Received: 31 August 2018 | Revised: 9 October 2018 | Accepted: 25 November 2018
DOI: 10.1002/cpe.5102

WILEY

SPECIAL ISSUE PAPER

Adversaries or allies? Privacy and deep learning in big data era

Bo Liu[1] | Ming Ding[2] | Tianqing Zhu[3] | Yong Xiang[4] | Wanlei Zhou[3]

[1]Department of Engineering, La Trobe University, Melbourne, VIC, Australia
[2]Data61, CSIRO, NSW, Australia
[3]School of Software, University of Technology Sydney, Sydney, NSW, Australia
[4]School of IT, Deakin University, Melbourne, VIC, Australia

Correspondence
Bo Liu, Department of Engineering, La Trobe University, Melbourne, VIC, Australia.
Email: b.liu2@latrobe.edu.au

Summary

Deep learning methods have become the basis of new AI-based services on the Internet in big data era because of their unprecedented accuracy. Meanwhile, it raises obvious privacy issues. The deep learning–assisted privacy attack can extract sensitive personal information not only from the text but also from unstructured data such as images and videos. In this paper, we proposed a framework to protect image privacy against deep learning tools, along with two new metrics that measure image privacy. Moreover, we propose two different image privacy protection schemes based on the two metrics, utilizing the adversarial example idea. The performance of our solution is validated by simulations on two different datasets. Our research shows that we can protect the image privacy by adding a small amount of noise that has a humanly imperceptible impact on the image quality, especially for images of complex structures and textures.

KEYWORDS

deep learning, image, neural networks, privacy

1 | INTRODUCTION

With the Facebook data privacy scandal occupying the recent headlines of major media,[1] the privacy issue once again raises people's attention. It prompts us to revisit the privacy challenges in a big data era with various intelligent technologies emerging. For example, the newly emerged deep learning technique will have a profound and long-lasting impact on the privacy problem. In more detail, recent rapid developments have shown that big dataset–assisted deep learning methods can detect objects' type and identify celebrities and landmarks, from users' personal photos posted on social networks with unprecedented accuracy.[2] This powerful tool might be both adversary and ally to privacy protection. On the one hand, deep learning methods can automatically collect and process millions of photos or videos to reveal private and sensitive information from social networks. Traditional privacy preserving method seems powerless when facing the large-scale deep learning tools. On the other hand, deep learning can also be used to find and delete sensitive information in the massive data and improve the efficiency of privacy protection. Therefore, to fully understand the interaction of privacy and deep learning requires an urgent treatment.

Some researchers have recently started to investigate the privacy problem in the context of deep learning. Most of them study the privacy problem from the viewpoint of the deep learning model, ie, focusing on the privacy challenge and risks of the models. For example,[3] Fredrikson et al[4] discovered a model-inversion attack which can rebuild images from a facial recognition system. Ababi et al[5] assumed that the adversaries would have access to the trained model and may have the full knowledge of the training mechanism with parameters. Phan et al[6] claimed that privacy leaks can stem from malicious inference with the model's inputs and outputs. These works presented possible attacks on deep learning models. There are several other research papers which do not fall in the scope of the deep learning model related privacy topic. For example, Yu et al[7] developed an approach to automatically identify privacy sensitive object classes and their privacy settings. Liu et al[8] proposed an algorithm of elaborating adversarial examples to resist the automatic detection system based on the Faster RCNN framework. Overall, the research in this area is still in its infancy and needs further investigation.

In light of the existing literature and our view of this research topic, there are three key challenges for privacy-preserving with the presence of deep learning technology.

1. Both the attacker and the privacy objective will undergo significant changes. New type of attackers may be aided with deep learning to achieve unprecedented accuracy during the information acquisition process. Meanwhile, the wide deployment of deep learning tools raises new privacy concerns related to the deep learning models and parameters.
2. Deep learning outperforms traditional methods, especially in the areas of computing vision and big data mining, which makes the privacy preservation a much difficult challenge. Traditional privacy preserving mechanism is not suitable in these areas. For example, even we can use cryptography to partly solve the problem, photo sharing in social networks still discloses private information.
3. Developing privacy-preserving schemes for individuals is difficult as different persons have diverse privacy perceptions and requirements.

In this paper, we will investigate the privacy protection problems in the context of deep learning. Especially, we focus on image privacy which has not been well studied in traditional privacy research but at the same time is one of the most important use cases in deep learning. Based on an overall investigation of the emerging challenges in this area, we will then focus on a privacy protection case which has rarely been studied yet, ie, the attacker is the automatic deep learning tool without human supervision. Our approach is to protect the privacy against deep learning network by adding a humanly imperceptible noise on the original image. To achieve privacy protection in this scenario, we revise the unstructured data to conceal specific features. Specifically, we apply a small but intentionally worst-case perturbation to the initial image that can mislead the deep learning network while the appearance of the image does not change to human eyes. This perturbed image[9] is called an "adversarial example." Based on the adversarial example concept, we will develop new methods that can manage different types of neural networks at the same time.

In summary, the contributions of this paper are as follows:

- For the first time, we develop an architecture which implements the idea of adversarial example for image privacy protection against deep learning image classification tools.
- We propose two privacy metrics: image classification probability metric and image classification entropy metric to measure the image privacy.
- Based on the proposed two metrics, we propose two privacy protection schemes, which perform well against adversaries equipped with deep learning tools.
- We evaluate and compare the performances of our proposed schemes on two different datasets and investigate the reasons for the different results.

The rest of the paper is organized as follows. Section 2 gives the preliminary background knowledge and briefly reviews the related work. Section 3 gives a detailed introduction to the system model, the two image privacy metrics and the problem formulation. The proposed image privacy protection schemes are described and analyzed in Section 4. Performance analysis and extensive numerical simulations are presented in Section 5. Finally, Section 6 draws our conclusions.

