Unsupervised Multi-criteria Adversarial Detection in Deep Image Retrieval

Yanru Xiao[1], Cong Wang[2(✉)], and Xing Gao[3]

[1] Old Dominion University, Norfolk, VA, USA
yxiao002@odu.edu
[2] Zhejiang University, Hangzhou, China
cwang85@zju.edu.cn
[3] University of Delaware, Newark, DE, USA
xgao@udel.edu

Abstract. The vulnerability in the algorithm supply chain of deep learning has imposed new challenges to image retrieval systems in the downstream. Among a variety of techniques, deep hashing is gaining popularity. As it inherits the algorithmic backend from deep learning, a handful of attacks are recently proposed to disrupt normal image retrieval. Unfortunately, the defense strategies in softmax classification are not readily available to be applied in the image retrieval domain. In this paper, we propose an efficient and unsupervised scheme to identify unique adversarial behaviors in the hamming space. In particular, we design three criteria from the perspectives of hamming distance, quantization loss and denoising to defend against both untargeted and targeted attacks, which collectively limit the adversarial space. The extensive experiments on four datasets demonstrate 2–23% improvements of detection rates with minimum computational overhead for real-time image queries.

Keywords: Deep hashing · adversarial detection · image retrieval

1 Introduction

Powered by neural networks, deep hashing enables image retrieval at a large scale [7,8,26,27,46,48]. By representing high-dimensional images with compact binary codes, retrieval becomes an efficient similarity computation of Hamming distance. Google [3], Bing [2], Pinterest [4], Taobao [1] have all incorporated image query as part of their products. Despite of its great success, deep hashing also inherits the vulnerabilities from neural networks [36] with new attack vectors and effects. By introducing adversarial perturbations either on the query or database images, normal requests can be diverted to an irrelevant (*untargeted attack*) [45] or specific category (*targeted attack*) [6,40,42], e.g., turning a query

The work was supported by the U.S. National Science Foundation under Grant 2245055.

H. Duan et al. (Eds.): SecureComm 2023, LNICST 567, pp. 145–164, 2025.
https://doi.org/10.1007/978-3-031-64948-6_8

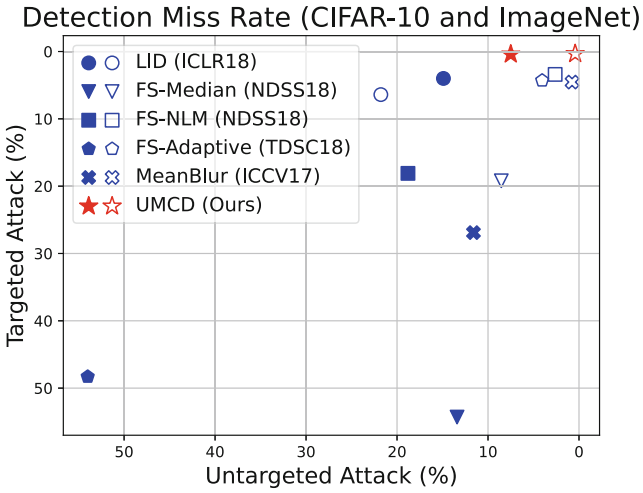


Fig. 1. Miss rate of different detections for untageted/targeted adversarial examples in deep hashing. The solid and hollow markers are for CIFAR-10 and ImageNet respectively. Our method (UMCD) has the lowest miss rate on both datasets.

of "husky dog" into retrieving a branded "dog food" so the attacker can advertise their products for free (Fig. 1).

With a handful of efforts on the attack side [6,40,42,45], deep hashing still falls short to defend against adversarial examples in the hamming space. *Adversarial training* and *detection* are the two common defenses in softmax classification. Yet, adversarial training has to deal with the non-trivial trade-off between robustness and accuracy [47]. According to our implementation (see appendix), finding the min-max saddle points becomes even more difficult under the hash function, which suffers from a large accuracy loss. On the other hand, detection aims to unveil the adversarial behaviors on different levels of raw pixel [15,17], feature distribution [17,23,28], softmax probabilities [19] and frequency components [39] in a *supervised* [11] or *unsupervised* manner [44]. Based on the prior knowledge of attack methods, supervised detection trains a classifier to distinguish the adversarial images, but is hard to extrapolate to the unknown attacks. To this end, we pursue the direction of unsupervised anomaly detection in this paper. Different from softmax classification on a closed set of class probabilities, deep hashing maps similar/dissimilar images into binary codes in an open Hamming space. Thus, the focus of our work is to tap into the unique adversarial behaviors in deep hashing to detect both untargeted and targeted attacks.

Starting from the untargeted attacks [45], we first theoretically deduce the hamming distance distribution from the adversarial image to other categories, which asymptotically approaches a Gaussian distribution. For targeted attacks, we discover an interesting adversarial behavior on the quantization loss: when the adversarial objective is to produce the same hash code of a targeted category [6,40], it unintentionally brings the quantization loss close to zero. Thus, we

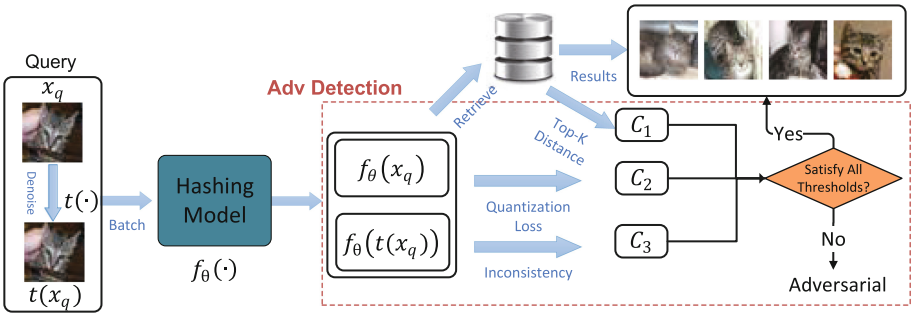


Fig. 2. The proposed detection framework: highlighted by the dash lines.

first develop two thresholding methods that take hamming distance and quantization loss as the proxies. Then we combine the two criteria with a denoising-based detection to measure the disagreement between an input and its denoised transformation. We demonstrate that this combination can successfully defend against both *gray-box* attackers, who have no prior knowledge of the detection method, and the strongest white-box attackers, who know the existence of the detection and can implement countermeasures. The overall framework is shown in Fig. 2.

