Robustness of Practical Perceptual Hashing Algorithms to Hash-Evasion and Hash-Inversion Attacks

Jordan Madden, Moxanki Bhavsar, Lhamo Dorje, and Xiaohua Li
Dept. of ECE, Binghamton University
Binghamton, NY 13902
{jmadden2, mbhavsa2, ldorje1, xli}@binghamton.edu

Abstract

Perceptual hashing algorithms (PHAs) are widely used for identifying illegal online content and are thus integral to various sensitive applications. However, due to their hasty deployment in real-world scenarios, their adversarial security has not been thoroughly evaluated. This paper assesses the security of three widely utilized PHAs—PhotoDNA, PDQ, and NeuralHash—against hash-evasion and hash-inversion attacks. Contrary to existing literature, our findings indicate that these PHAs demonstrate significant robustness against such attacks. We provide an explanation for these differing results, highlighting that the inherent robustness is partially due to the random hash variations characteristic of PHAs. Additionally, we propose a defense method that enhances security by intentionally introducing perturbations into the hashes.

1 Introduction

Perceptual hashing, also known as robust hashing, is a content fingerprinting technique that generates similar binary hashes for perceptually similar contents and different hashes for different contents [1]. Perceptual hashing is useful in various applications such as content moderation, authentication, and detection. Numerous perceptual hash algorithms (PHAs) have been developed or deployed for practical applications [2]. Early examples include the algorithm used on YouTube to identify copyrighted material [3] and algorithms used in online image search engines of Google, Microsoft Bing, etc [4].

Recently, PHAs have gained more critical importance, particularly in responding to the requiremnt of reporting illicit content. For instance, in cases involving Child Sexual Abuse Media (CSAM), US media service providers are legally obliged to examine suspected illegal content and report it to a "Cyber Tip Line" operated by the National Center for Missing and Exploited Children (NCMEC). Microsoft's PhotoDNA and Meta's PDQ were immediately deployed for this purpose [5][6][7].

With a growing emphasis on user privacy, technologies like end-to-end encryption have become prevalent. This shift has led to the deployment of PHAs on the client side before data encryption [8]. For example, Apple has proposed to deploy NeuralHash on all its iOS devices [9]. However, this client-side deployment has raised significant concerns among the client users about their data privacy, which inadvertently arouses a widespread criticism regarding the security and privacy-preserving capabilities of PHAs [10]. Criticism has been further fueled by recent studies highlighting the vulnerability of PHAs to adversarial attacks and the potential for information leakage through hash bits [4][11][12].

AdvML-Frontiers'24: The 3nd Workshop on New Frontiers in Adversarial Machine Learning@NeurIPS'24, Vancouver, CA.

Given the critical role PHAs play in sensitive applications, it is essential to ensure their security. Unfortunately, PHAs are often deployed hastily without sufficient security evaluation. This paper systematically assesses the adversarial robustness of three widely used PHAs—PhotoDNA, PDQ, and NeuralHash—against two types of attacks: hash-evasion attacks, aimed at enabling illicit content to evade detection, and hash-inversion attacks, aimed at reconstructing the original secret content from hashes. To support this evaluation, we introduce two new attack algorithms: a query-efficient black-box adversarial attack algorithm and a data-efficient hash inversion algorithm.

The major contributions of this paper are as follows.

- This study demonstrates that PHAs exhibit notable robustness against blackbox adversarial evasion attacks when realistic constraints on image distortion, query budget, and hash difference are applied. This finding challenges previous studies that overlooked these constraints, leading to unjustified criticisms of PHAs.
- This paper confirms the sufficient privacy-preserving capabilities of hash bits, countering claims from some existing studies that suggested hash bits disclose significant information about the content of the original images. The key point is how diverse the images are.
- This research reveals that the security of PHAs is partially due to the unique property of random hash variations. Based on this observation, a defense method is proposed to enhance security by intentionally randomizing hash bits.

The paper is organized as follows: Section 2 provides a literature review, Section 3 formulates the attack and defense algorithms, Section 4 presents the experiment results, and Section 5 concludes the paper. More detailed experiment data can be found in the Appendix. The source code of the experiments, including ways to access binary executables of the PHAs, can be found at `https://github.com/neddamj/nnhash`.

2 Literature Review

Among the numerous PHA's [2], this paper focuses on the following three major ones due to their widespread deployment in practice:

- **PhotoDNA**, released by Microsoft in 2009, is used by over 70 companies including Cloudflare and Dropbox [6][13][14]. It generates an 1152-bit hash for each input image;
- **PDQ**, launched by Meta in 2019 [7] and heavily inspired by pHash [2][15], produces a 256-bit hash for each input image.
- **NeuralHash**, released by Apple in 2021 with the goal of identifying CSAM [9], creates a more concise 96-bit hash for each input image.

The adversarial robustness of PhotoDNA, PDQ, and NeuralHash to either hash-evasion or hash-inversion attacks has not been thoroughly investigated, primarily due to their relative novelty and the absence of publicly available source code. For hash-evasion, [16] was the first to explore the robustness of NeuralHash, revealing its vulnerability to whitebox attacks, assuming the adversary has full access to the PHA. Additionally, [17] examined PDQ and demonstrated how adversarial images could be generated in the whitebox context. Furthermore, [18] utilized linear interpolation between two images to create adversarial examples.

Regarding blackbox hash-evasion attacks, where the adversary only has access to the input and output of the algorithm, [17] examined PDQ. It concluded that it was not robust against untargeted blackbox attacks. Similarly, [19] evaluated both PhotoDNA and PDQ, finding that they lacked robustness against both targeted and untargeted attacks. While these studies are closely related to the current paper, they overlooked the importance of adversaries' constraints on image distortion, query budget, as well as hash difference levels. By all means, unconstrained blackbox attacks may always be successful, particularly in the untargeted setting.

Regarding hash-inversion attacks or privacy-preserving capabilities of PHAs, Microsoft initially claimed that the hashes generated by PhotoDNA were irreversible and could not be used to reconstruct the original images [20]. However, a blog post [12] challenged this assertion, suggesting that PhotoDNA hashes might contain enough information to synthesize images from the hashes. To

the best of our knowledge, [12] is the only work addressing the reconstruction of original images from hashes, and it specifically focuses on PhotoDNA, which uses long hashes. Another method for investigating information leakage involves training classifiers to identify image content from hashes. Studies such as [16] and [18] demonstrated that NeuralHash hashes could indeed reveal information about their source images, such as identifying the object's class. These findings contrast with those of [21], which examined PhotoDNA hashes and concluded that they were resistant to such classification attacks.