2 | PRELIMINARY

2.1 | Deep neural networks

In the area of computer vision, the most advanced algorithms are based on deep neural networks (DNN). Moreover, as most of the successful DNN architectures adopt the convolutional structure, here we mainly discuss the convolutional neural network (CNN). CNNs are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linear activation function. The whole network still realizes a single differentiable score function: from the raw image pixels on one end to class scores at the other.[10]

There are several most commonly used CNN architectures. LeNet[11] is the first successful applications of CNNs. AlexNet[12] was submitted to the ImageNet ILSVRC challenge in 2012 and significantly outperformed the second runner-up.[10] It is this work that popularized CNNs in Computer Vision. Szegedy et al from Google proposed the GoogLeNet[13] in 2014 which introduced an Inception Module that dramatically reduced the number of parameters in the network (4M, compared to AlexNet with 60M). Additionally, this paper uses Average Pooling instead of Fully Connected layers at the top of the ConvNet, eliminating a large number of parameters. There are also several followup versions to the GoogLeNet, eg, Inception-v3,[14] Inception-v4.[15] Simonyan and Zisserman proposed VGGnet[16] and which shows that the depth of the network is a critical component for good performance. Residual Network[17] developed by He et al features special skip connections and heavy use of batch normalization. ResNets are currently used widely in practice.

The main-stream applications of DNNs include image classification,[12,15,16] object detection,[18] recognition,[19] and tracking,[20] Semantic Segmentation,[21] etc. Outputs of DNNs in these applications contain rich information such as type and position of objects, identity and action of people, thus make DNNs and privacy issues highly relevant. We will discuss the problem in detail in the following part.

2.2 | Privacy protection and deep learning in big data era

DNNs bring new challenges to privacy protection which cannot be tackled under the traditional privacy protection framework.[22] There has been some initial research in this area which can be divided into three categories.

TABLE 1 Different categories of privacy protection problem with deep learning context

Attacker	Target	Tools
human	data privacy	Deep learning
human+Deep learning	model privacy	Deep learning
Deep learning	data privacy	Adversarial noise

First, making the DNN private from all aspects. This will include the model parameter privacy and output privacy, and the privacy leakage may happen during training, publishing or prediction process. Researchers have tried to apply the differential privacy concept in deep learning models.[23] For example,[3] Abadi et al[5] clipped the objective function to bound its sensitivity and applied a moment accountant method to the objective function to form an optimal privacy composition. Phan et al[6] applied a functional mechanism to perturb the objective function and decrease noise. Shokri and Shmatikov[24] designed a distributed deep learning model training system that enables multiple parties to jointly learn an accurate neural network, and developed the differentially private SGD algorithm with convex objective functions.

Second, using DNNs to improve the privacy protection. For example, Yu et al[7] developed an approach to automate the process of identifying privacy sensitive object classes and their privacy settings.

Third, developing mechanisms to protect the user's privacy against deep learning, eg, Liu et al[8] proposed an algorithm of elaborating adversarial examples to resist the automatic detection system based on the Faster RCNN framework. However, the authors only discussed the object detection case.

The above three categories are not exclusive. There might be some cases fall into multiple categories. For example, the attacker might be a human equipped with deep learning tools and targets on the data privacy. In this case, the adversary first uses deep learning tools to dig information on a large scale, and then check the preliminary results manually. In this case, we need to consider noises which are applicable both to deep learning tools and human at the same time. Table 1 summarizes the different categories of privacy protection problems involving deep learning tools.

Our research in this paper falls into the third category. Different from the research of Liu et al,[8] we will focus on the image classification problem. It is the most widely and well-studied topic in deep learning and also poses the severest privacy challenge in practice.

2.3 | Adversarial examples and adversarial noise

Along with the development of DNNs, many researchers also find that there are limitations of deep learning. Specifically, it is found to be vulnerable to some well-designed inputs termed *adversarial examples*. Szegedy et al[25] first discovered that even an almost imperceptible noised added on the original image would cause DNNs to misclassify the tampered image into a completely unrelated category. Then, Goodfellow et al[9] proposed the fast gradient sign method (FGSM), which can be used to generate adversarial examples. They[9] also found that the adversarial examples have the transferability property. It means an adversarial image designed to mislead one model is very likely to mislead another as well. That is to say, it might be possible for us to craft adversarial perturbation in the circumstance of not having access to the underlying DNN model.

3 | SYSTEM MODEL AND PROBLEM FORMULATION

3.1 | System architecture

In this paper, we focus on the scenario of social networks. In more detail, users post images on a social network platform. Attackers are assumed to collect images by a crawler and use DNNs to dig sensitive information. Figure 1 presents an example of such system architecture. When a user shares an image on social networks without any preprocessing on the original image, an adversarial equipped with DNNs can automatically obtain useful information from this image (ie, this is a giant panda with high confidence, indicating a likely event of visiting a zoo). Other sensitive information such as the user's activity, location or even name can be detected by similar powerful deep learning models. In order to prevent the privacy leakage, we will add some noises to the original image, so the released image can mislead the DNN models to get the wrong information. Meanwhile, we hope to keep the noise as small as possible so that it has a minor impact on the image quality and user experience.

3.2 | Privacy metrics for image

In order to evaluate the performance of our scheme, we need to define privacy metrics first. The metrics have been discussed extensively,[26] with probability and entropy as the most frequently used mathematical measures. Also, traditional privacy metrics might be revised to define the privacy metric for the image. Especially, we consider the case of image classification problem in our paper.