The main contributions of this paper are summarized below. To the best of our knowledge, this is the first, unsupervised effort to defend against adversarial attacks in deep hashing. Based on the novel discoveries and analysis, we propose three criteria to unveil adversarial behaviors of targeted and untargeted attacks in the hamming space, and demonstrate their complementing relations against the strongest white-box attackers. The extensive experiments on CIFAR-10, ImageNet, MS-COCO and NUSWIDE datasets show that the proposed method surpasses the state-of-the-art defenses by up to 23% in detection rates with negligible computational overhead for real-time image queries.

2 Preliminary

This section illustrates the fundamentals of deep hashing and adversarial attacks.

2.1 Deep Hashing

Given a dataset of N samples $X = \{x_1, x_2, \ldots, x_N\}$, $x_i \in \mathbb{R}^D$ and their corresponding labels $Y = \{y_1, y_2, \ldots, y_N\}$, $y_i \in \mathbb{R}^C$, where x_i is the i-th sample and $y_{c,i} = 1$ if the i-th image is associated with class c. Deep hashing learns a function $f_\theta(x)$ that maps the input image x into a K-bit binary code $h(x)$ via a sign operation,

$$h(x) = sign(f_\theta(x)) \in \{-1, +1\}^K, \tag{1}$$

where θ are the parameters learned from minimizing the weighted combination of the similarity loss $\mathcal{L}_S$ and quantization loss $\mathcal{L}_Q$ [7,8,26,27,46,48],

$$\theta = \arg\min_{\theta} \mathcal{L}_S + \lambda \mathcal{L}_Q. \tag{2}$$

$\mathcal{L}_S$ represents the hamming distance $D_h(h(x_i), h(x_j))$ between two images x_i and x_j with their similarity $s(y_i, y_j)$,

$$s(y_i, y_j) = \begin{cases} +1, & \text{if } y_i y_j^T > 0 \\ -1, & \text{otherwise.} \end{cases} \tag{3}$$

$\mathcal{L}_Q$ is the quantization loss to minimize the difference between the continuous output of $f_\theta(x)$ and its binary code $h(x)$. The objective is to minimize the hamming distance $D_h(h(x_i), h(x_j))$ between two samples x_i and x_j when they are similar, maximize the hamming distance when they are dissimilar, and meanwhile, represent the continuous $f_\theta(x)$ as binary codes. Both $D_h(h_1, h_2)$ and $h(x)$ are non-differentiable regarding their inputs. A common technique is to use the differentiable form of $D_h(h_1, h_2)$ noted as $\frac{1}{2}(K - {h_1}^T h_2)$ during backpropagation, where h_1, h_2 are the continuous floating point representation in $[-1, +1]$, and the binary hash codes $h(x)$ are represented by the continuous output of $f_\theta(x)$. The gap between such continuous and binary representations is considered as the quantization loss $\mathcal{L}_Q$, which is minimized in Eq. (2).

Deep hashing consists of two main components, a *database* and a *model*. The database stores the images and their pre-computed hash codes. Given a query image x with hash code $h(x)$, the system returns the top-k images from the database which are $h(x)$'s k-nearest neighbors determined by hamming distance. The retrieval performance is calculated by the mean average precision (mAP), which is the ratio of images similar to x. In this paper, we base the hashing framework on the state-of-the-art method called Central Similarity Quantization (CSQ) [46]. CSQ pre-determines the optimal hash codes based on the Hadamard matrix and randomly selects a set of hash codes with sufficient distances from each other as the hash centers from the Hadamard matrix (or from a random binary matrix if the Hadamard matrix is not available). Since different hashing techniques share the general objective of Eq. (2), our defense applies to other techniques as well [7,8,26,27,48].

2.2 Adversarial Attacks

Untargeted Attack. [45] finds an adversarial image x' by maximizing the hamming distance between the hash codes of adversarial examples and original images, subject to the $\mathcal{L}_\infty$ bound of ϵ.

$$\begin{aligned} &\max_{x'} D_h\big(h(x'), h(x)\big) \\ &\text{s.t. } \|x - x'\|_\infty \leq \epsilon \end{aligned} \tag{4}$$

It works effectively to reduce the mAP by pushing the original image towards the furthest hamming distance in the hash space.

Targeted Attack. [6,40,42] attempts to minimize the hamming distance from x' to the targeted hash code h_t of a specific category,

$$\begin{aligned} &\min_{x'} D_h(h(x'), h_t) \\ &\text{s.t. } \|x - x'\|_\infty \leq \epsilon \end{aligned} \tag{5}$$

Once the attacker has embedded the adversarial images in the database, targeted attacks enable image retrieval from a specific category upon user queries. For example, as illustrated in [42], the database could mistakenly return the advertisements of branded beer from the database upon the query of facial lotions.

Attack Model. Attackers can carry out both untargeted and targeted attacks. In particular, we consider two types of *gray-box* and *white-box* attackers. *Gray-box* attackers have access to all the information including network architecture, weights and data, but are not aware of the existence of adversarial detection. The stronger *white-box* attackers are aware of both the model function/parameters and the existence of the detection. So they implement different bypassing strategies as discussed in Sect. 4.

3 Adversarial Behaviors in the Hamming Space

Among a variety of artifacts left by adversarial images in classification networks, one of the most evident "adversarial behaviors" is from the softmax function [19,23]. Due to the fast-growing exponentiation, it magnifies small changes in the logits [19] and becomes overconfident in the presence of adversarial images by regularizing other categories [23]. In contrast to softmax, which makes the decision from a closed set of categories, hashing maps similar images into compact hamming balls in an open hamming space of $\{-1, 1\}^K$. In this section, we define three criteria to identify adversarial behaviors in the hamming space.