3 Attack Algorithms and Defense Method

In a perceptual hash system, a service provider scans a client's unencrypted content, extracts hashes for each image using a perceptual hashing algorithm (PHA) and compares these hashes with a database of known illicit hashes. If a match is found, an alarm is raised or the unencrypted content is forwarded for further analysis [8]. In this context, we consider two types of adversaries: the *client adversary*, who aims to evade hash matching, and the *service provider (or third-party) adversary*, who seeks to reconstruct the client's original images from collected hashes. This paper evaluates the security of three widely used PHAs—PhotoDNA, PDQ, and NeuralHash—against two specific types of attacks: blackbox hash-evasion attacks, which are employed by the client adversary, and hash-inversion attacks, which are utilized by the service provider adversary. Additionally, we propose a defense mechanism that can be integrated into the perceptual hashing system.

3.1 A Query-Efficient Adversarial Blackbox Attack Algorithm for Hash Evasion

This study focuses exclusively on blackbox adversarial attacks, rather than whitebox attacks, to reflect the practical scenario where PHA implementations are proprietary. In many cases, the details of PHA architectures and model weights are not publicly available due to their commercial nature. For example, Microsoft enforces non-disclosure agreements for PhotoDNA users, providing only an emulation of the algorithm. Similarly, although [22] released weights for the first-generation NeuralHash, Apple has since updated the algorithm, and the details of the latest version remain confidential.

Consider the original image $\mathbf{I}_0$ with the hash value $\mathbf{h}_0 = \mathcal{H}(\mathbf{I}_0)$. The objective of the client adversary is to generate a very similar image $\mathbf{I}_{\text{adv}}$ whose new hash $\mathbf{h}_{\text{adv}} = \mathcal{H}(\mathbf{I}_{\text{adv}})$ is sufficiently different from $\mathbf{h}_0$ in untargeted attacks or sufficiently similar to some target hash $\mathbf{h}_{\text{tar}}$ in targeted attacks. We use normalized Hamming distance $D_h(\mathbf{h}_0, \mathbf{h}_{\text{adv}})$ to evaluate the similarity of two hashes, and use per-pixel normalized L_2 norm $L_2(\mathbf{I}_0, \mathbf{I}_{\text{adv}})$ to evaluate the similarity of two images.

Because the output of PHAs is hash bits, not logits or class decisions, we propose a joint soft-label hard-label attack (JSHA) algorithm that can drastically improve blackbox attack query efficiency. JSHA consists of two steps. The first step is to apply a soft-label blackbox attack to generate an initial adversarial image $\mathbf{I}_{\text{init}}$ so that its hash $\mathbf{h}_{\text{init}} = \mathcal{H}(\mathbf{I}_{\text{init}})$ is far away from $\mathbf{h}_0$ for untargeted attacks or close to $\mathbf{h}_{\text{tar}}$ for targeted attacks. This step is formulated as the optimization

$$\max_{\delta} \;\; D_h\left(\mathbf{h}_0, \mathcal{H}(\mathbf{I}_0 + \delta)\right) \text{ (untargeted)}, \quad \text{or,} \quad \min_{\delta} \;\; D_h\left(\mathbf{h}_{\text{tar}}, \mathcal{H}(\mathbf{I}_0 + \delta)\right) \text{ (targeted)} \tag{1}$$

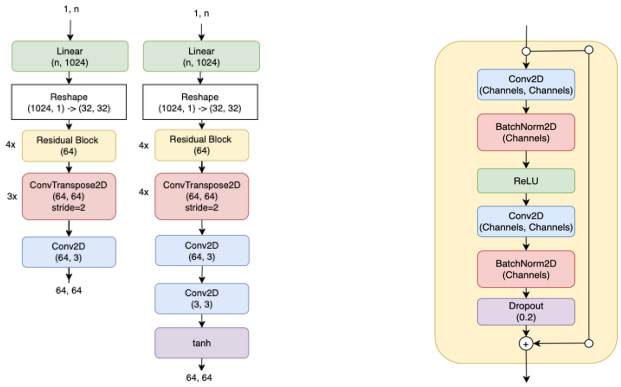
The optimization is conducted iteratively. In each iteration, the adversary queries the PHAs with its adversarial input $\mathbf{I}_0 + \delta$ to get the hash, and uses the hash to optimize the loss. After reaching the query number threshold or the hash distance threshold D_0, this first step stops with output $\mathbf{I}_{\text{init}} = \mathbf{I}_0 + \delta$.

The second step applies a hard-label blackbox attack to reduce the image distortion while maintaining the hash distance obtained in the first step. Specifically, this step solves the following optimization

$$\min_{\mathbf{I}_{\text{adv}}} \;\; L_2(\mathbf{I}_0, \mathbf{I}_{\text{adv}}), s.t.\; D_h(\mathbf{h}_0, \mathcal{H}(\mathbf{I}_{\text{adv}})) \geq D_0 \text{ (untargeted)}, D_h(\mathbf{h}_0, \mathcal{H}(\mathbf{I}_{\text{adv}})) \leq D_0 \text{ (targeted)} \tag{2}$$

Starting from $\mathbf{I}_{\text{adv}} = \mathbf{I}_{\text{init}}$, the adversary queries the PHAs to get the hash and uses the queried hash to optimize (2) until the query number threshold or the threshold of $L_2(\mathbf{I}_0, \mathbf{I}_{\text{adv}})$ is reached.

One of major advantages of this algorithm is high query efficiency. The first step is an unconstrained optimization and a large learning rate can be applied to get an initial adversarial sample quickly. In the second step, since we start from $\mathbf{I}_{\text{init}}$ which is only a noisy version of $\mathbf{I}_0$, the convergence is fast.



(a) (b)

Figure 1: (a) Hash inversion attack model (left: for Celeb Dataset. Right: for STL-10 dataset). (b) Architecture of residual block.

3.2 A Data-Efficient Hash-Inversion Attack Algorithm

GAN (generative adversarial network) was trained in [12] to invert CelebA images. However, the adversary may need more data-efficient hash-inversion attack algorithms because the training data may not be abundant. For this, we has developed new and more data-efficient generative models to reconstruct the original input images from the hashes. We designed a decoder-style deep model, similar to the Fast Style Transfer [23], to convert the N-bit hashes to $(64, 64)$ images. This model is mainly a combination of residual blocks depicted in Fig. 1(b), with fractionally-strided convolutions/deconvolutions for learned upsampling. The entire architecture is described in Fig. 1(a), where the left network is for the MNIST and CelebA dataset while the right network is used for the STL-10 dataset.

For a given training dataset, we first compute the hashes of each image offline and store the image-hash pairs. We then train the model outlined above to reconstruct the images as closely as possible by minimizing the mean-square-error (MSE) loss between the model output and the image from which the hash was generated.

3.3 A Defense Method based on Random Hash Perturbation

The adversaries rely on gradient estimation to solve the optimization problems. It is well known that hash bits of PHAs change slightly and randomly with respect to slight changes in input images. Based on existing work on random perturbation defense [24][25], we have the following proposition:

Proposition 1. Gradient estimation based on queried hash is not reliable due to the random variation of hash bits.

