machines

MDPI

Article

A Fast Method for Protecting Users' Privacy in Image Hash Retrieval System

Liang Huang [1], Yu Zhan [1], Chao Hu [1,2,*] and Ronghua Shi [1]

[1] School of Computer Science and Engineering, Central South University, Changsha 410083, China; hliang@csu.edu.cn (L.H.); zhanyu@csu.edu.cn (Y.Z.); shirh@csu.edu.cn (R.S.)
[2] Big Date Institute, Central South University, Changsha 410083, China
* Correspondence: huchao@csu.edu.cn

Abstract: Effective search engines based on deep neural networks (DNNs) can be used to search for many images, as is the case with the Google Images search engine. However, the illegal use of search engines can lead to serious compromises of privacy. Affected by various factors such as economic interests and service providers, hackers and other malicious parties can steal and tamper with the image data uploaded by users, causing privacy leakage issues in image hash retrieval. Previous work has exploited the adversarial attack to protect the user's privacy with an approximation strategy in the white-box setting, although this method leads to slow convergence. In this study, we utilized the penalty norm, which sets a strict constraint to quantify the feature of a query image into binary code via the non-convex optimization process. Moreover, we exploited the forward–backward strategy to solve the vanishing gradient caused by the quantization function. We evaluated our method on two widely used datasets and show an attractive performance with high convergence speed. Moreover, compared with other image privacy protection methods, our method shows the best performance in terms of privacy protection and image quality.

Keywords: adversarial attack; image hash; privacy protection; retrieval; penalty norm

Citation: Huang, L.; Zhan, Y.; Hu, C.; Shi, R. A Fast Method for Protecting Users' Privacy in Image Hash Retrieval System. *Machines* **2022**, *10*, 278. https://doi.org/10.3390/machines10040278

Academic Editors: Richard Hill and Zhuming Bi

Received: 16 January 2022
Accepted: 12 April 2022
Published: 14 April 2022

1. Introduction

Various types of user information, such as images, texts and videos, are shared on the internet, and have rapidly increased in popularity in recent years. Snapchat users upload 527,760 photos every minute, and more than 300 million photos are uploaded per day [1]. However, such a large amount of user information is at the risk of leakage. A search engine could store the users' uploaded image data for 7 days to improve their product, e.g., Google and Yahoo [2]. Moreover, Facebook have released up to 6.8 million private photos, demonstrating a considerable privacy risk [3]. These behaviors seriously violate the privacy of users, so there is an urgent need to strengthen research into image retrieval privacy protection.

At present, most image search engines on the market are mainly composed of DNN-based image hash retrieval systems [4–9] due to their high storage capacity and excellent retrieval performance. The DNN-based image hash retrieval system comprises convolution neural networks (CNNs) and multiple fully connected layers (Mul-FC), which convert high-dimensional image data into hash codes for large-scale database retrieval. Specifically, the CNN module extracts the spatial feature from the image and the Mul-FC module turns the continuous feature value into a discrete binary hash code.

Tradition work of protecting the user's privacy is to generate the masking [10] or scrambling [11], while leading to lower image visualization quality, potentially affecting the user's experience. Recent work [12–16] has proposed protecting when using the image hash retrieval system by the adversarial attack. Though adversarial attack, users can upload carefully modified query images, which are called adversarial images, that are entirely

different from private images to the image retrieval system to obtain the same query results as private images. The user's private images will not be returned when the malicious party sends a request to the service provider. Figure 1 shows the process of the potential threat of image privacy leakage and the work flow of protecting image privacy through adversarial attacks. In addition, adversarial attacks also have practical applications in the industry, and examples of privacy-preserving applications of images in the industry can be found in Appendix A. The adversarial attack was first introduced in [17], proving that the DNN is vulnerable to malicious and well-designed adversarial perturbations because of its approximate linear property and depth. Much work on adversarial attack has been carried out based on the image classification task [18–22] to improve the DNN robustness [22,23] and provide visualization interpretability [24,25]. Note that the adversary's goal of the image classification system is to cause error output of category confidence. Unlike the image hash retrieval system, the adversary aims to craft the target hash code by adversarial image and finally obtain the specific retrieval result. In other words, this attack process is a `mismatch` between the content of the query and the retrieval result. Taking advantage of this mismatch, adversarial attacks can play a role in protecting user privacy. The existing works concerning the attack of image hash retrieval systems under white-box settings can be divided into two sections. One section is the `non-target` attack [12–15] which is to tamper with the query image to any retrieval result differing from the original, while `target` attack [16,26,27] aim to create a specific retrieval result.

In this paper, we focus on the targeted adversarial attack of DNN-based hash retrieval systems, which was firstly proposed in [16]. Given a query image, the malicious adversary's goal is to synthesize a visually indistinguishable adversarial example to generate an adversarial hash code different from the original. This process is formulated to maximize the Hamming distance between the adversarial hash code and the original hash code. The Hamming distance function can be replaced by the inner product. The transformation of continuous values generated by DNN-based hash retrieval systems into binary discrete values is finished by the sign(·) function. Gradient-based optimization methods are often plagued by vanishing gradients, making the optimization difficult. The approximation strategy is widely used to solve the problem of the gradient vanishing in the optimization process of adversarial attacks. Specifically, the image hash retrieval system converts the continuous feature value into binary discrete hash code via `tanh(·)` function [5]. Moreover, a well-designed hyperparameter is used to control the distribution of tanh(·) and adjust it with an iterative optimization process. For example, the hyperparameter is set to 0.1 for the first 1000 iterations and gradually increase to 1 for the finally 1000 iterations. However, we note that non-adaptive hyperparameter tuning may have an impact on slow convergence. Further more, the approximate strategy based on tanh(·) cannot fundamentally solve the problem of gradient disappearance. In this paper, we are inspired by a recent work [28] that abandons the approximation strategy and hopes to train deep hash networks by constructing a new loss function with gradient derivation strategy.

Specifically, we follow the work of [16] to define the optimization objective of image privacy preservation as minimizing the Hamming distance between the hash code of the adversarial example and the target hash code. We identify the problem in the optimization process which is that the sign(·) function will hinder the calculation of the gradient and make the gradient-based optimization method fall into the difficulty of gradient vanishing. The chain rule is used to separate the gradient-hard formulation and transform it into a convex optimization problem with binary constraints. We obtain an analytical solution to the above convex optimization problem through the proximal operator and transform the problem to construct a new loss function in order to solve the gradient computation difficulty from sign(·). Finally, we design a penalty term-based loss function and exploit a gradient transfer path for obtaining adversarial gradients.

Our main contributions can be summarized as follows:

1. We reformulate the image hash retrieval system's targeted attack with the relaxation penalty norm to obtain better performance and convergence speed.

2. We introduce a forward–backward strategy to solve the gradient vanishing problem with the relaxation penalty norm for the adversarial attack.
3. We propose protecting the privacy of image semantic information and content with an adversarial attack method in an image hash retrieval system.
4. We exploit the PSNR metric, and gradient-based heatmap [29] to compare the pros and cons of traditional privacy protection methods.
5. We conducted experiments on the FLICKER-25K and NUS-WIDE datasets and verified that our method outperforms other adversarial attack methods.

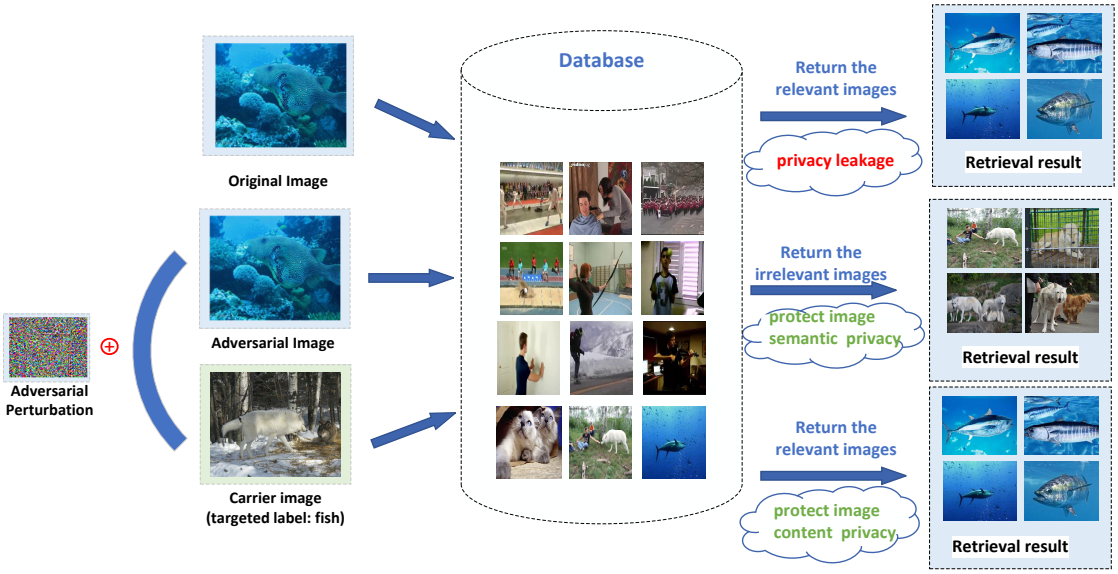


Figure 1. Three kinds of queries. The first row is the original query, which may encounter the threat of privacy leakage during the retrieval process. In the second row, we conduct adversarial attacks on the images uploaded by users to social platforms. This can ensure that the semantic information of images will not be retrieved by the retrieval system, resulting in the leakage of image semantic privacy. In the third row, we conduct a targeted adversarial attack on a carrier image. The target label is the same as the original image. This retrieval can be carried out by uploading the carrier image without uploading the original image, which can protect the image content privacy.

