examples to hinder unintentional face image identification from protecting privacy (Gafni, Wolf, and Taigman 2019; Zhang et al. 2020).

The technologies mentioned above allow users to leave anonymous, natural-looking image or video messages in public places without being identified unintentionally. However, one limitation of these technologies is that once features necessary for recognition are eliminated from the data or adversarial perturbation is inserted into data, no one except the holder of the original image can recover the original image. For example, (Yang et al. 2020) proposed face encryption by generating adversarial identity masks to protect users' face images from unauthorized face recognition systems. However, this type of face encryption can only be considered as face perturbation since there is no decryption process for it. Considering that the basic principle of privacy protection is to be able to control one's own information, it is desirable that users themselves can decide which of their images will be recognized and used by AI and which will not. Moreover, we try to not affect users' experience when users share the protected images on social media, and simultaneously conceal their information from unauthorized AI models.

Based on the above considerations, we propose a new

methodology to exploit the properties of adversarial examples to control how user’s data is recognized and used by AI. For this purpose, we propose reversible adversarial example (RAE), a new type of adversarial example. On the one hand, RAE is able to attack unauthorized AI models to prevent unauthorized access to users’ image data since RAE still functions as an adversarial example. On the other hand, authorized AI models can exactly recover the original image from RAE for recognition and further usage. We explain the functionalities of RAE by taking face recognition system as an example, but the idea of RAE works with any classification task. Suppose we have a face image \mathbf{X} (say, Alice’s face image). Using \mathbf{X} , we generate a RAE \mathbf{X}' with a classification model f , which is misrecognized as some label. Then,

- the appearance of RAE \mathbf{X}' is quite similar to original image \mathbf{X} to human,
- \mathbf{X}' works as an adversarial example for f (e.g., \mathbf{X}' is recognized as Bob),
- \mathbf{X}' is expected to work as an adversarial example for classifiers other than f due to transferability of adversarial examples, and
- some classifier f' specified by the one who created the RAE (say authorized classifier) can recover \mathbf{X} from \mathbf{X}' only, meaning only f' can recognize \mathbf{X}' correctly as Alice and use the original image \mathbf{X} .

In other words, RAE can be considered as encryption to computer vision. The reversibility of RAE guarantees the decryption of this type of encryption.

To realize RAE, we employ (1) reversible data hiding (Ni et al. 2006), which allows to embed secret information into an image without being able to detected, and (2) highly functional encryption (e.g., attribute-based encryption), which enables to generate ciphertexts in a way that the ones who can decrypt the ciphertexts can be selectively controlled by the private key holder.

We demonstrate by experiments on the ImageNet (Deng et al. 2009) dataset that RAEs can be misrecognized by unauthorized classifiers, while only the authorized classifier can recover the original images exactly. We demonstrate that the visual quality of RAEs: RAEs look overall similar to original images.

Preliminaries

Adversarial Example

First proposed by Szegedy *et al.* (Szegedy et al. 2014), various methods to generate adversarial examples have been presented. In this part, we outline the details of the adopted attack methods in this paper. We use $l(\mathbf{x}, y)$ to notate the cross-entropy loss function where \mathbf{X} is the input image and y is the true class for the input image.

FGSM (Goodfellow, Shlens, and Szegedy 2014) crafts an adversarial example under the ℓ_∞ norm as

$$\mathbf{X}^{\text{adv}} = \mathbf{X} + \epsilon \cdot \text{sign}(\nabla_{\mathbf{X}} l(\mathbf{X}, y)) \quad (1)$$

where $\nabla_{\mathbf{X}} l(\mathbf{X}, y)$ denotes the gradient of the loss function with respect to the input image.

BIM (Kurakin, Goodfellow, and Bengio 2016) extends FGSM by taking iterative gradient updates in the following equation:

$$\mathbf{X}^{(t+1), \text{adv}} = \text{clip}_{\mathbf{X}, \epsilon} \left(\mathbf{X}^{(t), \text{adv}} + \eta \cdot \text{sign} \left(\nabla_{\mathbf{X}} l \left(\mathbf{X}^{(t), \text{adv}}, y \right) \right) \right) \quad (2)$$

where $\text{clip}_{\mathbf{X}, \epsilon}$ guarantees the adversarial example to satisfy the ℓ_∞ constraint.

C&W adopts the original C&W loss (Carlini and Wagner 2017) based on the iterative mechanism of BIM to perform attack in classification tasks. In particular, the loss takes the form of

$$\mathbf{X}^{\text{adv}} = \underset{\mathbf{X}'}{\text{argmin}} \| \mathbf{X}' - \mathbf{X} \| \text{ subject to } Z(\mathbf{X}')_y < \max_{i \neq y} Z(\mathbf{X})_i - \kappa \quad (3)$$

where $Z(\mathbf{X})_i$ is the logit output of the classifier for the i th class.