An outline of the proof is in the Appendix A.1. The random variation of hash bits makes the adversaries' optimization struggle to converge, which will be demonstrated in the next section. Furthermore, considering this observation, we propose a defense method where the PHAs randomly invert q-portion of the hash bits to guarantee sufficient hash randomness, where $0 \leq q \leq 1$.

4 Experiment

4.1 Performance Metrics

We use the attack success rate for assessing the robustness against blackbox adversarial hash-evasion attacks. Given a set of M original images $\mathbf{I}_0^m$ with their corresponding hashes $\mathbf{h}_0^m$, the adversary generates adversarial images $\mathbf{I}_{\text{adv}}^m$ with hashes $\mathbf{h}_{\text{adv}}^m$. The attack success rate is defined as

$$\text{ASR}(p) = \frac{1}{M} \sum_{m=1}^{M} \mathbb{I}\left(D_h(\mathbf{h}_0^m, \mathbf{h}_{\text{adv}}^m) \gtrless p\right) \mathbb{I}\left(L_2(\mathbf{I}_0^m, \mathbf{I}_{\text{adv}}^m) < \theta\right) \tag{3}$$

Table 1: ASR(p) of Targeted Hash-Evasion Attacks. (**PH**: PhotoDNA. **PD**: PDQ. **NH**: NeuralHash)

ASR	SimBA	SimBA +HSJA	NES	NES+ HSJA	ZO-sign-SGD	ZOsign -SGD +HSJA
PH ASR(.1)	0%	0%	0%	0%	0%	0%
PD ASR(.1)	0%	0%	0%	0%	0%	0%
NH ASR(.1)	0%	0%	0%	0%	0%	0%
[19] PH ASR(.3) ($L_2 = 0.19$)			83%	(2×10^6 queries)		
[19] PD ASR(.3) ($L_2 = 0.42$)			100%	(6×10^5 queries)		

where $\mathbb{I}(\cdot)$ is the indicator function, $p \in [0, 1]$ is the pre-set hash distance threshold and θ is the pre-set image distortion threshold. For the operator "$\gtrless$", $>$ is used for untargeted attacks, $<$ is for targeted attacks. Small or zero ASR(p) means a PHA is robust to an attack.

For the thresholds, we set $\theta = 0.1$. On the other hand, we should apply large enough p to evaluate untargeted attacks because $\mathbf{h}_{\text{adv}}$ should be different from $\mathbf{h}_0$. It was found in [11] that most pairs of distinct images had $D_h(\mathbf{h}_0, \mathbf{h}_1) > 0.3$. Therefore, we primarily use $p = 0.3$ in (3) to determine the success of untargeted attacks. In contrast, we should use small enough p for targeted attacks to make sure $\mathbf{h}_{\text{adv}}$ is sufficiently close to $\mathbf{h}_{\text{tar}}$. Prior studies [11] indicate that many normal image edits would change an image's hashes to the level of $D_h(\mathbf{h}_0, \mathbf{h}_1) < 0.1$. Therefore, we mainly use $p = 0.1$ in (3) for assessing the success of targeted attacks. This important parameter p was unfortunately neglected by most existing adversarial robustness studies.

For hash-inversion attacks, we use the L_2 distortion, the SSIM (Structural Similarity Index Measure) [26], and the LPIPS (Learned Perceptual Image Patch Similarity) [27] between the reconstructed image and the true image to evaluate the performance. No ASR or query budget is involved.

4.2 Experiment on Hash-Evasion Attack and Defense

In implementing our proposed JSHA algorithm, we modified three different soft-label attack algorithms: SimBA [28], NES [29], and ZOsignSGD [30], to use in Step One, and modified the hard-label attack algorithm HSJA [31] to use in Step Two. For instance, "SimBA+HSJA" means our JSHA with a modified SimBA as Step One and a modified HSJA as Step Two. Modifications included removing/changing constraints and adjusting hyper-parameters, etc. We compared the proposed JSHA with the direct application of the original SimBA, NES, and ZOsignSGD algorithms. We also compared it with existing works [17] and [19].

The experiments were conducted on 100 randomly selected images from the ImageNet validation dataset. Step One was executed for a total of 3000 queries unless the early stopping criterion was met. Step Two was run for an additional 2000 queries unless meeting the early stopping criterion. The main challenge for us was the CPU-based non-parallel PHA executables, which limited the number of images or queries we could process. In comparison, [19] only utilized 30 images in experiments.

For targeted hash-evasion attacks, experiment results are presented in Table 1 and Appendix A.3. With a focus on ASR(0.1), we observed that all targeted attacks failed, yielding ASR$(0.1) = 0$. To our knowledge, no literature has ever reported non-zero ASR(0.1), suggesting that all the PHAs are robust against blackbox targeted hash-evasion attacks.

For untargeted attacks, experiment results are presented in Table 2 and Appendix A.4. With a focus on ASR(0.3), we see that PDQ was robust against all adversarial attacks, with a maximum ASR(0.3) of 1% only. It's worth noting that although [17] achieved a much higher attack success rate against PDQ, it required significantly more queries. The results also demonstrated that our defense method was effective because it drastically reduced ASR(0.3) and made the PHAs robust.

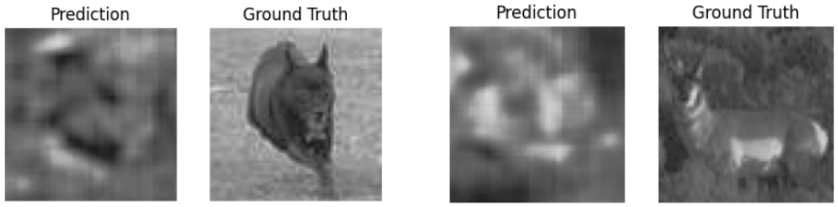
4.3 Experiment on Hash-Inversion Attack and Defense

Experiments were conducted over three datasets: MNIST, CelebA, and STL-10. For each hashing algorithm and dataset, we trained the inversion network depicted in Fig. 1 for 50 epochs, with a batch size of 64 and an initial learning rate of 0.005. We utilized the MSE loss function with the AdamW optimizer and the cosine annealing learning rate scheduler. While MNIST and CelebA models were

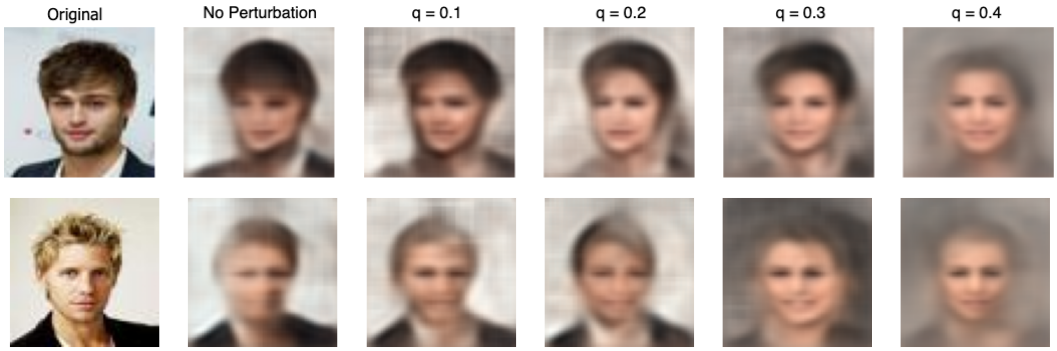
Table 2: Adversary's ASR(p) in Untargeted Blackbox Hash-Evasion Attacks, either without defense ($q = 0$) or with defense $q = 0.1$. (**PH**: PhotoDNA. **PD**: PDQ. **NH**: NeuralHash.)