2. Related

2.1. Traditional Method to Image Privacy Protection

The traditional method of image privacy protection is to add noise [10,11] such as blur, distortion, or mosaic to sensitive areas of the image. They are the simplest and most commonly used privacy protection methods but lead to many problems. These methods are irreversible, causing permanent damage to the image. For images uploaded to social networks, images with noise lose the capability of being shared on social platforms, leading to the decreased usability of images. For DNN-based image hash retrieval systems, these methods are invalid. Some works in the literature show that 96% of image data can be correctly recognized under the recognition of the blurred image by the convolutional neural network.

2.2. Dnn-Based Image Hash Retrieval System

Various image hash retrieval systems have obtained strong performance based on deep neural networks (DNN). It can be divided into supervised and unsupervised approaches by whether the data label information is given or not. We explore the supervised-based image hash retrieval system in this paper. The challenge of the DNN-based image hash retrieval system is to design an effective strategy that projects continuous values to discrete binary values. Ref. [4] exploits the regularization norm to encode discrete values of images, which can maximize the distinguishability of binary output space. Ref. [5] designed a novel

architecture to solve the ill-posed gradient problem in the training process via the adaptive activation function. Ref. [30] designed the paired cross-entropy loss based on the Cauchy distribution to significantly penalize different image pairs. One work [28] similar to this paper utilizes the greedy strategy with a novel encode layer to avoid gradient vanishing and quantization loss.

2.3. Privacy Protection of DNN-Based Image Hash Retrieval System

Unlike the adversarial attack in the image classification, which outputs the false result by the faulty unit, the adversarial attack in the image hash retrieval system causes the wrong result by outputting the adversarial hash code. The image hash retrieval system under the white box attack is as follows. Refs. [13,31] disrupts the matching relationship between the high-dimensional query image descriptor for universal attack [21] and the hash code space. Ref. [14] confused the retrieval rank list of query images by the reformulated normalized discounted cumulative gain (NDCG) to achieve a non-target attack. Refs. [27,32] embed label and hash code information into the objective semantic space to achieve flexible target attack and defense based on a generative adversarial network (GAN). The most relevant previous work [12,16] explored the adversarial attack by optimizing the hamming distance between the adversarial and the target hash code while the applying the `tanh` function and its hyperparameter leading to inefficiency.

3. Background

3.1. Image Hash Retrieval System

Suppose a query image $I_q \in \mathcal{R}^{W\times H\times C}$ where W, H, and C are the image width, height, and channel, respectively. A pre-trained convolution neural network (CNN), such as Alexnet [33], can extract the spatial features from the query image I_q. The hash layers consist of multiple fully connected layers (Mul-FC), which can quantify the feature values into binary hash code h_q, i.e., $h_q \in \{-1,+1\}^K$, where K is the length of hash code. We denote $\mathtt{C}(\cdot)$ as the sequential combination of pre-trained CNN and Mul-FC function. Therefore, the image hash retrieval model $\mathtt{H}(\cdot)$ can be formulated as $\mathtt{H}(I_q)$=$sign(\mathtt{C}(I_q))$, where sign(·) indicates that if the output bit of $\mathtt{C}(I_q)$ is larger than 0, then output 1, otherwise -1. The major challenge of obtaining an efficient binary hash code is to design an effective process method for preserving the discriminate feature information from images. Recently, the approximation method proposed by [5,34] presented high performance in converting the continuous feature values into discrete hash code, where tanh(·) is exploited to replace sign(·) for solving the gradient vanishing problem, because the discrete binary output cannot be directly optimized in the model training process. Hence, the image hash retrieval model can be reformulated as $\mathtt{H}(I_q) = \tanh(\alpha\mathtt{C}(I_q))$. The hyperparameter α gradually increases to positive infinity until it approaches the sign function.

Here, we give a detailed description of the image hash retrieval process. The image hash retrieval model $\mathtt{H}(\cdot)$ turns query images I_q into hash code h_q, and then the output hash code is compared with all the hash code h_{q*} of the corresponding image I_{q*} in the database. All the images I_{q*} of hash codes h_{q*} that satisfy the condition $\mathtt{D}(h_q, h_{q*}) \leq T_t$ will be returned, where T_t is the threshold. $\mathtt{D}(h_q, h_{q*})$ is the hamming distance function to measure the similarity between two hash codes. For example, if the calculation result of $\mathtt{D}(h_q, h_{q*})$ is lower than the threshold, it shows that the corresponding images I_q and I_{q*} are relevant in content; otherwise, they are not. The Hamming distance is generally represented by the inner product operator [34], which can be reformulated as $\mathtt{D}(h_q, h_{q*}) = \frac{1}{2}(K - \sum_z^K H^z(I_q) \cdot H^z(I_{q*}))$ in practice, where $H^z(\cdot)$ denotes the z-th hash code bit. It is worth mentioning that there still exists a concept of a weak class, which means an efficient image hash retrieval system will generate similar hash codes for images with the same label. More details can be found in [35].

3.2. Adversarial Attack

In our scenario, we assume that an adversary crafts malicious adversarial examples based on the query image I_q. We carried out our work under the white-box settings with all parameters and the structure of the retrieval system known. The adversary aims to craft malicious adversarial perturbations θ to synthesise adversarial images $I_q^{'}$, i.e., $I_q^{'} = I_q + \theta$. Hence, we can formulate the objective function below:

$$\min_{\theta} \mathcal{L}\left(\mathtt{C}(I_q^{'}), h_t\right) = -\frac{1}{K} sign(\mathtt{C}(I_q^{'})) \cdot h_t^T \quad s.t. \|\theta\|_\infty \leq \tau \,, \tag{1}$$

$\mathcal{L}$ is the inner product function to measure the similarity between hash codes, $\mathtt{C}$ is the deep hash model, and h_t is the target hash code generated from the goal image. The image hash retrieval system exploited the sign$(\cdot)$ function to quantify the feature value $\mathtt{C}(I_q)$ into binary hash codes. The constraint of $\|\cdot\|_\infty$ denotes that the maximum perturbation value of the image should be smaller than τ.

