Evade Deep Image Retrieval by Stashing Private Images in the Hash Space

Yanru Xiao
Old Dominion University
Norfolk, VA
yxiao002@odu.edu

Cong Wang*
Old Dominion University
Norfolk, VA
c1wang@odu.edu

Xing Gao
University of Memphis
Memphis, TN
xgao1@memphis.edu

Abstract

With the rapid growth of visual content, deep learning to hash is gaining popularity in the image retrieval community recently. Although it greatly facilitates search efficiency, privacy is also at risks when images on the web are retrieved at a large scale and exploited as a rich mine of personal information. An adversary can extract private images by querying similar images from the targeted category for any usable model. Existing methods based on image processing preserve privacy at a sacrifice of perceptual quality. In this paper, we propose a new mechanism based on adversarial examples to "stash" private images in the deep hash space while maintaining perceptual similarity. We first find that a simple approach of hamming distance maximization is not robust against brute-force adversaries. Then we develop a new loss function by maximizing the hamming distance to not only the original category, but also the centers from all the classes, partitioned into clusters of various sizes. The extensive experiment shows that the proposed defense can harden the attacker's efforts by 2-7 orders of magnitude, without significant increase of computational overhead and perceptual degradation. We also demonstrate 30-60% transferability in hash space with a black-box setting. The code is available at: `https://github.com/sugarruy/hashstash`

1. Introduction

A picture is worth a thousand words. The rapid growth of large image and video collections has made content-based image retrieval possible at a large scale, e.g. Google [1], Pinterest [2], Bing [3] and TinEye [4]. Powered by deep learning, they have been increasingly built into social networks [5], e-commerce [6, 7](e.g., Pailitao from Taobao [8]) and fashion design [9] to capture semantic similarities from visual queries for finer results. Social media, e-commerce websites and even the user's query are utilized as a rich mine of images to train these systems. For instance, 100M photos and videos are uploaded everyday on Instagram [10]; more than 1G products are listed on Ebay [11]. Google also claims a 7-day storage of queried and uploaded images and utilizes them for further analysis [12].

Although legislation (e.g. GDPR [13]) imposes restrictions on the usage of personal data, for the exploding volume of visual content, there still remains a vague definition of ownership as well as a weak legal boundary between what can be learned and what cannot, from an image. Further, users' awareness of their privacy remains quite subjective towards latent, but sensitive information in their images. The resourceful visual content can be exploited in bewildering ways to learn private information such as family member, location, income, personal interest or even sexual orientation for accurate contextual advertising [14–17] or spear phishing [18]. For example, Facebook has patented a new application of predicting household demographics based on image data [19]. Though these applications expedite search efficiency and product offering, they also compromise user privacy and make privacy trampling easier at a large scale. These issues stretch beyond social media and search engines: any platform with content-based image retrieval shares the same risk of privacy leakage.

Unfortunately, there is less incentive for the platforms to implement privacy guarantees, as long as they are faltering in the grey area of legislation. It is always up to the users to protect their own privacy. Previous approaches utilize image processing such as blurring, darkening and occlusion to evade face recognition [20] or disassociate friend tagging [21], at a sacrifice of degraded visual quality. Another thread of work is to establish a privacy-respecting protocol by an identifiable tag [22, 23], so anyone wearing the privacy tag is excluded from the image. The success of these systems relies on building sophisticated, trusted protocols between the users and the platform, that demands commitments from both sides.

These techniques may be fragile in the eyes of deep learning, which can still extract useful information from the local descriptors. The state-of-the-art image retrieval adopts *deep hashing* for efficient similarity search [24–28]. It quantizes images in the database into low-dimensional binary codes during training, computes the *hamming distance* from the queried image, and returns relevant images (inadvertently) gathered by the database. A well-trained model would return images with high similarity (usually from the same category). With some categorical information, e.g., gathering a few images from the targeted category, an ad-

*Corresponding author: Cong Wang, c1wang@odu.edu

versary can query the database and retrieve all the images including those private ones. Thus, to evade retrieval, privacy preservation entails opening the box of deep hashing while maintaining perceptual similarity.

In this paper, we aim to minimize the chances for the private images to be extracted by introducing a small, crafted perturbation on the original image. Studied in [29–32], deep neural networks are vulnerable to adversarial inputs - perturbations that are inconspicuous to human eyes can be added to cause misclassification. In principle, deep hashing should inherit these vulnerabilities by design. A recent work shows that maximizing the hamming distance from the original image in hash space would make the system return an irrelevant image to the query, which can be utilized directly to protect the private images. Nevertheless, by implementing the strategy, we find that it can only defend weak adversaries, who only exploit the original category. Strong adversaries are more common in reality; they could enumerate all the categories and expose the private images in brute force. To tackle this challenge, we propose a new *cluster-based weighted distance maximization* that can transform the hash code into the subspaces away from all the categories.

The main contributions are summarized below. First, we propose to utilize adversarial techniques for privacy preservation and identify the limitations of the existing approach against strong adversaries. Second, we develop a new mechanism to stash samples into the hash space that maximizes the hamming distance to all the classes, while maintaining perceptual similarity. Finally, we conduct experiments on various datasets and demonstrate that the proposed mechanism successfully hardens the attack efforts by 1-3 orders of magnitude compared to [33], while achieving minimal perceptual dissimilarity. We show that 30-60% of the protected images can successfully transfer to an unknown model in a blackbox setting.

The rest of the paper is organized as follows. Section 2 introduces the related works. Section 3 motivates this study by defining the threat model and identifying the limits of the existing approach. Section 4 presents a new defense against strong adversaries. Section 5 evaluates the proposed mechanism and Section 6 concludes this work.