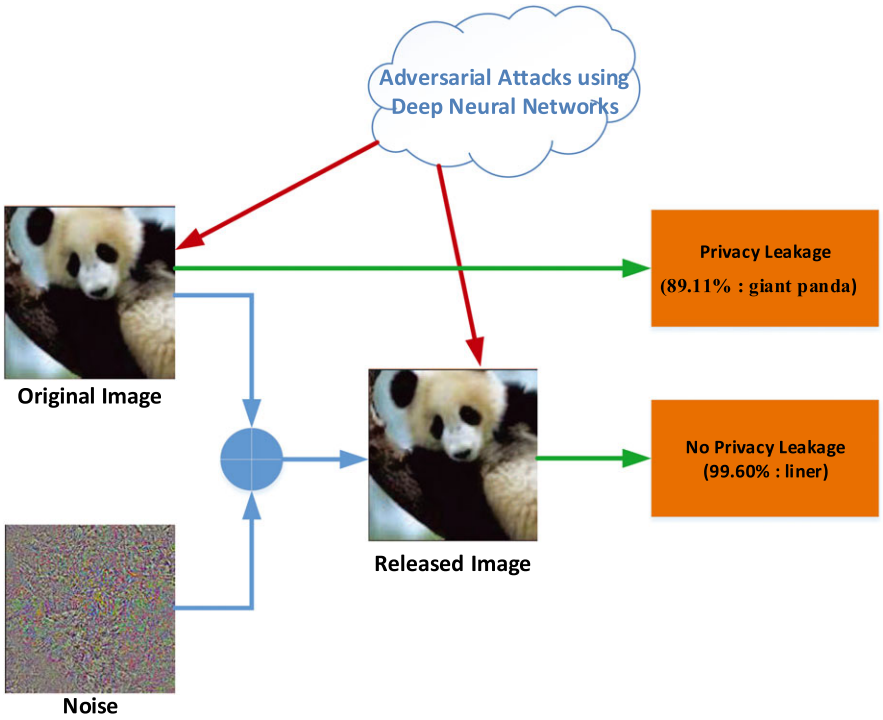

FIGURE 1 An example of the system architecture

3.2.1 | Probability-based metric

First, we define a probability-based metric. It is calculated as the probability of the adversarial DNNs obtaining the correct information, ie,

$$\Pr(class_p = class_{\boldsymbol{x}} | \boldsymbol{o}), \tag{1}$$

where $\boldsymbol{o}$ is the observation, $class_p$ is the predicted class of the adversary, and $class_{\boldsymbol{x}}$ is the true class of the original image $\boldsymbol{x}$.

As it is designed for the image classification problem, we name it the *image classification probability* (ICP) metric. Intuitively speaking, the smaller the ICP, the larger the false detection probability, and hence the higher level of privacy protection can be achieved.

3.2.2 | Entropy-based metric

On the other hand, the entropy-based metric can be computed by the posterior probability of the attacker's estimation based on the results from the DNNs:

$$-\sum_i \Pr(class_p = class_i | \boldsymbol{o}) \log \Pr(class_p = class_i | \boldsymbol{o}), \tag{2}$$

where i is any class in the DNN model.

As a result, we name this metric as the *image classification entropy* (ICE) metric. Intuitively speaking, the larger the ICE, the more ambiguity in classification, and hence the higher level of privacy protection can be achieved.

3.3 | Problem formulation

Based on the above two metrics, we can define the image privacy protection problems as optimization problems.

First, if we formulate the problem based on the ICP metric, the target is to minimize the value, so that the probability of the image being correctly classified by the attacker can be small and the user's privacy level will be high, ie,

$$\text{P1:} \quad \min \ \Pr(class_p = class_{\boldsymbol{x}} | \boldsymbol{o}). \tag{3}$$

The output of $P1$ will be a number between 0 and 1, where "0" means completely private and "1" indicates no privacy.

On the other hand, if the ICE metric is used, the image privacy protection problem will be formulated as the optimization one where the target is to maximize the entropy, ie,

$$\text{P2:}\quad \max -\sum_i \Pr(class_p = class_i|\boldsymbol{o}) \log \Pr(class_p = class_i|\boldsymbol{o}). \tag{4}$$

The optimal case is when the adversary finds all classes have equal probabilities based on his/her observation. Moreover, if the number of classes is N_c, the output of $P2$ will falls into the range between 0 and $\log(Nc)$. This indicates that if the number of classes is bigger, the user potentially has the opportunity to better protect his privacy. From this viewpoint, the ICE metric is better than ICP for measuring the image privacy.

4 | IMAGE PRIVACY PRESERVATION SCHEME AGAINST DNNS

As stated in Section 3.1, we aim to mislead the DNNs so that the privacy in the images can be preserved. During the research of DNNs, the so-called ***adversarial samples*** have been found to cause a state-of-art DNN to misclassify any input image to another class. Moreover, these adversarial examples can be generated simply by adding a small amount of "well designed" ***adversarial noise*** to the original image. The changes are imperceptible to human eyes but fool the DNNs.

Although the adversarial examples are treated as harmful from the perspective of DNN performance, it can be used to achieve our privacy protection targets. The basic idea is to generate an adversarial example as the released image. There are several methods to generate the noise for the adversarial example, among which the most widely used one is the fast gradient sign method (FGSM). In this section, we will first briefly introduce the FGSM and then present two different privacy protection schemes for the probability minimization problem and entropy maximization problem, respectively.

4.1 | Fast gradient sign method (FGSM)

Let θ be the parameters of a model, $\boldsymbol{x}$ the input to the model, y the targets associated with $\boldsymbol{x}$ (for machine learning tasks that have targets) and $J(\theta; \boldsymbol{x}; y)$ be the cost function (output) used to train the neural network.[9] The cost function can be linearized around the current value of θ, obtaining an optimal max-norm constrained perturbation of

$$\boldsymbol{\eta} = \epsilon \text{sign}(\nabla_{\boldsymbol{x}} J(\theta; \boldsymbol{x}; y)), \tag{5}$$

where $\nabla_{\boldsymbol{x}}$ is the gradient of the cost function J with regard to the input image *boldsymbol x* and ϵ is a small scalar which keeps the noise imperceptible to human eyes.

