Breaking Widely Deployed Perceptual Hash Functions: Black-Box Collisions in Apple NeuralHash and Microsoft PhotoDNA

Diane Leblanc-Albarel*
COSIC, KU Leuven

Bart Preneel
COSIC, KU Leuven

Abstract

Perceptual hash functions have been designed to detect multimedia copyright violations and illegal content. To achieve their purpose, they map inputs that are perceived as similar to close outputs. For many widely deployed schemes, however, both the design strategy and detailed specifications remain proprietary. Governments are now considering their extension to Client-Side Scanning (CSS) for end-to-end encrypted services, verifying content against illegal material before encryption. In 2021, Apple presented a detailed proposal for CSS based on the NeuralHash perceptual hash function. After strong criticism over privacy and security concerns, Apple withdrew the proposal, but NeuralHash remains deployed on all devices, with its current purpose undisclosed. In theory, brute-force collisions for NeuralHash (96-bit hash value) require 2^{48} evaluations. Shortly after the NeuralHash release, researchers showed it is easy to craft perceptually dissimilar collisions, to incriminate any user by sending an innocent image sharing the same hash value as illegal content. This work shows a more serious weakness: when inputs are restricted to human faces, we found several collisions between perceptually different images after only 2^{16} hash function evaluations. Unlike targeted attacks, our black-box approach requires no knowledge of the hash function design. We also demonstrate a high false negative rate (images that should share the same hash but do not). We further confirm the generality of our approach by studying PhotoDNA, Microsoft's widely deployed 1152-bit perceptual hash function. In the case of PhotoDNA, we found near-collisions at thresholds significantly lower than previously reported, appearing after between $2^{14.6}$ and 2^{17} evaluations depending on the threshold used. This is the first work to demonstrate exact natural collisions in NeuralHash and to identify natural near-collisions in PhotoDNA at such low thresholds. These results cast serious doubts on the suitability of these designs for large-scale client scanning, as they produce high false positive and false negative rates, and highlight the need to reassess their security and feasibility, particularly for large-scale applications where privacy risks and false positives have serious consequences.

Coordinated Vulnerability Disclosure. As part of our research, we followed coordinated vulnerability disclosure procedures by timely notifying Apple and Microsoft of our findings. Apple informed us that they had reproduced all of our results without exception, acknowledged the issues, and were investigating solutions. We have entered constructive discussions with Microsoft. We can confirm that the results presented for PhotoDNA in this paper are accurate for the version currently deployed. These exchanges have been focused on addressing the shortcomings we identified.

1 Introduction

Unlike cryptographic hash functions that produce completely different outputs for even small input changes, perceptual hash functions produce the same or similar outputs for perceptually similar inputs. This allows for the detection of perceptually similar multimedia content such as images, videos, or sounds.

Perceptual hash functions are widely used to detect copyright violations [30, 59], identify problematic content such as Child Sexual Abuse Material (CSAM) [3] or terrorist content [16], and allow biometric authentication [35]. Comparing hash values allows detection without exposing the content. The US National Center for Missing and Exploited Children (NCMEC) has reported a sharp increase in CSAM detections in recent years [53]. NCMEC and researchers such as Bursztein et al. [12] and Farid [26] have called for automated detection methods, including perceptual hashing, to address this problem. At the same time, the rise of end-to-end encryption has further hindered centralized detection.

In response, the UK [64] and the European Union [15] have proposed Client-Side Scanning (CSS), which scans content directly on user devices before encryption, enabling authorities to intervene before it is shared [3,41,44]. CSS has faced strong criticism from academics, industry and NGOs (see Abelson

*Contact author: diane.leblanc-albarel@kuleuven.be

et al. [1] for an overview). Experts warn that it enables mass surveillance, undermines privacy rights, and creates a chilling effect. Once deployed, such systems could be extended beyond CSAM detection to other criminal content, political dissent, or the identification of whistleblowers and journalists. Although some proposals aim to detect CSAM while preserving privacy [7], the technologies deployed offer weak privacy guaranties and remain vulnerable to false positives and evasion. Members of the EU Parliament [10] and experts have issued open letters and public statements highlighting these concerns [54, 55].

This paper evaluates two widely deployed perceptual hash functions: Apple's NeuralHash [3] and Microsoft's PhotoDNA [52]; the latter is used by NCMEC, hundreds of tech companies, and all Microsoft services for online CSAM detection. We focus on *false positives* (perceptually different images with identical or close hash values) and *false negatives* (perceptually similar images with different hash values). Earlier work on NeuralHash [60] showed that it is easy to cryptanalyse the function by creating false positives through image manipulation. Here, we consider large-scale deployments where legitimate unmodified image sharing could still lead to false accusations.

In its Q&A report [2] on NeuralHash, Apple addressed concerns about false positives:

> *"Will CSAM detection in iCloud Photos falsely report innocent people to law enforcement? No. The system is designed to be very accurate, and the likelihood that the system would incorrectly identify any given account is less than one in one trillion per year."*

The European Commission's 2023 report to Parliament and the Council on PhotoDNA [22] states:

> *"The most widely used tool is Microsoft PhotoDNA, used by over 150 organizations. PhotoDNA has been in use for more than 10 years and has a high level of accuracy. The rate of false positives is estimated at no more than 1 in 50 billion, based on testing."*

However, the estimate "1 in 50 billion"' originates from a 2019 report by its creator, Farid [25], which does not explain or justify the methodology used to derive this estimate.

To our knowledge, no independent and rigorous evaluation has confirmed these claims. We thus analyze the efficiency and accuracy of both functions at scale, assessing their suitability for real-world use. Our findings show that large-scale deployment of either NeuralHash or PhotoDNA would inevitably result in a substantial number of false positives, even in the absence of image manipulation.

The remainder of this paper is organized as follows. Section 2 introduces perceptual hashing and defines perceptually identical content. Section 3 examines the properties of NeuralHash, focusing on the collision behavior for different types of images. Section 4 presents quantitative results and collision examples for NeuralHash. Section 5 reports PhotoDNA collision results. Section 6 analyzes the impact of these collisions on large-scale deployments. Section 7 discusses the limitations of our work and the applicability of our results to CSAM detection. Finally, Section 8 summarizes our findings.

Disclaimer. We unequivocally condemn the creation and distribution of CSAM, which are serious crimes that must be addressed in our digital society. This paper aims to inform discussions on the efficacy and implications of deploying perceptual hashing and Client-Side Scanning at scale. Our analysis of NeuralHash and PhotoDNA is not a critique of Apple, Microsoft, or anti-CSAM initiatives, but highlights the risks of a large-scale deployment of perceptual hash functions. These functions serve as case studies underscoring the need for accuracy and robustness to avoid false positives. We hope that this work fosters further research and dialog to combat CSAM while balancing effectiveness and privacy.

2 Background

This section defines perceptually similar content, outlines the properties and building blocks of perceptual hash functions, and reviews major designs.

2.1 Perceptually Identical Content

Although content can consist of images, video, audio, and 3D or immersive environments, this paper limits itself to the first category which is the target of NeuralHash and PhotoDNA. Perceptually similar images are those that appear identical or nearly identical to a *human observer*. From a human perception perspective, similarity between images is typically characterized by the following factors, among others:

- **Color Consistency:** Images with minor differences in brightness, contrast, or color balance but the same overall composition are considered perceptually similar. For example, an image with slightly adjusted brightness remains perceptually similar to its original.
- **Structural Similarity:** Images with similar shapes, edges, and textures, even after minor geometric transformations such as rotations, translations, or small distortions, are considered perceptually similar. For instance, an image and its slightly rotated version would be perceptually similar.
- **Content Preservation:** Images with the same core content but reduced resolution, compression artifacts, or added overlays, such as watermarks or logos, are still perceptually similar, provided these modifications do not obscure the underlying scene. For example, a high-resolution image and a compressed or watermarked ver-

sion with some loss of detail are perceived as the same image.

- **Noise Robustness:** Images with minor noise additions, such as random pixel variations, or distortions, such as blurring, which do not significantly alter the perceived content, are considered perceptually the same. For instance, an original image and one with a limited amount of Gaussian noise are perceptually similar.

These four factors form the basis for our notion of perceptual similarity and are used consistently throughout the paper. They define the criteria by which images are classified as perceptually similar or distinct in our experiments, and all evaluations of collisions, near-collisions, and false positives or negatives are grounded in this definition to reflect human-perceived similarity rather than purely numerical differences.

Although some mathematical metrics [11, 14] can capture individual aspects of perceptual similarity, they do not account for all of these factors. We therefore assess perceptual similarity directly using the above criteria rather than relying solely on existing similarity metrics. Section 3 provides the exact definition used.

2.2 Properties

While cryptographic hash functions and perceptual hash functions share some similarities, they have different purposes and thus different properties. Both efficiently process large inputs and reduce them to short outputs in a deterministic way. However, in the context of perceptual hash functions the distinction between *natural collision* and *collision* is important. A *natural collision* refers to the event in which two inputs share the same output, without any intentional modification aimed at forcing their hash values to be equal. More generally, a *collision* corresponds to inputs sharing the same hash value regardless of whether the inputs have been modified or not. For simplicity, and unless stated otherwise, in this paper, we will refer to natural collisions simply as collisions, and to collisions resulting from modified inputs as *non-natural collisions*.

Given this terminology, the properties [21, 26] of perceptual hash functions in the context of image hashing are:

- **Preimage resistance:** Similar to the case of cryptographic hash functions, preimage resistance refers to the infeasibility of finding for a value $H(x)$ any input x' such that $H(x') = H(x)$. Note that this requirement extends to cases where equality is replaced by a distance condition $d(H(x), H(x')) \leq \tau$, where d denotes a metric such as the L_1 or L_2 distance (see Definitions 3 and 4 in Section 5 for details), and τ is a predefined (low) threshold. An additional requirement is the inability of an adversary to reconstruct or infer meaningful information about the original input x from its hash value $H(x)$.
- **Second preimage resistance:** Given x_1 and its hash value $H(x_1)$, it should be computationally infeasible to find an unmodified input x_2, perceptually different from x_1, such that $H(x_2) = H(x_1)$. Note that for a perceptual hash function, it is trivial to find a distinct but perceptually identical input x_2 with $H(x_2) = H(x_1)$. Similar to preimage resistance, this notion can be extended with a distance function.
- **Illegitimate collision resistance:** Perceptual hash functions should ensure that perceptually different inputs yield distinct or sufficiently distant hash values. More formally, it should be computationally infeasible to find two perceptually different inputs x_1 and x_2 such that $H(x_1) = H(x_2)$ or $d(H(x_1), H(x_2)) \leq \tau$. Note that this can be a natural collision (occurring between random inputs) or collision obtained by carefully modifying inputs.
- **Accuracy:** Perceptual hash functions should produce the same (or similar) hash values for perceptually identical or very similar inputs. This property ensures that minor variations in the input (such as slight changes in brightness or minor cropping in an image) do not result in significantly different hash values.

2.3 Perceptual Hashing Process

First-generation perceptual hash functions do not involve deep learning techniques. Examples include pHash [68] and Microsoft's PhotoDNA [59]. These designs typically rely on hand-crafted features and transformations to generate hash values that are robust to minor input changes.