2. Background and Related Works

2.1. Deep Image Retrieval

Traditional image retrieval works on a vector of handcrafted visual descriptors [34, 35], followed by a separate process of projection and quantization to encode feature vectors into binary codes. Propelled by the success of deep learning, the new deep image retrieval enables learning of pairwise similarity from end-to-end [24–27]. It transforms high-dimensional real-valued inputs into the binary hash codes so similarity search can be performed efficiently by calculating the *hamming distance*. These systems typically consist of a *database* and a *model*. The database contains a finite set of images as the retrieval results; the model accepts query and returns retrieved images. The objective is to learn a nonlinear hash function to map input $x \rightarrow h(x) \in \{-1, +1\}^m$ into an m-bit binary code. A typical range of m is between 16 to 128 depending on the application requirements, which is made less than the original image dimension.

In addition to the convolutional and densely connected layers, a hash layer is introduced for the binarization process, in order to mitigate the quantization error. It converts a continuous representation z into discrete hash code by the sign function $sgn(z)$. Since the sign function is not compatible with backpropagation due to non-smoothness, the key is to build a function for continuous approximation. For example, HashNet [26] adopts the hyperbolic tangent function, $sgn(z) = \lim_{\beta \rightarrow \infty} \tanh(\beta z)$. By tuning the scaling parameter β during the learning process, the function converges to the sign function when $\beta \rightarrow \infty$. Similar to deep features in their floating point format, hashing concentrates similar images into a Hamming ball. The system usually defines a *retrieval threshold* so any image with smaller hamming distance would be returned as the query results. We refer to the survey [28] for more details.

2.2. Adversarial Examples

In contrast to their super-human capabilities, neural networks are highly vulnerable to small perturbations, where purposely crafted perturbations added to the input can make the system misbehave at run-time [29–32]. An efficient attack is the *fast gradient sign method* [30]. It takes a large step in the gradient directions to maximize the loss function, by finding a perturbed image x' with small additive noise ϵ such that $f(x') \neq f(x)$.

$$x' = x + \epsilon \cdot sgn(\nabla_x L(\theta, x, y)), \tag{1}$$

where $L(\cdot)$ is the loss function. θ is the model parameter. ∇ is the gradient. x is the data and y is the true label. Instead of making one-step gradient ascent, the method is extended in [31] as the *basic iterative method* to apply (1) multiple times and clip the image within the ϵ-constraint. Empirical experiments demonstrate that these adversarial examples can not only "fool" the classifier, but also transfer between different models for black-box attacks [36, 37].

2.3. Privacy Preserving

Previous efforts of preserving privacy online mainly focus on web analytics [14, 15], mobile advertising [16, 17] and behavioral tracking [38, 39]. To balance privacy and utility, a popular approach is through differential privacy that introduces noise to answers so the service provider cannot detect the presence or absence of a user. Though these mechanisms offer provable foundations on a statistical basis, they are not specialized in protecting inference of a single record, such as the private image being retrieved from the database. Privacy is a growing concern with the wide adoption of deep learning based search methods. Only a

few works have utilized adversarial examples for privacy preservation. In [40], a strategy based on adversarial examples is developed to disable the object detections so it cannot identify objects at the first place. An adversarial technique is also developed in [33] to corrupt semantic relationships and make the retrieval system return irrelevant images. Our work extends [33] to tackle strong and adaptive adversaries.

3. Motivation

This section motivates the research by defining the threat model and investigating the mechanism in [33] as defense.

3.1. Threat Model

We first present the scenario and assumptions made in this paper. Platforms such as social networks and search engines usually collect user information including profile, email, IP address and most importantly, *pictures*. The platform has deployed a deep image retrieval system such as HashNet-ResNet50 [26] to match imagery content from visual queries for marketing purposes. For profit, the platform also opens an interface for third-party advertisers or data brokers (escalated by calling them *adversaries*) [2, 4], who can match and retrieve similar images from the database for accurate advertising [38, 39]. Since the service is rated per query, the platform does not impose any limit on the number of queries but the adversaries have a fixed amount of budget. Users (*defenders*) have no control over the privacy policy, therefore, they introduce perturbation to prevent personal images from being returned as the retrieval results. The flowchart is illustrated in Fig.1.

To maximize retrieval quality, the adversary collects a dataset (*attack set*) to resemble the database. Similarly, the user also collects a dataset to facilitate the generation of the perturbations. We assume both data sets are independent and identically distributed (i.i.d) with the training set. For simplicity, in this paper, it is implemented by random selections from the test set. As a first proof of concept in the hash space, we assume the user has complete knowledge about the model (white-box) as [32, 33], including the information of category, structure, parameters, hashing mechanism and loss function. Then we demonstrate the existence of black-box transferability of the proposed mechanism in hash space, when users estimate the model architecture and parameters at the best effort.

3.2. Hamming Distance Maximization as a Defense

The work of [33] fools the hash-based image retrieval system by adversarial examples, which can be also leveraged as a privacy-preserving technique. The objective is to maximize the distance between the perturbed image and the original one, such that the hamming distance exceeds the retrieval threshold for that category. More formally, it transforms x into x' by maximizing their hamming distance $\mathcal{D}_h(x, x')$. $\mathcal{D}_h(x, x')$ can be deduced from the inner product of the m-bit hash code [41],