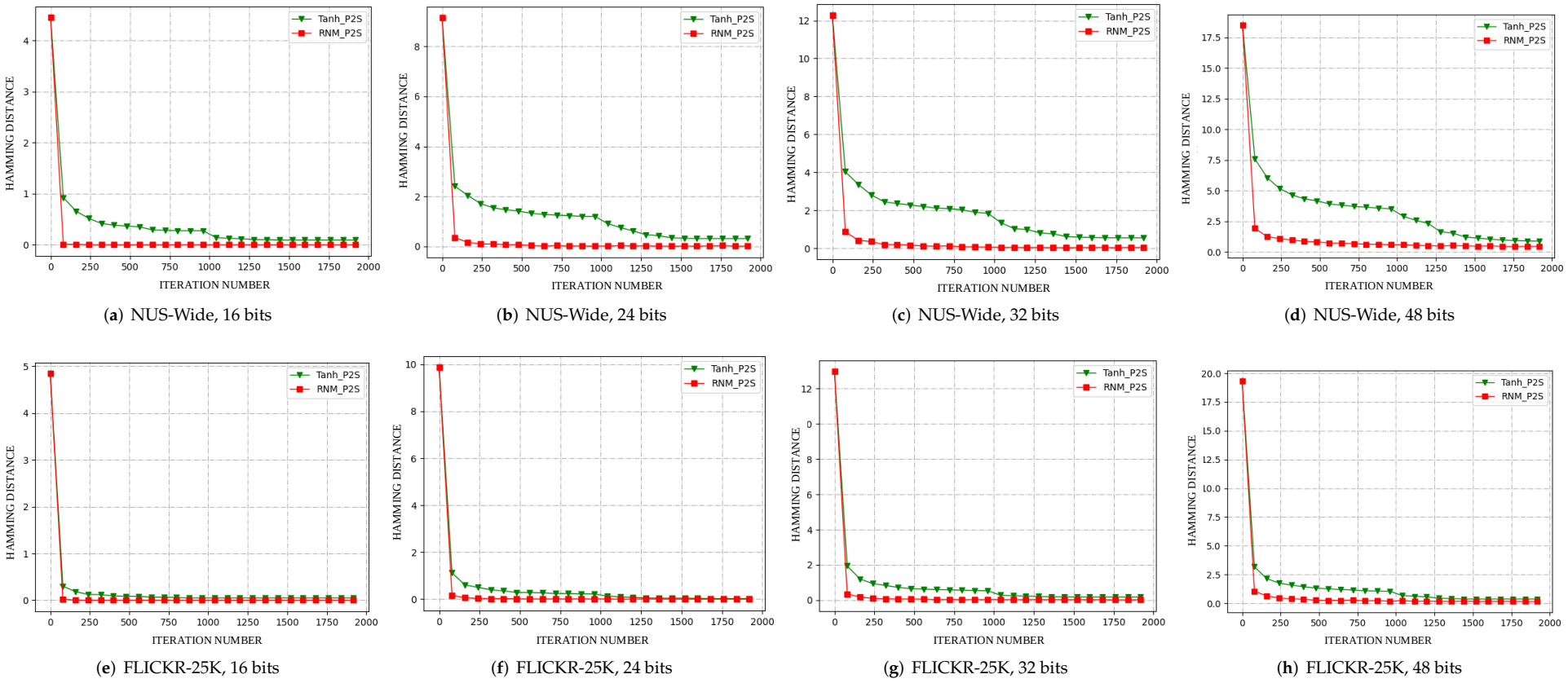
Recent work [12,16] on attacking the image hash retrieval system exploited the relaxation function tanh$(\cdot)$ to solve the gradient vanishing problem in the optimization process, and can be described as:

$$\min_{\theta} \mathcal{L}(\mathtt{C}(I_q^{'}), h_t) = -\frac{1}{K} \tanh\left(\alpha \mathtt{C}(I_q^{'})\right) \cdot h_t^T \qquad s.t. \|\theta\|_\infty \leq \tau \,, \tag{2}$$

α is the Hyperparameter. The optimization process goal is to minimize the Hamming distance by enlarging the inner product value. Various optimizers, such as ADAM [36], can effectively solve the optimization function (2) above. The optimization problem of minimizing the Hamming distance between the target and query hash code can be easily exploited with the tanh$(\cdot)$ function. We show the gradient-based solution process under the tanh$(\cdot)$ function. Specifically, let I_q be the query image; the gradient term $\frac{\partial \mathcal{L}}{\partial I_q^{'}}$ of $\mathcal{L}$ in Equation (2) can be decomposed via the chain rule by:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial I_q} &= \frac{\partial \mathcal{L}}{\partial tanh(\alpha \mathtt{C}(I_q))} \frac{\partial tanh(\alpha \mathtt{C}(I_q))}{\partial \mathtt{C}(I_q)} \frac{\partial \mathtt{C}(I_q)}{I_q} \\ &= \frac{\partial tanh(\alpha \mathtt{C}(I_q))}{\partial \mathtt{C}(I_q)} \frac{\partial \mathtt{C}(I_q)}{I_q} \cdot \alpha (1 - tanh(\alpha \mathtt{C}(I_q))^2) \end{aligned} \tag{3}$$

When $\alpha = 0.1$, the tanh$(\alpha C(I_q))$ function has a good gradient for neural network updates at $C(I_q) \in [-40, 40]$, while being out of range for gradient vanishing. The larger the value of α, the smaller the range of gradient saturation. Hence, the hyperparameter α is initialized as 0.1 in the first 1000 iteration steps in the optimization, gradually increasing within the remaining 1000 steps in the order of [0.2, 0.5, 0.7, 0.9, 1], which significantly relieves the vanishing gradient problem in the attacking process [12]. We present the convergence situation in Figure 2, and the Hamming distance value decreases step-by-step in 2000 iterations under the approximation strategy.



(**a**) NUS-Wide, 16 bits (**b**) NUS-Wide, 24 bits (**c**) NUS-Wide, 32 bits (**d**) NUS-Wide, 48 bits

(**e**) FLICKR-25K, 16 bits (**f**) FLICKR-25K, 24 bits (**g**) FLICKR-25K, 32 bits (**h**) FLICKR-25K, 48 bits

Figure 2. The value of the Hamming distance in the 2000-iteration optimization process under two datasets with 16, 24, 32, 48 bits. The green straight line is the Tanh-P2S method with anchor code, and the red dotted line is our method.

4. Approach

We describe the adversarial attack on the image hash retrieval system with the targeted attack under the white-box setting.

4.1. Explore Adversarial Gradient

Through the above process, we noticed that the attack method based on the approximate strategy is inefficient because it requires a high amount of computation to obtain the gradient of back propagation. To quickly solve the disadvantages of approximate functions, we aim to treat the process of converting continuous features into a binary hash as a non-convex optimization problem without the tanh(·) function. In general, we introduce a penalty term to make $\mathtt{C}(I_q^{'})$ approach the binary output gradually. Next, we will introduce the solution step by step. In general, if the gradient descent is performed directly on Equation (1), the following update formula is used to update the variable θ at the t-th step:

$$\theta^{t+1} = \theta^t - lr * \frac{\partial \mathcal{L}(sign(\mathtt{C}(I_q^{'})), h_t)}{\partial \theta} \tag{4}$$

where $\mathcal{L}(\cdot)$ is the inner product loss function and lr is the learning rate. However, in the case that the sign(·) function in above Equation (4) still hinders the derivation, we hope to further decompose the derivative term of the Equation (4) into the following formula according to the chain rule:

$$\theta^{t+1} = \theta^t - lr * \frac{\partial \mathcal{L}(sign(\mathtt{C}(I_q^{'})), h_t)}{\partial \mathtt{C}(I_q^{'})} * \frac{\partial \mathtt{C}(I_q^{'})}{\partial \theta} \tag{5}$$

Note that $\frac{\partial \mathtt{C}(I_q^{'})}{\partial \theta}$ can be easily obtained by back-propagation of deep neural networks. However, updating the $\frac{\partial \mathcal{L}(sign(\mathtt{C}(I_q^{'})), h_t)}{\partial \mathtt{C}(I_q^{'})}$ term will still be hindered by the existence of sign(·). Therefore, we focus on the solution of this term.

4.2. Discrete Proximal Linearized Minimization

We notice that this gradient term is the update solution of the inner product function $\mathcal{L}(\cdot)$ with the variable $\mathtt{C}(I_q^{'})$ which is constrained by a binary value $\{-1, 1\}^K$. In order to simplify this formulation, we initialize $\mathtt{C}(I_q^{'})$ to binary variable, i.e., $\mathtt{B}(I_q^{'}) = sign(C(I_q^{'}))$ and formulate the function below:

$$\min_{\mathtt{B}(I_q^{'})} \mathcal{L}\left(\mathtt{B}(I_q^{'}), h_t\right) = -\frac{1}{K}\mathtt{B}(I_q^{'}) \cdot h_t^T \quad s.t. \quad \mathtt{B}(I_q^{'}) \in \{-1, 1\}^K , \tag{6}$$

Note that if we can obtain the gradient of this formula, we also obtain the gradient term of interest in Equation (5). However, hindered by the binary constraint, we introduce an indicator function $\mathcal{I}(B) = \begin{cases} 0 & if\ B \in \{-1, 1\} \\ +\infty & otherwise \end{cases}$ in order to simplify the above objective function which is used to convert constrained optimization into unconstrained optimization. Therefore, Equation (6) can be reformulated as:

$$\min_{\mathtt{B}(I_q^{'})} \mathcal{L}\left(\mathtt{B}(I_q^{'}), h_t\right) = -\frac{1}{K}\mathtt{B}(I_q^{'}) \cdot h_t^T + \mathcal{I}(\mathtt{B}(I_q^{'})) \tag{7}$$