ASR	SimBA	SimBA +HSJA	NES	NES+ HSJA	ZO-sign-SGD	ZOsign -SGD +HSJA	Defense +SimBA +HSJA	Defense +NES +HSJA
PH ASR(.3)	1%	**92%**	0%	69%	0%	19%	27%	21%
PD ASR(.3)	0%	0%	0%	**1%**	0%	0%	0%	0%
NH ASR(.3)	0%	14%	0%	**28%**	0%	2%	0%	2%

[19] PH ASR(.3)= 90%, ($L_2 = 0.2$, HSJA with 3×10^4 queries)
[17] PD ASR(.3)= 73%, ($L_2 = 0.1$, up to 8×10^6 queries)



(a) STL-10 image samples with NeuralHash, without defense



(b) CelebA image samples with PhotoDNA, under various levels q of defense

Figure 2: Samples of true/original images and reconstructed images by the hash inversion algorithm.

relatively easy to train, we encountered substantial challenges when training STL-10 models. We had to degrade STL-10 images to black-and-white in order to get meaningful reconstructed images.

Some sample images are shown in Fig. 2 whereas major experiment results are left to Appendix A.5 to save space. In general, hash-inversion failed over STL-10 while some level of success could be seen in MNIST and CelebA over PhotoDNA hash only. The reason is that the MNIST and CelebA images have similar or regular formation. The reconstructed images tend to converge to such common formation when the hash is long enough. However, even in this case, the reconstructed CelebA images could not be used to recognize the original face.

The effectiveness of random hash perturbation defense can be clearly seen in Fig. 2(b). Even the slightest perturbation $q = 0.1$ made the reconstructed image quite different from the original image.

5 Conclusions

This paper evaluates the security of three widely used perceptual hashing algorithms (PHAs): PhotoDNA, PDQ, and NeuralHash. Contrary to many existing studies, our findings show that these PHAs demonstrate notable robustness against both adversarial blackbox hash-evasion attacks and hash-inversion attacks. We attribute this robustness, in part, to the random variations inherent in the hash generation process. Building on this insight, we propose enhancing security by introducing additional randomization to the hash bits.

References

[1] Mauro Barni, Patrizio Campisi, Edward J Delp, Gwenaël Doërr, Jessica Fridrich, Nasir Memon, Fernando Pérez-González, Anderson Rocha, Luisa Verdoliva, and Min Wu, "Information forensics and security: A quarter-century-long journey," *IEEE Signal Processing Magazine*, vol. 40, no. 5, pp. 67–79, 2023.

[2] Hany Farid, "An overview of perceptual hashing," *Journal of Online Trust and Safety*, vol. 1, no. 1, 2021.

[3] Sivic and Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings ninth IEEE international conference on computer vision*. IEEE, 2003, pp. 1470–1477.

[4] Qingying Hao, Licheng Luo, Steve TK Jan, and Gang Wang, "It's not what it looks like: Manipulating perceptual hashing based applications," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 69–85.

[5] Wayne A Logan, "Citizen searches and the duty to report," *Ohio St. LJ*, vol. 83, pp. 939, 2022.

[6] Tracy Ith, "Microsoft's photodna: Protecting children and businesses in the cloud," *Retrieved from Microsoft News Center: https://news. microsoft. com/features/microsofts-photodna-protecting-children-and-businesses-in-the-cloud*, 2015.

[7] Antigone Davis and Guy Rosen, "Open-sourcing photo-and video-matching technology to make the internet safer," *Facebook Newsroom*, 2019.

[8] Anunay Kulshrestha and Jonathan Mayer, "Identifying harmful media in {End-to-End} encrypted communication: Efficient private membership computation," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 893–910.

[9] Jennifer Cobbe, "Data protection, eprivacy, and the prospects for apple's on-device csam detection system in europe," *SocArXiv*, 2021.

[10] Harold Abelson, Ross Anderson, Steven M Bellovin, Josh Benaloh, Matt Blaze, Jon Callas, Whitfield Diffie, Susan Landau, Peter G Neumann, Ronald L Rivest, et al., "Bugs in our pockets: The risks of client-side scanning," *Journal of Cybersecurity*, vol. 10, no. 1, pp. tyad020, 2024.

[11] Sean McKeown and William J Buchanan, "Hamming distributions of popular perceptual hashing techniques," *Forensic Science International: Digital Investigation*, vol. 44, pp. 301509, 2023.

[12] Anish Athalye, "Inverting photodna," *https://anishathalye.com/inverting-photodna/*, 2021.

[13] Gannon Burgett, "Photodna lets google, fb and others hunt down child pornography without looking at your photos," *PetaPixel, August*, 2014.

[14] Charles Arthur, "Twitter to introduce photodna system to block child abuse images," *The Guardian*, vol. 22, 2013.

[15] Evan Klinger and David Starkweather, "phash: The open source perceptual hash library," 2013.

[16] Lukas Struppek, Dominik Hintersdorf, Daniel Neider, and Kristian Kersting, "Learning to break deep perceptual hashing: The use case neuralhash," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022, pp. 58–69.

[17] Shubham Jain, Ana-Maria Creţu, and Yves-Alexandre de Montjoye, "Adversarial detection avoidance attacks: Evaluating the robustness of perceptual hashing-based client-side scanning," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 2317–2334.

[18] Jagdeep Singh Bhatia and Kevin Meng, "Exploiting and defending against the approximate linearity of apple's neuralhash," *arXiv preprint arXiv:2207.14258*, 2022.

[19] Jonathan Prokos, Neil Fendley, Matthew Green, Roei Schuster, Eran Tromer, Tushar Jois, and Yinzhi Cao, "Squint hard enough: attacking perceptual hashing with adversarial machine learning," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 211–228.

[20] Robert Grimm, "Putting the count back into accountability: An audit of social media transparency disclosures, focusing on sexual exploitation of minors," *arXiv preprint arXiv:2402.14625*, 2024.

[21] Muhammad Shahroz Nadeem, Virginia NL Franqueira, and Xiaojun Zhai, "Privacy verification of photodna based on machine learning," in ,. 93y42, 2019.