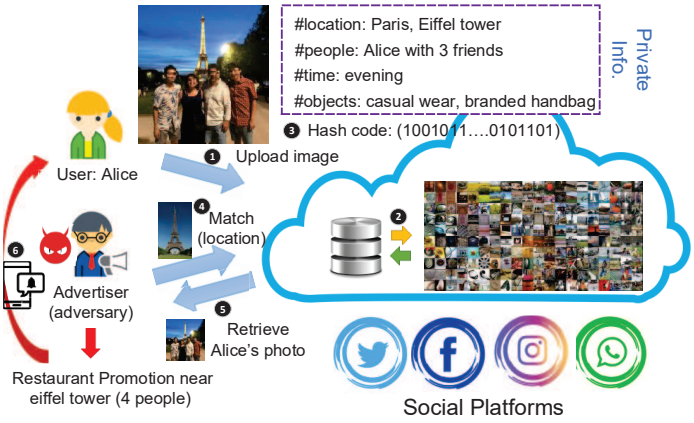


Figure 1: Illustration of the attack flow: ❶ user uploads a photo to the social platform; ❷ ❸ platform adds the photo into the database, generates hash code; ❹ ❺ advertiser matches the image via an identical query; ❻ advertiser exploits location privacy from the image and pushes nearby promotions onto the user's mobile (even though she has disabled location access on her phone).

$$\mathcal{D}_h(x, x') = \frac{1}{2}(m - h(x)h(x')^\top), h_i(x) \in \{1, -1\}^{1\times m} \tag{2}$$

where $i \in [1, m]$ and $m = 48$ bits for the HashNet-ResNet50 architecture. The goal is to adjust x' such that the hamming distance is maximized, $\max_{x'} \mathcal{L}(x', x) = -\frac{1}{m}h(x)h(x')^\top$. The problem can be re-written into a least-square style minimization function [33, 41], and shift the negative hash code by $+1$ to $\{0, 2\}$. The ϵ-constraint maintains the perceptual similarity between x and x'.

$$\min_{x'} \mathcal{L}_h(x', x) = \|\frac{1}{m}h(x)h(x')^\top + 1\|_2^2, \tag{3}$$

$$s.t.\ |x - x'| < \epsilon. \tag{4}$$

Though effective against trivial queries targeting at the *original category* of the protected image, the defense is vulnerable when the adversary enumerates through the rest categories and extracts the protected image by brute force. This is because simple maximization of hamming distance from the original image may unwittingly push the perturbed image into the vicinity of other categories. Fig.2 visualizes such cases in t-SNE on the MNIST dataset. As observed, simply hiding the private images into the subspaces of some irrelevant categories is still susceptible to stronger and adaptive adversaries. To gain more insights, we present some preliminary results based on MNIST [42] and CIFAR10 [43] in Fig. 3.

3.3. Key Observations

The adversary could expose all private images by enumerating through the entire attack set. Since the adversary is budget-limited, he wants to minimize such effort. Thus,

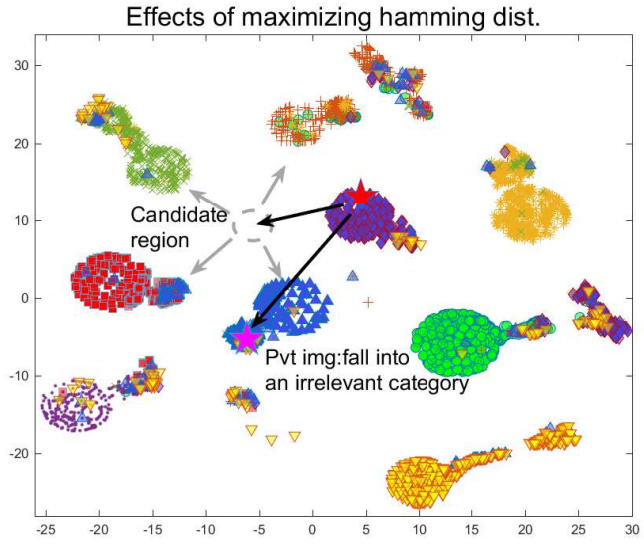
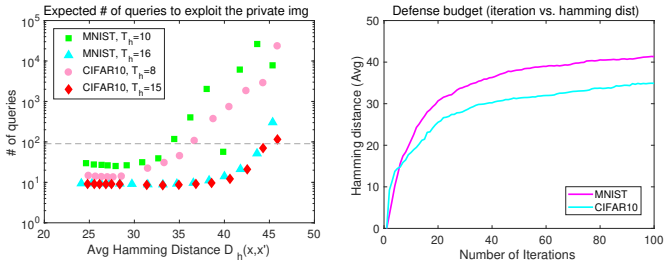


Figure 2: t-SNE visualization of learned hash codes from MNIST: hamming distance maximization has (accidentally) driven the private image into an irrelevant category.

we evaluate the average number of queries for the adversary to extract the private images, when a random image is queried from the attack set each time. If a private image is mapped to the vicinity of n images in the attack set of size N, the probability of retrieving this image is n/N. The expected number of queries is N/n.

Fig.3 shows the expected number of queries against strong attackers and the defense efforts in terms of iterations to generate the crafted perturbations [33]. The retrieval threshold T_h is selected according to the best F-1 score and precision.



(a) (b)

Figure 3: Brute-force attacks against [33] as a defense (a) expected number of queries to extract private images; (b) defense budget (# iterations).

Observation 1. The attack efforts trend up parabolically with the increasing hamming distance between x and x'. However, a strong adversary can still extract the private images within 100 queries for most of the hamming distances.

Observation 2. The average hamming distance is difficult to maximize further after a certain number of iterations. For example, its average saturates around 40 and 35 after 100 iterations on MNIST and CIFAR10 as observed in Fig.3(b), leaving a large gap to the total hash bits of $m = 48$.

Observation 3. When the categorical features are more dispersed in the hamming space, the protected image is more prone to fall into the retrieval threshold of some samples. It is validated in Fig. 3 given that the attacks on CIFAR10 require less effort than MNIST, due to higher intra-class diversity of CIFAR10. This makes the defense using hamming distance maximization flimsy in the real world, where data has complex and high intra/inter-class diversity.