Moreover, the release image is generated by

$$\boldsymbol{x}' = \boldsymbol{\eta} + \boldsymbol{x}. \tag{6}$$

The above method is referred to as the FGSM of generating adversarial examples. The most important step in FGSM is the calculation of the gradient. Normally, the gradient can be obtained by

$$\nabla_{\boldsymbol{x}} J(\theta; \boldsymbol{x}; y) = \frac{\partial J}{\partial \boldsymbol{x}}. \tag{7}$$

However, as there are many layers between the input and output of the DNN, it is difficult to directly derive the relationship between J and $\boldsymbol{x}$. Therefore, in a DNN, backpropagation is introduced and the chain rule is used to compute the gradients. Moreover, in practice, current popular DNN frameworks such as TensorFlow has built-in functions can automatically calculate the gradients, which greatly helps the implementation of FGSM.

Finally, it is important to note that this method can reliably cause a wide variety of models to misclassify their input.[9] Because when the direction $\boldsymbol{\eta}$ has positive dot product with the gradient of the cost function, and ϵ is large enough, the adversarial examples occur in broad subspaces. Thus they are abundant and the example misclassified by one classifier has a fairly high prior probability of being misclassified by another classifier.[9]

4.2 | ICP Metric–based privacy protection scheme

It is hard to minimize the probability of the true class, as presented in Equation (3). However, we can solve this problem from an opposite angle, ie, maximize the probability of a wrong class. We call this the class replacement scheme (CRS), as shown in Algorithm 1.

Algorithm 1 Class replacement scheme

Parameters: The number of classes in predication k.
Noise step size η_{step}.
Noise limit η_{max}.
Required score for the target class $score_{required}$.
Max iteration number. max_{iter}.
Input: The original image $\boldsymbol{x}$.
Output: The released privacy preserving image $\boldsymbol{x'}$.
Initialization: noise $\boldsymbol{\eta} = 0, \boldsymbol{x}_0 = \boldsymbol{x}$
$score_{top\_k} = prediction(class_{top\_k})$; (Calculate the scores for the top-k predictions using the DNN model).
$y = randint(class_{all-k})$; (Randomly select a target class y which does not belong to the top-k predictions).
for $1 \leq i \leq max_{iter}$ **do**
  $\boldsymbol{x}_i = \boldsymbol{\eta}_{i-1} + \boldsymbol{x}_{i-1}$;
  $score_{target} = prediction(y)$;
  if $score_{target} < score_{required}$ **then**
    $\boldsymbol{\eta}_i = \boldsymbol{\eta}_{i-1} + \eta_{step} * \text{sign}(\nabla_{\boldsymbol{x}_i} J(\theta; \boldsymbol{x}; y))$. (clip the element in $\boldsymbol{\eta}_i$ if it exceeds η_{max})
  end
  else
    $\boldsymbol{x'} = \boldsymbol{x}_i$;
    break.
  end
end
if $score_{target} < score_{required}$ **then**
  Return to step 10 and repeat the process. (If not successful, select another target class and repeat the process.)
end

The CRS essentially performs optimization with gradient descent. We first randomly pick up a target class y, and then the noises is iteratively increased using the gradient of the loss function with regard to the image generated in the current iteration. In each step, the noise takes the classification result further from the correct one and closer to the selected class, until it reaches the target score (eg, 99%). The target class y is randomly picked rather than using an arbitrary value.

The current digital images often use 8 bits per pixel so the maximum value will be 255. Then, the noise limit η_{max} is generally set as a percentage (ie, 1%) of 255. Moreover, if the image is a color image with 3 channels (RGB), the process will be performed on all channels in the same way, because an attacked in practice will know the image has been protected if he/she find all sub-channel images belong to the same class.

4.3 | ICE Metric–based privacy protection scheme

Now we design another scheme to solve the problem presented by Equation (4). The entropy will be maximized when the probabilities of all classes are equal. In the case that the number of classes is small, we can try all possible combinations to find the optimal solution. However, in practice, there might be applications with thousands of classes. In this case, we switch our target to make the probabilities of n classes closed to each other, then the true information becomes "indistinguishable" among these n classes. We name this the class indistinguishability scheme (CIS).

As shown in Algorithm 2, if we have a big number of classes, we will first generate a target class list. n candidate classes are randomly picked up. We exclude the $top - k$ classes given by the DNN model on the original image, because we found that $top - k$ classes are sometimes quite close to the true class and is not a good candidate as an "adversary." Then for each candidate class, we apply the CIS and store the noise. Finally, we add all noises to the original image to obtain the released image.

The essential process of CIS is quite similar to that of CRS, except that we add noises for several targeted classes instead of a single one. This will make the sensitive information invisible from the DNN attacks.

5 | PERFORMANCE EVALUATION AND DISCUSSIONS

5.1 | Datasets and evaluation setup

We evaluate our privacy protection schemes using two different databases, with two different neural network models.

Algorithm 2 Class indistinguishable scheme

Parameters: The number of target indistinguishability classes in predication n.
Noise step size η_{step}.
Noise limit η_{max}.
Required score for the target class $score_{required}$.
Max iteration number max_{iter}.
Input: The original image $\boldsymbol{x}$.
Output: The released privacy preserving image $\boldsymbol{x'}$.
$score_{top\_k} = prediction(class_{top\_k})$; (Calculate the scores for the top-k predictions using the DNN model).
$class\_list_{target} = randint(class_{all}, n)$; (Randomly select a target class list which contains n components).
for $y \in class\_list_{target}$ **do**
  $\boldsymbol{\eta} = 0, \boldsymbol{x}_0 = \boldsymbol{x}$
  for $1 \leq i \leq max_{iter}$ **do**
    $\boldsymbol{x}_i = \boldsymbol{\eta}_{i-1} + \boldsymbol{x}_{i-1}$;
    $score_{target} = prediction(class_{target})$;
    if $score_{target} < score_{required}$ **then**
      $\boldsymbol{\eta}_i = \boldsymbol{\eta}_{i-1} + \eta_{step} * \text{sign}(\nabla_{\boldsymbol{x}_i} J(\theta; \boldsymbol{x}; y))$. (clip the element in $\boldsymbol{\eta}_i$ if it exceeds η_{max})
    end
    else
      $\boldsymbol{\eta}_y = \boldsymbol{\eta}_i$;
      break.
    end
  end
  if $score_{target} < score_{required}$ **then**
    Return to step 9 and repeat the process. (If not successful, select another target class and repeat the process.)
  end
end
$\boldsymbol{x'} = \boldsymbol{x}$;
for $y \in class\_list_{target}$ **do**
  $\boldsymbol{x'} = \boldsymbol{x'} + \boldsymbol{\eta}_y$;
end

The first one is Modified National Institute of Standards and Technology database (MNIST) database.[27] It is a large commonly used handwritten digits database. We divide the dataset into three parts: training-set (55000 images), test-set (10000 images) and validation-set (5000 images). We build a simple neural network model which consists of 2 convolutional layers, 1 fully-connected layer, and 1 softmax classifier to finish the digits recognition job and achieve around 96% accuracy rate. Then we apply our privacy protection schemes on top of this neural network.