[22] A. Ygvar, "Apple neural hash to onnx," *https://github.com/AsuharietYgvar/ AppleNeuralHash2ONNX*, 2021.

[23] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, 2016, pp. 694–711.

[24] Zeyu Qin, Yanbo Fan, Hongyuan Zha, and Baoyuan Wu, "Random noise defense against query-based black-box attacks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7650–7663, 2021.

[25] Manjushree B Aithal and Xiaohua Li, "Mitigating black-box adversarial attacks via output noise perturbation," *IEEE Access*, vol. 10, pp. 12395–12411, 2022.

[26] Alain Hore and Djemel Ziou, "Image quality metrics: Psnr vs. ssim," in *2010 20th international conference on pattern recognition*. IEEE, 2010, pp. 2366–2369.

[27] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[28] Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger, "Simple black-box adversarial attacks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2484–2493.

[29] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin, "Black-box adversarial attacks with limited queries and information," in *International conference on machine learning*. PMLR, 2018, pp. 2137–2146.

[30] Sijia Liu, Pin-Yu Chen, Xiangyi Chen, and Mingyi Hong, "signsgd via zeroth-order oracle," in *International Conference on Learning Representations*, 2018.

[31] Jianbo Chen, Michael I Jordan, and Martin J Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," in *2020 ieee symposium on security and privacy (sp)*. IEEE, 2020, pp. 1277–1294.

[32] Martin Steinebach, "An analysis of photodna," in *Proceedings of the 18th International Conference on Availability, Reliability and Security*, 2023, pp. 1–8.

A Appendix / supplemental material

A.1 Proof of Proposition 1 of Section 3.3

Without loss of generality, consider the optimization problem (1). A common approach involves utilizing K randomly perturbed images to query the PHAs and then using the queried hashes to estimate the gradient as follows

$$\mathbf{g} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{\beta}\mathbf{u}_k D_h(\mathbf{h}_{\text{tar}}, \mathcal{H}(\mathbf{I}_0 + \delta + \beta\mathbf{u}_k)) \tag{4}$$

where $\beta\mathbf{u}_k$ represents random perturbation [29]. Firstly, due to the nature of PHAs, overly small perturbations may not induce any change in hash. Hence, relatively large β is required, but this renders the gradient estimation unreliable. Secondly, the hash $\mathcal{H}(\mathbf{I}_0 + \delta + \beta\mathbf{u}_k)$ randomly varies with the small perturbation $\beta\mathbf{u}_k$. The random variation can be modeled as a noise term Z with variance σ^2. Following the analysis of [25], this noise is amplified by $1/\beta$, leading to the gradient estimate $\mathbf{g}$ being affected by a noise power σ^2/β^2, hence making it unreliable.

Due to the random nature of PHAs, it is almost impossible to limit the change of hash to just one or a few hash bits by perturbations. Practical σ is usually around 0.1. β may be around 0.001. As per [25], the signal-to-noise-ratio (SNR) of the gradient estimation can fall well below -40 dB, which makes the adversary's optimization not convergent or converge extremely slowly. This implies that PHAs are theoretically robust to blackbox attacks under the query budget constraint.

A.2 Robustness Problem of PHAs to Normal Image Editing

One of the special issue of PHAs is that they seem not robust to some normal imaging editing methods, for which [32] examined PhotoDNA, while [11] investigated PDQ and NeuralHash. Both studies demonstrated that PHAs were robust to certain edits like compression but vulnerable to others like cropping. While such prior works have already extensively investigated image editing on PHAs, a new evaluation is necessary due to ongoing algorithm updates. Notably, NeuralHash has received at least one update since its first release. This study addresses the gap in knowledge by evaluating the robustness of the updated algorithms and, more importantly, by providing the first comparison of the three PHAs.

Assume the original input to the PHA is an image $\mathbf{I}_0$, and the true output is an N-bit hash $\mathbf{h}_0 = \mathcal{H}(\mathbf{I}_0)$, where $\mathcal{H}(\cdot)$ denotes the hashing algorithm. The client adversary applies standard image editing operations over $\mathbf{I}_0$ to obtain an adversarial image $\mathbf{I}_{\text{adv}}$ with a new N-bit hash $\mathbf{h}_{\text{adv}} = \mathcal{H}(\mathbf{I}_{\text{adv}})$. A successful attack means $\mathbf{h}_{\text{adv}}$ is sufficiently different from $\mathbf{h}_0$.

We experimented with four image editing operations that can keep the original high image quality: JPEG compression with random quality factors, resizing images with random scaling factors, filtering images with the vignette filter, or rotating images to random degrees. We calculated the hashes $\mathbf{h}_0$ and $\mathbf{h}_1$ of the images before and after the editing, and derived their distance $D_h(\mathbf{h}_0, \mathbf{h}_1)$. Using 1000 images from the ImageNet validation dataset, we obtained ASR(p) for various p based on (3), wherein the L_2 condition was removed by setting θ to infinity.

Experiment results are presented in Tables 3. The data for ASR(.3) is highlighted in black because $p = 0.3$ is the threshold we suggested to focus on. It is evident that all three PHAs were robust to compression and resizing, but not to filtering and rotation. In conjunction with the findings in [11][16], the PHAs are robust to many standard image editing operations except the following:

- PhotoDNA is not robust to: filtering, rotating, resizing, mirroring, bordering, cropping;
- PDQ is not robust to: filtering, rotating, mirroring [11], bordering [11], cropping [11];
- NeuralHash is not robust to: filtering, rotating, mirroring [11].

NeuralHash was the most robust among the three. Surprisingly, the newest edition of NeuralHash that we experimented with was still not robust to filtering, rotating, and mirroring, similar to the first edition experimented in [11][16]. Nevertheless, we believe this robustness problem can be mitigated by training the model to be invariant to such standard operations with data augmentations.

Table 3: Adversary's ASR(p) with Image Editing Attacks

ASR	Compress	Resize	Filter	Rotate
PhotoDNA ASR(.1)	3%	99%	100%	100%
PhotoDNA ASR(.2)	0%	55%	100%	100%
PhotoDNA ASR(.3)	**0%**	**0%**	**94%**	**100%**
PhotoDNA ASR(.4)	0%	0%	37%	98%
PDQ ASR(.1)	0%	27%	88%	100%
PDQ ASR(.2)	0%	0%	67%	100%
PDQ ASR(.3)	**0%**	**0%**	**44%**	**100%**
PDQ ASR(.4)	0%	0%	27%	100%
NeuralHash ASR(.1)	0%	22%	96%	100%
NeuralHash ASR(.2)	0%	0%	82%	100%
NeuralHash ASR(.3)	**0%**	**0%**	**57%**	**96%**
NeuralHash ASR(.4)	0%	0%	12%	31%

A.3 Extra Experiment Data of Targeted Hash-Evasion Attacks in Section 4.2

Besides the data shown in Section 4.2 and Table 1, more experiment data of targeted attacks are presented in Table 4. We find that the robustness is extremely strong because of ASR$(p) \approx 0$ across all p. The only exception was NeuralHash, which exhibited ASR(0.4) of 43% with SimBA. However, this isn't indicative of successful adversarial attacks; rather, it's due to numerous hashes generated by NeuralHash having $D_h(\mathbf{h}_0, \mathbf{h}_1)$ as low as 0.3. Under NeuralHash, many images are inherently adversarial when evaluated at $p = 0.4$.