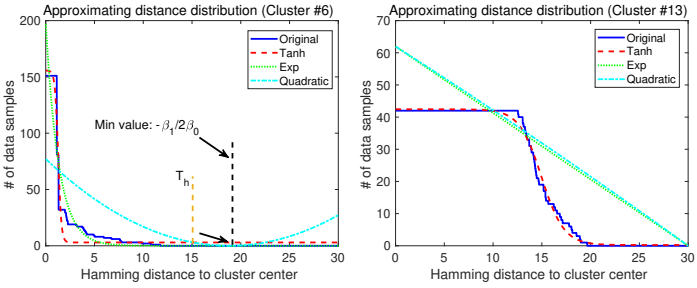
We can see from these observations that defense is challenging against strong adversaries. Instead of naive maximization from the original category, the optimization should be guided within a narrow subspace to avoid being: 1) exposed from the original category; 2) extracted via querying the rest categories; 3) degrading visual quality. To meet these requirements, we propose a new mechanism in the next section.

4. Cluster-based Weighted Distance Maximization

We propose a new mechanism called *cluster-based weighted distance maximization.* The idea is parallel to the center loss [44], which aims to enhance the discrimination of inter-class features and pull the intra-class features towards their centers for better classification. Here, however, we are learning through the adversarial lens for generating a hashcode via perturbing the input image, such that the distance to the hash centers is maximized. To account for intra-class variations, we represent each class with several centers, rather than a single one [44]. The hamming distance to the centers also exhibits heterogenous distributions across various categories. Samples may have high density around the center for some categories while others may scatter more evenly. Thus, the optimization should be aware of the intra-class distributions and their hamming distance to the center; otherwise, the protected image may fall into high-density regions, where all the samples have similar hash codes. The attacker can easily exploit these regions to retrieve the private image with high chances.

Our Mechanism. To address intra-class variations, we further partition the hash codes by a clustering method. For the set of hash codes $\{h(x_i)\}_{i=1,\cdots,N}$, we re-organize them into a number of k distinct clusters $\mathcal{C}_i$, $1 \leq i \leq k$. Existing clustering techniques such as k-means [45] and density-based DBSCAN [46] can be adopted (their pros and cons will be compared in Sec. 5.1).

After the clusters are found, we develop a weighted loss function to characterize the in-cluster hamming distance distribution. The goal is to push x' away from the cluster centers such that the number of samples returned by a query using x' is minimized. Because hamming distance is symmetric, this is equivalent to our original intention that maximizes $\mathcal{D}_h(x, x')$, so x' is not returned by the query, when the attacker queries any image x from $\mathcal{C}_i$. Define $F_i(d)$ as the cumulative distribution of the number of samples with distance d from center c_i. For a total number of k clusters,



(a) (b)

Figure 4: Least square approximation of in-cluster sample distributions using hyperbolic tangent, exponential and quadratic functions on CIFAR10 with k-means clustering (a) Cluster #6; (b) Cluster #13.

the new objective minimizes a new loss function $\mathcal{L}_c$ defined as,

$$\min_{x'} \mathcal{L}_c(x') = \sum_{i=1}^{k} \|F_i(\frac{1}{2}(m - h(x')h(c_i)^\top))\|_2^2, \quad (5)$$

$$s.t. |x - x'| < \epsilon. \quad (6)$$

where $h(c_i)$ is the hash code of the i-th cluster center c_i, m is the total hash bits ($m = 48$).

Optimization. Set the initial image of x' as x, x' can be updated in an iterative manner,

$$x' = clip_{x,\epsilon}\big(x' + \epsilon \cdot \nabla_x \mathcal{L}_c(\theta, h(x'), \{h(c_i)\}_{i=1}^k)\big) \quad (7)$$

The gradient of the loss function can be calculated as,

$$\frac{\partial \mathcal{L}_c}{\partial h(x')} = \sum_{i=1}^{k} 2F_i\big(\mathcal{D}_h(x', c_i)\big) \frac{\partial F_i(\mathcal{D}_h(x', c_i))}{\partial \mathcal{D}_h(x', c_i)} \frac{\partial \mathcal{D}_h(x', c_i)}{\partial h(x')}$$
$$= -\sum_{i=1}^{k} F_i\big(\mathcal{D}_h(x', c_i)\big) \frac{\partial F_i\big(\mathcal{D}_h(x', c_i)\big)}{\partial \mathcal{D}_h(x', c_i)} h(c_i). \quad (8)$$

To use gradient-based optimization, $F_i(\cdot)$ should be a differentiable function. We learn a least square regression for each cluster $1 \le i \le k$, based on the hamming distance j to the center ($j \in [1, 30]$) and the number of samples y_j,

$$\hat{F}_i = \arg\min_{F_i} \sum_{j=1}^{m} \|F_i(d_j) - y_j\|_2^2. \quad (9)$$

The parameters can be derived by a closed form solution, $\hat{\boldsymbol{\beta}} = (\boldsymbol{d}^T\boldsymbol{d})^{-1}\boldsymbol{d}^T\boldsymbol{y}$.

To examine the effectiveness of regression, we plot the relationships between the d and y (shown as "Original") in Fig.4 for CIFAR10. In most of the cases, the images are concentrated around the cluster centers (Fig.4(a)). There are also some clusters that samples are more scattered (Fig.4(b)). To minimize the square error, it is tempting to adopt high-order polynomials for better characterization, but they would slow down the defense process due to high computations. For training stability, we adopt quadratic regression and compare them with nonlinear regressions of hyperbolic tangent and exponential in Fig. 4. The quadratic regression demonstrates empirical advantages summarized by the following properties.