The second dataset is the ILSVRC-2012 database which includes 48238 images.[28] Moreover, GoogleLeNet InceptionV3 DNN model[14] is used to finish the image classification task. The Inception v3 model takes weeks to train on a monster computer with 8 Tesla K40 GPUs so it is impossible to train it on our own. We instead download the pre-trained Inception model, and we implement our own algorithms on top of the model.

All the codes in the experiment are implemented in Python, using the Tensorflow framework.[29] Moreover, we run our experiment on a Dell Laptop with Intel Core i5-7300HQ Quad Core and 32 GB DDR4 RAM.

5.2 | Performance of MNIST Dataset

First, we consider the handwritten digits recognition task. In practice, it is performed as a classification task, ie, find the class of the given image among all possible classes $0 - 9$. From the privacy protection viewpoint, we need to add noise to the image so that the neural network will be mislead to spit out a false class.

5.2.1 | Performance of class replacement scheme

First, we implement CRS on this dataset, ie, finding a noise pattern for an arbitrary class, then most of the figures added this noise will be misled to this target noises. As there are only 10 classes in total, we have found the noise-patterns for all target-classes, as shown in Figure 2. The average noise added on each pixel is 0.013.

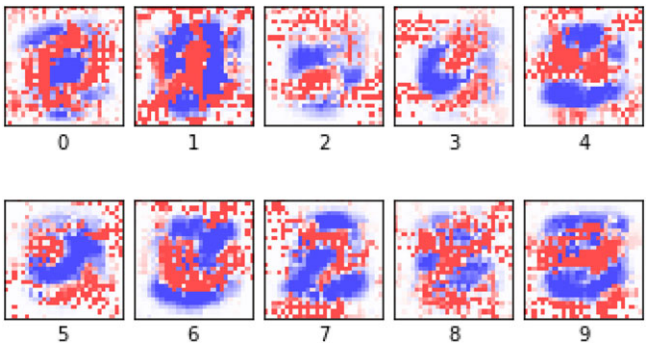

FIGURE 2 Illustration of noises for each target class

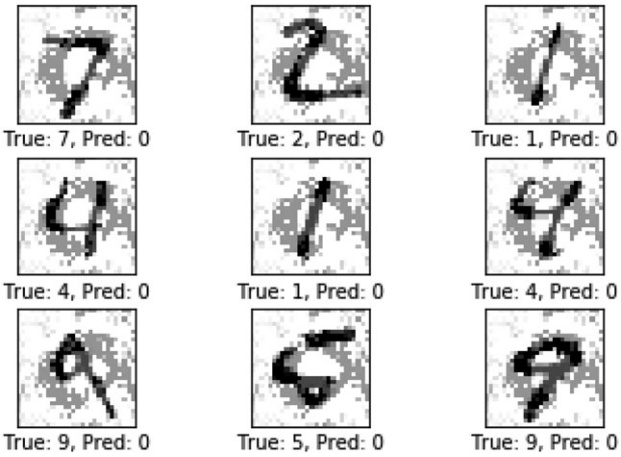

FIGURE 3 Examples of errors when the target class is 0

TABLE 2 Accuracy of handwritten digits recognition with adversarial noise applied on single target class

Target Class	0	1	2	3	4	5	6	7	8	9
Accuracy	17.8%	56.4 %	12.3%	11.6%	21.6%	13.0%	28.7 %	23.9%	12.9%	22.2%

The images in MNIST are gray scale with size 28 × 28 pixels. The impact of the noise is to increase or decrease the value which represents the pixels' intensities (ie, shades of gray). For illustration purposes, we use red pixels for positive noise values, and blue pixels for negative noise values in Figure 2.

As can be seen, the noise-patterns for the MNIST data-set were clearly visible to the human eye. In some of these noisy images we can even see traces of the numbers. For example, the noise for target-class 0 shows a red circle surrounded by blue. It means that a little noise will be added to the input image in the shape of a circle, and it will dampen the other pixels. This is sufficient for most input images in the MNIST data-set to be misclassified as a 0 (the accuracy of handwritten digits recognition comes down to 17.8%). Figure 3 gives some examples of errors when the target class is 0. It shows that the digits are still easily identified by a human with the existence of adversarial noise, but the neural network will misclassify the images with high probabilities.

Table 2 lists the accuracy of handwritten digits recognition with adversarial noise applied to each target class. It shows that the impacts of applying adversarial noises for different target classes are quite different. Noises of some classes have obvious visual implications (eg, 2,3, and 8), while others have much less visual implications (eg, 1). A possible explanation is that the "distance" between "1" and any other digit is far, so it is difficult to mislead the DNN to recognize "1" as anything else. However, how to accurately measure the "distance" is an interesting open question.

5.2.2 | Performance of class indistinguishability scheme

In the case of CRS, we mislead the DNN to a single target class. We now test the CIS scheme on the dataset. First, we try to sum noises for all target classes, as shown in Figure 4. It is now difficult to see any obvious shape of the red part. Moreover, the accuracy of handwritten digits recognition now comes back to 82.1%. Moreover, when we check the error examples as shown in Figure 5, we can see that the DNN is more likely to mislead to the classes "4" or "8." It means that the adversarial for each target class is not independent. When added together, they are blended together to some extent and then become close to the adversarial noises of "4" and "8."