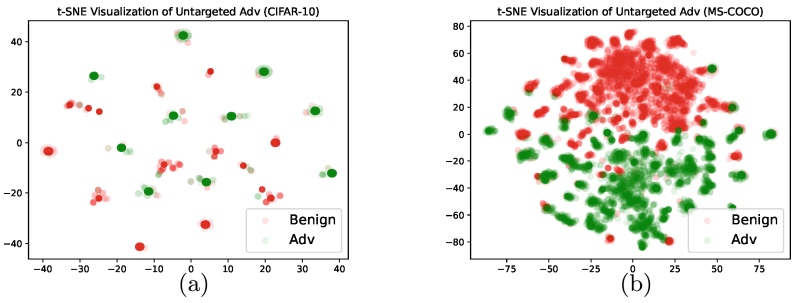


Fig. 3. t-SNE visualization of untargeted adversarial images vs. original images of different datasets (a) CIFAR-10. (b) MS-COCO.

3.1 Detecting Untargeted Attacks (C_1)

We start with untargeted attacks that maximize the hamming distance between x' and x [45]. Though such behavior is straightforward to discern, we seek a theoretical answer to the distribution of $h(x')$ when the attacking capacity is maximized.

Assumption 1. The network is capable of learning *perfect* hash codes with the minimum intra-class distance (i.e., equals to zero) and maximum margin between each other.

If Assumption 1 holds, what is the hamming distance from the adversarial $h(x')$ to the rest of the hash codes? To answer this question, we first establish the distribution of the maximum inter-class distance as illustrated in the following Lemma.

Lemma 1. Given a number of C classes in the K-bit hamming space with (ideally) compact hash codes, the inter-class hamming distance follows a Binomial Distribution of $\mathcal{X} \sim B(K, p)$, where $p = \frac{C}{2(C-1)}$ and $p \approx \frac{1}{2}$ when C is large.

Proof. For all C classes, consider only one bit location at a time. The maximum hamming distance is achieved when there is an equal number of $\frac{C}{2}$, $\{+1\}$ and $\{-1\}$ codes among the C classes. The hamming distance between two bits is either 0 or 1. Thus, among $\binom{C}{2}$ selection of pairs, the probability that the hamming distance equals to 1 is $(\frac{C}{2} \cdot \frac{C}{2})/\binom{C}{2} = \frac{C}{2(C-1)}$. Since all K bits can be selected independently, the probability of the inter-class hamming distance between h_i and h_j equals to d is,

$$Pr\big(D_h(h_i, h_j) = d\big) = \binom{K}{d} p^d (1-p)^{K-d}, \quad p = \frac{C}{2(C-1)}, \tag{6}$$

in which mean value is Kp and variance is $Kp(1-p)$.

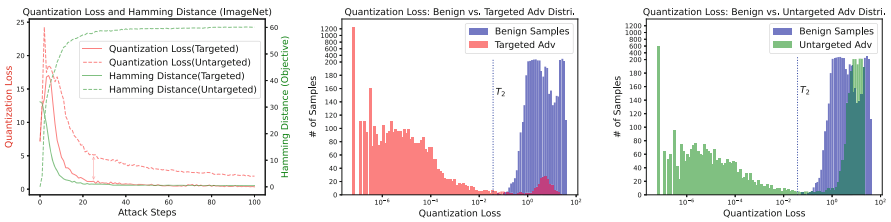
From *Lemma* 1, we can further deduce the next theorem.

Theorem 1. For untargeted attacks, the hamming distance from the adversarial image to any other classes follows a Gaussian distribution $\mathcal{N} \sim (K(1-p), Kp(1-p))$.

Proof. In the ideal situation, the untargeted attack maximizes the hamming distance from $D_h(h(x'), h(x_i))$ to K. Thus, for any other hash codes $h(x_j)$, the hamming distance is $D_h(h(x'), h(x_j)) = K - D_h(h_i, h_j)$, which is also a Binomial distribution with the mean of $K(1-p)$ and the same variance. When the hash bits K is a large value, it can be approximated by a Gaussian distribution $\mathcal{N} \sim (K(1-p), Kp(1-p))$ [34].

Example. When $K = 64$ bits, and C is large ($p \to \frac{1}{2}$), using the three-sigma rule, the confidence interval is $(K(1-p) - 3\sqrt{Kp(1-p)}, K(1-p) + 3\sqrt{Kp(1-p)})$.

In other words, there is 99.73% confidence that the hamming distance from an untargeted adversarial image to any other classes would be within the $[20, 44]$ hamming distance interval with the mean of $K/2 = 32$, which is sufficiently distinguishable in the hamming space. Note that the above analysis serves a theoretical upper bound because achieving Assumption 1 is still an ongoing effort [20,46]. To see some examples, we visualize the t-SNE of untargeted adversarial images vs. benign images on CIFAR-10 and MS-COCO in Fig. 3. It is observed that despite of a few samples, the majority of the adversarial images are sufficiently distinguishable based on hamming distance. Hence, we formalize the first detection criterion.



(a) (b) (c)

Fig. 4. Example of identifying targeted attacks based on quantization loss on ImageNet. (a) Targeted attacks push the quantization loss to zero compared to untargeted attacks. (b) The quantization loss for targeted attacks concentrates around zero vs. the benign samples. (c) 60% of the untargeted attacks also concentrate around zero.

Criterion 1 (Hamming Distance). For query x, collect the set of top-k hash codes $\mathcal{H}_k$ and calculate the average hamming distance to $h(x)$.

$$C_1 = \frac{1}{|\mathcal{H}_k|} \sum_{h(x_k) \in \mathcal{H}_k} D_h\big(h(x_k), h(x)\big) \tag{7}$$

C_1 is the average hamming distance of the top-k retrieval results, i.e., a scalar value and we can compare it with a threshold $\mathcal{T}_1$ calculated on benign samples. The computational process of C_1 follows the normal retrieval procedures using the top-k hash codes. To detect targeted attacks, we develop the next criterion.

3.2 Detecting Targeted Attacks (C_2)

While untargeted attacks attempt to induce a bit flip that makes $h(x') = -h(x)$, targeted attacks minimize the hamming distance between $h(x')$ and an arbitrary target code h_t (e.g., such as computed from consensus voting [6] of a category). To find an appropriate metric to identify them, we have the following observation.

Observation 1. For the quantization loss of benign images $\mathcal{L}_Q^b$ and targeted images $\mathcal{L}_Q^t$, the relation $\mathcal{L}_Q^b > \mathcal{L}_Q^t \approx 0$ holds.