Property 1. The convexity of quadratic function facilitates the convergence of the loss function. Though both hyperbolic tangent and exponential functions fit the distribution better (almost perfectly for $\tanh$), they are not stable during training.

For $\tanh$, the gradient vanishes for most of the clusters and the loss function is unable to converge. We conjecture that the failure is due to the original distance distribution having a high concentration of samples close to the cluster center and a flat, long tail with gradients almost equal to zero. Since tanh fits such distribution perfectly, the flat tail is causing the gradient to vanish and no subsequent updates from the backpropagation. For the exponential function, it tends to overfit when d is small (overshoots around the cluster center with small distance). Our test indicates that when $d \to 0$, $F_i(d) \to \infty$ for some clusters and this brings instability to the backpropagation process.

Property 2. Denote the quadratic parameters as $(\beta_0^i, \beta_1^i, \beta_2^i), 1 \le \forall i \le k$, and the largest hamming distance to the center (radius) as, $r_i = \max \mathcal{D}_h(x, c_i), x \in \mathcal{C}_i$. If $-\frac{\beta_1^i}{2\beta_0^i} > r_i + T_h$ and x is mapped to x' such that $\mathcal{L}_c$ is minimized, it is guaranteed that x' will not be returned as query results.

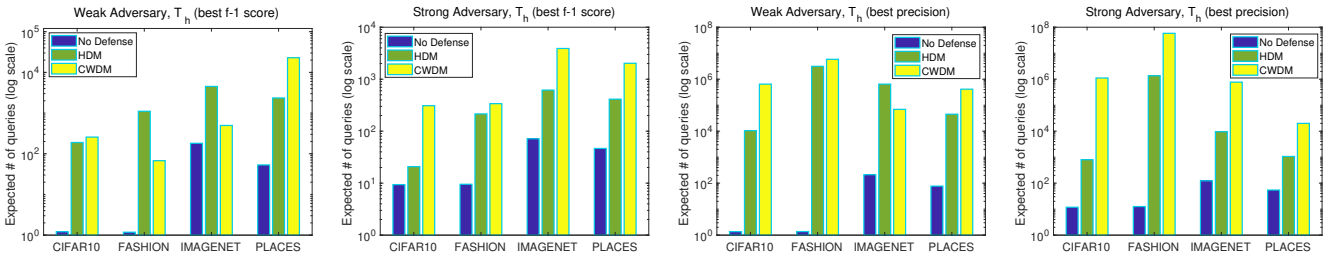
In Fig.4(a), $-\frac{\beta_1^i}{2\beta_0^i}$ is the distance corresponds to the minimum value of the quadratic function, which is the optimization goal. If it is larger than the sum of the retrieval threshold and the radius, using any samples from the cluster will not be able to fetch x'. This condition holds for most clusters because the samples tend to concentrate in high density around the centers. For the rest of clusters like shown in Fig.4(b), though optimization is able to reach the minimum value (around 30 in hamming distance), it has to balance the influence from other clusters as well. That is, maximizing the distance to a single cluster may accidently push the protected image into the proximity of other clusters. Our loss function is designed in a way to mitigate such effects based on the in-cluster distributions.

5. Evaluation

The main goal of evaluation is to investigate: 1) effectiveness of the proposed mechanism in both white-box and black-box settings; 2) defense budget in terms of computational efforts; 3) perceptual similarity from the original image.

Dataset. We conduct the experiments on four datasets: CIFAR10 [43], Fashion-MNIST [47], ImageNet [48] and Places365 [49]. Places365 mimics the scenario when privacy is exploited from location similarity. Following [26], we randomly select 10% categories of ImageNet and Places365.

Implementation Details. We train HashNet-ResNet50 for CIFAR10/Fashion and HashNet-ResNet152 for ImageNet/Places365. We randomly select 500 images from the



(a) (b) (c) (d)

Figure 5: Expected number queries to expose the private images with strong and weak adversaries (larger indicates higher robustness) (a) Weak adversary (T_h with best f-1 score); (b) Strong adversary (T_h with best f-1 score); (c) Weak adversary (T_h with best precision); (d) Strong adversary (T_h with best precision).

test set as the *private* images to be protected and use the rest of the test set as the *attack set*. The retrieval threshold T_h is selected when the best F-1 score ($T_h = 15, 16, 12, 10$) and the best precision ($T_h = 8, 6, 8, 8$) are achieved for the four datasets, respectively.

Baselines. We compare our mechanism with a combination of baselines: *no defense* and *hamming distance maximization* [33] against the *weak adversary* and *strong adversary*. The weak adversary has some knowledge about the private image so he only queries the original category. The strong adversary enumerates through the entire test set of all categories.

Metrics. Based on the threat model, the adversary randomly picks images from the attack set to expose the private images. The mechanisms are evaluated thoroughly based on the following metrics: 1) Expected number of queries of weak adversary E_w,

$$E_w = \frac{\text{total \# attack images}}{\text{avg \# img retrieved (same class)}}.$$

2) Expected number of queries of strong adversary E_s,

$$E_s = \frac{\text{total \# attack images}}{\text{avg \# img retrieved (all class)}}.$$

These metrics quantify the efforts from the attacker. 3) Defense effort in terms of the number of iterations and computational time using a sole Nvidia GTX1070 GPU. 4) Perceptual difference between x and x' by the two metrics, a) *mean square error*, MSE $= \sum_i (x'_i - x_i)^2/N$, where x_i, x'_i are the normalized pixel values of the original and protected images and N is the dimensionality of the image; b) *Structural similarity index* that captures structural similarities to emulate human visual [50].