Deep perceptual hash functions [45, 46], on the other hand, are based on a Machine Learning (ML) model. They involve training deep neural networks to learn feature representations that capture perceptual similarity. Deep perceptual hash functions are less common, as they are more recent. Research includes hashing for image retrieval [69], for label prediction [66], and for CSAM detection [3]. Perceptual hash functions typically follow similar sequences of steps to generate hash values [21].

The first step is preprocessing, which prepares the input image by normalizing it into a standardized format. This often involves resizing the image to fixed dimensions, such as (360×360) pixels, and normalizing pixel values to a specific range, e.g., $[-1, 1]$. Additional preprocessing may include computing image gradients or converting the image to grayscale. In deep perceptual hash functions, preprocessing relies on an ML model to extract features, in addition to or in place of these traditional steps.

The next step is hash value extraction, where specific image features are selected to derive the hash value. The features depend on the application. Common methods include computing the average color of the image, its gradient, or using

techniques such as Locality-Sensitive Hashing [28, 34] and Binary Reconstructive Embeddings [40].

Finally, the generated hash values are converted to binary strings and compared to other hash values using a comparison technique. The Euclidean distance [21, 26, 59] is the most common metric for comparisons.

2.4 Perceptual Hash Function Designs

Perceptual hash functions have been developed and deployed since the early 2000s [21]. We mention in the following some of the most notable designs and their applications.

One of the earliest implementations of perceptual hashing for content identification was YouTube's Content ID [30]. Content ID identifies copyrighted material to assist copyright holders in managing their rights.

The 2010 thesis of Zauner [39, 68] provides a detailed introduction to perceptual hash functions and introduces the open-source construction pHash, which inspired several other designs.

PhotoDNA [52] was developed by Microsoft and Farid in 2009. It is widely used for content moderation [59] and on platforms such as Gmail, Twitter, Facebook, Reddit, and Discord to detect illegal content, particularly CSAM. Despite its widespread use, some successful attacks have been reported (see Section 2.7).

eGlyph [16], based on PhotoDNA, was implemented by the Counter Extremism Project (CEP), a nonprofit international policy organization combating extremist ideologies, with the help of Farid. In 2018, eGlyph was used to detect extremist videos on YouTube.

In 2019, Facebook (now Meta) released PDQ and TMK+PDQF [23] as open-source perceptual hash functions based on pHash [18]. These functions are designed to enhance the identification and moderation of prohibited content, in particular CSAM, across Facebook's platform and beyond. The PDQ function is designed for images, while TMK+PDQF targets videos.

Apple introduced in 2021 NeuralHash [3], a perceptual hash function specifically designed for CSAM detection [9]. NeuralHash uses a convolutional neural network (CNN) to generate hash values from images, with the aim of detecting illegal content.

2.5 NeuralHash

NeuralHash has been designed to identify known CSAM images stored in iCloud Photos by comparing image hash values on the device to a database of known CSAM hash values provided by child safety organizations such as NCMEC. The deployment of NeuralHash involved integrating the hashing algorithm directly into the iOS operating system. When an image is uploaded to iCloud Photos, NeuralHash generates a hash value that is compared against the database. If a sufficiently high number of matches are found to meet a predefined threshold, the corresponding content can be decrypted, reviewed, and, if confirmed, the user may be reported to the authorities. The Apple approach has raised significant public debate and controversy. As a consequence, Apple officially withdrew the proposal and postponed the large-scale deployment of the CSAM detection system. Nonetheless, NeuralHash has been embedded in all Apple devices.

NeuralHash was designed to operate in conjunction with the Apple CSS framework. At a high level, Apple employs a mechanism based on derived cryptographic keys and threshold secret sharing to match content on user devices with encrypted reference material stored on the server. This approach is intended to ensure that a minimum number of matches is required before any decryption occurs. For a detailed description of the CSS protocol, including Apple's use of cryptographic headers and threshold cryptography, we refer the reader to the Apple technical report [3] and to Appendix A of the full version of this paper [43] .

For our experiments, we extracted the neural network and hash matrix used by NeuralHash from an iPhone with firmware version 16.2. We then used [67] to convert the NeuralHash model into ONNX format, allowing us to run NeuralHash across platforms and devices.

2.6 Microsoft PhotoDNA

Unlike NeuralHash, PhotoDNA relies exclusively on traditional image processing operations. Although Microsoft has never publicly disclosed the exact algorithm, several leaks have enabled researchers to obtain equivalent black-box implementations. The publicly available code allows retrieval of corresponding PhotoDNA hash values for any image input. However, except for some high-level descriptions of PhotoDNA [25], as of the date of submission there is no publicly available information detailing the mathematical steps used to compute a PhotoDNA hash value. The exact algorithm therefore remains undisclosed, and PhotoDNA can only be computed in a fully black-box setting. Nevertheless, the high-level description of PhotoDNA, as presented by Farid in [24], is summarized in Appendix A of the full version of this paper [43].

In this paper, we use leaked code [37] to execute PhotoDNA in a black-box setting. This implementation, derived from previously disclosed materials, has also been referenced in prior work [5, 57, 59].

2.7 Related Work

Perceptual hash functions have been studied in academic papers for over two decades. Farid provided a comprehensive introduction to the topic in 2021 [26], detailing the fundamental techniques and challenges. Du, Ho, and Cong present

in [21] a survey of perceptual hashing techniques and their applications.

Recent research in perceptual hashing has primarily focused on non-natural collisions (as defined in Section 2.2) and information leakage. Information leakage refers to the risk of leakage of information about the input from its hash value.

One prevalent method of attacking perceptual hash functions is through gradient-based hash attacks, which imperceptibly alter pixels to achieve specific outputs [13, 17, 19, 61, 62]. Various optimization techniques have been proposed to improve the effectiveness of these attacks [29, 48, 56].

Several papers have analyzed the resistance of perceptual hash functions to image modifications and the difficulty to recover original images from their hash values. This includes research on the robustness of perceptual hash functions derived from pHash [20, 36] and attempts to reconstruct original images using Binary Reconstructive Embeddings instead of Locality-Sensitive Hashing [65].

Recent work has revealed a series of attacks that exploit vulnerabilities in the internal structure of NeuralHash and other perceptual hash functions. In addition to information leakage and preimage reconstruction attacks [8, 32], Struppek et al. [60] showed how to create second preimages for attacking NeuralHash, where input images were imperceptibly modified to produce a non-natural collision. They also published a proof-of-concept implementation (other tools for second preimages can be found in [6, 38]). Struppek et al. further described classification attacks, in which hash values are used to categorize inputs, achieving a maximum success rate of 52% in some categories.

Similarly, attacks on pHash have shown how images can be manipulated to produce specific hash values [31]. Additionally, partial inversion of the inputs from PhotoDNA hash values has been demonstrated using neural networks [5].

Despite this research on cryptanalyzing perceptual hash functions and NeuralHash and PhotoDNA in particular, there remains a significant gap in the evaluation of these functions concerning empirical false positive rates and their performance in black-box settings. This gap underscores the necessity of our analysis, which aims to evaluate the performance, accuracy, and false positive rate of NeuralHash in the context of large-scale CSAM detection.

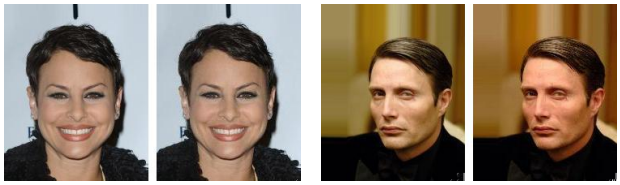
3 Analysis of Perceptual Hash Functions

This section introduces the datasets used for both NeuralHash and PhotoDNA, and defines what we refer to as a *perceptually identical image*. We then report our observations on the distribution of NeuralHash outputs and the (non-)independence of the hash bits. For both the distribution and independence analyzes, we begin with observations for different image types and then estimate the corresponding theoretical probability of collisions. For brevity, we present detailed observations only for NeuralHash; however, we obtained similar observations and identical conclusions for PhotoDNA.

3.1 Image Sets

We are using two distinct datasets in our analysis. The first consists of non-human images from the PASS dataset [4]. The second contains celebrity face images from the CelebA dataset [47]. Within the CelebA dataset ($N = 202\,599$ images), we identified at least $N_{\text{dup}} = 4\,629$ duplicates or perceptually similar images, i.e., images that should ideally map to the same hash value. These databases were chosen because they contain relevant data without violating the privacy of normal users. The size is sufficient to obtain statistically meaningful results while still allowing for manual verification of similarity.

Examples of identical and perceptually similar pairs are shown in Figure 1. Our primary goal is to evaluate whether hashing face images with perceptual hash functions yields different results compared to non-human images. Specifically, we aim to analyze the number of collisions and the statistical properties of individual bits of the hash values.



(a) Identical images

(b) Different filter

(c) Very close pose

Figure 1: Perceptually similar images

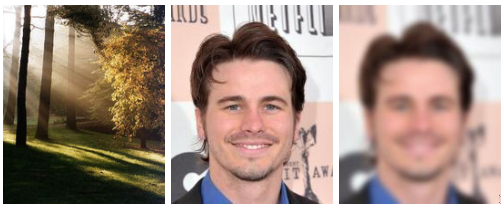
We selected face images for two main reasons: first, to approximate a use case related to CSAM detection, as such images often contain faces; second, to reflect their prevalence on mobile devices, making this a realistic scenario.

We also introduced different levels of blurring to stress-test the stability of the hashing functions. Facial blurring is likely to occur in CSAM (to hide the identity of perpetrators); our results show that blurring can induce a disproportionately high number of false positives and false negatives, reinforcing the instability we observe. Including blurred faces, therefore, strengthens the robustness evaluation by ensuring that the analysis covers realistic degradations encountered in practice.

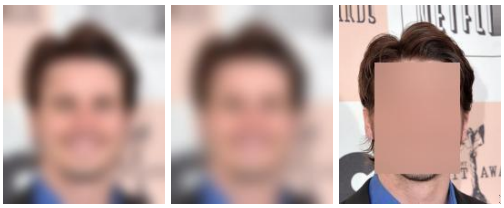
Based on these criteria, we constructed six image sets: non-human images, human face images, and human face images

with varying levels of blurring. An example of each type is shown in Figure 2. Throughout the paper, the abbreviation *BF* stands for *blurred faces*, and we refer to the six image types as follows:

- *Non Human*: images from the PASS [4] dataset.
- *Non BF*: images from the CelebA dataset [47].
- *Light BF*: CelebA images with light blur.
- *Medium BF*: CelebA images with medium blur.
- *High BF*: CelebA images with high blur.
- *BF only*: CelebA images with blur applied only to the face region.



(a) Non Human (b) Non BF (c) Light BF



(d) Medium BF (e) High BF (f) BF only

Figure 2: Examples of the six types of images used

3.2 Statistical Properties of Hash Values

This section presents both a theoretical analysis and experimental results to assess the properties of perceptual hash functions. We analyze the statistical distribution of hash values and test the independence of individual bits of the hash values using probabilistic models. In parallel, we conducted experiments on large-scale datasets to validate these theoretical findings, using NeuralHash as a primary example before extending our evaluation to PhotoDNA.