To illustrate this observation, recall that the original targeted objective in Eq. (5) is not differentiable regarding the targeted binary code of x'. The implementation approximates via a continuous relaxation and the goal is to minimize the distance between the continuous output from the $\tanh(\cdot)$ function and the target code [8]. As more gradient descent steps are taken, the quantization loss $\mathcal{L}_Q^t \to 0$, when their inter-distance is minimized. This is in close analogy with the adversarial images on softmax classifications while the targeted probabilities become overconfident [23], and surprisingly, similar phenomenon is reflected on the quantization loss in deep hashing. In contrast, for all the benign samples, it is difficult to find the optimal model parameters to push $\mathcal{L}_Q^b$ towards zero during the training process. An example of ImageNet is shown in Fig. 4(a) as the targeted attacks leave a distinguishable gap from benign samples. It is also interesting to compare with the quantization loss of untargeted attacks in Fig. 4(a), which is larger than zero. This is because for untargeted attacks, finding an adversarial subspace that reduces the mAP to zero can be achieved before flipping all the bits, which is much easier than targeted attacks. Based on these observations, we develop the second detection criterion.

Criterion 2 (Quantization Loss). Calculate the $\boldsymbol{L}_p$ distance from the output of network $f_\theta(x)$ (logits before the sign function) and its hash code $h(x)$,

$$C_2 = \|h(x) - f_\theta(x)\|_p \tag{8}$$

C_2 is the quantization loss of query x. Here, we use the $\boldsymbol{L}_1$ distance ($p = 1$) and obtain a threshold $\mathcal{T}_2$ on benign samples offline. Figures 4(b)(c) show the distribution of quantization loss between the adversarial and the benign images. As $C_2 \to 0$ for targeted attacks, we can see that using $\mathcal{T}_2$ can effectively identify most of the attacks; using C_2 also identifies about 60% of the untargeted attacks.

3.3 Detecting Prediction Inconsistency (C_3)

C_1 and C_2 alone are not sufficient. In principle, detection works by limiting the attacker's action space in a confined region. Perturbation can be generally treated as an artificial noise with high-frequency components [39]. Thus, a common approach is to apply local, non-local smoothing filters [44], auto-encoder denoiser [30], color bit reduction [44], quantization [25], and measure the response sensitivity to the denoised images. The adversarial images are more prone to produce a different result, while the benign samples are less sensitive. These denoising operations reduce the entropy (randomness) and the input dimensions of adversarial space that the perturbations can act upon.

We extend this principle in deep hashing to formulate Criterion 3. Denote the transformation [25,30,44] as $t(\cdot)$. For query x, C_3 measures the hamming distance between a transformed $t(x)$ and x based on the output before the sign function.

Criterion 3 (Prediction Inconsistency)

$$C_3 = D_h\big(f_\theta(t(x)), f_\theta(x)\big) \tag{9}$$

In other words, C_3 quantifies the disagreement between the original and transformed inputs, which can be evaluated against a threshold $\mathcal{T}_3$ calculated offline on benign samples.

3.4 Put Everything Together

The overall detection combines the three criteria: given a query image x, we calculate $\{C_1, C_2, C_3\}$ and compare with thresholds $\{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3\}$. If (a) $C_1 < \mathcal{T}_1$; (b) $C_2 > \mathcal{T}_2$; (c) $C_3 < \mathcal{T}_3$, the input is considered as benign; otherwise, if any of them is not satisfied, the input is rejected as an adversarial example. The computation time is bounded by C_3 since it requires two retrievals. To minimize the compute time, the system can combine the original query and its denoised copy into a batched query. In case the GPU has sufficient resources, it should have minimum overhead as discussed in Sect. 4.5.

4 Experiments

4.1 Implementation

We evaluate our mechanism on the CIFAR-10, ImageNet, MS-COCO and NUSWIDE datasets that are commonly used for deep hashing [7,8,26,27,46,48] and adopt CSQ [46] with ResNet50 [18] as the base model. The RMSProp optimizer [37] with learning rate 10^{-5} is used for training of 150 epochs. The weight of the quantization loss is set to 10^{-4}. For four datasets, our trained models achieve mAP of 0.854, 0.883, 0.884, and 0.843, respectively.

We compare with several benchmarks originally designed for softmax classification: Local Intrinsic Dimensionality (LID) [28], Median Smoothing (FS-Median), Non-local Means (FS-NLM) [44], FS-Adaptive [25], and MeanBlur [23]. FS-Adaptive uses the entropy of input as a metric to adaptively reduce the input space using scalar quantization and smoothing spatial filter. We select MeanBlur as the denoising technique for our method. True Positive Rate (TPR), False Negative Rate (FNR, Miss Rate), and Area-Under-Curve (AUC) are used as the evaluation metric, where adversarial examples are considered as Positive and benign samples are considered as Negative. Thus, a detected adversarial example is counted as a true positive, while a misidentified benign sample is counted as false positive.

All detection methods are evaluated against both untargeted [45] and targeted [6] deep hashing adversarial attacks (based on the PGD attack) and an untargeted deep hashing CW [10] attack[1]. The step sizes of the former two are set to 1.0 with 100 steps, limited by an $\boldsymbol{L}_\infty$ norm of perturbation $\epsilon = 8$. For CW attack, the learning rate is set to 0.01 with 500 steps.

[1] The CW attack is initially designed for softmax classification thus only a few logits are optimized. But for deep hashing, all hashing bits need to be optimized which rarely succeed for targeted CW attacks. Thus, we focus on the untargeted CW attack.

Table 1. A comparison with state-of-the-arts methods of *Detection Miss Rate* (False Negative Rate) against adversarial attacks for deep hashing, when allowing 5% FPR on benign samples. Lower miss rate is better. Top two numbers of each column are **bolded**, with the best in **red** and the second in **blue**.