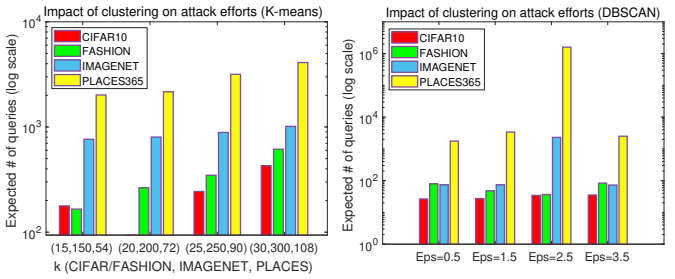
5.1. Attack Efforts

Fig.5 compares the attack efforts of the cluster-based weighted hamming distance maximization (CWDM) with the hamming distance maximization (HDM) [33] and the no defense baseline. We can see that with "no defense", the adversary can simply extract the private images from the database within 10 queries for CIFAR10/Fashion and 100 queries for ImageNet/Places365. For weak adversary, our mechanism CWDM is a little worse or on par with HDM. This is because the hash code found by CWDM is not as far as HDM from the original image, because CWDM has to consider distance from the rest categories to defend strong adversary. As a result, for strong adversary, CWDM effectively hardens the attack effort by 1-3 orders of magnitude than HDM, and 2-7 orders of magnitude than "no defense". E.g, for best precision, 1.1M, 58M, 0.77M and 20K number of queries are required on average, which are prohibitive for attackers with finite resources. In practice, adversaries may not know exactly what categories the private images are from, so a viable way is to explore all possible categories. CWDM successfully enlarges the attack efforts in this case.

Clustering Techniques. We assess the impact from the clustering techniques in Fig.6 between k-means [45] and DBSCAN [46]. For k-means, we increase the number of clusters k from 15 to 30 for CIFAR10/Fashion, 150 to 300 for ImageNet and 54 to 108 for Places365; for DBSCAN, we increase EPS (maximum distance between two samples in order to be clustered) from 0.5 to 3.5. With a larger k, k-means tends to result more compact clusters with less intra-cluster distance, which leads to a general trend of higher robustness against strong adversaries. DBSCAN is more sensitive to distribution density and the value of EPS (e.g., the surge when $eps = 2.5$). k-means offers better predictability, and performance than DBSCAN by almost an order of magnitude. The main reason is because DBSCAN explicitly categorizes samples with distance larger than the EPS as outliers; CWDM does not account for these outliers during optimization thus leaving some risks, if the attack sample is identical to the outliers. An exception is Places365 where the learned hashcodes of selected categories are more concentrated than ImageNet and the outliers are less. This makes DBSCAN better than k-means on Places365.

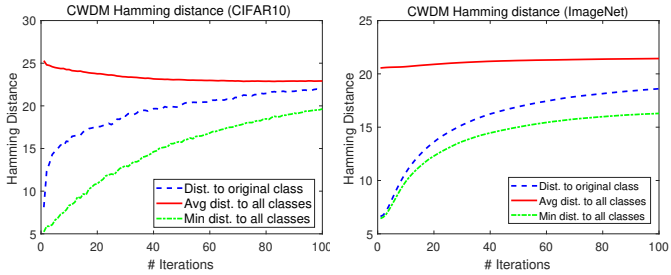
5.2. Defense Efforts

Defense efforts are measured by the hardness of finding the adversarial example in hash space. HDM only pays attention to the original category, thus should be much easier to optimize in general (which takes about 20 iterations to reach equilibrium as shown in Fig.3(b)). On the other hand, CWDM balances the influence from all the class distributions and the quadratic regression introduce additional com-



(a) (b)

Figure 6: Impact of clustering techniques on attack efforts (a) k-means; (b) DBSCAN.



(a) (b)

Figure 7: Defense budget (convergence) - hamming distance from the protected image to different classes (a) CIFAR10; (b) ImageNet.

putation overhead. Fig.7 traces the convergence of hamming distance to the original class, average and minimum distance to all classes. It is observed that CWDM quickly enlarges the distance from the original class beyond the retrieval threshold (preventing retrieval from weak adversaries). "Minimum distance" tracks the class with the minimum hamming distance. It converges a little slower than the original class, because it represents those hard classes during optimization. This explains why a few samples from irrelevant classes could still fall into the retrieval threshold and lead to the success of strong adversaries. Overall, the "average distance" summarizes the convergence from all classes, and reaches a value larger than the retrieval threshold so most of the queries should return no result for protected images. Computationally, using the Nvidia GTX1070 GPU, an image takes about 4s with 100 iterations, which is quite practical in real applications. A speed-up strategy is to increase the learning rate, but at a cost of degraded success rate of generating the protected image.

5.3. Useability

Perturbation Artifacts. We compare the amount of perturbations introduced onto the private images with some examples in Fig. 8. To visualize the noise clearly, we scale up their values by four times with a 0.5 uplift to offset any negative adversarial values. We can see that noise from CWDM concentrates more around the object, whereas HDM tends to distribute the noise across the entire image. To quantify the nuances, we further evaluate the average MSE and SSIM from the original image in Table 1. The MSE is av-

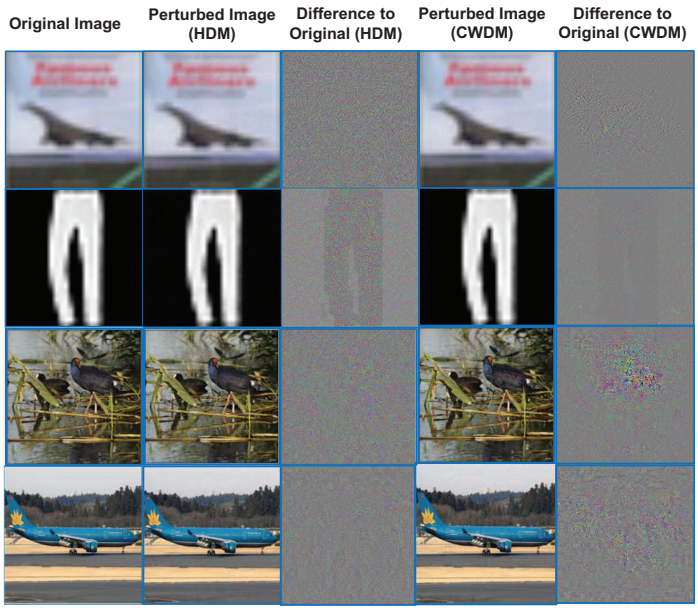


Figure 8: Perturbed image using HDM, CWDM and their normalized difference to the original image.