For non-smooth indicator functions, we can solve the above equation by the proximal operator $Prox_\lambda^I(z) = argmin_h\ g(h) + \frac{\lambda}{2}||hz||^2$ which was proven successful in [37]. Hence, we can update $\mathtt{B}(I_q^{'})$ at t-th step as follow:

$$\mathtt{B}^{j+1}(I_q^{'}) = Prox_{\lambda}^{I}(z)(\mathtt{B}^{j}(I_q^{'}) - \frac{1}{\lambda}\frac{\partial \mathcal{L}\left(\mathtt{B}(I_q^{'}), h_t\right)}{\partial \mathtt{B}(I_q^{'})}) \tag{8}$$

where the optimization process of Equation (8) is the forward–backward splitting algorithm. Furthermore, the Equation (8) has the analytical solution as follow:

$$\mathtt{B}^{j+1}(I_q^{'}) = sign(\mathtt{B}^{j}(I_q^{'}) - \frac{1}{\lambda}\frac{\partial \mathcal{L}\left(\mathtt{B}(I_q^{'}), h_t\right)}{\partial \mathtt{B}(I_q^{'})}) \tag{9}$$

From this analytical solution, we can find the update solution of $\mathtt{B}(I_q^{'})$ in Equation (6) which can be obtained by the sign function after each step of gradient descent. We regard this process as a greedy strategy which makes $\mathtt{B}(I_q^{'})$ generated by each iteration to approach the binary value. However, we can easily obtain an optimal binary solution using Equation (9) while it is difficult to perform back propagation hindered by the sign$(\cdot)$ function. This encourage us to find a new loss function that makes the output value of $\mathtt{B}^{j}(I_q^{'}) - \frac{1}{\lambda}\frac{\partial \mathcal{L}\left(\mathtt{B}^{j}(I_q^{'}), h_t\right)}{\partial \mathtt{B}^{j}(I_q^{'})}$ in Equation (9) infinitely closing to the binary value, then we can use the gradient descent to directly solve Equation (6).

4.3. Back Propagating with Adversarial Penalty Norm

Inspired by [28], which proposed to add a penalty term to the loss function $\mathcal{L}$, we made the output approach the binary value in each iteration. We reformulate the loss function in Equation (1) by constructing a new loss function $\mathcal{P}(\cdot)$. We aim to make the output $\mathtt{C}(I_q^{'})$ approach the binary value, i.e., $\mathcal{P}(\mathtt{C}(I_q^{'}), h_t) = \alpha||\mathtt{C}(I_q^{'}) - sign(\mathtt{C}(I_q^{'}))||_p^p + \mathcal{L}(sign(\mathtt{C}(I_q^{'})), h_t)$ where α is the balance constant. Hence, we reformulate the Equation (6) to simply replace $\mathcal{L}(\cdot)$ by the loss function $\mathcal{P}(\cdot)$ and we also present the solution as follows:

$$\begin{aligned}
\mathtt{C}^{j+1}(I_q^{'}) &= \min_{\mathtt{C}(I_q^{'})} \mathcal{P}\left(\mathtt{C}(I_q^{'}), h_t\right) \\
&= \min_{\mathtt{C}(I_q^{'})} \alpha||\mathtt{C}(I_q^{'}) - sign(\mathtt{C}(I_q^{'}))||_p^p + \mathcal{L}\left(sign(\mathtt{C}^{j}(I_q^{'})), h_t\right) \\
&= \mathtt{C}^{j}(I_q^{'}) - p\alpha||\mathtt{C}^{j}(I_q^{'}) - sign(\mathtt{C}^{j}(I_q^{'}))||_p^{p-1} - lr * \frac{\partial \mathcal{L}\left(sign(\mathtt{C}^{j}(I_q^{'})), h_t\right)}{\partial \mathtt{C}^{j}(I_q^{'})} \\
&\approx sign(\mathtt{C}^{j}(I_q^{'})) - lr * \frac{\partial \mathcal{L}\left(sign(\mathtt{C}^{j}(I_q^{'})), h_t\right)}{\partial \mathtt{C}^{j}(I_q^{'})}
\end{aligned} \tag{10}$$

Note that the addition of the penalty remainder will make $\mathtt{C}(I_q^{'})$ approach the binary value. Furthermore, we set the equivalence for $\frac{\partial \mathcal{L}\left(sign(\mathtt{C}^{j}(I_q^{'})), h_t\right)}{\partial \mathtt{C}^{j}(I_q^{'})} = \frac{\partial \mathcal{L}\left(sign(\mathtt{C}^{j}(I_q^{'})), h_t\right)}{\partial sign(\mathtt{C}^{j}(I_q^{'}))}$. The new iterative solution can be formulated as follows:

$$\mathtt{C}^{j+1}(I_q^{'}) = sign(\mathtt{C}^{j}(I_q^{'})) - lr * \frac{\partial \mathcal{L}\left(sign(\mathtt{C}^{j}(I_q^{'})), h_t\right)}{\partial sign(\mathtt{C}^{j}(I_q^{'}))} \tag{11}$$

Hence, we can obtain a optimal solution on t-th step by performing $sign(\mathtt{C}^{j+1}(I_q^{'}))$ which is equal to Equation (9). This means that we can obtain the value of $\frac{\partial \mathcal{L}(\mathtt{C}(sign(\mathtt{C}(I_q^{'})), h_t)}{\partial \mathtt{C}(I_q^{'})}$ in Equation (5) by calculating $\frac{\partial \mathcal{P}(sign(\mathtt{C}(I_q^{'})), h_t)}{\partial \mathtt{C}(I_q^{'})}$. Therefore, we can regard $\mathcal{P}(\cdot)$ as our new objective function as follows:

$$\min_{\theta} \mathcal{P}(\mathtt{C}(I_q^{'}), h_t) = \alpha||\mathtt{C}(I_q^{'}) - sign(\mathtt{C}(I_q^{'}))||_p^p + \mathcal{L}(sign(\mathtt{C}(I_q^{'})), h_t) \quad (12)$$

Formally, given the query image and then obtain the backward gradient, the adversarial image can be easily generated by the PGD attack [38]. We summarize our process in the Algorithm 1.

Algorithm 1: Optimization Process

Input: Query image I_q, target hash code h_t, adversarial perturbation variable θ

Output: Adversarial Image $I_q^{'}$;

Initialize θ as 0 and iteration variable i as 0;

Initialize the adversarial image $I_q^{'}$ as $I_q^{'}$;

for $i \leq Max\ Iterations$ **do**

  Calculate the loss function $\mathcal{P}(\mathtt{C}(I_q^{'}), h_t) = \alpha||\mathtt{C}(I_q^{'}) - sign(\mathtt{C}(I_q^{'}))||_2^2 + \mathcal{L}\Big(sign(\mathtt{C}(I_q^{'})), h_t\Big)$;

  Calculate the backward gradient $\frac{\partial \mathcal{P}(\mathtt{C}(I_q^{'}), h_t)}{\partial \mathtt{C}(I_q^{'})} = 2\alpha||\mathtt{C}(I_q^{'}) - sign(\mathtt{C}(I_q^{'}))||_2 + \frac{\partial \mathcal{L}(sign(\mathtt{C}(I_q^{'})), h_t)}{\partial \mathtt{C}(I_q^{'})}$;

  Set the gradient equivalent $\frac{\partial \mathcal{L}\left(sign(\mathtt{C}(I_q^{'})), h_t\right)}{\partial \mathtt{C}(I_q^{'})} = \frac{\partial \mathcal{L}\left(sign(\mathtt{C}(I_q^{'})), h_t\right)}{\partial sign(\mathtt{C}(I_q^{'}))}$;

  Calculate the gradient $\frac{\partial \mathcal{P}(\mathtt{C}(I_q^{'}), h_t)}{\partial \mathtt{C}(I_q^{'})} = 2\alpha||\mathtt{C}(I_q^{'}) - sign(\mathtt{C}(I_q^{'}))||_2 + \frac{\partial \mathcal{L}(sign(\mathtt{C}(I_q^{'})), h_t)}{\partial sign(\mathtt{C}(I_q^{'}))}$;

  Calculate the adversarial gradient $\frac{\partial \mathcal{P}(\mathtt{C}(I_q^{'}), h_t)}{\partial \theta} = \frac{\partial \mathcal{P}(\mathtt{C}(I_q^{'}), h_t)}{\partial \mathtt{C}(I_q^{'})} * \frac{\partial \mathtt{C}(I_q^{'})}{\partial \theta}$;

  Given the adversarial gradient $\frac{\partial \mathcal{P}(\mathtt{C}(I_q^{'}), h_t)}{\partial \theta}$, update adversarial image $I_q^{'}$ with PGD attack;

  i = i + 1

return the adversarial image $I_q^{'}$;

5. Experiment

5.1. Datasets Description

We evaluated our method on two popular datasets. One dataset is NUS-WIDE [39], which consists of 269,648 images in 81 categories. Significantly, the training samples were exploited as our retrieval database, and the evaluating samples as queries. The other dataset is FLICKER-25K [40], which contains 25,000 images with 38 classes. We randomly sampled 500 as query images in each dataset and kept the remaining images as retrieval and training databases. We randomly selected target images different from the original label of each carrier image.