Reversible Data Hiding

Data hiding (Zeng 1998) is referred to as a method to hide secret information into an image for covert communication. Reversible data hiding (RDH) (Ni et al. 2006) is a special type of data hiding, which allows to recover the original image without any distortion from the marked image and extract the embedded hidden data. Classical RDH algorithms are mainly divided into three categories, compression-embedding(Fridrich, Goljan, and Du 2002), difference expansion (Tian 2003), and histogram modification (Ni et al. 2006). We adopt the histogram modification based RDH in this paper. Intuitively, the algorithm of RDH leverages the fact that the color histograms of natural images distribute unevenly. By shifting the bins of the color histogram in a way that the visual appearance of the image does not change significantly, we can encode a certain amount of information into the image without being detected.

Let $\mathbf{X} \in R^{H \times W \times C}$ be a C -channel image (referred to as a cover image) where H and W denotes the height and width of the image, respectively. Let $M \in \{0, 1\}^*$ be a bit-string message of arbitrary length to be encoded into the cover image. The RDH process consists of two functions: encoding and decoding. We use $\text{RDH} : R^{H \times W \times C} \times \{0, 1\}^* \rightarrow R^{H \times W \times C}$ to denote the encoding method, which takes an image and a secret bit-string as inputs and outputs an encoded image (referred to as a marked image) with the same size. $\text{RDH}^{-1} : R^{H \times W \times C} \rightarrow R^{H \times W \times C} \times \{0, 1\}^*$ denotes the decoding method, which takes an encoded image as input, and outputs the exact recovery of the original image and the secret bit-string.

For a given integer a^1 , the encoding procedure in one color channel is processed as follows. For $i = 1, \dots, H$ and

¹In the process of recovery, a is a key parameter for recovery. So a small part of pixels in the cover image are preserved to embed the value of a utilizing the data hiding technique. Moreover, the information of recovering these preserved pixel values also needs to be treated as a secret message and embedded into the other part of pixels in the cover image utilizing RDH.

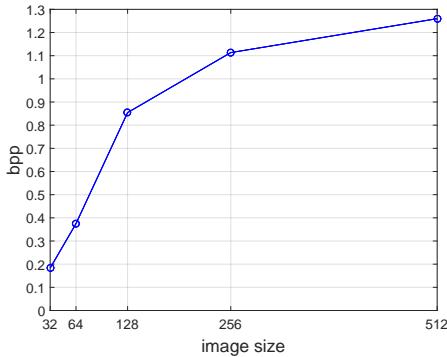


Figure 1: Capacity of RDH with different image sizes.

$$j = 1, \dots, W$$

$$\mathbf{X}_{i,j}^{\text{RDH}} = \begin{cases} \mathbf{X}_{i,j} - 1, & \text{if } \mathbf{X}_{i,j} < a \\ \mathbf{X}_{i,j} - M_{(i-1)W+j}, & \text{if } \mathbf{X}_{i,j} = a \\ \mathbf{X}_{i,j}, & \text{if } \mathbf{X}_{i,j} > a \end{cases} \quad (4)$$

where (i, j) represents the pixel coordinate in the cover image and $\mathbf{X}_{i,j}$ denotes the pixel value at the specified index. The length of binary string M is usually set as shorter than $H \times W \times C$ in RDH (see Figure 1 for the detail). When a is set as the peak of the color histogram, this process shifts the bins less than a towards smaller by 1 to create a vacant bin for data embedding. Each secret bit in M is embedded into the pixel whose value is a from left to right, top to bottom in the cover image. By taking a as the histogram peak, the capacity of embedding is known to be maximized. As a result, each bit of the string M is embedded into the pixel whose value is a of the cover image and the marked image \mathbf{X}^{RDH} is generated as $\mathbf{X}^{\text{RDH}} = \text{RDH}(\mathbf{X}, M)$.

On the decoder side, extraction of embedded bits M and recovery of the cover image \mathbf{X} is processed as follows:

$$M = \begin{cases} M \parallel 0, & \text{if } \mathbf{X}_{i,j}^{\text{RDH}} = a \\ M \parallel 1, & \text{if } \mathbf{X}_{i,j}^{\text{RDH}} = a - 1 \\ M, & \text{otherwise} \end{cases} \quad (5)$$

$$\mathbf{X}_{i,j} = \begin{cases} \mathbf{X}_{i,j}^{\text{RDH}} + 1, & \text{if } \mathbf{X}_{i,j}^{\text{RDH}} \leq a - 1 \\ \mathbf{X}_{i,j}^{\text{RDH}} & \text{otherwise} \end{cases} \quad (6)$$

where \parallel denotes concatenation. We denote these operations by $[\mathbf{X}, M] = \text{RDH}^{-1}(\mathbf{X}^{\text{RDH}})$.