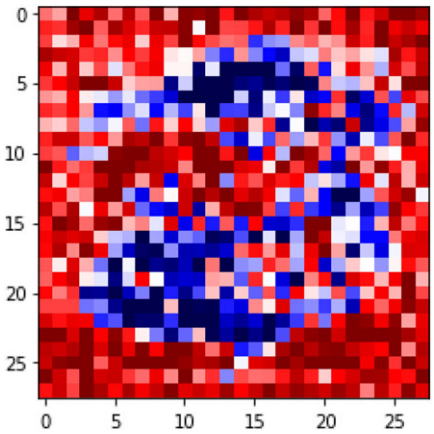

FIGURE 4 Illustration of sum of the noises for all target classes

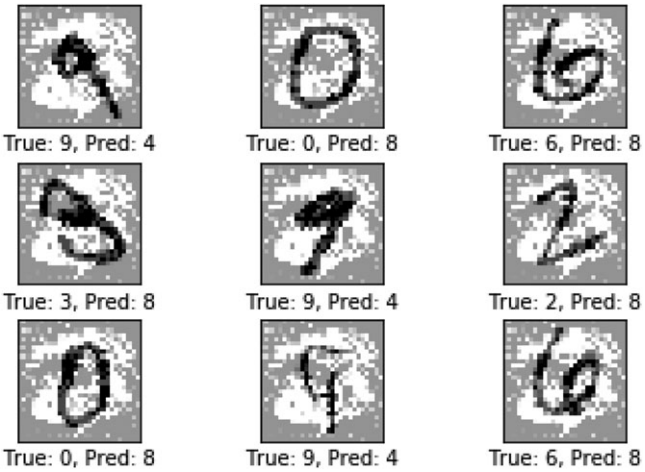

FIGURE 5 Examples of errors when using the sum noises

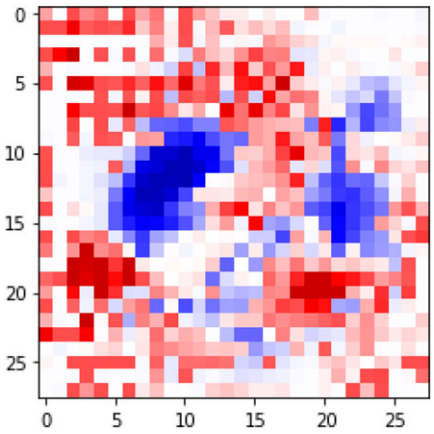

FIGURE 6 Illustration of sum of the noises from target classes "2" and "3"

Since combining noises for all classes does not work well, we try adding noises from any two target classes. Figure 6 and Figure 7 show the sum of the noises from target classes "2" and "3" and examples of errors when using such a noise. It can be seen from the figures that the DNN, in this case, will be led to the classes "2" and "3." However, the accuracy of handwritten digits recognition is higher (21.7%) than the cases of using adversarial noises of any single target class (12.3% for "2" and 11.6% for "3"). If we use noises from three target classes, for example, "2," "3," and "8," the accuracy of handwritten digits recognition is (32.7%). Actually, our experiments show that all combinations of noises from multiple target classes have the same property. This phenomenon may also be caused by the similarity of these two digits.

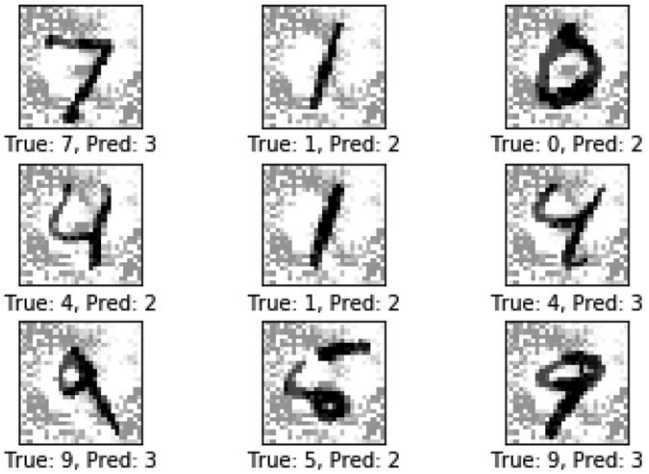

FIGURE 7 Examples of errors when using the sum noises from target classes "2" and "3"

5.3 | Performance of ILSVRC-2012 dataset

We now investigate the performance of our proposed schemes on the ILSVRC-2012 dataset. This dataset is far more complicated than the MNIST. Luckily, as we use the pre-trained DNN models in our scheme, the complexity of CRS and CIS mainly come from the computation of gradients, which is linear in the number of parameters, the size of the input, and the size of each hidden layers in the DNN. The iteration numbers will be restricted to below 100. Therefore, the complexity of our proposed schemes are quite low and an adversarial image can be generated in several seconds in a personal laptop.

There are 1000 image classes in the ILSVRC-2012 dataset. Moreover, the adversarial noise in ILSVRC-2012 was found through an optimization process for each individual image. Because the noise was specialized for each image, it may not generalize and have any effect on other images.

5.3.1 | Performance of class replacement scheme

Figure 8 gives an example of the performance of CRS. The Inception V3 model has high confidence (96.23%) to classify the original image as "black swan." Moreover, when we add a small noise using FGSM, it will be misclassified as a "cowboy boot" with even higher confidence (99.01%). Figure 9 shows a similar result for the image of a minibus.

Table 3 shows the result of running our proposed CRS on the large-scale image dataset. The Inception V3 model can correctly classify the original images with an average probability of 76.7%, while it gives less than 0.1% probability of the released images being the correct class. Moreover, it will be misled to another incorrect class with high confidence (99.45%). The average noise added on each pixel is only 0.0015. The computation complexity is also low. In more detail, the computation time per image is about 5s with an average iteration number of 17.205.