5.2. Baseline and Metrics

In this paper, we focus on evaluating the effectiveness of the different attacking methods. Our method can be easily deployed in the Point-To-Point (P2P) and Point-To-Set(P2S) approaches [16]. We set two baseline methods, including the Point-To-Point (P2P) [16] method, which exploits the tanh(·) function with the hyperparameter α, referred to as Tanh-P2P; see Equation (2). The maximum optimization iteration number was set as 2000. The hyperparameter α was initialized as 0.1 during the first 1000 iterations and updated every 200 iterations according to [0.2, 0.3, 0.5, 0.7, 1.0] during the last 1000 iterations. Moreover, we randomly selected nine images with the same label to generate the anchor code [16] for the Tanh-P2S baseline.

Here, we introduce three metrics to evaluate attacking performance. We follow the established method to set MAP (mean average precision), a widely used criterion in image retrieval. The higher the MAP value, the stronger the performance of the retrieval system. Moreover, we exploited the t-MAP (target mean average precision), similar to MAP, which set the target label setting in advance instead of the original label used in MAP. Finally, the `Hamming distance` is extensively used in the adversarial attack of image retrieval, and was used to reflect the convergence speed in our experiment. Significantly, the smaller the Hamming distance, the better the attack effect.

5.3. Experiment Settings

All experiments were processed on the Pytorch platform with RTX2070 Super, Intel(R) Core(TM) i7-9700 CPU @ 3.00 GHz. For image hashing, we chose the DPSH [41] method to build the target hashing model, which is one of the most representative deep hashing methods. We set the VGG-11 [33] as the backbone and set the length of the hash code as 16, 24, 32, 48 bits for evaluation. Specifically, we exploited the random noise sampled from the uniform noise within the $[-\tau,\tau]$ value, Tanh-P2P, Tanh-P2S, and our Relaxation Norm Method (RNM). The hyperparameter β in Equation (12) was set as $1/K$, where K is the length of the hash bits. The maximal perturbation magnitude ϵ was 0.032. We adopted PGD [18] to optimize the Tanh method and our method. We set the maximum iteration as 2000 with a learning rate of 1 for comparison.

5.4. Results

Convergence analysis: Hamming distance. We show the Hamming distance between the generated adversarial hash code and the target hash code over the entire iteration optimization in Figure 2. We summarize the following two points: First, our method has faster convergence than the strategy using the Tanh(·) function. For example, in Figure 2d, demonstrating the convergence speed on the NUS-WIDE dataset with the 48-bit hash code, our method can minimize the Hamming distance within 100 iterations. According to the description in Equation (2), the hyperparameter α with the tanh(·) function will gradually increase. Specifically, the α was set as 0.1 for the first 1000 iterations, and sequentially became [0.2, 0.5, 0.7, 0.9] in the last 1000 iterations. However, the convergence rate of the Hamming distance in the first 1000 iterations was fast at the beginning and then appeared to be slow. The optimal solution was reached after 2000 iterations. Obviously, our method showed a better performance in terms of the convergence speed. Second, although our method performs well on both datasets, it is noticed that the Tanh(·) function strategy also performs relatively well on the FLICKR-25K dataset. We consider the reason for this to be that the continuous feature value $C(I_q^{'})$ of the retrieval model on the FLICKR-25K dataset is more concentrated, so that the optimization process can be performed smoothly even when $\alpha = 0.1$ in the first 1000 iterations. Through the analysis of convergence, it is concluded that our proposed method can generate adversarial examples faster and start privacy protection in a shorter amount of time.

Privacy protection effect: t-MAP. Table 1 lists the t-MAP and original MAP of FLICKER-25K and NUS-WIDE datasets with [16, 24, 32, 48] hash bits in length. t-MAP shows the attack efficiency, and the higher the t-MAP value, the better the privacy protection. The MAP was computed based on the original labels and represents the original performance of the retrieval model. The t-MAP values of noise on the FLICKR-25K and NUS-WIDE datasets are very small, which indicates that the image with random noise cannot cause the deep hash model to return the target image. In contrast, all t-MAP values of RNM and Tanh are higher than the "Original" MAP value, which validates the effectiveness of adversarial attacks. In addition, the method with the P2S property is more effective than the P2P method, which means that our method can be generally applicable to the adversarial attack method of deep hash retrieval. However, we noticed that the the t-MAP of RNM is slightly higher than Tanh under 16, 24, 32, and 48 bits. For example, in 16 bits, Tanh-P2S is 85.79%, while RNM-P2S is 85.92%. A similar situation can be seen in another setting.

We selected the same target hash code for each adversarial example of the four methods. After sufficient optimization iterations for both Tanh and RNM, the generated adversarial hash codes were very close to the target hash codes. The results show the superiority of the adversarial attack method from the side. Targeted adversarial attack methods can be used for image content privacy protection. By retrieving the carrier image with the adversarial attack, the same retrieval content as the original image can be obtained, and the returned experimental result may even be better than the original image.

Table 1. t-MAP(%) on targeted attack methods and MAP(%) of query object with the target hash code of query images under 16, 24, 32, 48 bits hash code on two images datasets.

Method	Metric	FLICKER-25k				NUS-Wide			
		16 bits	24 bits	32 bits	48 bits	16 bits	24 bits	32 bits	48 bits
Noise	t-MAP	0.76%	0.57%	1.31%	0.29%	0.57%	0.57%	0.61%	0.96%
Tanh-P2P	t-MAP	83.22%	84.39%	85.25%	85.92%	73.97%	76.10%	76.66%	77.49%
Tanh-P2S	t-MAP	85.79%	88.44%	88.76%	89.18%	78.05%	79.53%	80.18%	81.10%
RNM-P2P	t-MAP	83.95%	84.32%	83.22%	85.26%	74.02%	76.03%	76.93%	77.25%
RNM-P2S	t-MAP	85.92%	88.45%	88.89%	89.49%	77.96%	79.60%	80.30%	81.33%
Original	MAP	78.88%	80.69%	81.01%	81.48%	70.94%	73.01%	73.61%	74.11%

5.5. Ablation Study

Impact of Perturbation Magnitude on Privacy Protection. We explore the relationship between privacy protection and adversarial perturbation magnitude. Table 2 shows the different maximal magnitude of τ in Equation (1). We evaluate the experiment on 32 bits with the different value $\tau \in \{0.01, 0.02, 0.03, 0.04, 0.05\}$ under two datasets and show them in Table 2. It can be seen that the t-MAP will increase with the increasing τ value, proving that there is a trade off between the attack performance and the perturbation magnitude. The larger the perturbation magnitude, the better the privacy protection.

Table 2. The t-MAP (%) of targeted attack with variable parameter τ on NUS-WIDE dataset with 24-bit hash code.

Method	Metrics	0.01	0.02	0.03	0.04	0.05
Noise	t-MAP	1.24%	2.78%	1.31%	1.25%	1.14%
Tanh-P2P	t-MAP	54.95%	63.11%	75.24%	75.90%	77.23%
RNM-P2P	t-MAP	52.44%	67.03%	76.19%	76.13%	76.95%

Image Semantic Privacy Protection: Non-targeted attack. The non-targeted attack can also be treated as a particular target attack requiring the target label to be different from the original. We exploit the above method to explore the non-target attack and show it in Table 3; it can be seen that our method can achieves excellent results on both datasets. Untargeted attacks can be used to protect the privacy of image semantic information.