Capacity of RDH We test the capacity of RDH on ImageNet (Deng et al. 2009). We adopt the state-of-the-art RDH method (Zhang et al. 2013). (Zhang et al. 2013) is a histogram modification method for RDH which embeds the data by recursively utilizing the decompression and compression processes of an entropy coder. We separately use 1000 images with 3 color channels randomly selected from ImageNet to perform the experiments. We resize the images of ImageNet into different sizes: $32 \times 32 \times 3$, $64 \times 64 \times 3$, $128 \times 128 \times 3$, $256 \times 256 \times 3$ and $512 \times 512 \times 3$. For each image, we gradually increase the length of to-be-embedded

bit-string until RDH cannot embed any bit into the image. Then we obtain the maximum length of a bit string that RDH can embed on each image and calculate the embedding rate. Finally, we average the embedding rate of 1000 images to measure the capacity of RDH. Figure 1 illustrates the capacity of RDH with different image sizes. The x-axis is the image size and the y-axis is the average bits per pixel (bpp) to be embedded. The result shows that around 1 bit per pixel can be embedded into an image on average. For example, when the size of an image is $299 \times 299 \times 3$ in ImageNet, about 270,000 bits can be embedded covertly.

Highly Functional Encryption

RDH allows us to embed secret bit-string into an image covertly; however, it does not necessarily mean that the security is guaranteed. We need to use cryptography to guarantee the security of the embedded message. More specifically, we encrypt the secret bits before embedding it into the cover image. Also, we can compress the bit-string before encryption when the size of the bit-string exceeds the capacity of RDH of the cover image. We can use any algorithm for encryption and compression. Unless necessarily mentioned, we suppose the bit-string is compressed and encrypted in the following.

Although any cryptosystem works with RDH, we can introduce highly functional cryptosystem in order to selectively control the AI to recognize the image. Suppose a communication of secret information between a sender and receiver. Public key cryptosystem is a cryptosystem that uses a pair of public keys pk (which may be known to others, including the sender) and private keys sk (which should be never known to anyone except the receiver).

In message transmission, anyone can encrypt a message $m \in \{0, 1\}^*$ using the receiver's public key as $c \leftarrow \text{Enc}_{\text{pk}}(m)$ where c is the resulting ciphertext, but the ciphertext can only be decrypted with the receiver's private key as $m \leftarrow \text{Dec}_{\text{sk}}(c)$.

When a sender wants to specify multiple receivers that can decrypt ciphertexts, attribute-based encryption can be used. Attribute-based encryption, such as (Bethencourt, Sahai, and Waters 2007; Okamoto and Takashima 2010), enables to encrypt and distribute a message so that the corresponding ciphertext can be decrypted only by entities that satisfy a policy specified at the timing of encryption (e.g., public sector only, authorized services only). Attribute-based encryption enables enforcement of policies by attributes in a non-interactive manner. Also, we can use timed-release encryption scheme (Dent and Tang 2007; Matsuda, Nakai, and Matsuura 2010) in which the sender specifies the time at which the receiver can decrypt the ciphertext.

In the following sections, we employ RSA instead of a highly functional encryption for simplicity, however this can be exchanged with any kind of high functional encryption without making any modifications.

Proposed Method

Overall Framework

In our problem setting, we have three stakeholders: user, authorized classifier and unauthorized classifier. The abstract

process proceeds as Algorithm 1².

Algorithm 1: Reversible Adversarial Example Creation

Input: image X

Output: reversible adversarial example X^{RDH}

- 1: All classifiers independently create a key pair of public-key cryptosystem and every classifier distributes its public key.
 - 2: User chooses an authorized classifier f and obtains f 's public key pk_f .
 - 3: User generates an adversarial example X' of image X .
 - 4: User encrypts adversarial perturbation $\Delta U = X' - X$ and auxiliary information R necessary for recovery with the f 's public key as $I = \text{Enc}_{\text{pk}_f}(\Delta U, R)$. Here we simply use RSA algorithm as the encryption method to achieve $\text{Enc}_{\text{pk}_f}(\Delta U, R)$.
 - 5: User embeds I into the adversarial example X' and obtains RAE as $X^{\text{RDH}} = \text{RDH}(X', I)$.
-

How the RAE is recognized is summarized as follows:

- When a human see X^{RDH} , it appears quite similar to X since RDH preserves visual quality of the cover image as demonstrated later.
- When an unauthorized classifier obtains X^{RDH} and recognizes X^{RDH} with his unauthorized classifier, X^{RDH} is expected to be misclassified because X^{RDH} works as an adversarial example. He might be able to obtain I by the decoding process of RDH, however, he can learn nothing from I because I is encrypted by a cryptosystem. So even when the unauthorized classifier learns I , exact recovery cannot be obtained.
- When an authorized classifier obtains X^{RDH} , he obtains I and can obtain $\Delta U, R = \text{Dec}_{\text{sk}_f}(I)$. With adversarial perturbation ΔU and auxiliary information R , the authorized classifier can obtain the exact recovery X as $X = X^{\text{RDH}} - \Delta U$.

In the following, we show the detailed realization of each step.