Therefore, contrary to the prior claims that PHAs were not robust against targeted attacks [19], our findings demonstrate the strong robustness of all PHAs against targeted blackbox attacks. This is significant to the practical application of PHAs because the client adversary needs targeted attacks to ensure that its adversarial hashes do not inadvertently match illicit hashes in the database. The strong robustness means it is hard for the adversary to undertake either untargeted or targeted attacks without being caught.

Table 4: Adversary's ASR(p) in Targeted Blackbox Hash-Evasion Attacks. (**PH**: PhotoDNA. **PD**: PDQ. **NH**: NeuralHash)

ASR	SimBA	SimBA +HSJA	NES	NES+ HSJA	ZO-sign-SGD	ZOsign -SGD +HSJA
PH ASR(.1)	0%	0%	0%	0%	0%	0%
PH ASR(.2)	0%	0%	0%	0%	0%	0%
PH ASR(.3)	0%	0%	0%	0%	0%	0%
PH ASR(.4)	1%	**2%**	0%	0%	0%	0%
PD ASR(.1)	0%	0%	0%	0%	0%	0%
PD ASR(.2)	0%	0%	0%	0%	0%	0%
PD ASR(.3)	0%	0%	0%	0%	0%	0%
PD ASR(.4)	0%	0%	0%	0%	0%	0%
NH ASR(.1)	0%	0%	0%	0%	0%	0%
NH ASR(.2)	0%	0%	0%	0%	0%	0%
NH ASR(.3)	0%	0%	0%	0%	0%	0%
NH ASR(.4)	**43%**	11%	0%	0%	0%	0%
[19] PH ASR(.3) ($L_2 = 0.19$)			83%	(2×10^6 queries)		
[19] PD ASR(.3) ($L_2 = 0.42$)			100%	(6×10^5 queries)		

A.4 Extra Experiment Data of Untargeted Blackbox Hash-Evasion Attacks in Section 4.2

A.4.1 Experiment data and sample images of untargeted attacks without defense

Besides the experiment data shown in Section 4.2 and Table 2, extra experiment data are presented in Table 5. Sample adversarial images generated with our proposed JSHA with various pre-set threshold

p are depicted in Fig. 3 and Fig. 4, with their corresponding distortion metrics listed in Table 6. Notably, the images with $p = 0.4$ were deemed invalid adversarial images as their distortion was not under the threshold $\theta = 0.1$, resulting in extremely noisy images. Similarly, for PDQ, the adversarial images with $p = 0.2$ and 0.3 were also invalid.

Therefore, contrary to previous studies suggesting that PHAs were not robust against untargeted attacks, we found that some PHAs like PDQ and NeuralHash could be considered as robust if realistic constraints regarding image distortion and query budget are taken into account.

Table 5: Adversary's ASR(p) in Untargeted Blackbox Hash-Evasion Attacks. (**PH**: PhotoDNA. **PD**: PDQ. **NH**: NeuralHash.)

ASR	SimBA	SimBA +HSJA	NES	NES+ HSJA	ZO-sign-SGD	ZOsign -SGD +HSJA
PH ASR(.1)	100%	100%	18%	100%	1%	98%
PH ASR(.2)	80%	100%	2%	98%	0%	90%
PH ASR(.3)	1%	**92%**	0%	69%	0%	19%
PH ASR(.4)	0%	0%	0%	1%	0%	0%
PD ASR(.1)	0%	45%	0%	64%	0%	8%
PD ASR(.2)	0%	1%	0%	10%	0%	0%
PD ASR(.3)	0%	0%	0%	**1%**	0%	0%
PD ASR(.4)	0%	0%	0%	0%	0%	0%
NH ASR(.1)	4%	88%	0%	99%	0%	94%
NH ASR(.2)	2%	23%	0%	77%	0%	42%
NH ASR(.3)	0%	14%	0%	**28%**	0%	2%
NH ASR(.4)	0%	0%	0%	0%	0%	0%
[19] PH ASR(.3)= 90%, ($L_2 = 0.2$, HSJA with 3×10^4 queries)						
[17] PD ASR(.3)= 73%, ($L_2 = 0.1$, up to 8×10^6 queries)						

Table 6: Distortion of the Adversarial Image Samples in Fig. 3(b).

Measure	Threshold p	PhotoDNA	PDQ	NeuralHash
L_2	$p = 0.1$	0.02	0.07	0.02
	$p = 0.2$	0.02	**0.11**	0.04
	$p = 0.3$	0.05	**0.14**	0.07
	$p = 0.4$	**0.17**	**0.19**	**0.19**

A.4.2 Experiment data of untargeted attacks under proposed defense

Observing that some untargeted attack algorithms such as SimBA+HSJA achieved relataively high ASR(0.3) over some PHAs such as PhotoDNA, we experimented with the hash perturbation defense method proposed in Section 3.3. The level of our random hash perturbation was denoted as q, where $0 \leq q \leq 1$, which stand for the ratio of the hash bits that were randomly flipped.

Besides the experiment results presented in Section 4.2 and Table 2, more experiment data are presented in Tables 7, 8, and 9. It can be seen that random perturbation reduced ASR(p) drastically, which means the attacks became much less successful or even failed. Even a slight perturbation with $q = 0.1$ could prevent most of the attacks.

A.5 Extra Experiment Results of Hash-Inversion Attacks in Section 4.3

A.5.1 Experiment data and sample images of hash-inversion attacks without defense

Besides the experiment sample images presented in Section 4.3 and Fig. 2, a complete set of experiment data of image reconstruction quality are in Table 10. In addition, Fig. 5 shows sample images of MNIST dataset reconstruction, Fig. 6 shows sample images of STL-10 dataset reconstruction, whereas Fig. 7 shows sample images of CelebA dataset reconstruction. All these experiment results were obtained without the defense being applied.

Table 7: Adversary's ASR(0.1) in Untargeted Blackbox Hash-Evasion Attacks when Defense was Applied. (**PH**: PhotoDNA. **PD**: PDQ. **NH**: NeuralHash.)