Comparison with Privacy Protection Methods. We introduced PSNR (Peak Signal-to-Noise Ratio) and MAP to assess the visual quality and privacy protection effect of the processed images, and show that higher PSNR values have better visual quality. The low MAP have better privacy protection effect. We set up traditional privacy protection methods such as mosaic, brightness, transparency, and noise. For our method, we generated adversarial examples with values of hyperparameter τ of [0.1, 0.2, 0.3, 0.4] for comparison and show the examples in Figure 3. We notice that the traditional privacy-preserving strategy is less effective in terms of visual quality and privacy protection, and the best performance in terms of traditional strategies is adding random noise with PSNR = 24.53 dB. In adversarial strategies, PSNR and map performs better, for example, when τ = 0.2, and PSNR = 43.51 dB, MAP = 19%, which is 18.98 dB and 71% higher than the best traditional method on visual quality.

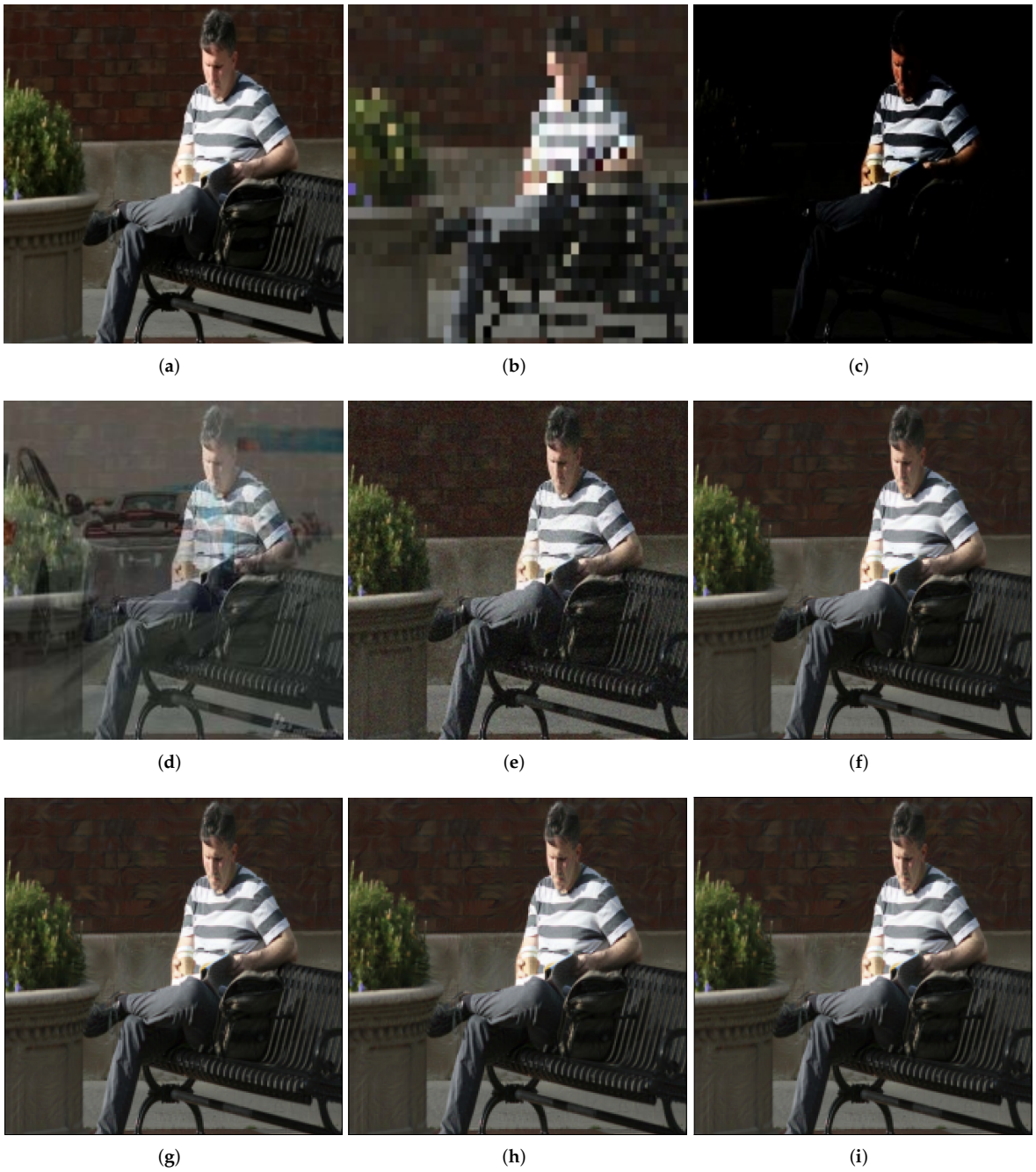
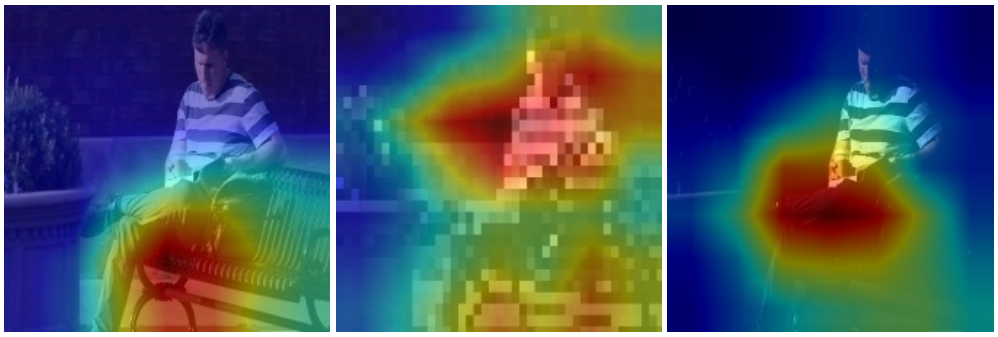


Figure 3. Examples under the action of traditional privacy-preserving strategies, such as mosaic (**b**), brightness (**c**), transparency (**d**), and noise (**e**). We also show examples of adversarial attacks when τ = [0.01, 0.02, 0.03, 0.04]. We demonstrate visual quality and the effect of privacy-preserving via the PSNR metric and MAP. (**a**) Original images, MAP=91%; (**b**) mosaic, PSNR = 16.66 dB, MAP = 23%; (**c**) brightness, PSNR = 13.02 dB, MAP = 89%; (**d**) transparency, PSNR = 18.28 dB, MAP = 88%; (**e**) noise, PSNR = 24.53 dB, MAP = 90%; (**f**) τ = 0.01, PSNR = 44.45 dB, MAP = 20%; (**g**) τ = 0.02, PSNR = 43.51 dB, MAP = 19%; (**h**) τ = 0.03, PSNR = 43.32 dB, MAP = 17%; (**i**) τ = 0.04, PSNR = 43.22 dB, MAP = 16%.

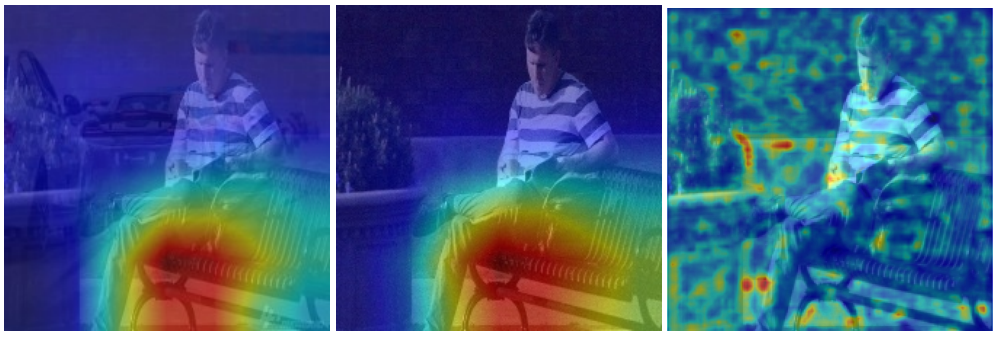
A Visual Explanation of Why Adversarial Attack Shows the Better Performance in Terms of Privacy Protection: Heatmap. Heatmaps can show the decisions of the DNN regarding objects from a visual perspective. To verify the general applicability of adversarial examples in privacy protection, we constructed a deep hash retrieval model with Resnet50 [42] and Densenet161 [43] as the backbone. Furthermore, we utilized the widely used method Grad-Cam [29] for generating heatmaps of traditional methods and adversarial examples, as shown in Figure 4. We found similar results on the two benchmark frameworks. Specifically, we noticed that only the heatmap generated by the mosaic method is significantly different from the heatmap generated by the original image in the traditional method. However, the example generated by the mosaic method is visually different from the original image. Moreover, the heatmaps generated by the remaining traditional methods are relatively similar to those of the original images. This shows that the examples generated by traditional methods cannot protect the privacy of users well, because the DNN-based hash model can still identify the examples normally. It is observed that in the examples generated by adversarial attack, the warm-colored pixels are dispersed, which means that the attention of the DNN-based hashing model is diverted. Therefore, the examples generated by users using adversarial methods can enable the DNN-based hashing model to identify errors, thereby effectively protecting user privacy.