Generation of Adversarial Perturbation for RDH

A direct idea to achieve RAE is to embed adversarial perturbation into the adversarial image utilizing RDH scheme so that the receiver can invalidate the adversarial perturbation. However, RDH is only well suited to embedding a short length of information into a large image. In RDH, Figure 1 shows that the average embedding rate is around 1 bit per pixel. Since the size of adversarial perturbation is usually the same size as the cover image, the capacity of RDH is usually

²Since I is encoded as a bit-string, we can use any lossless information compression algorithm to reduce the size of information to embed. In our experiments, we use arithmetic coder (Howard and Vitter 1994) for compression.

not enough to directly embed the adversarial perturbation into the image. For example, if we set the ℓ_∞ norm perturbation of BIM as 2, the required embedding rate is $\log_2 4 = 2$ bits per pixel. In order to solve this problem, we propose to divide the images into super-pixels (e.g., treat pixels in 2×2 tile as a single pixel by smoothing) and then embed adversarial perturbation generated for the super-pixels.

We denote X as the original image with size $H \times W \times C$ and X' as its adversarial example with size $H \times W \times C$. Each pixel can take an integer value in $\{0, 1, \dots, 255\}$. For each color channel, C (C is either of r , g or b), the original image X and the adversarial image X' are divided into non-overlapping tiles with the same size $h \times w$, which are called super-pixels. When $h = 2$ and $w = 2$, the super-pixel consists of a set of four neighboring pixels. Let $P_{i,j}$ and $P'_{i,j}$ be the (i, j) th super-pixel of the original image and corresponding adversarial example where $1 \leq i \leq \lceil H/h \rceil$, $1 \leq j \leq \lceil W/w \rceil$. We consider a type of adversarial perturbation where the pixel values of adversarial perturbation are smoothed over each super-pixel. Due to smoothing, the amount of information contained in such adversarial perturbation can be reduced up to $\frac{1}{h \times w}$, which is sufficiently small to embed with RDH.

Post Smoothing Method This post smoothing super-pixel adversarial attack method is the most straightforward way to realize adversarial example over super-pixels. We first generate an adversarial example in an arbitrary way. Letting $n = h \times w$, we denote the collection of the super-pixels of the image and corresponding adversarial example by $P_{i,j} = \{p_1, p_2, \dots, p_n\}$ and $P'_{i,j} = \{p'_1, p'_2, \dots, p'_n\}$, respectively. First, calculate the average value of adversarial perturbations of all pixels in each super-pixel and round it to get the closest integer $\Delta U_{i,j}$

$$\Delta U_{i,j} = \text{round} \left(\frac{1}{n} \sum_{k=1}^n p_k - \frac{1}{n} \sum_{k=1}^n p'_k \right). \quad (7)$$

We call this smoothing method as post smoothing.

Then we get the adversarial example generated with the post smoothing method as $p''_k = p_k + \Delta U_{i,j}$. Note that the pixel value p''_k should be an integer between 0 and 255, so the transformation may result in some overflow/underflow pixel values. For exact recovery, the truncation information for each super-pixel is recorded as $R_{i,j} = \{r_1, r_2, \dots, r_n\}$ where $n = h \times w$.

After transformation and truncation, we get a new tile $P''_{i,j}$. We get a super-pixel adversarial image X'' by completing transformations and truncations for all the tiles $P''_{i,j}$ where $1 \leq i \leq \lceil H/h \rceil$ and $1 \leq j \leq \lceil W/w \rceil$.

In-the-loop Smoothing Method One drawback of the post-smoothing method is that the adversarial perturbation is smoothed after finishing optimization, which can lower the attack success rate. To avoid this, we introduce the in-the-loop smoothing super-pixel adversarial attack method, which is more costly, but is expected to have less effect to the attack success rate. To describe how to generate the super-pixel adversarial perturbation, we take BIM (Kurakin,

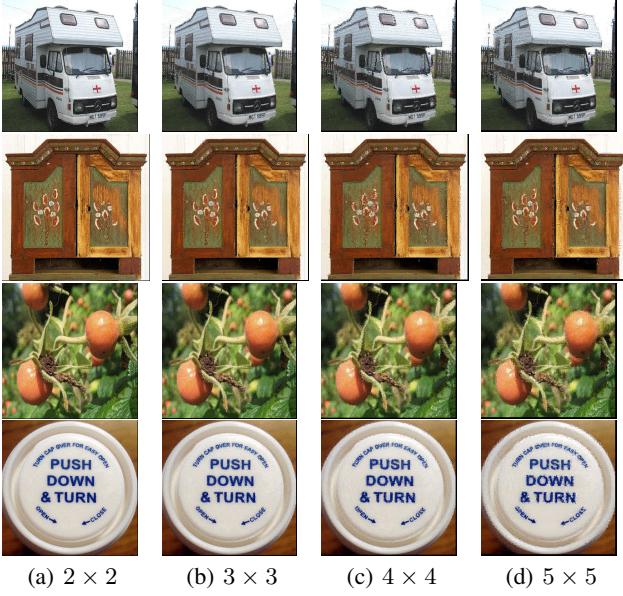


Figure 2: Visual quality of reversible adversarial examples with different super-pixel sizes.