3.2.1 Distribution

We first experimentally analyze the distribution, in particular the uniformity and variance, of NeuralHash bit values for different types of image. We then introduce three propositions that help determine the theoretical number of hash function evaluations required before obtaining an illegitimate collision.

Experiments. As stated in Section 2, the distribution of the individual output bits of a hash function must be uniform with a small variance to prevent information leakage about the input and to resist second preimage and collision attacks.

To compute the average distribution of NeuralHash bits for each image type, we hashed 30 000 randomly selected images from each set. For each type of image, we count, for each of the 96 bits, how often it equals 1 or 0. The results for the Non Human and Non blurred faces types are presented in Figure 3a, where the percentage of times each bit is equal to 1 is plotted. Under a Bernoulli model with $n = 96$ bits and $p = \frac{1}{2}$, the expected proportion is 50%, with standard deviation $\sigma = 1/(2\sqrt{n})$; hence, all points should lie within the range 35%–65% (i.e., $\pm 3\sigma$).

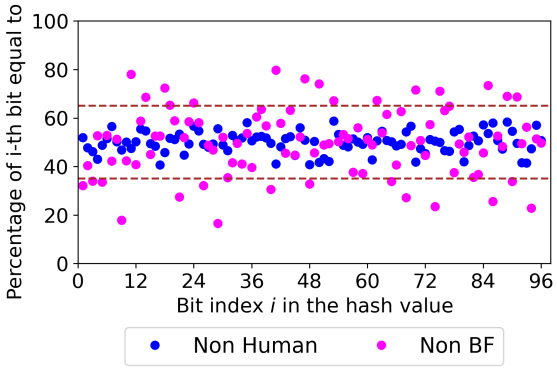
For Non Human images, the results meet expectations, with the percentage of bits equal to 1 uniformly distributed around 50%. In contrast, for Non blurred human face images (pink dots in Figure 3a), a significant number of bits deviate from being equal to 1 about 50% of the time. For Non Human images, none of the output bits equals 1 less than 35% or more than 65% of the time. In contrast, for human face images, 30 bits fall outside this range, with even 2 bits being less than 20% or more than 80% of the time equal to 1.

The results for the four remaining image types are presented in Figure 3b. Hash values are even less uniformly distributed than for Non blurred faces. For lightly blurred images (orange dots), 39 bits fall outside the $35\% - 65\%$ range. For images with only the face blurred (green dots), 49 bits fall outside the range. For medium blurred images (black dots), 51 bits fall outside the range, and for highly blurred images, 61 out of the 96 bits fall outside the $35\% - 65\%$ range.

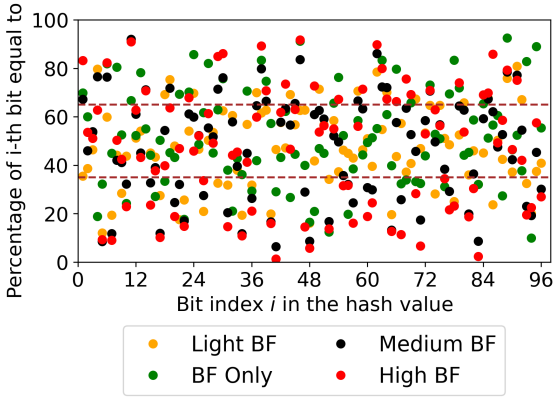
Theoretical Collision Probability with the independence hypothesis. An illegitimate collision corresponds to the event in which the hash values of two perceptually different images are the same. For regular hash functions, this event should be exceedingly rare. For a uniformly distributed 96-bit hash value (NeuralHash), where each bit is independent of the others, the birthday paradox [27] states that the average number of hash values required before encountering an illegitimate collision is 2^{48}.

Assuming that all bits of the hash value are mutually independent and considering the observed distribution for each dataset, Proposition 1 provides the probability that two hash values are identical. Proposition 2 provides the expected number of hash values required before the first illegitimate collision. To make this result more practical, Proposition 3 provides a tight bound on this expectation.

Proposition 1. *Consider a hash value of n bits and a vector p for which the i-th element p_i denotes the probability that the i-th bit of the hash value is equal to* 1 *($0 \leqslant p_i \leqslant 1$). If E_p^n*



(a) Non Human vs. Non blurred faces (Non BF)



(b) Blurred Human Faces

Figure 3: Bit distributions for each image type

denotes the event that two hash values are equal, then

$$\mathbb{P}(E_p^n) = \prod_{i=1}^{n} (p_i^2 + (1-p_i)^2).$$

Proof. The proof is provided in Appendix A. □

Proposition 2. *Denote by $\mathbb{P}(E_p^n)$ the probability that two n-bit hash values with distribution p are equal and define $U = 2^n$. The expected number of hash values to compute before the first collision $\mathbb{E}(D_p^n)$ is equal to:*

$$\mathbb{E}(D_p^n) = 2 + (U+1) \times y^{\frac{U(U+1)}{2}} + \sum_{x=1}^{U-1} y^{\frac{x(x+1)}{2}}$$

$$\text{with } y = 1 - \mathbb{P}(E_p^n).$$

Proof. The proof is provided in Appendix A. □

The exact computation of $\mathbb{E}(D_p^n)$ is infeasible due to the sum over the U elements ($U = 2^{96}$ for NeuralHash and $U = 2^{1152}$ for PhotoDNA). However, Proposition 3 gives a bound for $\mathbb{E}(D_p^n)$ that is computationally efficient.

Proposition 3. *Denote by $\mathbb{P}(E_p^n)$ the probability that two n-bit hash values with distribution p are equal and define $U = 2^n$. The expected number of hash values to compute before the first collision $\mathbb{E}(D_p^n)$ is upper bounded by:*

$$\mathbb{E}(D_p^n) \leq 1 + (U+1) \times y^{\frac{U(U+1)}{2}} + \frac{\theta_2(0; y^{\frac{1}{2}})}{2 \times y^{\frac{1}{8}}},$$

with $\theta_2(0; y^{\frac{1}{2}})$ the Jacobi theta function $\theta_2(z;q)$ with $z = 0$ and $q = y^{\frac{1}{2}}$ and with $y = 1 - \mathbb{P}(E_p^n)$.

Proof. The proof is provided in Appendix A. □

Conclusion on distribution. Using Proposition 3 together with the NeuralHash distributions observed in Figure 3, we construct Table 1. Since the observed distributions deviate from the uniform distribution (i.e., bits are not equal to 1 exactly 50% of the time), the table provides the theoretical number of hash evaluations required before the first illegitimate collision for each image type.

Table 1: Expected number of hash operations before the first illegitimate collision using NeuralHash

Type	Non Human	Non BF	Light BF	BF only	Medium BF	High BF
$\mathbb{E}(D_p^n)$	$2^{47.8}$	$2^{43.5}$	$2^{41.1}$	$2^{38.9}$	$2^{37.5}$	$2^{34.1}$

From Table 1, we conclude that, except for Non Human images, the expected number of hash values to compute before the first NeuralHash illegitimate collision is significantly lower than the theoretical value 2^{48}. For other image types, the expected number of hash values to compute before an illegitimate collision varies between $2^{43.5}$ (Non blurred faces) and $2^{34.1}$ (High blurred faces). Note that these values assume that every hash value bit is independent of the others. If this independence does not hold, the expected number of hash values before reaching an illegitimate collision decreases substantially.

3.2.2 Independence of Bits

In the previous section, we assumed the independence of the hash bits. Here, we test this hypothesis. If the bits are truly independent, the value of any bit j should not provide information about the value of another bit i with $j \neq i$.

Simple Matching Coefficient. The simple matching coefficient (*SMC*) is a statistic ranging from 0 to 1 used to compare the similarity of symmetric binary vectors. When two vectors are identical, the *SMC* is equal to 1; when they differ completely (opposite values at all positions), the *SMC* is equal to 0. For two independent random binary vectors, the expected *SMC* is 0.5. It is worth noting that while the *SMC* can detect non-independence, it cannot confirm independence.

For two bits from the same hash value h, Definition 1 gives $m_{i,j}(h)$, the simple matching value computed for bits i and j only, denoted $h(i)$ and $h(j)$ respectively. Definition 2 then extends this notion to define the *SMC* over a sample of N hash values of n bits.

Definition 1. *For a hash value h, $m_{i,j}(h)$ is defined as:*

$$m_{i,j}(h) = \begin{cases} 1 \text{ if } h(i) = h(j) \\ 0 \text{ if } h(i) \neq h(j). \end{cases}$$

Definition 2. *Consider a set D_p^n of ℓ ordered hash values of length n. Given h_k, the k-th element of D_p^n, the SMC of bits i and j is defined as:*

$$SMC_{i,j}(D_p^n) = \frac{1}{N} \cdot \sum_{k=1}^{\ell} m_{i,j}(h_k).$$

By definition, $\forall i, j$, $SMC_{i,j}(D_p^n) = SMC_{j,i}(D_p^n)$ and $SMC_{i,j}(D_p^n) = 1$ when $i = j$.

The Simple Matching Matrix (*SMM*) of a set D_p^n is obtained by computing the *SMC* for every pair of bits. This *SMM*is used to compute the *similarity* between every pair of bits of hash values from each of the six image types.

Bits Sample Similarity. For each image type, the *SMM* is computed using Definition 2 and from 30 000 random hash values per set. Table 2 summarizes, for each set, the values of the NeuralHash *SMM* that fall within different ranges, from 0.0–0.1 (pairs of bits almost always opposite) to 0.9–1.0 (pairs almost always equal). For independent bits, 100% of these values should fall within the range 0.4–0.6.

Conclusion on Independence. Table 2 shows that for Non Human images, only 0.22% of NeuralHash values fall outside the 0.4–0.6 range, corresponding to 20 of the 9120 possible pairs. A sufficient number of values are close to 0.5, which is not enough to conclude dependence.

However, for all other image types, a significant number of NeuralHash values fall outside the 0.4-0.6 range. For Non blurred face images, 13.97% of pairs are out of range. For blurred faces, this percentage ranges from 23.3% (Light BF) to 47.75% (High BF). For the blurred face only set, 30.39% of the values lie outside the range.

These results indicate that, except for Non Human images, NeuralHash bits are not independent. Consequently, Table 1 likely overestimates the number of hash values required before an illegitimate collision.

4 NeuralHash Collisions Observed in Practice

The results from Section 3 show that, except for Non Human images, NeuralHash collisions will occur far earlier than the theoretical bound of 2^{48} evaluations.

Therefore, we hashed all 202 599 images of each set, searching for illegitimate collisions between hash values. We classified collisions as *legitimate* when the colliding hash values correspond to images that are identical or perceptually similar, and as *illegitimate* (false positives) when the colliding hash values correspond to perceptually different images. In the few cases where colliding images could not clearly be classified as perceptually similar or not, we conservatively treated them as legitimate. We then determined the number of illegitimate collisions (false positives) and illegitimate non-collisions (false negatives). For the Non Human dataset, we observed no illegitimate collisions for either NeuralHash or PhotoDNA, resulting in a false positive rate of zero. Consequently, this dataset is omitted from the tables and figures in the following sections, which focus on datasets exhibiting non-zero false positive and false negative rates.