Table 3. MAP (%) of non-targeted attack with various attack methods for queries on the FLICKER-25K and NUS-WIDE datasets with 16, 24, 32, and 48 bits hash code. The best results are marked in bold, and the second best results are underlined.

Method	FLICKER-25k				NUS-WIDE			
	16 bits	**24 bits**	**32 bits**	**48 bits**	**16 bits**	**24 bits**	**32 bits**	**48 bits**
Noise	74.16%	76.54%	78.47%	80.36%	70.08%	70.25%	71.48%	72.02%
Tanh-P2P	1.92%	3.54%	**3.15**%	2.01%	9.74%	7.26%	6.97%	9.50%
RNM-P2P	1.69%	2.42%	3.32%	**1.90**%	9.65%	6.47%	5.59%	9.75%
HAG	**1.36**%	**2.64**%	4.53%	1.97%	**3.79**%	**3.64**%	**3.71**%	**3.12**%

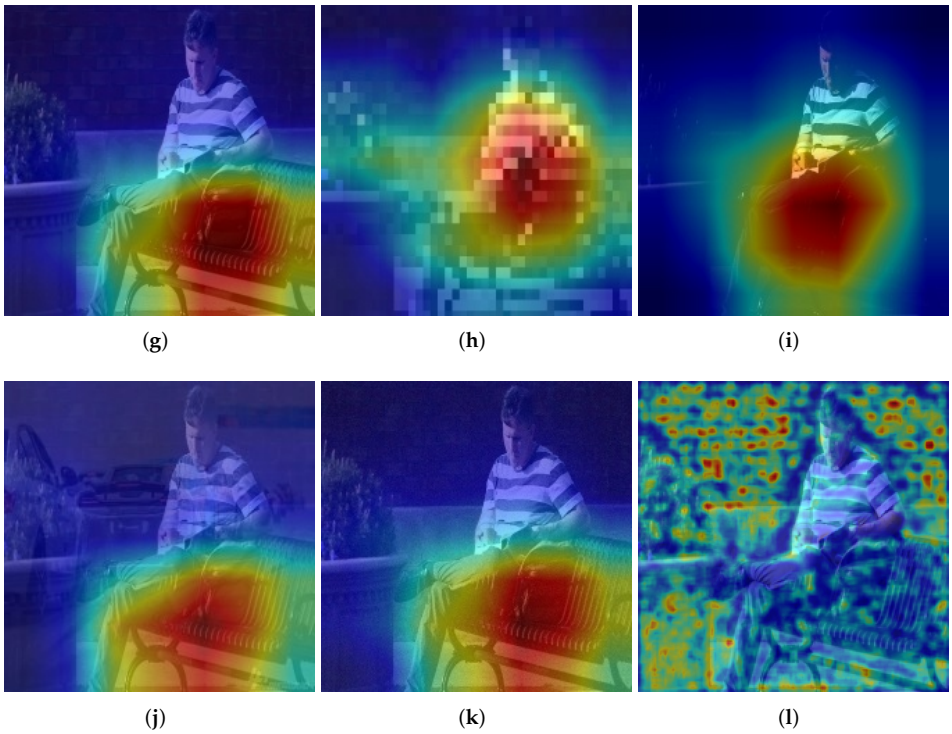


(**a**) (**b**) (**c**)



(**d**) (**e**) (**f**)

Figure 4. *Cont.*



(**g**) (**h**) (**i**)

(**j**) (**k**) (**l**)

Figure 4. Examples generated by traditional methods and adversarial methods through the visualization method, including mosaic, brightness, transparency, noise, and adversarial attack. We set up Resnet50 and Densenet161 as the backbone for constructing a deep hash network. The warmer colors in the examples indicate the attention of the deep neural network has been attracted. (**a**) Original (Resnet50); (**b**) mosaic (Resnet50); (**c**) brightness (Resnet50); (**d**) transparency (Resnet50); (**e**) noise (Resnet50); (**f**) adversarial example (Resnet50); (**g**) original (Densenet161); (**h**) mosaic (Densenet161); (**i**) brightness (Densenet161); (**j**) transparency (Densenet161); (**k**) noise (Densenet161); (**l**) adversarial example (Densenet161).

6. Conclusions

This paper explores the use of adversarial attacks to mislead image hash retrieval systems in order to protect user privacy. Previous work used adversarial attacks to protect user privacy; an approximation strategy based on the tanh function for binary output to address the problem of vanishing gradients during optimization. We abandoned the approximation strategy above by using a penalty term to strictly project image features into binary values. We utilized PSNR and gradient-based heatmaps to compare the advantages of adversarial-attack-based privacy-preserving methods with those of traditional methods. Our method achieves satisfactory performance in several aspects, and can be easily extended to more industrial privacy protection fields, such as image sensors, and the Appendix A illustrates the industrial application of our method.

Author Contributions: Conceptualization, L.H. and Y.Z.; methodology, L.H.; software, Y.Z. and L.H.; validation, L.H. and Y.Z.; formal analysis, L.H.; investigation, L.H. and Y.Z.; resources, L.H.; data curation, L.H.; writing—original draft preparation, L.H.; writing—review and editing, C.H. and Y.Z.; visualization, L.H. and Y.Z.; supervision, R.S.; project administration, C.H. and L.H.; funding acquisition, C.H. and R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 62177046; National Natural Science Foundation of China, grant number 61977062; and Hunan Natural Science Foundation, grant number 2021JJ30866.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

To demonstrate the practicability of our method, we apply our method in an industrial scenario. Taking image sensor transmission as an example, Figure A1 shows the privacy protection during image transmission in industrial scenarios.

In order to protect the privacy of the images in the image sensor, the output of the image by the sensor can be directly transmitted to the image privacy protection system, and the image privacy protection system can generate adversarial examples. It can be seen from the image that the generated adversarial examples are visually indistinguishable from the original images, but the hash codes generated are entirely different. Therefore, using the generated adversarial examples for retrieval can effectively protect the privacy of the image sensor.

In particular, the image privacy protection system can generate specific adversarial examples according to the usage of the image.

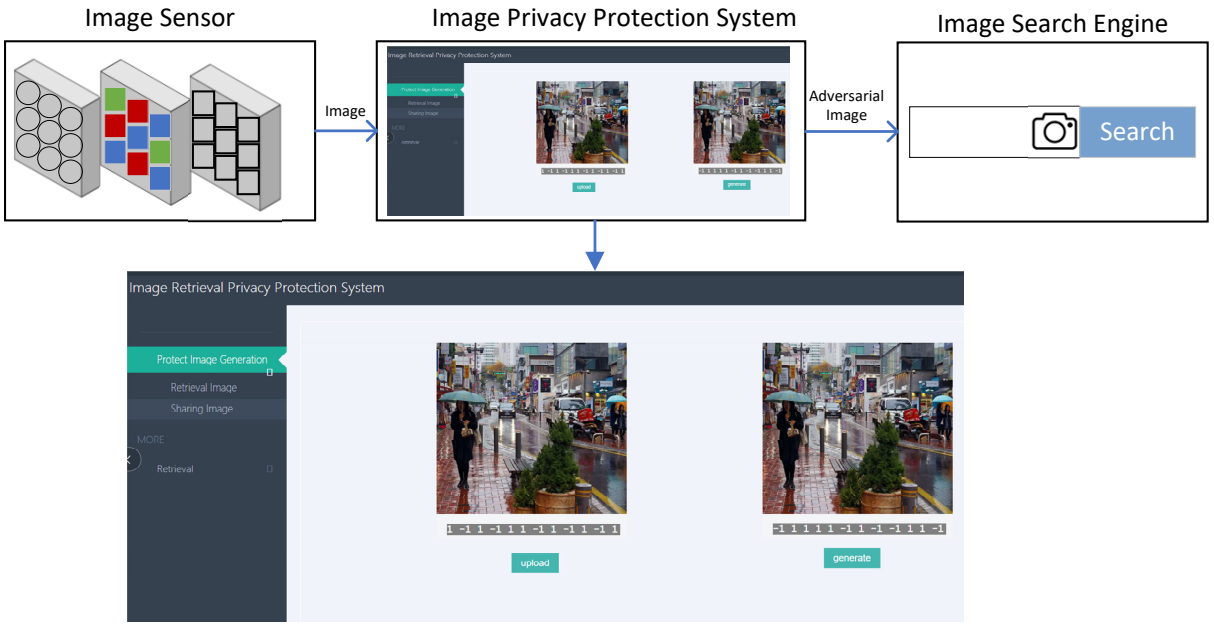


Figure A1. Above figure shows the industrial application scenario of the adversarial attack method. In order to protect the privacy of the images collected by the image sensor, the output image of the image sensor is input into the image privacy protection system to generate adversarial examples. It can be seen from the figure that the hash code of the original image is different from the hash code of the adversarial example. So, using adversarial examples for image retrieval can protect the semantics of the original image's privacy.