		Untgt [45]	Tgt [6]	Untgt CW [10]
CIFAR-10	LID [28]	14.90	**4.00**	12.16
	FS-Median [44]	13.40	54.30	**2.40**
	FS-NLM [44]	18.80	18.10	14.41
	FS-Adaptive [25]	54.00	48.30	52.55
	MeanBlur [23]	**11.60**	26.90	**2.40**
	Ours	**7.50**	**0.40**	**0.15**
ImageNet	LID [28]	21.78	6.40	75.95
	FS-Median [44]	8.54	19.20	1.41
	FS-NLM [44]	2.62	**3.40**	8.02
	FS-Adaptive [25]	4.05	4.30	13.68
	MeanBlur [23]	**0.76**	4.54	**0.94**
	Ours	**0.42**	**0.34**	**0.47**
MS-COCO	LID [28]	3.24	**1.20**	54.59
	FS-Median [44]	0.26	7.10	**0.00**
	FS-NLM [44]	**0.00**	1.72	0.08
	FS-Adaptive [25]	**0.00**	2.84	0.08
	MeanBlur [23]	**0.00**	2.12	**0.00**
	Ours	**0.00**	**1.12**	**0.00**
NUSWIDE	LID [28]	25.34	**0.08**	47.96
	FS-Median [44]	3.52	16.33	**0.00**
	FS-NLM [44]	**0.00**	2.90	**0.00**
	FS-Adaptive [25]	**0.00**	4.57	**0.00**
	MeanBlur [23]	0.19	4.29	**0.00**
	Ours	**0.00**	**1.29**	**0.00**

4.2 Detection of Gray-Box Attacks

Our Method. Table 1 shows the detection miss rate of different methods when we fix the FPR at 5%. The proposed method can make robust detection of targeted attacks with less than 1% miss rate in the most cases. Compared to the SOTA benchmarks, our method is 2.13% to 23.44% better than other methods on average. Untargeted attacks are relatively easier to detect, so the overall miss rates are lower than targeted attacks. Our method can still improve the baselines by 1.15%–14.33%.

Compare with LID. Although LID performs well on most of the targeted attacks, it does not generalize to the CW attack. Furthermore, different from

the softmax networks that the last few layers often offer better detection [28], LID is quite sensitive to which layers should those features be extracted in deep hashing, which adds extra configuration overhead. In sum, LID is not always effective against all types of attacks.

Compare with Spatial Denoising Methods. LID does not take advantage of the spatial information like the rest four benchmarks using non-local mean, median, etc. As shown in Table 1, though they generally have less than 10% miss rates on untargeted attacks, there is a 0.5–17.8% gap in detecting targeted attacks. Such gaps can be explained by the attack mechanisms as targeted attacks take more gradient steps. This lands the image deep into the adversarial space, which is more robust to pixel-level modifications such as denoising [21,42]. However, our method provides an extra layer of defense from C_2 to specifically monitor the value of quantization loss as targeted attacks bring it to zero, thereby complementing C_3 when targeted attacks push the inputs deep into the adversarial space.

Table 2. Ablation study: detection rates of different combinations ($\epsilon = 32$)

	CIFAR-10		ImageNet		MS-COCO		NUSWIDE		*Avg. Gain*	
	Untgt	Tgt	Untgt	Tgt	Untgt	Tgt	Untgt	Tgt	Untgt	Tgt
C_3 Alone	0.8160	0.7460	0.9110	0.8338	0.9954	0.9076	0.9757	0.8904	–	–
$C_1 + C_3$	0.9870	0.7580	0.9522	0.8460	0.9966	0.9098	0.9757	0.8904	*0.0533*	*0.0066*
$C_2 + C_3$	0.8170	0.9830	0.9504	0.9828	0.9992	0.9784	0.9961	0.9847	*0.0161*	*0.1377*
$C_1 + C_2 + C_3$	0.9880	0.9950	0.9916	0.9956	0.9992	0.9784	0.9961	0.9847	*0.0692*	*0.1439*

4.3 Detection of White-Box Attacks

Next, we demonstrate the detection of white-box attackers, who know the existence of our detection and conduct countermeasures accordingly. The attacker adopts *backward pass differentiable approximation* [5] to estimate the gradients and develops different strategies against C_1, C_2 and C_3:

Against C_1. C_1 relies on the hamming distance between hash codes to detect outliers. Thus, an effective evasion is to drive the adversarial examples into the neighborhoods of benign images, e.g., generating the same binary hash codes of certain targeted images h_t:

$$\mathcal{L}_1 = \underbrace{D_h(f_\theta(x'), h_t)}_{\text{adv loss}} \tag{10}$$

Against C_2. C_2 detects near-zero quantization loss by accessing the logits before the sign function. To bypass this detection, the attacker aims to maximize the quantization loss, which amortizes the adversarial behavior identified from C_2.

$$\mathcal{L}_2 = -\underbrace{\|h(x') - f_\theta(x')\|_1}_{\text{quantization loss}} \tag{11}$$

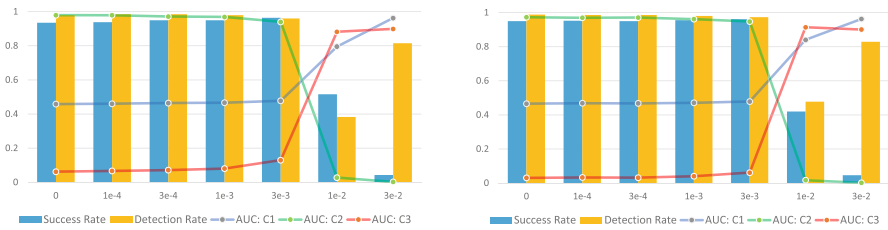
Against C_3. C_3 detects the disagreement between $f_\theta(x')$ and the denoised copy $f_\theta(t(x'))$. The attacker minimizes such difference by enforcing distance between $f_\theta(x')$ and $f_\theta(t(x'))$ to be small,

$$\mathcal{L}_3 = \underbrace{D_h(f_\theta(t(x')), f_\theta(x'))}_{\text{denoised adv loss}} \tag{12}$$

We use MeanBlur [23] as the transformation $t(\cdot)$ here. By combining them, the white-box attacker constructs a joint optimization objective,

$$\min_{x'} \mathcal{L} = \mathcal{L}_1 + \lambda_1 \mathcal{L}_2 + \lambda_2 \mathcal{L}_3 \tag{13}$$

Detection of White-Box Attacks. Optimizing (13) turns out to be quite difficult by finding λ_1, λ_2 that breach all three criteria. We demonstrate the best effort in Fig. 5 that fixes λ_2 to 0.3 and 3 and adjusts λ_1 from 0 to 0.03. The best case is when λ_1=0.01, the detection rate has been lowered to around 0.4 (the yellow bars). However, the number of adversarial examples that can successfully optimize (13) is also decreased to around 45% (the blue bars of success rate). Hence, although the strongest white-box attackers still have some chances, our detection has successfully confined the adversarial space by enlarging the attacker's efforts. It is also interesting to see that different criteria form compensating relations as indicated by the AUC values. When C_2 (green curve) declines, C_1, C_3 quickly rise and vice versa. This relation is further validated by the ablation study next.