4.1 Illegitimate Collisions (False Positives)

We effectively found numerous collisions between perceptually different images, confirming the existence of false positives in practice. The number of illegitimate collisions for each image type, as well as the number of hash evaluations required before the first collision, are presented in Table 3. Since some collisions involve more than two images (multiple images sharing the same hash), we count the number of illegitimate collisions as the number of images sharing their hash values with at least one perceptually different image.

Table 3 shows that out of the $N = 202\,599$ hash values computed for each image type, the Non Human set is the only one without any illegitimate collisions. All other image types exhibit several illegitimate collisions. The number of false positives observed in sets of only $N = 202\,599$ images suggests much higher collision rates when millions or billions of images are hashed. Section 6 presents estimates for large-scale scenarios.

Figure 4 presents an example of a collision obtained for each image type. In particular, for all blurred face images (light, medium, and high), there are hash values shared by three or more perceptually different images. For example, three light blurred images sharing the same hash are shown in Figure 4e. For the medium blurred face set, eight different hash values are each shared by three images, and four hash values are each shared by four images. For the highly blurred face set, such collisions are even more frequent, with two hash values each shared by five different images.

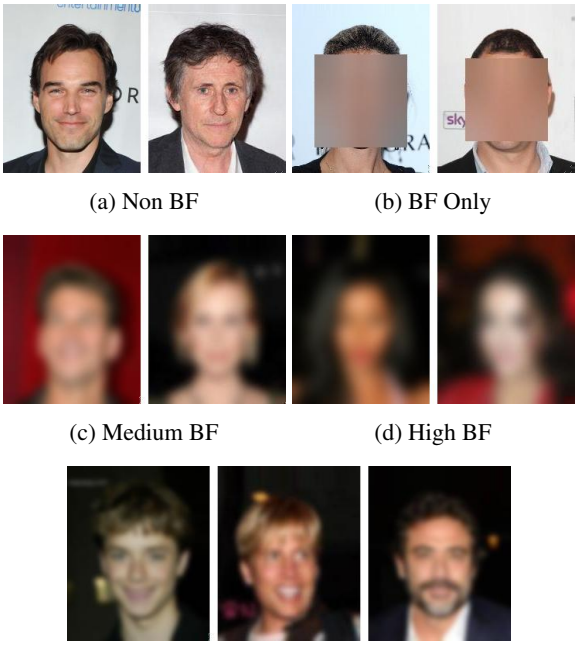
A new version of NeuralHash appears to have been deployed on macOS devices since early 2024, without any official explanation from Apple. We evaluated the false positive rates of this updated implementation; detailed results are provided in Appendix B. These results are very similar to those

Table 2: Similarity score (*SMC*), in percentage, for each type of image

Type \ *SMC*	0–0.1	0.1–0.2	0.2–0.3	0.3–0.4	0.4–0.5	0.5–0.6	0.6–0.7	0.7–0.8	0.8–0.9	0.9–1
Non Human	0.0	0.0	0.0	0.18	48.6	51.18	0.04	0.0	0.0	0.0
Non BF	0.0	0.0	0.29	6.97	43.05	42.98	6.56	0.13	0.02	0.0
Light BF	0.0	0.02	1.1	10.72	38.62	38.07	10.48	0.96	0.02	0.0
BF Only	0.0	0.2	2.7	11.64	33.93	35.68	12.7	2.89	0.26	0.0
Medium BF	0.0	0.33	3.53	15.7	31.47	29.69	14.76	3.88	0.64	0.0
High BF	0.2	1.69	6.67	15.75	26.45	25.8	13.64	7.61	2.08	0.11

Table 3: NeuralHash False positives and false negatives for each image type

	Non BF	Light BF	BF only	Medium BF	High BF
# illegit. coll.	12	12	6	126	270
# images involved in illegit. coll. (q)	24	25	12	260	588
False pos. rate (%)	0.01	0.01	0.005	0.13	0.29
# hashes before first illegit. coll.	$2^{16.1}$	$2^{15.2}$	$2^{16.1}$	$2^{12.6}$	$2^{11.6}$
# legit. coll. (N_{legit})	1 559	1 513	1 005	1 864	2 333
False neg. rate (%)	64.7	65.9	77.3	57.9	47.3



(a) Non BF (b) BF Only

(c) Medium BF (d) High BF

(e) Light BF

Figure 4: Examples of NeuralHash collisions observed for each image type

obtained with the previous model, leading us to the same conclusions regarding false positive rates.

4.2 False Negatives

As defined in Section 3.1, we identified at least $N_{\text{dup}} = 4\,629$ images in the CelebA dataset that should produce legitimate collisions. Given N_{legit} observed legitimate collisions, the minimum false negative rate (FNR) is computed as in Equation (1):

$$\text{FNR} = 100 \cdot \frac{N_{\text{dup}} - N_{\text{legit}}}{N_{\text{dup}}}. \quad (1)$$

Table 3 reports the legitimate collisions observed and the resulting FNR. Note that N_{dup} is a lower bound: in the medium and highly blurred sets, additional legitimate collisions may occur when distinct images become perceptually identical after blurring. Despite this, many such cases are missed, indicating a substantial false negative issue.

For the Non blurred face, Light blurred face, and Blurred faces Only image types, a significant number of perceptually identical images do not share the same hash values. In the Blurred faces Only set, this rate reaches 77.3%. These results indicate that the function does not reliably detect perceptually identical images. This ratio should be considered together with the false positive rate. For the Blurred faces Only set, for instance, the number of illegitimate collisions is lower than in other sets, but the number of legitimate collisions is also lower. This suggests that the function is particularly inaccurate when only faces are blurred.

It should be noted that our experiments consider only *exact collisions* in NeuralHash, whereas Apple's published documents [3] indicate that *near-collisions* are also taken into account. Including near-collisions would likely reduce the false negative rate, since more perceptually identical images would map to sufficiently close hashes. However, because the false positive rate is already prohibitively high in practice, accepting near-collisions would only exacerbate the problem by dramatically increasing the number of false positives.

5 PhotoDNA Collisions Observed in Practice

To assess whether the vulnerabilities identified in NeuralHash are inherent to the design of perceptual hashing, we conducted similar black-box experiments on Microsoft's PhotoDNA. Like NeuralHash, PhotoDNA is widely deployed for content moderation and is the primary function used by NCMEC to detect CSAM. PhotoDNA produces a 1152-bit hash value and, as indicated by Microsoft [25, 26, 52], near-collisions under a given threshold are used to flag similar images. Given the large size of the hash value, exact collisions are expected only for identical images, making near-collisions the mandatory mechanism for detecting perceptually similar ones.

In this section, we present the near-collisions and false negatives observed for PhotoDNA under different thresholds. For clarity and brevity, we focus on Non blurred face images, since the results already vary significantly with the threshold. As with NeuralHash, even slight blurring further increases the number of illegitimate collisions.

5.1 Experimental Setup

PhotoDNA's design requires comparing hash values using a distance metric, typically L_1 or L_2, since perceptually similar images yield close but not identical hash values. Thus, the choice of threshold directly determines whether two images are considered near-collisions.

Definition 3. *Given two hash vectors $x, y \in \mathbb{R}^n$, the L_1 distance is defined as:*

$$L_1(x,y) = \sum_{i=1}^{n} |x_i - y_i|,$$

Definition 4. *Given two hash vectors $x, y \in \mathbb{R}^n$, the L_2 distance is defined as:*

$$L_2(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}.$$

Circumventing PhotoDNA's detection mechanism requires finding perceptually different images whose hash values fall below the similarity threshold. Microsoft has not disclosed the specific distance metric used for PhotoDNA matching or the precise threshold values that they apply. However, previous research [36, 59] has analyzed the function and recommended using the L_2 distance metric with a threshold between 150 and 175. According to this study, these values prevent false positives while ensuring more than 99% true positive detection. Additionally, [57] employs the L_1 distance metric and suggests that setting an L_1 threshold of 1800 allows to avoid all false positives.

On the basis of these observations, we conducted experiments using the CelebA dataset. Specifically, we computed PhotoDNA hash values for all Non blurred face images using the implementation provided in [37]. We then computed the distances L_2 and L_1 between all pairs of images, counting both legitimate collisions (cases where perceptually identical images yield similar hash values) and illegitimate collisions (cases where perceptually distinct images yield similar hash values). This methodology mirrors the approach we employed for NeuralHash, enabling a direct comparison of results.

In Sections 5.2 and 5.3, we present the results using the L_2 distance. The results for the L_1 distance, with a threshold at 1 800 as in [57], are less favorable to PhotoDNA. Thus, for brevity and to remain conservative, we present the results for the L_2 distance in this section. Results for the L_1 distance are provided in Appendix F of the full version of the paper [43].

5.2 Illegitimate Collisions (False Positives)

Figure 6 presents the number of legitimate and illegitimate collisions according to the L_2 distance. Although [59] extends the analysis up to a distance of 500, we limited our study to 225 as beyond this threshold the number of illegitimate collisions increases dramatically.

Table 4 summarizes the number of illegitimate collisions and the number of hash evaluations required before the first illegitimate collision for the two representative thresholds (150 and 175). Figure 5 provides concrete examples of near-collisions observed at thresholds 150 (Figure 5a) and 175 (Figure 5b). For a threshold of 150, only 1 illegitimate collision is observed, corresponding to a false positive rate of 0.001%, with the first illegitimate collision appearing after approximately 2^{17} hash evaluations. Increasing the threshold to 175 results in 32 illegitimate collisions (false positive rate 0.03%) and reduces the number of hash operations before the first illegitimate collision to $2^{14.6}$.

(a) Threshold 150 (b) Threshold 175

Figure 5: Examples of PhotoDNA near-collisions observed with L_2 distance at thresholds 150 and 175

Previous work [59] reports the first illegitimate collision at an L_2 distance between 200 and 225, claiming no false positives below 200. In contrast, our experiments identify the first illegitimate collision at a much lower distance of 144 and show that raising the threshold even moderately (e.g., to 175) significantly increases the false positive rate. Although the rates reported in Table 4 may seem low for small-scale deployments, they would become significant when hashing millions or billions of images, as discussed in Section 6.

Table 4: False positive and false negative results for PhotoDNA with L_2 distance at thresholds 150 and 175

Threshold	150	175
# illegit. collisions	1	32
# images involved in illegit. coll. (q)	2	64
False positive rate (%)	0.001	0.03
# hashes before first illegit. coll.	2^{17}	$2^{14.6}$
# legit. collisions (N_{legit})	3 506	3 800
False negative rate (%)	24.1	17.8

5.3 False Negatives

Table 4 reports the number of legitimate collisions for each threshold and the corresponding false negative rates, computed using Equation (1) with N_{legit} from Table 4. For threshold 150, the false negative rate is 24.1%, while at threshold 175 it decreases to 17.8%. These results contradict previous claims in [57, 59] that thresholds in the range 150–175 can maintain a false negative rate below 1% while eliminating false positives. While a threshold of 150 does indeed produce a relatively low false positive rate (one illegitimate collision after only 2^{17} hash operations, which is a high rate for large-scale applications), it comes at the cost of missing nearly one quarter of perceptually identical images. Conversely, increasing the threshold to 175 improves the number of legitimate collisions but also substantially raises the false positive rate. This trade-off limits the effectiveness of PhotoDNA for large-scale deployment and contrasts sharply with the claims made by Microsoft [25] and previous studies [59].