References

1. Bernardmarr. Available online: https://bernardmarr.com/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/ (accessed on 11 January 2022).
2. Seroundtable. Available online: https://www.seroundtable.com/google-search-by-image-storage-14101.html (accessed on 11 January 2022).
3. Theverge. Available online: https://www.theverge.com/2018/12/14/18140771/facebook-photo-exposure-leak-bug-millions-users-disclosed (accessed on 11 January 2022).
4. Liu, H.; Wang, R.; Shan, S.; Chen, X. Deep supervised hashing for fast image retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2064–2072.

5. Cao, Z.; Long, M.; Wang, J.; Yu, P.S.Hashnet: Deep learning to hash by continuation. In Proceedings of the IEEE International Conference on Computer Vision(ICCV), Venice, Italy, 22–29 October 2017; pp. 5608–5617.
6. Lai, Z.; Chen, Y.; Wu, J.; Wong, W.K.; Shen, F. Jointly sparse hashing for image retrieval. *IEEE Trans Image Process.* **2018**, *27*, 6147–6158. [CrossRef]
7. Shen, F.; Xu, Y.; Liu, L.; Yang, Y.; Huang, Z.; Shen, H.T. Unsupervised deep hashing with similarity-adaptive and discrete optimization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 3034–3044. [CrossRef]
8. Chen, Z.; Yuan, X.; Lu, J.; Tian, Q.; Zhou, J. Deep hashing via discrepancy minimization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 6838–6847.
9. Gu, W.; Gu, X.; Gu, J.; Li, B.; Xiong, Z.; Wang, W. Adversary guided asymmetric hashing for cross-modal retrieval. In Proceedings of the 2019 on International Conference on Multimedia Retrieval(ICMR), Ottawa, ON, Canada, 10–13 June 2019; pp. 159–167.
10. Wickramasuriya, J.; Alhazzazi, M.; Datt, M.; Mehrotra, S.; Venkatasubramanian, N. Privacy-protecting video surveillance. In Proceedings of the Real-Time Imaging IX, San Jose, CA, USA, 18–20 January 2005; pp. 64–75.
11. Elkies, N.; Fink, G.; Bärnighausen, T. "Scrambling" geo-referenced data to protect privacy induces bias in distance estimation. *Popul. Environ.* **2015**, *37*, 83–98. [CrossRef]
12. Yang, E.; Liu, T.; Deng, C.; Tao, D. Adversarial examples for hamming space search. *IEEE Trans. Cybern.* **2018**, *50*, 1473–1484. [CrossRef] [PubMed]
13. Tolias, G.; Radenovic, F.; Chum, O. Targeted mismatch adversarial attack: Query with a flower to retrieve the tower. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 5037–5046.
14. Li, J.; Ji, R.; Liu, H.; Hong, X.; Gao, Y.; Tian, Q. Universal perturbation attack against image retrieval. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 4899–4908.
15. Xiao, Y.; Wang, C.; Gao, X. Evade Deep Image Retrieval by Stashing Private Images in the Hash Space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 16–18 June 2020; pp. 9651–9660.
16. Bai, J.; Chen, B.; Li, Y.; Wu, D.; Guo, W.; Xia, S.T.; Yang, E.H. Targeted attack for deep hashing based retrieval. In Proceedings of the European Conference on Computer Vision(ECCV), Glasgow, UK, 23–28 August 2020; pp. 618–634.
17. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014.
18. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial examples in the physical world. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
19. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. Deepfool: A simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2574–2582.
20. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The limitations of deep learning in adversarial settings. In Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P), Saarbruecken, Germany, 21–24 March 2016; pp. 372–387.
21. Moosavi-Dezfooli, S.M.; Fawzi, A.; Fawzi, O.; Frossard, P. Universal adversarial perturbations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1765–1773.
22. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the 2017 IEEE Security and Privacy (SP), San Jose, CA, USA, 22–24 May 2017; pp. 39–57.
23. Xie, C.; Wu, Y.; Maaten, L.V.D.; Yuille, A.L.; He, K. Feature denoising for improving adversarial robustness. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 501–509.
24. Xu, K.; Liu, S.; Zhao, P.; Chen, P.; Zhang, H.; Fan, Q.; Erdogmus, D.; Wang, Y.; Lin, X. Structured adversarial attack: Towards general implementation and better interpretability. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
25. Fan, Y.; Wu, B.; Li, T.; Zhang, Y.; Li, M.; Li, Z.; Yang, Y. Sparse adversarial attack via perturbation factorization. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 35–50.
26. Bai, J.; Chen, B.; Wu, D.; Zhang, C.; Xia, S.T. Universal Adversarial Head: Practical Protection against Video Data Leakage. In Proceedings of the ICML 2021 Workshop on Adversarial Machine Learning, Online, 18–24 July 2021.
27. Wang, X.; Zhang, Z.; Wu, B.; Shen, F.; Lu, G. Prototype-supervised Adversarial Network for Targeted Attack of Deep Hashing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 16357–16366.
28. Su, S.; Zhang, C.; Han, K.; Tian, Y. Greedy hash: Towards fast optimization for accurate hash coding in cnn. In Proceedings of the 32nd International Conference on Neural Information Processing Systems(NIPS), Montréal, QC, Canada, 4–5 December 2018; pp. 806–815.
29. Selvaraju, R.R.; Cogswell, M.; Das, A. Grad-cam: Visual explanations from deep networks via gradient-based. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
30. Cao, Y.; Long, M.; Liu, B.; Wang, J. Deep cauchy hashing for hamming space retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 1229–1237.

31. Liu, Z.; Zhao, Z.; Larson, M. Who's afraid of adversarial queries? The impact of image modifications on content-based image retrieval. In Proceedings of the 2019 on International Conference on Multimedia Retrieval (ICMR), Ottawa, ON, Canada, 10–13 June 2019; pp. 306–314.
32. Wang, X.; Zhang, Z.; Lu, G.; Xu, Y. Targeted Attack and Defense for Deep Hashing. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGTR), Online, 11–15 July 2021; pp. 2298–2302.
33. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural INF Process Syst.* **2012**, *25*, 1097–1105. [CrossRef]
34. Gu, Y.; Ma, C.; Yang, J. Supervised recurrent hashing for large scale video retrieval. In Proceedings of the 24th ACM International Conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 272–276.
35. Xiao, Y.; Wang, C. You See What I Want You To See: Exploring Targeted Black-Box Transferability Attack for Hash-Based Image Retrieval Systems. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 1934–1943.
36. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
37. Shen, F.; Zhou, X.; Yang, Y.; Song, J.; Shen, H.T.; Tao, D. A fast optimization method for general binary code learning. *IEEE Trans. Image Process.* **2016**, *25*, 5610–5621. [CrossRef] [PubMed]
38. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv* **2017**, arXiv:1706.06083.
39. Chua, T.S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; Zheng, Y. Nus-wide: A real-world web image database from national university of singapore. In Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR), Santorini Island, Greece, 8-10 July 2009; pp. 1–9.
40. Huiskes, M.J.; Lew, M.S. The mir flickr retrieval evaluation. In Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval (MM), Vancouver, BC, Canada, 30–31 October 2008; pp. 39–43.
41. Li, W.J.; Wang, S.; Kang, W.C. Feature learning based deep supervised hashing with pairwise labels. *arXiv* **2015**, arXiv:1511.03855.
42. He, K.; Zhang, X.; Ren, S. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, USA, 27–30 June 2016; pp. 770–778.
43. Huang, G.; Liu, Z.; Van Der Maaten, L. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.