Goodfellow, and Bengio 2016) as example and propose the in-the-loop smoothing version of BIM.

BIM generates adversarial perturbation by repeatedly update \mathbf{X} by the gradient of the loss function with respect to \mathbf{X} so that the image is recognized as a wrong label. The idea of the in-the-loop smoothing is to take the gradient with respect to a noise vector $\Delta \mathbf{U}$ whose length is the super-pixel number. We initialize the noise vector $\Delta \mathbf{U}$ by randomly sampling with uniform distribution in $[-\epsilon/2, \epsilon/2]$.

$$\Delta \mathbf{U}^{(t)} = \Delta \mathbf{U}^{(t-1)} + \eta \cdot \text{sign} \left(\nabla_{\Delta \mathbf{U}} l \left(\mathbf{X}^{(t-1), \text{adv}}, y \right) \right) \quad (8)$$

After update, we obtain the t th update $\mathbf{X}^{(t), \text{adv}}$ by

$$\mathbf{X}^{(t), \text{adv}} = \text{clip}_{\mathbf{X}, \epsilon} \left(\mathbf{X} + f_{\text{pad}}(\Delta \mathbf{U}^{(t)}, h, w) \right) \quad (9)$$

where function f_{pad} is a mapping function which fills each element of the noise vector $\Delta \mathbf{U}^{(t)}$ into a $h \times w$ super-pixel to get the full-size image of adversarial perturbation padded with the super-pixel values. Note that $\text{clip}_{\mathbf{X}, \epsilon}$ operates on super-pixels of \mathbf{X} and guarantees that the perturbations within each super-pixel are still same. A super-pixel adversarial example is generated by alternate iterations of eq. 8 and eq. 9.

Experiments

In this section, we experimentally evaluate to what extent the RAEs can deceive the DNN. If RAEs have a similar attack success rate to regular adversarial examples, we can say unauthorized classifiers cannot recognize given images correctly.

Thanks to RDH, the scheme of the proposed method guarantees that the authorized classifier can exactly recover the

Table 1: The attack success rate (%) of adversarial examples (AE) and reversible adversarial examples (RAE) on the ImageNet dataset in the white-box setting (Inception-v3).

Attack Method	Parameter	AE	RAE
FGSM	$\epsilon = 4$	76.70	71.58
	$\epsilon = 8$	83.82	79.87
	$\epsilon = 16$	88.18	86.08
BIM (post)	$\epsilon = 4$	96.71	73.84
	$\epsilon = 8$	97.83	83.61
	$\epsilon = 16$	99.78	89.45
BIM (in-the-loop)	$\epsilon = 4$	96.71	81.01
	$\epsilon = 8$	97.83	94.67
	$\epsilon = 16$	99.78	97.41
C&W (post)	$\kappa = 0$	98.62	28.68
	$\kappa = 50$	99.20	81.43
	$\kappa = 100$	99.68	95.45

original image from the RAE without any distortion. So we do not evaluate the classification accuracy of adversarial from the viewpoint of the authorized classifier. We confirmed that RAEs generated for our experiments were recovered to be the original images exactly.

We perform our experiments on 100,000 images randomly selected from the ImageNet (Deng et al. 2009). Since only correctly classified images are considered for evaluation of the attack ability, the classification accuracy of the original set is 100%. The pretrained Inception-v3 (Szegedy et al. 2016) is adopted as the default target attack model, which is evaluated with top-1 accuracy. The values of pixels per color channel of the images range from 0 to 255. The adversarial attack methods used in the experiments are FGSM, BIM, and C&W, and we perform untargeted attacks in the experiments. For FGSM, we set the perturbation budget ϵ as 4/255, 8/255 and 16/255. For BIM, we set the perturbation budget ϵ as 4/255, 8/255 and 16/255, the number of iterations as 20, the step size η as 1/255. We use ℓ_∞ norm distance metric for FGSM and BIM. For C&W, we use ℓ_2 distance metric and $\kappa = 0, 50, 100$ where κ is the parameter of attack confidence.

Selection of Super-pixel Size

Intuitively, smaller super-pixel size leads to less influence on the attack success rate of the generated RAE while larger super-pixel size leads to larger amount of data can be embedded using RDH. In addition, we illustrate several samples of RAEs with different super-pixel sizes. To have a clearer difference, in-the-loop smoothing version of BIM with $\epsilon = 4$ is used here. As shown in Figure 2, smaller super-pixel size achieves better visual quality of RAEs. So we choose smaller super-pixel size in the experiments to achieve higher attack success rate and better visual quality of RAEs.