	MSE(per pixel 10^{-5})		SSIM ($[0,1]$)	
	HDM	CWDM	HDM	CWDM
CIFAR10	3.4071	2.0022	0.8971	0.9751
Fashion	2.9107	2.0757	0.8038	0.8907
ImageNet	3.0957	3.2244	0.9614	0.9611
Places365	2.3470	3.0031	0.9721	0.9628

Table 1: Perturbations measured by MSE and SSIM

eraged per pixel value by dividing $224 \times 224 \times 3$. SSIM falls in the range of $[0, 1]$, where 1 means the image is identical to the original one, and a less value means the distortion is higher. For CWDM, the MSE is 37% less than HDM for CIFAR10/Fashion and SSIM is almost identical to the original image by reaching a score over 0.9 on average. This is because the objective of maximized hamming distance would push the protected image further away from the original sample in hash space, whereas CWDM is more moderately looking for a subspace not far from the decision boundaries (retrieval threshold). The noise for ImageNet/Places365 is slightly higher because more scattered samples and clusters make it harder to find a subspace to hide, thereby demanding strengthened perturbations. Fortunately, the additive noise does not turn out to be significant measured by SSIM (last two columns).

Classification Tasks. Social platforms also provide functions such as automatic photo classification, object and text recognition. These tasks typically adopt different loss functions (e.g., softmax). Since the perturbations are applied globally, we show that they do not transfer to the normal feature space, and mislead softmax classifications. Table 2 demonstrates the accuracy loss of classification tasks by applying CWDM samples to the original models. The first

	Original		Adversarial (CWDM))	
	HashNet	Softmax	HashNet	Softmax
CIFAR10	0.870	*0.904*	0.097	**0.831**
Fashion	0.896	*0.936*	0.191	**0.934**
ImageNet	0.882	*0.908*	0.008	**0.817**
Places365	0.862	*0.853*	0.143	**0.731**

Table 2: Evaluation of potential accuracy loss on classification tasks.

two columns are the baselines of the original retrieval accuracy and softmax classification respectively (100 random categories for ImageNet). The third column shows the effectiveness of CWDM that reduces retrieval accuracy below 20%. When the protected image are applied to softmax classification (the fourth column), the result does not render significant accuracy loss (compared to the 2nd column). It is interesting to see that, though the hash space perturbations have influence in the normal feature space, neural networks can treat them as random noise in general so their existence should not impact other smart applications.

5.4. Black-box Defense

The previous subsections evaluate the scenario when users have full knowledge of the architecture and parameters of the model on the server (white-box). In practice, the proprietary model usually remains a *black box* to the users, such that they can only make their best guess of the *target model*. Empirical evidence has shown that adversarial perturbations can *transfer* across models in normal feature space [36, 37]. Here, we demonstrate transferability of our mechanism in the hash space.

We fix the target model (server side) and generate the protected image using different source models (user side). Black-box transferability is difficult given that the source and target models usually have different decision boundaries. Strong adversary from the server side can further take advantages of any nuance in such boundaries to expose the protected image. Thus, we consider the black-box scenario to be successful, as long as there are less than n samples that can be exploited to extract a protected image. We define the *defense success rate* as the ratio between the number of protected images that have less than n retrieval results in the target model and the total number of protected images. We set $n = 100$ here since it would take the adversary considerable efforts to find these 100 images from the 50K/60K/100K/36K attack sets. We adopt different architectures on the four datasets due to the performance gap from the original HashNet, i.e., ResNet50 and architectures of less complexity exhibit much lower accuracy on the ImageNet. Thus, we set ResNet50* and ResNet152* as the target models for CIFAR10/Fashion and ImageNet/Places365 respectively, use ResNet18, ResNet34 and VGG16 as the source model for CIFAR10/Fashion, and ResNet50, ResNet101 and ResNext101 [51] for ImageNet/Places365.

Table 3 shows the success rate of transferability to target models (col.2-4) and benchmarks the results with the "no defense" baseline as the lower bound (col.5). For CIFAR10/Fashion, CWDM-protected image can successfully transfer with a rate of 40-60% within the ResNet family, possibly due to similar decision boundaries. Transferability also exists for ImageNet/Places365 about 20-40%. VGG16 has less chance to transfer. Thus, in a blackbox setting, if the user makes the correct guess of the target architecture, her images can be protected almost perfectly based on the previous white-box experiments; if the guess falls off a little, she still enjoys nearly 30-50% on average, which offers considerable improvement over "no defense" (col. 5).

ResN50*	ResN18	ResN34	VGG16	No def.
CIFAR10	44.13	40.96	22.39	9.8
Fashion	56.99	60.32	51.85	5.8
ResN152*	ResN50	ResN101	ResNext101	No def.
ImageNet	22.40	36.40	33.40	13.20
Places365	45.09	40.36	30.79	11.86

Table 3: Defense success rate of black-box transferability from ResNet50* and ResNet152* to different architectures (%).