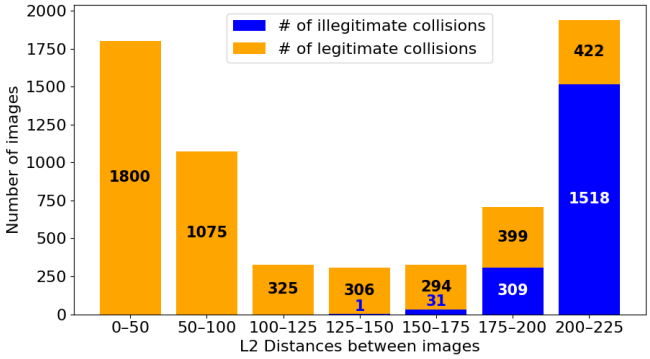


Figure 6: PhotoDNA near-collisions according to L_2 distance

6 Estimation for Large-Scale Applications

In this section, we use the empirical results obtained on the CelebA dataset, consisting of $N = 202\,599$ images, to estimate the expected false positive and false negative rates in the context of large-scale CSAM detection. We first present the probabilistic model used for our estimations. We then apply this model to NeuralHash and PhotoDNA.

6.1 Model Used for Approximation

To estimate the number of illegitimate collisions when hashing more images than in our current sets, we approximate the security by the equivalent output length of a uniform hash function with independent bits. This approach provides a worst-case scenario, ensuring that our estimates are conservative. In [42], Lamberger et al. propose a similar method for estimating the effective size of uniform, independent hash values when accounting for collisions and near-collisions.

Given the birthday paradox [27], we can estimate the number of images that share hash values. Let q denote the number of images that illegitimately share their hash values with at least one other image. For uniformly distributed and independent hash values, Equation (2) provides this estimate for a set of N images.

$$q = N\left(1 - \left(\frac{2^n - 1}{2^n}\right)^{N-1}\right). \tag{2}$$

To address the non-uniformity and interdependence of bits observed in both NeuralHash ($n = 96$) and PhotoDNA ($n = 1152$), we approximate each function by an *ideal* hash producing $n' < n$ uniformly distributed and independent bits. If the observed collision rate exceeds the 2^{-n} rate expected from an ideal n-bit hash function, then the function must exhibit reduced entropy (due to biased or correlated bits or other structural patterns) and thus behaves, in terms of collision probability, as though it had only n' effective independent bits. Approximating it by an ideal uniform hash of length n' preserves the relevant collision behavior while yielding a conservative upper bound, as any deviation from uniformity can only increase the likelihood of collision. This allows us to apply standard probabilistic models to estimate the probability of a collision.

We derive Equation (3) as a reformulation of Equation (2):

$$n' \approx -\log_2\left(1 - \left(1 - \frac{q}{N}\right)^{\frac{1}{N-1}}\right). \tag{3}$$

We compute n' from Equation (3) using the values of q given in Table 3 (for NeuralHash) and Table 4 (for PhotoDNA), with $N = 202\,599 - N_{\text{dup}} = 197\,970$ unique images,[1] where $N_{\text{dup}} = 4\,629$.

The obtained value n' represents the size of an equivalent hash function with uniformly distributed and independent output bits. We then apply the standard birthday bound (Equation (2)) for any number N of hash values to approximate the expected number of illegitimate collisions for each image

[1] We subtract the N_{dup} duplicate images, as these correspond to legitimate collisions.

type, and we validate this approximation against our experimental dataset. Tables 5 and 6 present the resulting n' values for NeuralHash and PhotoDNA, respectively.

6.2 Estimation of NeuralHash Performance at Large Scale

We extrapolate the number of illegitimate collisions for each image type to assess false positives in large-scale deployments. Figure 7 compares the approximated (blue) and observed (orange) number of collisions for medium and highly blurred images, showing a close match between theory and experiment. For other types (Non BF, Light BF and BF only), the number of false positives on the full 96-bit output is too small for a meaningful comparison; however, as validated on 56-bit subsets in Appendix C, the approximation remains accurate.

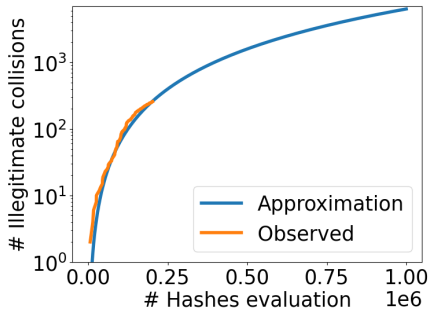
The consistency between observed and predicted false positives confirms the reliability of our approximation method. Table 5 summarizes the effective hash size n' for each image type and the corresponding estimated false positive rates for sets of 1 million, 10 million, and 100 million images; in what follows, M denotes one million.

Table 5: Estimated NeuralHash equivalent hash size (n') and false positive rates

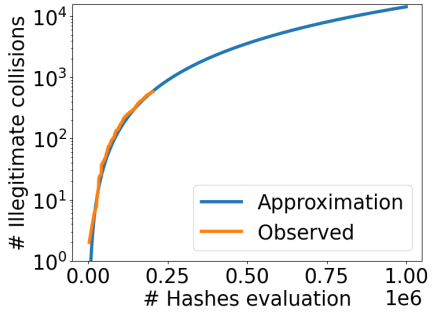
Type	Non BF	Light BF	BF only	Medium BF	High BF
Size of equivalent hash (n')	30.7	30.6	31.9	27.2	26.0
Estimated false positive rate					
Approx. for 1 M hashes	0.06%	0.06%	0.03%	0.65%	1.48%
Approx. for 10 M hashes	0.55%	0.63%	0.25%	6.36%	13.84%
Approx. for 100 M hashes	5.46%	6.18%	2.52%	48.25%	77.51%

From Table 5, we observe that the estimated false positive rates range from 0.03% to 1.48% for 1 M hash values. For 10 M hash values, the rates increase significantly, ranging from 0.25% to 13.84% depending on the image type. At 100 M hash values, the false positive rates become critical, with values between 2.52% and 77.51%, indicating that large-scale deployments would inevitably lead to substantial numbers of illegitimate collisions.

These results highlight a clear trend: the probability of false positives increases rapidly with the number of hash values considered. While these estimates focus on the behavior of NeuralHash in isolation, Apple's system-level deployment incorporates additional safeguards beyond single-hash matching. Public statements by Apple engineers indicate that an account is only flagged after a threshold number X of matching images, and that further verification steps may be applied before a report is generated.[2] Such mechanisms can substantially reduce the effective false positive rate by requiring multiple independent matches and enabling subsequent human review. Nevertheless, our results demonstrate that NeuralHash itself exhibits weak collision resistance relative to other perceptual hashing functions. Similar, though less severe, limitations have been observed for related schemes such as PDQ in prior work [18, 23, 57]. One potential mitigation strategy is to increase the effective output length of these hash functions; however, our findings indicate that careful design is required to achieve sufficient robustness at large scale. Moreover, longer hash outputs necessarily increase the amount of information leaked about the input images.



(a) Medium BF



(b) High BF

Figure 7: Estimated and observed number of illegitimate collisions on 96-bit NeuralHash

6.3 Estimating PhotoDNA Performance at Large Scale

Using Equations (2) and (3), we analyzed PhotoDNA results for the Non-Blur Face type, considering thresholds 150 and 175. The results are presented in Figure 8. Table 6 reports effective hash sizes: ∼34 bits (threshold 150) and ∼29 bits (threshold 175).

[2] In public technical presentations, Apple engineers have stated that this threshold is on the order of 30 matching images, although no formal specification has been released.

Figure 14 in Appendix F of the full version of this paper [43], compares estimates and observations for thresholds 175 and 220. Since only one collision is observed for a threshold of 150, the comparison is not meaningful in that case. For thresholds 175 and 220, the estimates closely match the observations, validating the approximation.

Table 6 also reports false positive rates for sets of 1 M, 10 M, and 100 M images. For 1 M hash values, the false positive rate remains low (below 0.2%), but it already increases noticeably for 10 M hash values, reaching 0.05% for threshold 150 and 1.6% for threshold 175. For 100 M hash values, the false positive rates increase dramatically: 0.51% for threshold 150 and 16.1% for threshold 175. These values indicate that even under the stricter threshold of 150, the number of false positives becomes non-negligible at large scales,[3] while under the threshold of 175, the collision rate becomes prohibitively high. At threshold 175 (as suggested in [59]), false positive rates exceed those of NeuralHash, despite the much larger 1152-bit output of PhotoDNA. Most importantly, Table 6 shows that for 100 M images, false positive rates are prohibitive for both thresholds.

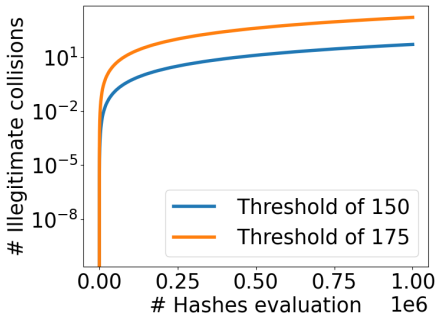


Figure 8: Estimated number of PhotoDNA illegitimate collisions for Non blurred faces images at thresholds 150 and 175

Table 6: Estimated PhotoDNA equivalent hash size (n') and false positive rates for thresholds 150 and 175

Threshold	150	175
Size of equivalent hash (n')	34.2	29.2
	False positive rate	
Approx. for 1 M hashes	0.005%	0.16%
Approx. for 10 M hashes	0.05%	1.6%
Approx. for 100 M hashes	0.51%	16.1%

Our analysis of PhotoDNA reinforces the conclusions drawn from NeuralHash: perceptual hash functions currently deployed for content moderation exhibit false positive and false negative rates that become problematic when applied to human faces. In particular, operating PhotoDNA at a threshold of 175 leads to error rates that are incompatible with reliable automated moderation, while lowering the threshold to 150 reduces false positives at the cost of a substantial increase in false negatives. Although such a configuration may still be manageable with extensive human verification and an appeal process, the number of false positives would still be significant, raising concerns about the scalability and robustness of these approaches as well as privacy concerns.

[3] It nevertheless remains impractical, as even a single illegitimate collision among 202 599 images is already unacceptable.

7 Limitations and Applicability to CSAM Detection

There may be concerns whether our experiments on face images directly generalize to CSAM detection. We contacted stakeholders with access to CSAM data, but at the time of submission had received no positive response. We plan to continue discussions with these stakeholders to seek a solution that enables independent evaluation while respecting privacy constraints.

Next, we provide arguments supporting the relevance of our results to CSAM detection. We focused on human faces, a category directly relevant to CSAM, which likely contributes to the high number of (near-)collisions observed. Evaluating additional image categories is left as future work, as is the study of potential classification attacks should similarly biased or distinctive hash distributions arise for other content types.