For FGSM, the required embedding rate for RDH is $\log_2 2 = 1$ bits per pixel. This embedding rate is within the capacity of RDH. So we set the super-pixel size of FGSM as 1×1 . That is to say, we use RDH to directly embed the adversarial perturbation of FGSM into the adversarial image

Table 2: The attack success rate (%) of adversarial examples and reversible adversarial examples on the ImageNet dataset. * indicates the target model.

Model	white-box	black-box			
		Inc-v3*	Inc-v4	IncRes-v2	Inc-v3 _{adv}
FGSM	83.82	53.25	55.90	55.08	
RAE-FGSM	79.87	48.28	50.54	50.12	
BIM	97.83	40.72	37.23	30.82	
RAE-BIM (post)	83.61	27.34	24.37	19.67	
RAE-BIM (in-the-loop)	94.67	35.31	30.73	25.57	
C&W	99.68	2.33	1.75	0.94	
RAE-C&W (post)	95.45	1.12	0.85	0.34	

to generate the RAE. So the in-the-loop smoothing method for FGSM turns to be same as the post smoothing method for FGSM when super-pixel size of FGSM is 1×1 .

For BIM, the required embedding rate of $\epsilon = 4$ is $\log_2 8 = 3$ bits per pixel, the required embedding rate of $\epsilon = 8$ is $\log_2 16 = 4$ bits per pixel and the required embedding rate of $\epsilon = 16$ is $\log_2 32 = 5$ bits per pixel. So we set the super-pixel size of BIM with $\epsilon = 4$ as 1×3 , $\epsilon = 8$ as 2×2 and $\epsilon = 16$ as 1×5 . Here we use the post smoothing method and the in-the-loop smoothing method for BIM.

For C&W, the bit length of adversarial perturbation for different images changes a lot. So we gradually increase the super-pixel size for each adversarial image from 1×1 to 1×2 , 1×3 and 2×2 until the capacity of RDH is enough to embed the perturbation. Here we use the post smoothing method for C&W because the optimization of the in-the-loop smoothing method did not complete within the practical computation time. We suspect the reason may be that in-the-loop version of C&W is not so effective in finding the suitable adversarial perturbations for super-pixels.

Attack Ability in the White-box Setting

To evaluate the attack ability, we adopt the attack success rate against the pretrained classification models on ImageNet dataset. Table 1 shows results on the attack success rate of reversible adversarial examples (RAE) generated by FGSM, BIM and C&W with parameters of different values in the white-box setting. The column ‘‘AE’’ is the result of adversarial examples while ‘‘RAE’’ is the result of reversible adversarial examples. We compare three baselines FGSM, BIM and C&W with our corresponding RAE. For BIM, we presented the results of two variants, post smoothing and in-the-loop smoothing. For FGSM, the in-the-loop smoothing method is same as the post smoothing method when super-pixel size of FGSM is 1×1 . For C&W, only the result of the post smoothing method is shown.

From the result, we can see that the gap of the attack success rate between AE and RAE of FGSM is less than 6 percent. For the in-the-loop smoothing method of BIM, the gap of the attack success rate between AE and RAE is less than 4 percent when the perturbation budget ϵ is not smaller than $8/255$. For C&W, the gap of the attack success rate between AE and RAE is less than 5 percent when κ is set as 100. So the attack success rates of regular adversarial examples and RAEs in the white-box setting are quite close when we set

larger perturbation budgets, indicating RAEs can work correctly. Moreover, the success rate of RAE generated by the in-the-loop smoothing method of BIM is at least 7 percent higher than the post smoothing method of BIM, which reveals that the results of in-the-loop smoothing gives better attack success rate compared to the post smoothing method. This is because adversarial perturbation generated with the in-the-loop smoothing is made considering the super-pixel strategy while that of post smoothing ignores this in the process of adversarial perturbation generation. Finally, we get the conclusion from Table 1 that RAE generated by larger budget adversarial perturbations can achieve comparable attack ability with regular adversarial examples in the white-box setting.

Visual Results

To evaluate the visual quality of RAE with different perturbation budgets, we show the visual results of in-the-loop smoothing method for BIM with $\epsilon = 4/255$, $\epsilon = 8/255$ and $\epsilon = 16/255$ respectively in Figure 3, 4 and 5. For each row, we show the original images (top), the adversarial examples (middle) and the RAEs (bottom) respectively. We can see that the adversarial images and their corresponding RAEs look almost same. When the perturbation budget becomes larger, the visual quality of RAE becomes worse to human eyes. But human eyes can hardly perceive the difference between RAEs and the original images when the perturbation budget ϵ is not larger than 8. These visual results reveal that RAE can still achieve good visual quality to human eyes when the adversarial perturbations are not large.