5.3.2 | Performance of class indistinguishability scheme

Figure 10 and Figure 11 give two examples of the performance of CIS. Instead of adding noise for one target class, we use multiple target classes to generate the noise. In this case, the DNN model cannot distinguish the image from different classes because the probabilities are close. For

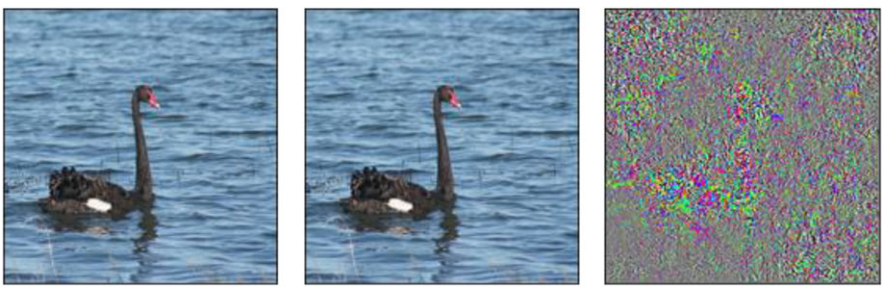

FIGURE 8 Illustration of CRS on the black swan image (the colors of noises are amplified by normalization otherwise they would be hard to see)

FIGURE 9 Illustration of CRS on the minibus image (the colors of noises are amplified by normalization otherwise they would be hard to see)

TABLE 3 Performance of CRS on large-scale image dataset

Term		Value
$\Pr(class_p = class_x)$	Original image	76.70%
	Released image	0.09%
$\Pr(class_p = y)$		99.45%
Noise		0.0015
Iteration number		17.205
Computation time per image (s)		0.585

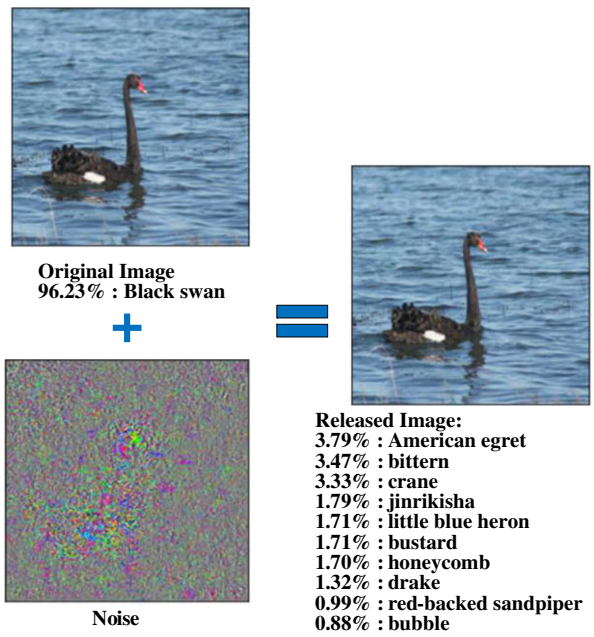

FIGURE 10 Illustration of CIS on the black swan image

example, as shown in Figure 10 the deep learning classifier thinks the released image might be an American egret (3.79%) or a bittern (3.47%), or even a jinrikisha (1.79%). Figure 11 illustrates another case with the original image being a minibus.

We also evaluate the performance of CIS on ILSVRC-2012 validation dataset. As shown in Figure 12, the entropy of the original image is about 0.7 (the case when the number of target class equals 0). If we mislead the DNN to one particular class (the case when the number of target class equals 1), the entropy reduces to a small value as the DNN has a guess on the target class with high confidence. When we introduce multiple target classes, the entropy will increase accordingly. Only two target classes can provide an obvious increase of the entropy. Adding more classes will not bring a significant increase. Additionally, the amount of noise in CIS is proportional to the number of target classes according to an approximately linear law, which is quite small in our simulation. Overall, we find that using two target classes are good enough to achieve satisfactory performance.

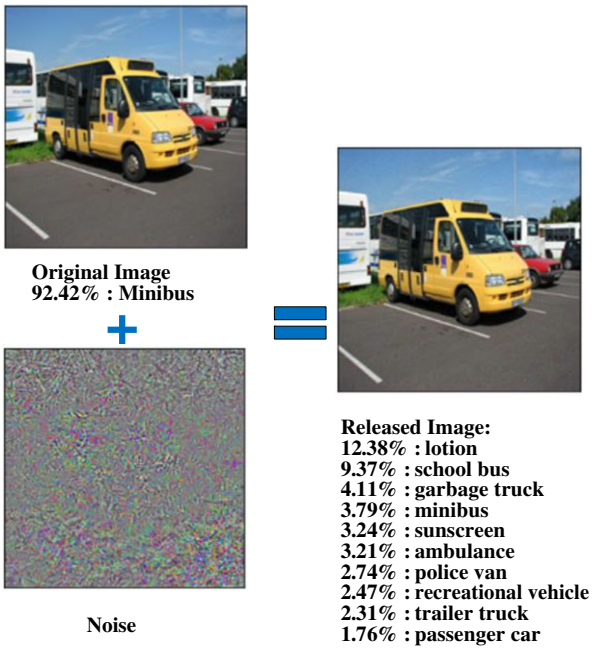

FIGURE 11 Illustration of CIS on the minibus image

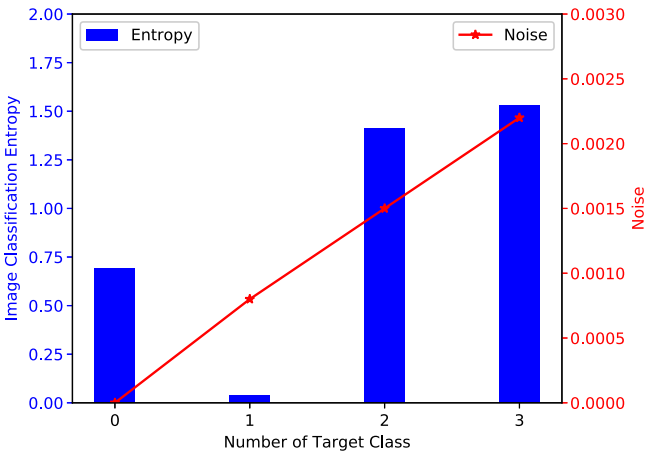

FIGURE 12 Performance of CIS on ILSVRC-2012 validation dataset

5.4 | Discussions

From the above evaluations, we can see that our privacy protection schemes exhibit various technical features, which can be summarized as follows:

- While the noise generated for the ILSVRC-2012 dataset is almost invisible to the human eye, the noise-patterns for the MNIST dataset are much clearer. The reason could be the clear structural characteristics of the handwriting digits.
- The CIS scheme does not work as well on the MNIST dataset as on the ILSVRC-2012 dataset.