We further observe that the false positive and false negative rates measured on relatively small datasets raise concerns about large-scale deployment. While real CSAM datasets may exhibit somewhat lower error rates, the discrepancy with commonly claimed rates, differing by several orders of magnitude, suggests that such claims may not accurately reflect operational behavior. This interpretation is consistent with Meta transparency reports: in 2024, Meta reported a false positive rate of 0.12% after appeals [51], with similarly low but non-zero rates in 2023 and 2022 [49, 50]. These values are in line with our estimates in Table 6, but they are achieved at the cost of substantial human verification, appeal mechanisms, and temporary storage of incorrectly flagged content, raising privacy concerns. Moreover, such systems have already led to documented cases with severe consequences for falsely flagged individuals [33]. In parallel, PhotoDNA is continually updated to address these issues, with ongoing efforts aimed at reducing false positives and improving appeal procedures.

The use of the CelebA dataset may introduce biases, as it is relatively narrow (mostly frontal celebrity faces with limited contextual and demographic diversity), which could influence collision rates; hence, it is probable that it does not capture the variability present in real CSAM material. In addition, CelebA has a significantly imbalanced racial composition, with more than 70% of images belonging to the white racial group [58], and it is unclear whether the individuals depicted

have provided explicit consent for their inclusion. These factors raise broader concerns about representativeness, potential demographic bias, and ethical data collection practices.

Nonetheless, CelebA provides a standardized human-centered corpus that supports controlled experiments and manual verification, which is essential given the practical constraints of evaluating false positives at scale. We therefore use it as a starting point rather than as a definitive model of real-world data. Future work will extend our evaluation to more diverse datasets to assess the possibility of generalizing our conclusions.

8 Conclusion

By analyzing NeuralHash and PhotoDNA as examples of real-world perceptual hash functions, our work has demonstrated fundamental flaws in their application to CSAM detection. Specifically, we showed that NeuralHash and PhotoDNA exhibit extremely high false positive rates when applied to human faces, regardless of whether they are blurred, and simultaneously suffer from very high false negative rates, even for unmodified facial images. These findings indicate that these functions are not reliable in detecting perceptually identical content. The impact of these observations is larger, since our attacks do not require the specific structure of description of these perceptual hash functions. In addition, in 2008, Microsoft, NCMEC, and Farid jointly defined explicit requirements for perceptual hash functions for CSAM detection, as formulated in [24]:

> "Any technology must satisfy the following requirements:
>
> 1. Analyze an image in under two milliseconds (500 images/second);
> 2. Misclassify an image as child pornography (CP) at a rate of no more than one in 50 billion;
> 3. Correctly classify an image as CP at a rate of no less than 99%; and
> 4. Do not extract or share any identifiable image content (because of the sensitive nature of CP)."

The Requirement 4 has already been challenged in the literature [5]. Our work is the first to demonstrate that both NeuralHash and PhotoDNA fail to satisfy requirements 2 and 3. In particular, instead of achieving the "one in 50 billion" false positive rate (i.e., 2×10^{-11}), we observe false positive rates ranging from 10^{-3} to more than 10^{-1} depending on the dataset and scale, i.e., *many orders of magnitude higher*. Similarly, instead of meeting the 99% correct classification requirement, both functions exhibit false negative rates that are consistently above 20%, and in some cases exceed 60%. Importantly, Apple has confirmed our findings for NeuralHash through their coordinated vulnerability disclosure process. These results reinforce the urgent need for transparency: both the design and the testing procedures of perceptual hash functions should be made public, especially given that they are already deployed in practice.

The scale of the inaccuracies we identify raises important concerns for real-world deployment. In particular, a large-scale use of NeuralHash would likely result in a substantial number of innocent users being flagged, while remaining relatively easy to circumvent for individuals deliberately sharing illegal content. This imbalance undermines both the effectiveness and the fairness of the system.

For PhotoDNA, the trade-offs are intrinsic to the design of the system. The thresholds currently used in practice appear to achieve a manageable false positive rate, but only at the cost of higher false negative rates and a significant reliance on human verification and appeal processes. These procedures can take considerable time, during which affected users may experience account suspension or deactivation, potentially leading to reputational harm and loss of access to personal data. Even when accounts are eventually restored, such consequences raise concerns regarding proportionality, user trust, and privacy. Together, these observations suggest that weak perceptual hash functions, such as NeuralHash and PhotoDNA, present substantial challenges when used for large-scale CSAM detection. The potential harms and privacy implications therefore warrant careful consideration when weighing the benefits of deploying such systems.

It remains an open problem whether new perceptual hash functions with longer hash values can be designed that provide a better distribution of outputs for relevant inputs such that they resist black-box attacks. Note that increasing the output size means that more information is provided about the inputs, increasing the risk of preimage attacks or leakage of information about the inputs.

The design of perceptual hash functions that can resist white-box attacks is a much harder open problem, in particular because Client-Side Scanning means that the design cannot be kept secret. Recent research has shown that, for current designs, it is easy to construct false positives and false negatives in a white-box setting.

Even if accurate and reliable perceptual hash functions were available, the use of Client-Side Scanning remains highly problematic due to the risk of function creep and abuse.

Acknowledgements

Diane Leblanc-Albarel is funded by a fellowship from the Research Foundation Flanders (FWO–Vlaanderen), supported by the Flemish Government, under project number 12AD826N. This work was supported in part by CyberSecurity Research Flanders under reference number VOEWICS02.

Ethical Considerations

The research for this work complies with the ethical guidelines of USENIX. Our research was conducted in the context of the recent proposals of the European Union for the mandatory detection of Child Sexual Abuse Material using client-side scanning mechanisms, similar to the NeuralHash system proposed by Apple in 2021. The study was carried out with the purpose of evaluating the feasibility, effectiveness, and potential risks of using perceptual hash functions in this setting, as described in relevant technical reports and draft regulations. More in particular, the research intends to verify the public claims of false positive rates (1 in 50 billion for PhotoDNA and 1 in 1 trillion for NeuralHash, cf. Section 1); to the best of our knowledge, there is no scientific evidence available to substantiate these claims.

In our investigation, no human subjects were involved and we have restricted our use of human images to a public image set of celebrities. As discussed in Section 3.1, the use of the CelebA dataset introduces constraints on the demographic representativeness of the analysis, which should be taken into account when interpreting our results.

Consistent with responsible research practices, we document the public sources that we have used and limit access to additional information that could enable misuse (cf. Open Science section below).

We have followed coordinated vulnerability disclosure procedures by timely notifying Apple and Microsoft of our findings. At the time of submission, Apple had confirmed that they had reproduced all of our results without exception, acknowledged the issues, and were investigating solutions. We have entered constructive discussions with Microsoft. We can confirm that the results presented for PhotoDNA in this paper are accurate for the version currently deployed. These exchanges have focused on addressing the shortcomings we identified.

As recommended in the Menlo Report, we assessed both the potential benefits and harms of the research we conducted for all stakeholders, namely CSAM victims, innocent users who may be falsely accused of possession or dissemination of CSAM, and companies developing and deploying perceptual hash functions.

After careful consideration, we conclude that our research does not pose a negative impact on CSAM victims. Our work does not affect, and is not expected to affect the false negative rate in a way that would reduce the detection of CSAM. Moreover, in light of our findings and given that the overall trade-off is widely regarded as positive by industry stakeholders and reflected in existing legislation, it is unlikely that companies would discontinue CSAM detection as a consequence of our work.

The potential impact of our research on companies is primarily reputational, as all other potential impacts were communicated to Apple and Microsoft in a timely manner, enabling them to account for the identified issues and explore possible mitigation strategies. In particular, organizations and providers with access to CSAM data can conduct additional research to assess whether the observed increase in false positives and false negatives extends to their datasets. False positives affect the privacy of innocent users, whereas false negatives reduce the effectiveness of CSAM detection. If the resulting parameters would not be acceptable in terms of true and false positive rates, improved designs for perceptual hash functions can be developed and deployed. In this way, the impact of our results on innocent users and service providers involved in CSAM detection can be minimized, while victims of CSAM can be better protected.

Finally, we expect our work to have a positive impact on innocent users by serving as an independent evaluation of perceptual hash functions. By pointing out the risks of increased false positives our work may result in the introduction of additional mitigation measures. We hope that our findings will be considered by institutions and policymakers when proposing regulations that mandate the use of perceptual hash functions.

Our results indicate that the large-scale deployment of currently available perceptual hash functions for automated client-side CSAM detection may present significant societal risks and potential harm to a large number of citizens.

Open Science

All datasets and perceptual hash implementations used in this study are from publicly available sources. In particular, we used the CelebA dataset, which can be downloaded at: https://www.kaggle.com/datasets/jessicali9530/celeba-dataset

For applying NeuralHash, we rely on publicly released implementations available at:

- AppleNeuralHash2ONNX: https://github.com/AsuharietYgvar/AppleNeuralHash2ONNX
- nhcalc: https://github.com/KhaosT/nhcalc

For computing PhotoDNA hash values, we use the open-source implementation:

- jPhotoDNA: https://github.com/jankais3r/jPhotoDNA/

In accordance with open science principles, we only use publicly accessible datasets and code. We do not release any additional datasets, trained models, or artifacts that could enable misuse. Any derived materials, code, or scripts specific to our experiments will be made available to legitimate researchers upon request, following ethical review.

A Proof of Propositions 1 to 3

Proof of Proposition 1

Proposition 1. ***(Restated)*** *Consider a hash value of n bits and a vector p for which the i-th element p_i denotes the probability that the i-th bit of the hash value is equal to 1 ($0 \leqslant p_i \leqslant 1$). If E_p^n denotes the event that two hash values are equal, then*

$$\mathbb{P}(E_p^n) = \prod_{i=1}^{n} (p_i^2 + (1-p_i)^2)\,.$$

Proof. Let $H_{i,j}^n$ denote the first j bits of the i-th hash value of n bits, and let a_i^n, b_i^n be the first i bits of two given hash values of size n. E_p^n is thus the event that the two strings a_n^n and b_n^n are equal. We have:

$$\begin{aligned}\mathbb{P}(E_p^n) &= \sum_{h=H_{i,n}^n} \mathbb{P}(a_n^n = h)\mathbb{P}(b_n^n = h) = \sum_{h=H_{i,n}^n} \mathbb{P}(a_n^n = h)^2 \\ &= (p_n^2 + (1-p_n)^2) \sum_{h=H_{i,n-1}^n} \mathbb{P}(a_{n-1}^n = h)^2\,.\end{aligned}$$

The result follows by recurrence over the remaining $n-1$ bits.

□

Proof of Proposition 2

Proposition 2. ***(Restated)*** *Denote with $\mathbb{P}(E_p^n)$ the probability that two n-bit hash values with distribution p are equal and define $U = 2^n$. The expected number of hash values to compute before the first collision $\mathbb{E}(D_p^n)$ is equal to:*

$$\mathbb{E}(D_p^n) = 2 + (U+1)y^{\frac{U(U+1)}{2}} + \sum_{x=1}^{U-1} y^{\frac{x(x+1)}{2}} \text{ with } y = 1 - \mathbb{P}(E_p^n).$$

Proof. $\mathbb{P}(E_p^n)$ denotes the probability of a collision between two hash values, and D_p^n denotes the random variable representing the number of hash evaluations required before encountering the first collision. The probability that the first collision occurs after x hash function is denoted by $\mathbb{P}[D_p^n = x]$.