Attack Ability in the Black-box Setting

In real applications, the black-box setting is more important for RAE. So we also evaluate the attack success rate of RAEs with larger perturbation budget in the black-box setting. The intention of the experiment setting is that users mainly rely on the transferability of RAEs in real applications since users usually can not have direct access to the authorized/unauthorized classifiers. In Table 2, we generated RAEs with Inc-v3, and evaluated the attack success rate with Inception-v4 (Szegedy et al. 2017), Inception Resnet-v2 (Szegedy et al. 2017), and Inception-v3_{adv}. Here, Inception-v3_{adv} denotes Inception-v3 (Szegedy et al. 2016) model with ensemble adversarial training (Tramer

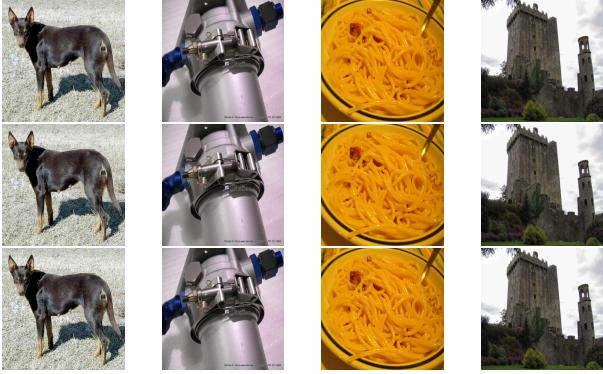


Figure 3: Visual results of original images (top), adversarial examples generated by BIM with $\epsilon = 4/255$ (middle) and reversible adversarial examples generated by in-the-loop smoothing method for BIM with $\epsilon = 4/255$ (bottom).

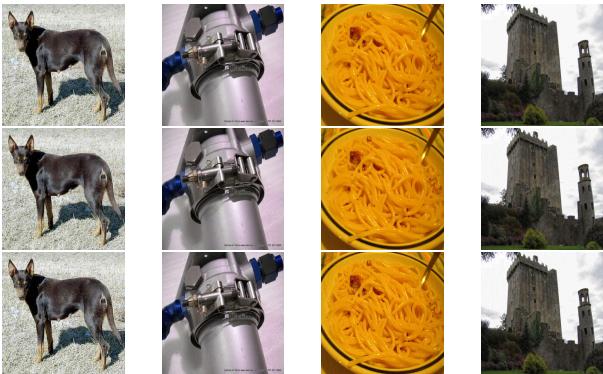


Figure 4: Visual results of original images (top), adversarial examples generated by BIM with $\epsilon = 8/255$ (middle) and reversible adversarial examples generated by in-the-loop smoothing method for BIM with $\epsilon = 8/255$ (bottom).

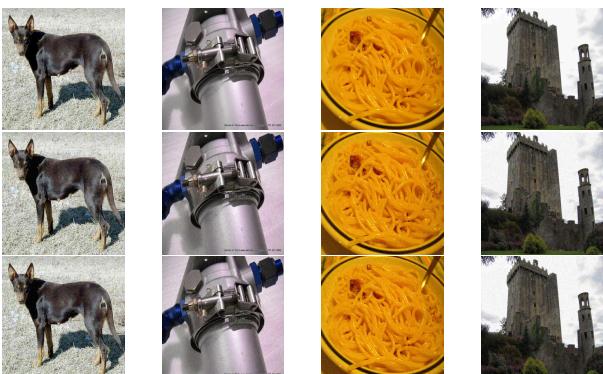


Figure 5: Visual results of original images (top), adversarial examples generated by BIM with $\epsilon = 16/255$ (middle) and reversible adversarial examples generated by in-the-loop smoothing method for BIM with $\epsilon = 16/255$ (bottom).

et al. 2017) which is known to be robust against adversarial attack. We set the perturbation budget ϵ as $8/255$ for FGSM and BIM, and the attack confidence κ as 100 for C&W. We can confirm that attack success rate in the black-box setting is lowered compared to the white-box setting in both regular adversarial examples and RAEs. Also, the gap between regular adversarial examples and RAEs is enlarged. To have better attack success rate in the black-box setting, introducing techniques to increase the transferability (Xie et al. 2019; Huang et al. 2019) into RAE is straightforward.

Discussion

This paper proposes the first prototype framework of RAE and makes a preliminary attempt in the black-box setting. To have better attack success rate in the black-box setting, introducing techniques to increase the transferability can be a straightforward solution. In addition, reducing the drop of attack ability in black-box setting can also be exploited by improving the reversible data hiding algorithm or using more efficient encryption schemes, such as “key agreement protocol” + “symmetrical encryption”, to reduce the length of information that reversible data hiding needs to embed.

The purpose of RAE is to control how user’s data is recognized by AI. We believe this proposal would have positive societal impact. However, attackers with malicious purpose may use RAE to attack AI. Since RAE is one type of adversarial example, it can have similar potential negative societal impacts as adversarial example. So RAE will also pose a potential threat to safety and security critical applications. To deal with the potential threat of RAE, adversarial defense methods can be adopted as countermeasures to protect deep neural networks.