5.5. Discussion

Without knowing our defense, the attacker strives to collect a dataset that resembles the training set to extract the private, similar images. Our mechanism successfully defends against these attackers when their attack sets are i.i.d. with the training set. Due to space limit, we will conduct more experiments to evaluate active attackers when they deviate from the i.i.d settings. The accuracy from the original HashNet also affects the effectiveness of the defense. For example, ResNet18-50 do not achieve satisfied accuracy to map semantically similar images into identical and compact hash codes on ImageNet/Places365. The learned codes are more scattered, thereby squeezing the optimization space to successfully perturb the image and prevent from retrieval. Since the service providers typically finetune their models, we expect the proposed mechanism to be effective against the production models with high accuracy.

6. Conclusion

In this paper, we describe efforts to protect private images from malicious deep image retrieval. We first identify and experimentally validate the effectiveness of using adversarial perturbations as a defense in the hash space. By showing vulnerabilities against strong adversaries, we propose a new mechanism to find an alternative subspace that maximizes the weighted hamming distance to all the classes. We evaluate the efforts from both the attack and defense perspectives, useability, and black-box transferability with extensive experimental results.

7. Acknowledgement

This work was supported in part by the U.S. National Science Foundation under grant number CCF-1850045.

References

[1] "Google image search," https://www.google.com/imghp.

[2] "Pinterest visual search tool," https://www.pinterest.com/.

[3] "Bing," https://www.bing.com/.

[4] "Tin eye," https://www.tineye.com.

[5] D. Lu, X. Liu, and X. Qian, "Tag-based image search by social re-ranking," *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1628–1639, 2016.

[6] D. Shankar, S. Narumanchi, H. Ananya, P. Kompalli, and K. Chaudhury, "Deep learning based large scale visual recommendation and search for e-commerce," *arXiv preprint arXiv:1703.02344*, 2017.

[7] X. Ji, W. Wang, M. Zhang, and Y. Yang, "Cross-domain image retrieval with attention modeling," in *Proceedings of the 25th ACM international conference on Multimedia*. ACM, 2017, pp. 1654–1662.

[8] "Alibaba's pailitao," http://www.pailitao.com/.

[9] "Deepfashion: attribute prediction dataset," https://bit.ly/2N4ZGQP.

[10] "Instagram by the numbers," https://bit.ly/2wnRfJ1.

[11] "Ebay by the numbers," https://bit.ly/2NzaEif.

[12] "How google uses the picture you search with," https://support.google.com/websearch/answer/1325808.

[13] "Eu general data protection regulation," https://eugdpr.org/.

[14] R. Chen, A. Reznichenko, P. Francis, and J. Gehrke, "Towards statistical queries over distributed private user data," in *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation NSDI 12)*, 2012, pp. 169–182.

[15] A. Reznichenko and P. Francis, "Private-by-design advertising meets the real world," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 116–128.

[16] M. Hardt and S. Nath, "Privacy-aware personalization for mobile advertising," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 662–673.

[17] S. Nath, F. X. Lin, L. Ravindranath, and J. Padhye, "Smartads: bringing contextual ads to mobile apps," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM, 2013, pp. 111–124.

[18] Y. Han and Y. Shen, "Accurate spear phishing campaign attribution and early detection," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. ACM, 2016, pp. 2079–2086.

[19] W. Bullock, L. Xu, and L. Zhou, "Predicting household demographics based on image data," Apr. 30 2019, uS Patent App. 10/277,714.

[20] M. J. Wilber, V. Shmatikov, and S. Belongie, "Can we still avoid automatic face detection?" in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–9.

[21] P. Ilia, I. Polakis, E. Athanasopoulos, F. Maggi, and S. Ioannidis, "Face/off: Preventing privacy leakage from photos in social networks," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 781–792.

[22] L. Zhang, K. Liu, X.-Y. Li, C. Liu, X. Ding, and Y. Liu, "Privacy-friendly photo capturing and sharing system," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 524–534.

[23] C. Bo, G. Shen, J. Liu, X.-Y. Li, Y. Zhang, and F. Zhao, "Privacy. tag: Privacy concern expressed and respected," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*. ACM, 2014, pp. 163–176.

[24] K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen, "Deep learning of binary hash codes for fast image retrieval," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 27–35.

[25] H. Zhu, M. Long, J. Wang, and Y. Cao, "Deep hashing network for efficient similarity retrieval," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[26] Z. Cao, M. Long, J. Wang, and P. S. Yu, "Hashnet: Deep learning to hash by continuation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5608–5617.

[27] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2064–2072.

[28] J. Wang, T. Zhang, N. Sebe, H. T. Shen *et al.*, "A survey on learning to hash," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 769–790, 2017.

[29] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[30] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[31] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[32] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.

[33] E. Yang, T. Liu, C. Deng, and D. Tao, "Adversarial examples for hamming space search," *IEEE transactions on cybernetics*, 2018.

[34] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[35] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005.

[36] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. ACM, 2017, pp. 506–519.

[37] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," *arXiv preprint arXiv:1704.03453*, 2017.

[38] I. E. Akkus, R. Chen, M. Hardt, P. Francis, and J. Gehrke, "Non-tracking web analytics," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 687–698.

[39] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz, "The web never forgets: Persistent tracking mechanisms in the wild," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 674–689.

[40] Y. Liu, W. Zhang, and N. Yu, "Protecting privacy in shared photos via adversarial examples based stealth," *Security and Communication Networks*, vol. 2017, 2017.

[41] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2074–2081.

[42] "Mnist dataset," http://yann.lecun.com/exdb/mnist/.

[43] "Cifar10 dataset," https://www.cs.toronto.edu/~kriz/cifar.html.

[44] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*. Springer, 2016, pp. 499–515.

[45] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[46] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[47] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[48] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 248–255.

[49] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[50] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[51] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.