As $\mathbb{P}[D_p^n < 2] = 0$ and $\mathbb{P}[D_p^n > U+1] = 0$, and as there are a maximum of $U = 2^n$ different hash values. By definition:

$$\sum_{i=1}^{U+1} \mathbb{P}[D_p^n = i] = 1 \;\texttt{and:}\; \mathbb{E}(D_p^n) = \sum_{x=2}^{U+1} x \times \mathbb{P}[D_p^n = x]\,.$$

We first define $\mathbb{P}[D_p^n = n]$. The probability $\mathbb{P}[D_p^n = 2]$ is the probability that a collision occurs after 2 hash evaluations, which is simply: $\mathbb{P}[D_p^n = 2] = \mathbb{P}(E_p^n)$. The probability $\mathbb{P}[D_p^n = 3]$ is the probability that a collision occurs after exactly 3 hash evaluations. This is the probability that the first two hash values are different, multiplied by the probability that the third hash value matches one of the first two. Therefore:

$$\mathbb{P}[D_p^n = 3] = (1 - \mathbb{P}(E_p^n)) \times (1 - (1 - \mathbb{P}(E_p^n))^2)\,.$$

Similarly, the probability $\mathbb{P}[D_p^n = 4]$ equals:

$$\mathbb{P}[D_p^n = 4] = (1 - \mathbb{P}(E_p^n))^3 \times (1 - (1 - \mathbb{P}(E_p^n))^3)\,.$$

This approach generalizes to obtain $\mathbb{P}[D_p^n = n]$, the probability that a collision occurs after exactly n hash operations with $(3 \leq n \leq U+1)$:

$$\mathbb{P}[D_p^n = n] = \left(\prod_{i=1}^{n-2} (1 - \mathbb{P}(E_p^n))^i\right) \times (1 - (1 - \mathbb{P}(E_p^n))^{n-1})\,,$$

which simplifies to:

$$\mathbb{P}[D_p^n = n] = (1 - \mathbb{P}(E_p^n))^{\frac{(n-2)(n-1)}{2}} - (1 - \mathbb{P}(E_p^n))^{\frac{(n-1)n}{2}}\,.$$

The expected number of hash values to compute before the first collision is given by $\mathbb{E}(D_p^n)$, which is expressed as:

$$\mathbb{E}(D_p^n) = \sum_{x=2}^{N+1} x \times \mathbb{P}[D_p^n = x]\,,$$

which is equivalent to:

$$\mathbb{E}(D_p^n) = 2 \times \mathbb{P}(E_p^n) + \sum_{x=3}^{N+1} x \times \mathbb{P}[D_p^n = x]\,.$$

Substituting the expression for $\mathbb{P}[D_p^n = x]$ and expanding the sum, we obtain:

$$\begin{aligned}\mathbb{E}(D_p^n) &= 2\mathbb{P}(E_p^n) + \left(3(1 - \mathbb{P}(E_p^n)) - 3(1 - \mathbb{P}(E_p^n))^3\right) + \ldots \\ &+ \left((U+1)(1 - \mathbb{P}(E_p^n))^{\frac{(U-1)U}{2}} - (U+1)(1 - \mathbb{P}(E_p^n))^{\frac{U(U+1)}{2}}\right)\end{aligned}$$

Reordering the terms, we then obtain:

$$\begin{aligned}\mathbb{E}(D_p^n) &= 2 \times \mathbb{P}(E_p^n) + 2 \times (1 - \mathbb{P}(E_p^n)) + (1 - \mathbb{P}(E_p^n)) \\ &+ (U+1) \times (1 - \mathbb{P}(E_p^n))^{\frac{U(U+1)}{2}} \\ &+ \sum_{x=4}^{U+1} \left((x+1)(1 - \mathbb{P}(E_p^n))^{\frac{(x-2)(x-1)}{2}} - x(1 - \mathbb{P}(E_p^n))^{\frac{(x-2)(x-1)}{2}}\right)\,.\end{aligned}$$

Simplifying the terms of the sum and shifting the bounds by changing the variable x to $x-2$, we include the term $(1 - \mathbb{P}(E_p^n))$ as the first term of the sum:

$$\mathbb{E}(D_p^n) = 2 + (U+1)(1 - \mathbb{P}(E_p^n))^{\frac{U(U+1)}{2}} + \sum_{x=1}^{U-1} (1 - \mathbb{P}(E_p^n))^{\frac{x(x+1)}{2}}$$

With $y = 1 - \mathbb{P}(E_p^n)$, we conclude that

$$\mathbb{E}(D_p^n) = 2 + (U+1) \times y^{\frac{U(U+1)}{2}} + \sum_{x=1}^{U-1} y^{\frac{x(x+1)}{2}}\,.$$

□

Proof of Proposition 3

Proposition 3. ***(Restated)*** *Denote with $\mathbb{P}(E_p^n)$ the probability that two n-bit hash values with distribution p are equal and define $U = 2^n$. The expected number of hash values to compute before the first collision $\mathbb{E}(D_p^n)$ is upper bounded by:*

$$\mathbb{E}(D_p^n) \leq 1 + (U+1) \times y^{\frac{U(U+1)}{2}} + \frac{\theta_2(0; y^{\frac{1}{2}})}{2 \times y^{\frac{1}{8}}},$$

with $\theta_2(0; y^{\frac{1}{2}})$ the Jacobi theta function $\theta_2(z;q)$ with $z = 0$ and $q = y^{\frac{1}{2}}$ and with $y = 1 - \mathbb{P}(E_p^n)$.

Proof. From Proposition 2 we have:

$$\mathbb{E}(D_p^n) = 2 + (U+1) \times y^{\frac{U(U+1)}{2}} + \sum_{x=1}^{U-1} y^{\frac{x(x+1)}{2}}.$$

As $\sum_{n=0}^{U-1} y^{\frac{n(n+1)}{2}} \leq \sum_{n=0}^{\infty} y^{\frac{n(n+1)}{2}}$ we thus have:

$$\sum_{n=1}^{U-1} y^{\frac{n(n+1)}{2}} \leq -1 + \sum_{n=0}^{\infty} y^{\frac{n(n+1)}{2}}. \tag{4}$$

Using the Jacobi theta function $\theta_2(z;q)$ with $z = 0$ and $q = y^{\frac{1}{2}}$ we obtain:

$$\theta_2(0; y^{\frac{1}{2}}) = 2 \times y^{\frac{1}{8}} \times \left(\sum_{n=0}^{\infty} y^{\frac{n(n+1)}{2}} \right).$$

By introducing the Jacobi function in the right hand side of Equation (4), we conclude that $\mathbb{E}(D_p^n)$ satisfies

$$\mathbb{E}(D_p^n) \leq 1 + (U+1) \times y^{\frac{U(U+1)}{2}} + \frac{\theta_2(0; y^{\frac{1}{2}})}{2 \times y^{\frac{1}{8}}}.$$

□

B False Positives for the 2023 NeuralHash Design

The NeuralHash files deployed on macOS devices have been updated around the end of 2023. The new model uses seeds in fp16 and is trained with fp16 precision (instead of fp32 for the previous model). Apple has not provided any official communication regarding this change. This update is surprising as Apple announced in December 2022 that it canceled its plan to scan photos on Apple devices for CSAM.

Previous attacks on NeuralHash were conducted in a white-box setting, meaning the attackers had full knowledge of the model. With this new model, which has not yet been reverse-engineered, it is currently impossible to fully understand or manipulate the inner workings of the algorithm, especially the CNN involved.

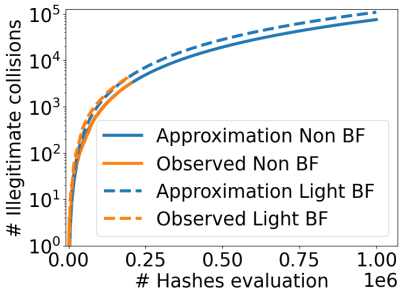
However, it is still possible to run NeuralHash using the files present on a device through code available on the GitHub project [63], without detailed knowledge of the internal processes. We applied our collision attacks using this method, and the results in Table 7 are very similar to those obtained in the previous section.

Table 7: False positives for each image type for the fp16 model of NeuralHash

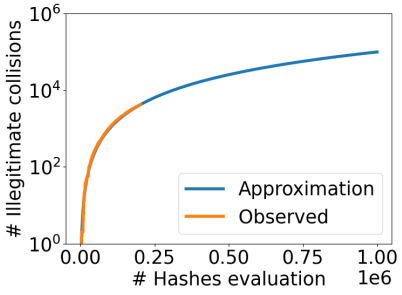
Type	Non human	Non BF	Light BF	BF only	Medium BF	High BF
# illegit. collisions	0	12	50	90	> 500	> 1000

The results indicate that, except for non-blurred face images where the rate of illegitimate collisions decreased, all other types showed a significant increase in illegitimate collision rates, particularly for medium and highly blurred face images. Therefore, the use of this new model does not alter the conclusions. On the contrary, it tends to exacerbate the issues highlighted.

C Approximation and observation of Illegitimate Collision on NeuralHash



(a) Non BF and Light BF



(b) BF only

Figure 9: Observed and aproximated number of illegitimate collisions on randomly selected 56 bits NeuralHash

For some image types, the number of illegitimate collisions observed on the full 96-bit NeuralHash output is too low

to reliably verify that the approximation from Equation (3) matches the experimental results. To address this, we repeat the analysis using only a randomly selected subset of 56 bits from the hash values, which increases the expected number of collisions and allows a meaningful comparison between theory and experiment.

We still report the 96-bit results for completeness, even though the collision counts are insufficient for firm conclusions (Figure 7). The 56-bit verification results, shown in Figure 9, confirm that the observed number of illegitimate collisions closely follows the values predicted by the birthday paradox when assuming a uniformly distributed and independent hash output. This supports the use of the same approximation for the full 96-bit results and for all image types in the subsequent analysis.

References

[1] Harold Abelson, Ross J. Anderson, Steven M. Bellovin, Josh Benaloh, Matt Blaze, Jon Callas, Whitfield Diffie, Susan Landau, Peter G. Neumann, Ronald L. Rivest, Jeffrey I. Schiller, Bruce Schneier, Vanessa Teague, and Carmela Troncoso. Bugs in Our Pockets: The Risks of Client-Side Scanning. *J. Cybersecur.*, 2024.

[2] Apple. Expanded Protections for Children, Frequently Asked Questions. https://www.apple.com/child-safety/pdf/Expanded_Protections_for_Children_Frequently_Asked_Questions.pdf, 2021. Accessed July, 2025.

[3] Apple. CSAM Detection – Technical Summary. https://www.apple.com/child-safety/pdf/CSAM_Detection_Technical_Summary.pdf, 2021. Accessed July, 2025.

[4] Yuki M. Asano, Christian Rupprecht, Andrew Zisserman, and Andrea Vedaldi. PASS: An ImageNet Replacement for Self-Supervised Pretraining Without Humans. *NeurIPS*, 2021.

[5] Anish Athalye. Inverting PhotoDNA. https://www.anishathalye.com/2021/12/20/inverting-photodna/, 2021. Accessed July, 2025.

[6] Anish Athalye. NeuralHash Collider. https://github.com/anishathalye/neural-hash-collider, 2021. Accessed July, 2025.

[7] James Bartusek, Sanjam Garg, Abhishek Jain, and Guru-Vamsi Policharla. End-to-End Secure Messaging with Traceability Only for Illegal Content. In *EUROCRYPT*, 2023.