Conclusion

In this paper, we propose reversible adversarial example to control how user’s data is recognized and used by AI. The AI specified by the user can exactly recover the original image from the RAE while other AIs cannot recognize the RAE correctly. This paper proposes the concept of RAE and give the first prototype framework to verify its feasibility. The proposed method combines adversarial example, reversible data hiding and encryption to realize RAE. Experiments show that the proposed method can achieve comparable attack ability with the corresponding adversarial attack method and similar visual quality with the original image. In the future, we will further investigate how to increase the attack ability of the RAE in the black-box setting.

References

- Bethencourt, J.; Sahai, A.; and Waters, B. 2007. Ciphertext-policy attribute-based encryption. In *2007 IEEE symposium on security and privacy (SP’07)*, 321–334. IEEE.
- Cao, Y.; Xiao, C.; Yang, D.; Fang, J.; Yang, R.; Liu, M.; and Li, B. 2019. Adversarial objects against lidar-based autonomous driving systems. *arXiv preprint arXiv:1907.05418*.

- Carlini, N.; and Wagner, D. 2017. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy*, 39–57.
- Deb, D.; Zhang, J.; and Jain, A. K. 2019. Advfaces: Adversarial face synthesis. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, 1–10. IEEE.
- Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.
- Dent, A. W.; and Tang, Q. 2007. Revisiting the security model for timed-release encryption with pre-open capability. In *International Conference on Information Security*, 158–174. Springer.
- Dong, Y.; Su, H.; Wu, B.; Li, Z.; Liu, W.; Zhang, T.; and Zhu, J. 2019. Efficient decision-based black-box adversarial attacks on face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7714–7722.
- Eykholz, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; and Song, D. 2018. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1625–1634.
- Fridrich, J.; Goljan, M.; and Du, R. 2002. Lossless data embedding for all image formats. *electronic imaging*, 4675: 572–583.
- Gafni, O.; Wolf, L.; and Taigman, Y. 2019. Live face de-identification in video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9378–9387.
- Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- Howard, P. G.; and Vitter, J. S. 1994. Arithmetic coding for data compression. *Proceedings of the IEEE*, 82(6): 857–865.
- Huang, Q.; Katsman, I.; He, H.; Gu, Z.; Belongie, S.; and Lim, S.-N. 2019. Enhancing adversarial example transferability with an intermediate level attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4733–4742.
- Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*.
- Matsuda, T.; Nakai, Y.; and Matsuura, K. 2010. Efficient generic constructions of timed-release encryption with pre-open capability. In *International Conference on Pairing-Based Cryptography*, 225–245. Springer.
- Miyato, T.; Maeda, S.-i.; Koyama, M.; and Ishii, S. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8): 1979–1993.
- Ni, Z.; Shi, Y.; Ansari, N.; and Su, W. 2006. Reversible data hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(3): 354–362.
- Oh, T.-H.; Dekel, T.; Kim, C.; Mosseri, I.; Freeman, W. T.; Rubinstein, M.; and Matusik, W. 2019. Speech2face: Learning the face behind a voice. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7539–7548.
- Okamoto, T.; and Takashima, K. 2010. Fully secure functional encryption with general relations from the decisional linear assumption. In *Annual cryptology conference*, 191–208. Springer.
- Sharif, M.; Bhagavatula, S.; Bauer, L.; and Reiter, M. K. 2016. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 1528–1540.
- Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.
- Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.
- Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. *International Conference on Learning Representations*.
- Tian, J. 2003. Reversible data embedding using a difference expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(8): 890–896.
- Tramer, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; and McDaniel, P. 2017. Ensemble Adversarial Training: Attacks and Defenses. *arXiv: Machine Learning*.
- Xie, C.; Tan, M.; Gong, B.; Wang, J.; Yuille, A. L.; and Le, Q. V. 2020. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 819–828.
- Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; and Yuille, A. L. 2019. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2730–2739.
- Xu, K.; Zhang, G.; Liu, S.; Fan, Q.; Sun, M.; Chen, H.; Chen, P.-Y.; Wang, Y.; and Lin, X. 2020. Adversarial t-shirt! evading person detectors in a physical world. In *European Conference on Computer Vision*, 665–681. Springer.
- Yang, X.; Dong, Y.; Pang, T.; Su, H.; Zhu, J.; Chen, Y.; and Xue, H. 2020. Towards Face Encryption by Generating Adversarial Identity Masks. *arXiv preprint arXiv:2003.06814*.
- Zeng, W. 1998. Digital watermarking and data hiding: technologies and applications. In *Proc. Int. Conf. Inf. Syst., Anal. Synth*, volume 3, 223–229.
- Zhang, J.; Sang, J.; Zhao, X.; Huang, X.; Sun, Y.; and Hu, Y. 2020. Adversarial Privacy-preserving Filter. In *Proceedings of the 28th ACM International Conference on Multimedia*, 1423–1431.

Zhang, W.; Hu, X.; Li, X.; and Yu, N. 2013. Recursive Histogram Modification: Establishing Equivalency Between Reversible Data Hiding and Lossless Data Compression. *IEEE Transactions on Image Processing*, 22(7): 2775–2785.