ASR	SimBA +HSJA	NES+ HSJA	ZOsign -SGD +HSJA
PH q =0.1	100%	100%	100%
PH q =0.2	100%	100%	100%
PH q =0.3	100%	100%	100%
PD q =0.1	0%	0%	0%
PD q =0.2	0%	0%	0%
PD q =0.3	0%	0%	0%
NH q =0.1	21%	18%	15%
NH q =0.2	20%	18%	15%
NH q =0.3	21%	18%	15%

Table 8: Adversary's ASR(0.2) in Untargeted Blackbox Hash-Evasion Attacks when Defense was Applied. (**PH**: PhotoDNA. **PD**: PDQ. **NH**: NeuralHash.)

ASR	SimBA +HSJA	NES+ HSJA	ZOsign -SGD +HSJA
PH q =0.1	98%	99%	95%
PH q =0.2	98%	99%	95%
PH q =0.3	98%	99%	95%
PD q =0.1	0%	0%	0%
PD q =0.2	0%	0%	0%
PD q =0.3	0%	0%	0%
NH q =0.1	5%	5%	2%
NH q =0.2	5%	6%	3%
NH q =0.3	5%	6%	2%

Generally, the hash-inversion attacks worked with some success when there was a lot of regularity in the dataset (e.g., MNIST and CelebA). In this case, the hash-inversion algorithm gave an image of such regularity only. The longer hash values leaked more information and allowed better hash-inversion. Nevertheless, even in the best case, it was still impossible to match the reconstructed human faces with the true faces in the CelebA images.

When the dataset was diverse (e.g., STL-10), we were not able to learn any meaningful inversions. This demonstrated that the PHAs were robust against hash-inversion attacks in practical applications. In addition, PHAs with smaller number of output bits were always preferable because natural images were not very standardized, so the smaller number of bits could help to prevent information leakage/hash inversion.

The reason for [12] to draw the false conclusion of successful hash-inversion was because only CelebA images were used over the long PhotoDNA hashes, and the authors overlooked the fact that the faces can not be discriminated based on the reconstructed images.

A.5.2 Experiment data hash-inversion attacks under the proposed defense

To mitigate the potential hash inversion attacks over regular images, we experimented with the proposed random hash perturbation defense algorithm. Some sample images were shown in Fig. 2(b) and explained briefly in Section 4.3. Extra experiment data are shown in Tables 11, 12 and 13.

Compare these data with those in Table 10, it is clear that as we increased the level of perturbation to the hash bits, the quality of reconstructed images decreased as expected. Therefore, the random perturbation defense does help to defend against hash-inversion attacks.

Table 9: Adversary's ASR(0.3) in Untargeted Blackbox Hash-Evasion Attacks when Defense was Applied. (**PH**: PhotoDNA. **PD**: PDQ. **NH**: NeuralHash.)

ASR	SimBA +HSJA	NES+ HSJA	ZOsign -SGD +HSJA
PH q =0.1	27%	21%	5%
PH q =0.2	27%	21%	5%
PH q =0.3	27%	21%	5%
PD q =0.1	0	0%	0%
PD q =0.2	0%	0%	0%
PD q =0.3	0%	0%	0%
NH q =0.1	0%	2%	0%
NH q =0.2	0%	2%	0%
NH q =0.3	0%	0%	0%

Table 10: Quality of Adversarial Images Created in Hash Inversion Attacks (**PH**: PhotoDNA. **PD**: PDQ. **NH**: NeuralHash)

Dataset	Hash	L_2	SSIM	LPIPS
MNIST	PH	$\mathbf{0.08} \pm 0.02$	$\mathbf{0.79} \pm 0.02$	$\mathbf{0.18} \pm 0.03$
	PD	-	-	-
	NH	0.19 ± 0.05	0.35 ± 0.10	0.33 ± 0.08
CelebA	PH	$\mathbf{0.13} \pm 0.03$	$\mathbf{0.57} \pm 0.08$	$\mathbf{0.40} \pm 0.07$
	PD	0.23 ± 0.07	0.44 ± 0.07	0.43 ± 0.05
	NH	0.23 ± 0.05	0.29 ± 0.09	0.53 ± 0.07
STL-10	PH	$\mathbf{0.14} \pm 0.03$	$\mathbf{0.41} \pm 0.09$	$\mathbf{0.56} \pm 0.06$
	PD	0.24 ± 0.05	0.26 ± 0.05	0.65 ± 0.02
	NH	0.23 ± 0.05	0.14 ± 0.09	0.67 ± 0.07

Table 11: Quality of Inverted Images when Defended with q = 0.1 (**PH**: PhotoDNA. **PD**: PDQ. **NH**: NeuralHash)

Dataset	Hash	L_2	SSIM	LPIPS
STL-10	PH	0.14 ± 0.03	0.41 ± 0.00	0.56 ± 0.06
	PD	0.24 ± 0.05	0.26 ± 0.05	0.65 ± 0.02
	NH	0.23 ± 0.05	0.14 ± 0.07	0.67 ± 0.03

Table 12: Quality of Inverted Images when Defended with q = 0.2 (**PH**: PhotoDNA. **PD**: PDQ. **NH**: NeuralHash)

Dataset	Hash	L_2	SSIM	LPIPS
STL-10	PH	0.18 ± 0.04	0.26 ± 0.08	0.68 ± 0.05
	PD	0.23 ± 0.05	0.24 ± 0.08	0.65 ± 0.04
	NH	0.23 ± 0.05	0.17 ± 0.08	0.77 ± 0.07

Table 13: Quality of Inverted Images when Defended with q = 0.3 (**PH**: PhotoDNA. **PD**: PDQ. **NH**: NeuralHash)

Dataset	Hash	L_2	SSIM	LPIPS
STL-10	PH	0.20 ± 0.06	0.21 ± 0.09	0.71 ± 0.05
	PD	0.23 ± 0.06	0.20 ± 0.08	0.68 ± 0.04
	NH	0.23 ± 0.05	0.17 ± 0.08	0.78 ± 0.04

(a) Original image $\mathbf{I}_0$.

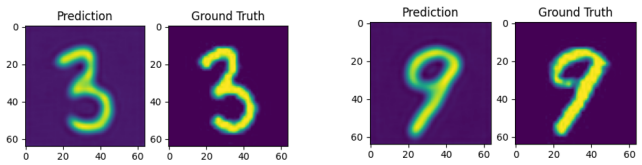
PhotoDNA PDQ NeuralHash

(b) Adversarial images $\mathbf{I}_{\mathrm{adv}}$.

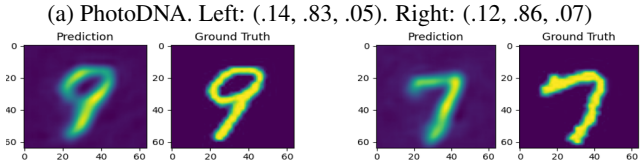
Figure 3: Samples of an original image and the adversarial images created by the proposed untargeted blackbox attack algorithm JSHA (NES+HSJA).

PhotoDNA PDQ NeuralHash

Figure 4: Samples of an original image and the adversarial images created by the proposed untargeted blackbox attack algorithm JSHA (NES+HSJA).