[8] Jagdeep Singh Bhatia and Kevin Meng. Exploiting and Defending Against the Approximate Linearity of Apple's NeuralHash. *arXiv preprint arXiv:2207.14258*, 2022.

[9] Abhishek Bhowmick, Dan Boneh, Steve Myers, Kunal Talwar, and Karl Tarbe. The Apple PSI System. https://www.apple.com/child-safety/pdf/Apple_PSI_System_Security_Protocol_and_Analysis.pdf, 2021. Accessed July, 2025.

[10] Patrick Breyer, Alviina Alametsä, Rosa D'Amato, Pernando Barrena, Saskia Bricmont, Antoni Comín, Gwendoline Delbos-Corfield, Francesca Donato, Cornelia Ernst, Claudia Gamon, Markéta Gregorová, Francisco Guerreiro, Svenja Hahn, Irena Joveva, Petra Kammerevert, Marcel Kolaja, Moritz Körner, Karen Melchior, Clara Ponsatí, and Mikuláš Peksa. Cross-Party Letter of Members of the European Parliament Against General Monitoring. https://www.patrick-breyer.de/wp-content/uploads/2021/11/20211020_Letter_General_Monitoring.pdf, 2021. Accessed July, 2025.

[11] Dominique Brunet, Edward R. Vrscay, and Zhou Wang. On the Mathematical Properties of the Structural Similarity Index. *IEEE Trans. Image Process.*, 2012.

[12] Elie Bursztein, Einat Clarke, Michelle DeLaune, David M. Elifff, Nick Hsu, Lindsey Olson, John Shehan, Madhukar Thakur, Kurt Thomas, and Travis Bright. Rethinking the Detection of Child Sexual Abuse Imagery on the Internet. 2019.

[13] Nicholas Carlini and David A. Wagner. Towards Evaluating the Robustness of Neural Networks. In *S&P*, 2017.

[14] Damon M. Chandler and Sheila S. Hemami. VSNR: A Wavelet-Based Visual Signal-to-Noise Ratio for Natural Images. *IEEE Trans. Image Process.*, 2007.

[15] Council of the European Union. Proposal for a Regulation of the European Parliament and of the Council Laying Down Rules to Prevent and Combat Child Sexual Abuse, 2024. Accessed July, 2025.

[16] Counter Extremism Project. How CEP's eGLYPH Technology Works. https://www.counterextremism.com/video/how-ceps-eglyph-technology-works, 2016. Accessed July, 2025.

[17] Francesco Croce and Matthias Hein. Minimally Distorted Adversarial Examples with a Fast Adaptive Boundary Attack. In *ICML*, 2020.

[18] Janis Dalins, Campbell Wilson, and Douglas Boudry. PDQ & TMK + PDQF - A Test Drive of Facebook's Perceptual Hashing Algorithms. *CoRR*, 2019.

[19] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting Adversarial Attacks With Momentum. In *CVPR*, 2018.

[20] Andrea Drmic, Marin Silic, Goran Delac, Klemo Vladimir, and Adrian Satja Kurdija. Evaluating Robustness of Perceptual Image Hashing Algorithms. In *MIPRO*, 2017.

[21] Ling Du, Anthony T. S. Ho, and Runmin Cong. Perceptual Hashing for Image Authentication: A Survey. *Signal Process. Image Commun.*, 2020.

[22] European Commission. Report from the Commission to the European Parliament and the Council on the implementation of Regulation (EU) 2021/1232. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52023DC0797. Accessed July, 2025.

[23] Facebook. The TMK+PDQF Video-Hashing Algorithm and the PDQ Image-Hashing Algorithm. https://github.com/facebook/ThreatExchange/blob/main/hashing/hashing.pdf, 2020. Accessed July, 2025.

[24] Hany Farid. Reining in Online Abuses. *Technology and Innovation*, 2018.

[25] Hany Farid. Testimony: Fostering a Healthier Internet to Protect Consumers. Written statement submitted to the U.S. House Committee on Energy and Commerce. https://www.congress.gov/116/meeting/house/110075/witnesses/HHRG-116-IF16-Wstate-FaridH-20191016.pdf, 2019. Accessed July, 2025.

[26] Hany Farid. An Overview of Perceptual Hashing. *Journal of Online Trust and Safety*, 2021.

[27] Philippe Flajolet and Andrew M. Odlyzko. Random mapping statistics. In *EUROCRYPT*. Springer-Verlag, 1989.

[28] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity Search in High Dimensions via Hashing. In *VLDB*, 1999.

[29] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *ICLR*, 2015.

[30] Google Inc. How Content ID works. https://support.google.com/youtube/answer/2797370?hl=en, 2007. Accessed July, 2025.

[31] Qingying Hao, Licheng Luo, Steve T. K. Jan, and Gang Wang. It's Not What It Looks Like: Manipulating Perceptual Hashing Based Applications. In *CCS*, 2021.

[32] Sophie Hawkes, Christian Weinert, Teresa Almeida, and Maryam Mehrnezhad. Perceptual Hash Inversion Attacks on Image-Based Sexual Abuse Removal Tools. *IEEE Security & Privacy*, 2024.

[33] Kashmir Hill. A Dad Took Photos of His Naked Toddler for the Doctor. Google Flagged Him as a Criminal. *The New York Times*, August 2022.

[34] Piotr Indyk and Rajeev Motwani. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In *STOC*, 1998.

[35] Shubham Jain, Ana-Maria Cretu, Antoine Cully, and Yves-Alexandre de Montjoye. Deep Perceptual Hashing Algorithms with Hidden Dual Purpose: When Client-Side Scanning Does Facial Recognition. In *S&P*, 2023.

[36] Shubham Jain, Ana-Maria Cretu, and Yves-Alexandre de Montjoye. Adversarial Detection Avoidance Attacks: Evaluating the Robustness of Perceptual Hashing-Based Client-Side Scanning. In *USENIX*, 2022.

[37] Jan Kaiser. pyPhotoDNA. https://github.com/jankais3r/pyPhotoDNA, 2023. Accessed Feb, 2025.

[38] Yannic Kilcher. Neural Hash Collision Creator. https://github.com/yk/neural_hash_collision, 2021. Accessed July, 2025.

[39] Evan Klinger and David Starkweather. pHash: The Open Source Perceptual Hash Library. https://www.phash.org/, 2010. Accessed July, 2025.

[40] Brian Kulis and Trevor Darrell. Learning to Hash with Binary Reconstructive Embeddings. In *NeurIPS*, 2009.

[41] Anunay Kulshrestha and Jonathan R. Mayer. Identifying Harmful Media in End-to-End Encrypted Communication: Efficient Private Membership Computation. In *USENIX*, 2021.

[42] Mario Lamberger and Elmar Teufl. Memoryless Near-Collisions, Revisited. *Information Processing Letters*, 2013.

[43] Diane Leblanc-Albarel and Bart Preneel. Black-box Collision Attacks on Widely Deployed Perceptual Hash Functions. ePrint, https://eprint.iacr.org/2024/1869, 2024.

[44] Ian Levy and Crispin Robinson. Thoughts on Child Safety on Commodity Platforms. https://arxiv.org/abs/2207.09506, 2022. Accessed July, 2025.

[45] Venice Erin Liong, Jiwen Lu, Gang Wang, Pierre Moulin, and Jie Zhou. Deep Hashing for Compact Binary Codes Learning. In *CVPR*, 2015.

[46] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep Supervised Hashing for Fast Image Retrieval. *Int. J. Comput. Vis.*, 2019.

[47] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *ICCV*, 2015.

[48] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*, 2018.

[49] Meta. EU CSAM Derogation Report 2022. https://transparency.meta.com/sr/eu-csam-derogation-report-2023/, 2023. Accessed July, 2025.

[50] Meta. EU CSAM Derogation Report 2023. https://transparency.meta.com/sr/eu-csam-derogation-report-2024/, 2024. Accessed July, 2025.

[51] Meta. EU CSAM Derogation Report 2024. https://transparency.meta.com/sr/eu-csam-derogation-report-2025/, 2025. Accessed July, 2025.

[52] Microsoft Corporation. PhotoDNA. https://www.microsoft.com/en-us/photodna, 2025. Accessed July, 2025.

[53] NCMEC. Statement Regarding End-to-End Encryption. https://www.missingkids.org/theissues/end-to-end-encryption, 2025. Accessed July, 2025.

[54] Joint Statement of Scientists and Researchers on EU's Proposed Child Sexual Abuse Regulation. https://edri.org/wp-content/uploads/2023/07/Open-Letter-CSA-Scientific-community.pdf, 2023. Accessed July, 2025.

[55] Joint Statement of Scientists and Researchers on EU's New Proposal for the Child Sexual Abuse Regulation. https://nce.mpi-sp.org/index.php/s/eqjiKaAw9yYQF87, 2024. Accessed July, 2025.

[56] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The Limitations of Deep Learning in Adversarial Settings. In *EuroS&P*, 2016.

[57] 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 *USENIX*, 2023.

[58] Tian Qiu, Arjun Nichani, Rasta Tadayontahmasebi, and Haewon Jeong. Gone With the Bits: Revealing Racial Bias in Low-Rate Neural Compression for Facial Images. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, 2025.

[59] Martin Steinebach. An Analysis of PhotoDNA. In *ARES*, 2023.

[60] Lukas Struppek, Dominik Hintersdorf, Daniel Neider, and Kristian Kersting. Learning to Break Deep Perceptual Hashing: The Use Case NeuralHash. In *FAccT*, 2022.

[61] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One Pixel Attack for Fooling Deep Neural Networks. *IEEE Trans. Evol. Comput.*, 2019.

[62] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing Properties of Neural Networks. In *ICLR*, 2014.

[63] Khaos Tian and Malcolm Hall. nhcalc. Compute NeuralHash for a given image. https://github.com/KhaosT/nhcalc, 2021. Accessed July, 2025.

[64] UK Department for Science, Innovation & Technology. Online Safety Act: Explainer. https://www.gov.uk/government/publications/online-safety-act-explainer/online-safety-act-explainer, 2024. Accessed July, 2025.

[65] Yongwei Wang, Hamid Palangi, Z. Jane Wang, and Haoqian Wang. RevHashNet: Perceptually De-hashing Real-Valued Image Hashes for Similarity Retrieval. *Signal Process. Image Commun.*, 2018.

[66] Dayan Wu, Zheng Lin, Bo Li, Mingzhen Ye, and Weiping Wang. Deep Supervised Hashing for Multi-Label and Large-Scale Image Retrieval. In *ICMR*, 2017.

[67] Asuhariet Ygvar. AppleNeuralHash2ONNX. Convert Apple NeuralHash model for CSAM Detection to ONNX. https://github.com/AsuharietYgvar/AppleNeuralHash2ONNX, 2021. Accessed July, 2025.

[68] Christoph Zauner. Implementation and Benchmarking of Perceptual Image Hash Functions, MsC Thesis, University of Applied Sciences, Hagenberg, Austria, 2010.

[69] Fang Zhao, Yongzhen Huang, Liang Wang, and Tieniu Tan. Deep Semantic Ranking Based Hashing for Multi-Label Image Retrieval. In *CVPR*, 2015.