(a) $\lambda_2 = 0.3$ (b) $\lambda_2 = 3$

Fig. 5. White-box attack results. x-axis is the λ_1 value and y-axis is the percentage. Once the white-box attacks achieve a lower detection rate at λ_1 =1e-2, the success rate of generating such adversarial examples also drops significantly. The three detection criteria compensate each other by observing the AUC values: when C_1, C_3 are low, C_2 is high and vice versa. (Color figure online)

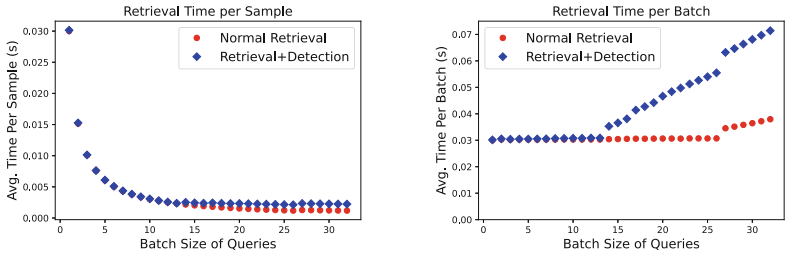


Fig. 6. Computation time of different batch size: a) per sample; b) per batch.

4.4 Ablation Study

We present the ablation study to quantify the contribution of each criterion. We use C_3 alone as the baseline and add C_1 and C_2 with their averaged gain shown in Table 2. The result is consistent with the defense objectives as the addition of C_1 and C_2 helps improve the detection rates of the untargeted and targeted attacks by **0.0533** and **0.1377**, respectively. Meanwhile, C_1 and C_2 contribute almost independently on the overall detection, e.g., $C_1 + C_3 = \mathbf{0.0533}$ plus $C_2 + C_3 = \mathbf{0.0161}$ is equal to $C_1 + C_2 + C_3 = \mathbf{0.0692}$ for untargeted attacks and the same also holds for targeted attacks. This validates that all three criteria act as indispensable parts in the detection.

4.5 Computational Time

Finally, we evaluate the computational overhead of the detection mechanism. In practice, the system can accumulate queries into a batch to enhance the utilization of GPU resources and reduce cost. Figure 6 shows the average retrieval time per sample/batch. First, it is observed that the average time per sample is under 50 ms and further reduced as we increase the batch size. Once the batch size is small, detection introduces negligible overhead because the GPU is underutilized; as the batch size increases, an additional retrieval from the denoised copy in C_3 enlarges the gap between normal retrieval since the GPU resources have been fully utilized. Thus, our detection introduces minimum overhead when the system accumulates relatively small batch and responds to queries in real-time.

4.6 Variants of White-Box Attacks

The white-box case is always a challenging open problem because the attacker can devise different variants after knowing the system details. Here we consider a variation of the attack by slightly modify $\mathcal{L}_3$ while keeping $\mathcal{L}_1$ and $\mathcal{L}_2$ the same. Recall that C_3 detects the disagreement between $f_\theta(x')$ and the denoised copy $f_\theta(t(x'))$, and the attacker aims to directly minimize such difference. When this objective becomes difficult to optimize, the attacker can resort to an anchor

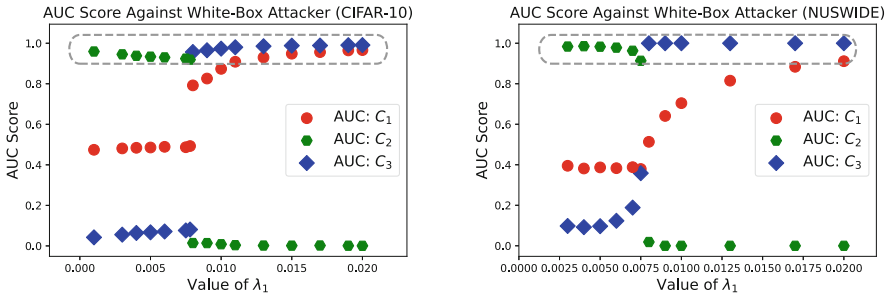


Fig. 7. AUC Scores from different criteria by tuning λ_1 against white-box attackers: a) CIFAR-10; b) NUSWIDE.

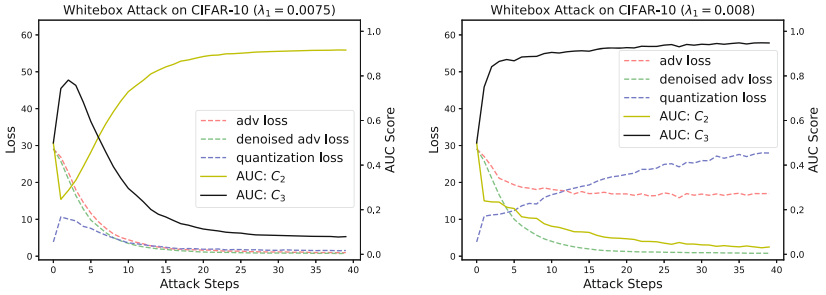


Fig. 8. Trace of the white-box attack process of sub-objectives between two cases on the threshold (a) $\lambda_1 = 0.0075$. (b) $\lambda_2 = 0.008$.

hash code h_a and bring $f_\theta(x')$ and $f_\theta(t(x'))$ close to h_a. Hence, $\mathcal{L}_3$ is replaced by $\mathcal{L}'_3$,

$$\mathcal{L}'_3 = D_h(f_\theta(t(x')), h_a). \tag{14}$$

By combining them, the white-box attacker constructs a joint optimization objective,

$$\min_{x'} \mathcal{L} = \mathcal{L}_1 + \lambda_1 \mathcal{L}_2 + \lambda_2 \mathcal{L}'_3 \tag{15}$$

For λ_1, a larger value penalizes C_2 more, and vice versa. To reduce the searching effort, we set λ_2 to 1 since $\mathcal{L}_1$ and $\mathcal{L}'_3$ carry equal importance in the evaluation.