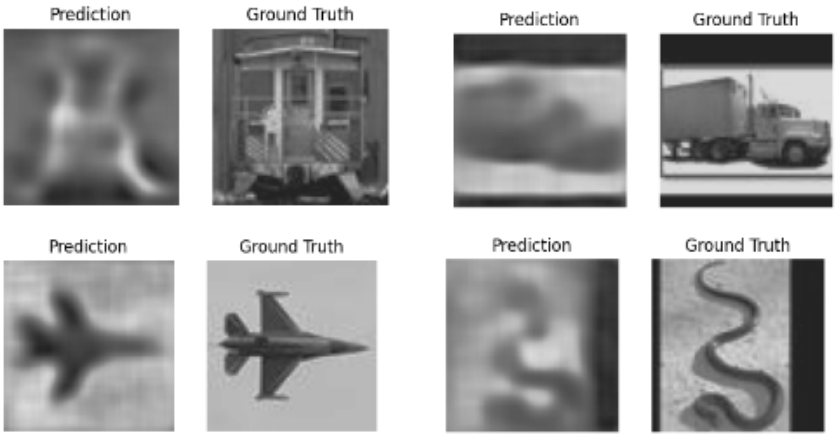


(a) PhotoDNA. Left: (.14, .83, .05). Right: (.12, .86, .07)

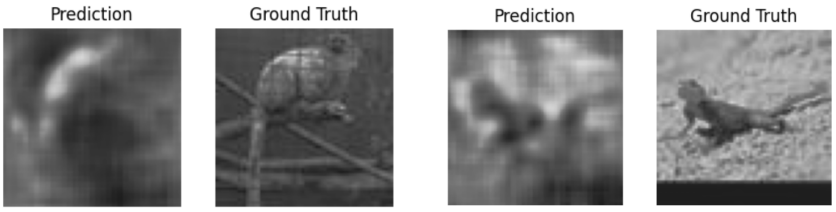


(b) NeuralHash. Left: (.26, .61, .10). Right: (.34, .52, .14)

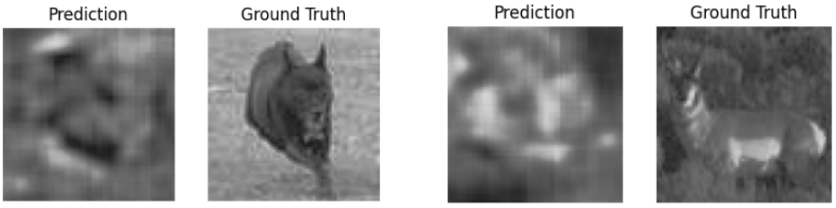
Figure 5: Samples of true MNIST images and adversarial images generated by hash inversion attacks based on hashes of (a) PhotoDNA, and (b) NeuralHash. The numbers are (L_2, SSIM, LPIPS) measures.



(a) Using PhotoDNA Hash bits

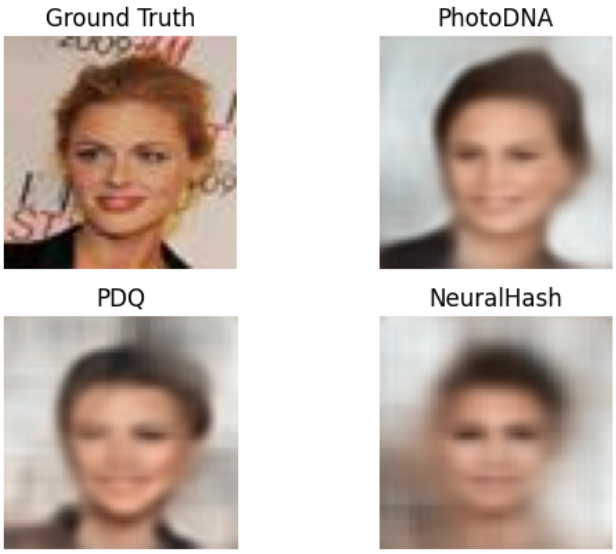


(b) Using PDQ Hash bits

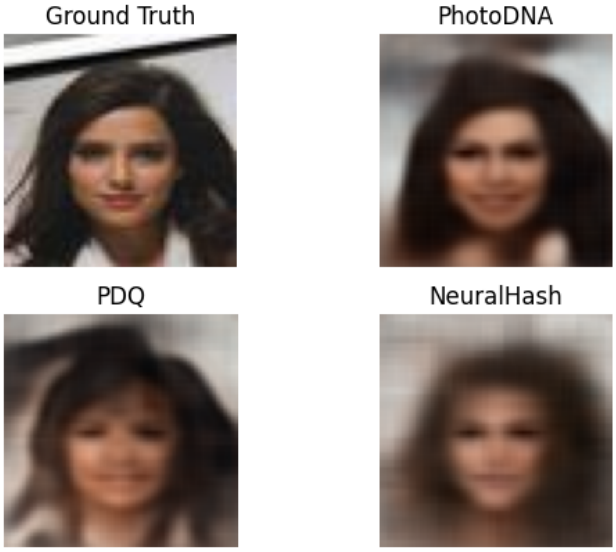


(c) Using NeuralHash hash bits

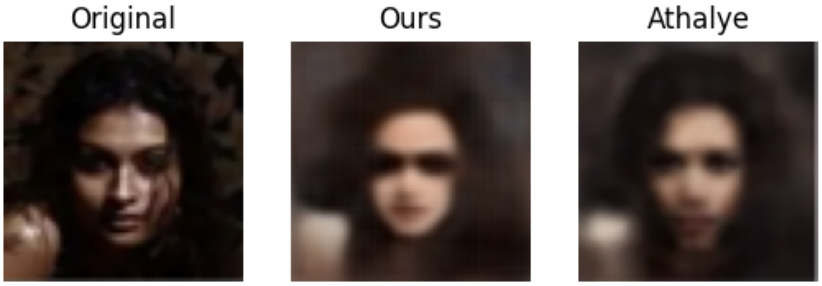
Figure 6: Samples of true STL-10 images and adversarial images generated by hash inversion attacks.



(a) Our model. $\{L_2$, SSIM, LPIPS$\} = \{$**.14**, **.65**, **.34**$\}$ (PhotoDNA), $= \{.15, .60, .36\}$ (PDQ), $\{.20, .40, .50\}$ (NeuralHash)



(b) Our model. $\{L_2$, SSIM, LPIPS$\} = \{$**.17**, **.63**, **.34**$\}$ (PhotoDNA), $= \{.21, .50, .34\}$ (PDQ), $\{.20, .47, .45\}$ (NeuralHash)



(c) Compare images generated by our model and [12].

Figure 7: Samples of true CelebA images and reconstructed images generated by hash inversion attacks from hashes of PhotoDNA, PDQ, and NeuralHash. The numbers are quality metrics $\{L_2$, SSIM, LPIPS$\}$ of reconstructed images.