These differences are caused by the features of the two datasets and the tasks performed on them:

- Images in MNIST are very simple: both the useful information and background information. While images in ILSVRC-2012 are complex and diverse.
- There are only 10 classes in the MNIST classification task, while there are 1000 classes in the ILSVRC-2012 classification task.

Due to the two above features, it seems to us that the "distances" of different classes in MNIST are farther away than those in ILSVRC-2012, and the adversarial noises in MNIST are more dependent. This leads to the fact that it is more difficult to protect sensitive information in simple

images. Hence, more sophisticated privacy preservation techniques other than adding noises might be needed to treat simple images in the context of deep learning.

6 | CONCLUSION

The rapid development of deep learning has brought forth both significant challenges and opportunities for privacy protection, especially for sensitive information in images shared on social network platforms. To solve this problem, we propose a framework to protect that uses the adversarial example idea to protect image privacy. The contributions of this paper are two-fold. First, we define two new image privacy metrics. Second, we design two privacy protection schemes to optimize the two proposed metrics. Our results show that we can achieve good privacy protection against deep learning tools at the cost of adding a small amount of noise that is imperceptible to human eyes. The proposed method is particularly effective for images of complex structures and textures. We believe that privacy protection in deep learning era is an important topic and this work sheds some new insight on the research in this area.

ORCID

Bo Liu https://orcid.org/0000-0002-3603-6617

REFERENCES

1. Facebook privacy breach. https://www.ft.com/content/87184c40-2cfe-11e8-9b4b-bc4b9f08f381. Accessed April 13, 2018.
2. Weyand T, Kostrikov I, Philbin J. PlaNet-photo geolocation with convolutional neural networks. In: *Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*. Cham, Switzerland: Springer International Publishing; 2016;37-55.
3. Zhu T, Li G, Zhou W, Yu PS. *Differential Privacy and Applications*. Cham, Switzerland: Springer International Publishing; 2017.
4. 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; 2015; Denver, CO.
5. Abadi M, Chu A, Goodfellow I, et al. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security; 2016; Vienna, Austria.
6. Phan N, Wang Y, Wu X, Dou D. Differential privacy preservation for deep auto-encoders: an application of human behavior prediction. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence; 2016; Vienna, Austria.
7. Yu J, Zhang B, Kuang Z, Lin D, Fan J. iPrivacy: image privacy protection by identifying sensitive objects via deep multi-task learning. *IEEE Trans Inf Forensics Secur*. 2017;12(5):1005-1016.
8. Liu Y, Zhang W, Yu N. Protecting privacy in shared photos via adversarial examples based stealth. *Secur Commun Netw*. 2017;2017.
9. Goodfellow I, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. 2014. arXiv preprint arXiv 1412.6572.
10. Convolutional Neural Networks for Visual Recognition. http://sungsoo.github.io/2016/05/29/convolutional-neural-networks.html. Accessed August 30, 2018.
11. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE*. 1998;86(11):2278-2324.
12. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012*. Red Hook, NY: Curran Associates; 2012:1097-1105.
13. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. Paper presented at: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015; Boston, MA.
14. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. Paper presented at: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016; Las Vegas, NV.
15. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA. Inception-v4, inception-ResNet and the impact of residual connections on learning. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence Volume 4 (AAAI); 2017; San Francisco, CA.
16. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. https://arxiv.org/abs/1409.1556
17. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016; Las Vegas, NV.
18. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems; 2015; Montreal, Canada.
19. Cimpoi M, Maji S, Vedaldi A. Deep filter banks for texture recognition and segmentation. Paper presented at: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015; Boston, MA.
20. Wang L, Ouyang W, Wang X, Lu H. Visual tracking with fully convolutional networks. Paper presented at: 2015 IEEE International Conference on Computer Vision (ICCV); 2015; Santiago, Chile.
21. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. Paper presented at: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015; Boston, MA.
22. Liu B, Zhou W, Zhu T, Gao L, Xiang Y. Location privacy and its applications: a systematic study. *IEEE Access*. 2018;6:17606-17624.
23. Papernot N, Goodfellow I. Privacy and machine learning: two unexpected allies? cleverhans-blog; 2018.
24. Shokri R, Shmatikov V. Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security; 2015; Denver, CO.

25. Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks. 2013. arXiv preprint arXiv:1312.6199.
26. Wagner I, Eckhoff D. Technical privacy metrics: a systematic survey. 2015. arXiv preprint arXiv:1512.00327.
27. LeCun Y, Jackel L, Bottou L, et al. Comparison of learning algorithms for handwritten digit recognition. Paper presented at : International Conference on Artificial Neural Network; 1995; Paris, France.
28. Russakovsky O, Deng J, Su H, et al. Imagenet large scale visual recognition challenge. *Int J Comput Vis*. 2015;115(3):211-252.
29. Abadi M, Agarwal A, Barham P, et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. White Paper. TensorFlow; 2015. tensorflow.org

How to cite this article: Liu B, Ding M, Zhu T, Xiang Y, Zhou W. Adversaries or allies? Privacy and deep learning in big data era. *Concurrency Computat Pract Exper*. 2019;31:e5102. https://doi.org/10.1002/cpe.5102