Figure 7 demonstrates the AUC scores with different λ_1 values of CIFAR-10 and NUSWIDE datasets. As highlighted, for a fixed λ_1, there is at least one of the criteria with high AUC scores (larger than 0.9), which guarantees our method is robust against white-box attackers. E.g., in CIFAR-10, when λ_1 increases from 0.0075 to 0.008, C_2 drops from 0.9 to near zero; however, C_3 quickly gaps up above 0.9 to compensate the weaknesses from C_2, and C_1 also increases. This phenomenon is the same as we observed earlier as it indicates that the white-box attacker can barely optimize C_2 and C_3 at the same time, i.e., it is difficult

to find an adversarial space that satisfies both: 1) its quantization loss is much larger than zero; 2) the adversarial example and its denoised copy that yield the same hash code.

Recall that the attacker's goal is to minimize both *adv loss* and *denoised adv loss*, but to maximize *quantization loss*. Figure 8 further demonstrates the trace of the optimization process for each sub-objective in Eq. (15) when λ_1 changes from 0.0075 to 0.008 (the threshold from above). When $\lambda_1 = 0.0075$, the *quantization loss* is not being maximized compared to *adv* and *denoised adv loss*. Hence, we see that white-box attacker is detected by C_2. When λ_1 increases to 0.008, the optimization starts to weigh more on the *quantization loss* but the original *adv loss* is no longer minimal, which leads to successful detection from C_3 instead.

Table 3. Detection rate (TPR) of the adversarial examples against white-box attacks ($\epsilon = 32$, $\lambda_1 = 0.0075$)

Detector	FPR	CIFAR10	ImageNet	MS-COCO	NUSWIDE
MeanBlur [23]	0.1	0.0340	0.0218	0.0044	0.0057
	0.2	0.0480	0.0250	0.0076	0.0076
UMCD(Ours)	0.1	0.9560	0.8824	0.4648	0.6204
	0.2	0.9790	0.9180	0.6742	0.7900

We present additional results of the white-box attacks. Since the attacker need to design specific objectives regarding the detection methods, we choose MeanBlur [23] as a benchmark for comparison. The detection rate is True Positive Rate (TPR) with a certain threshold calculated by False Positive Rate (FPR) on benign samples. We use the targeted attack with $||\epsilon||_\infty = 32$. For the best success rate and ability to bypass the detection, we choose $\lambda_1 = 0.0075$. In Table 3, we show the detection results on four datasets evaluated by TPR. Given FPR= 0.1 and FPR= 0.2, our method achieves an average of 0.7309 and 0.8403 white-box detection rates, whereas the detection rates for MeanBlur is close to zero. This further validates that our method is robust against the strongest white-box attackers. It is also interesting to see that both methods perform relatively better on single-label datasets (CIFAR-10 and ImageNet) than multi-label datasets (MS-COCO and NUSWIDE), which is different from the gray-box attacks. The reason behind that need to be further studied in the future.

5 Related Work

5.1 Deep Hashing

Image retrieval uses nearest neighbor search to return the semantically related images of query inputs. Traditionally, it relies on hand-crafted visual descriptors

to reduce the computational cost of similarity measure [14,31]. Powered by deep learning, end-to-end hash learning improves the performance to a new level [7,8,26,27,46,48]. They use the similarities between image pairs to train deep hashing models in a supervised manner by transforming the high-dimensional images into compact hash codes, on which neighboring search can be efficiently performed based on hamming distance. To convert the continuous outputs into discrete binary codes, common approaches use continuous relaxation such as sigmoid or hyperbolic tangent functions to approximate the discrete binary thresholding [7,8,26,27,46,48]. Our work exploits the adversarial behaviors originated from this approximation process, thus can be applied to a variety of deep hashing models.

5.2 Adversarial Attacks

Deep neural networks are known to be vulnerable to the non-perceptible perturbations [36]. The Fast Gradient Sign Method (FGSM) [16] generates perturbations in the direction of the signed gradient to maximize the loss function in one-shot computation. The Basic Iterative Method (BIM) [22] and Projected Gradient Descent (PGD) [29] take iterative steps (from random initialization) to achieve higher attack success. There are several other variants [5,10,32,38], e.g., the CW attack aims at minimizing the perturbations to evade detection.

By using the deep learning backends, deep hashing inherits the vulnerability from neural networks. With some slight adaptation, recent works have shown that adversarial attacks can also mislead image retrieval systems [6,40,42,43,45]. The attacks can be generally categorized into *untargeted* and *targeted attacks*. *Untargeted attacks* divert the query away from the correct results, which make the system retrieve irrelevant images or simply nothing. [45] proposes an untargeted attack to maximize the hamming distance between adversarial and benign samples. [12,24] craft adversarial examples based on iterative retrievals from a black-box model. [43] hides private images in the database into a non-retrievable subspace by minimizing the number of samples around the private images. *Targeted attacks* make the systems return images from a targeted category, different from the inputs. [6,40] minimize the average hamming distance between the adversarial examples and a set of images with a target label. [42] enhances targeted transferability to a black-box model via injecting random noise into the adversarial generation. Our work defends against both untargeted and targeted attacks in deep hashing.

5.3 Adversarial Defenses

Most of the defense mechanisms are based on softmax classification. As proactive measures, gradient masking [33] and adversarial training [13,29,35,41] learns a robust model. The early defense of [33] starts with an incorrect conjecture that ascribes adversarial example to high nonlinearity/overfitting, and develops defensive distillation to reduce the variations around input. The method is quickly subverted by [5,9,10] as argued in [16] that the primary cause is due to local linearity of neural networks instead.

Hence, a large body of works focus on adversarial training [13,29,35,41] by solving a min-max saddle point problem. However, it is non-trivial to tackle the trade-off between robustness and accuracy [47], which often leads to significant loss on clean image accuracy, with extensive training efforts. Applying adversarial training into the deep hashing domain could suffer from even higher accuracy loss. For image retrieval system, as long as the adversarial images are detected at the input, we can equivalently thwart the attacks without accuracy loss and training complexities.

Adversarial detections extract the artifacts left by the adversarial examples at different levels: raw pixels [15,17], feature distributions [17,23], softmax distributions [19] and frequency components [39]. By analyzing the contrastive distributions of the adversarial and natural images, a detector can be efficiently trained in a supervised or unsupervised manner. Another thread of works rely on the prediction inconsistency by exploiting denoise method and measuring the disagreement between the results [25,30,44]. All these works are based on softmax classification. In this work, we discover adversarial behaviors from the hamming space and propose a set of detection criteria including defending against the strongest white-box attackers.

6 Conclusion

In this paper, we propose an efficient, unsupervised detection of adversarial examples in deep hashing based image retrieval. We design three criteria to identify adversarial behaviors of both targeted and untargeted attacks in the hamming space and consider white-box attackers who are aware of the existence of the defense. The extensive evaluations demonstrate that the proposed detection can surpass previous defense techniques by a large margin and is also robust against white-box attacker by limiting its action space.

References

1. Alibaba's pailitao. http://www.pailitao.com. Accessed 03 Feb 2022
2. Bing image search. https://www.bing.com. Accessed 03 Feb 2022
3. Google image search. https://www.google.com/imghp. Accessed 03 Feb 2022
4. Pinterest visual search tool. https://www.pinterest.com. Accessed 03 Feb 2022
5. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. In: International Conference on Machine Learning, pp. 274–283. PMLR (2018)
6. Bai, J., et al.: Targeted attack for deep hashing based retrieval. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 618–634. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_36
7. 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, pp. 1229–1237 (2018)
8. 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, pp. 5608–5617 (2017)

9. Carlini, N., Wagner, D.: Defensive distillation is not robust to adversarial examples. arXiv preprint arXiv:1607.04311 (2016)
10. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 39–57. IEEE (2017)
11. Carrara, F., Becarelli, R., Caldelli, R., Falchi, F., Amato, G.: Adversarial examples detection in features distance spaces. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11130, pp. 313–327. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11012-3_26
12. Chen, M., Lu, J., Wang, Y., Qin, J., Wang, W.: DAIR: a query-efficient decision-based attack on image retrieval systems. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1064–1073 (2021)
13. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: International Conference on Machine Learning, pp. 2206–2216. PMLR (2020)
14. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 1, pp. 886–893. IEEE (2005)
15. Gong, Z., Wang, W., Ku, W.-S.: Adversarial and clean data are not twins. arXiv preprint arXiv:1704.04960 (2017)
16. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
17. Grosse, K., Manoharan, P., Papernot, N., Backes, M., McDaniel, P.: On the (statistical) detection of adversarial examples. arXiv preprint arXiv:1702.06280 (2017)
18. 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, pp. 770–778 (2016)
19. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv preprint arXiv:1610.02136 (2016)
20. Hoe, J.T., Ng, K.W., Zhang, T., Chan, C.S., Song, Y.-Z., Xiang, T.: One loss for all: deep hashing with a single cosine similarity based learning objective. In: Advances in Neural Information Processing Systems, vol. 34 (2021)
21. Hu, S., Yu, T., Guo, C., Chao, W.-L., Weinberger, K.Q.: A new defense against adversarial images: turning a weakness into a strength. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
22. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533 (2016)
23. Li, X., Li, F.: Adversarial examples detection in deep networks with convolutional filter statistics. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5764–5772 (2017)
24. Li, X., et al.: QAIR: practical query-efficient black-box attacks for image retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3330–3339 (2021)
25. Liang, B., Li, H., Miaoqiang, S., Li, X., Shi, W., Wang, X.: Detecting adversarial image examples in deep neural networks with adaptive noise reduction. IEEE Trans. Dependable Secure Comput. **18**(1), 72–85 (2018)
26. Lin, K., Yang, H.-F., Hsiao, J.-H., Chen, C.-S.: Deep learning of binary hash codes for fast image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 27–35 (2015)

27. 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, pp. 2064–2072 (2016)
28. Ma, X., et al.: Characterizing adversarial subspaces using local intrinsic dimensionality. arXiv preprint arXiv:1801.02613 (2018)
29. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018)
30. Meng, D., Chen, H.: Magnet: a two-pronged defense against adversarial examples. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 135–147 (2017)
31. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. Int. J. Comput. Vis. **42**(3), 145–175 (2001)
32. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pp. 506–519 (2017)
33. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE Symposium on Security and Privacy (SP), pp. 582–597. IEEE (2016)
34. Ross, S.M.: A First Course in Probability, vol. 6. Prentice Hall, Upper Saddle River (2002)
35. Shafahi, A., et al.: Adversarial training for free! arXiv preprint arXiv:1904.12843 (2019)
36. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
37. Tieleman, T., Hinton, G., et al.: Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA: Neural Netw. Mach. Learn. **4**(2), 26–31 (2012)
38. Tramèr, F., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: The space of transferable adversarial examples. arXiv preprint arXiv:1704.03453 (2017)
39. Wang, H., Wu, X., Huang, Z., Xing, E.P.: High-frequency component helps explain the generalization of convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8684–8694 (2020)
40. 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, pp. 2298–2302 (2021)
41. Wong, E., Rice, L., Zico Kolter, J.: Fast is better than free: Revisiting adversarial training. arXiv preprint arXiv:2001.03994 (2020)
42. 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, pp. 1934–1943 (2021)
43. 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, pp. 9651–9660 (2020)
44. Xu, W., Evans, D., Qi, Y.: Feature squeezing: detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155 (2017)
45. Yang, E., Liu, T., Deng, C., Tao, D.: Adversarial examples for hamming space search. IEEE Trans. Cybern. **50**, 1473–1484 (2018)

46. Yuan, L., et al.: Central similarity quantization for efficient image and video retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3083–3092 (2020)
47. Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., Jordan, M.: Theoretically principled trade-off between robustness and accuracy. In: International Conference on Machine Learning, pp. 7472–7482. PMLR (2019)
48. Zhu, H., Long, M., Wang, J., Cao, Y.: Deep hashing network for efficient similarity retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30 (2016)