Robustness and Collision-Resistance of PhotoDNA

Martin Steinebach

Fraunhofer SIT, Germany
E-mail: martin.steinebach@sit.fraunhofer.de

Received 04 January 2024; Accepted 23 January 2024;
Publication 09 April 2024

Abstract

PhotoDNA is a widely utilized hash designed to counteract Child Sexual Abuse Material (CSAM). However, there has been a scarcity of detailed information regarding its performance. In this paper, we present a comprehensive analysis of its robustness and susceptibility to false positives, along with fundamental insights into its structure. Our findings reveal its resilience to common image processing techniques like lossy compression. Conversely, its robustness is limited when confronted with cropping. Additionally, we propose recommendations for enhancing the algorithm or optimizing its application. This work is an extension on our paper [21].

Keywords: Robust hash, perceptual hash, CSAN detection.

1 Motivation

Robust hashing plays a crucial role in the ongoing effort to combat Child Sexual Abuse Material (CSAM) by facilitating the identification and removal of such content from online platforms. CSAM encompasses various forms of visual or digital media, including images, videos, or computer-generated

Journal of Cyber Security and Mobility, Vol. 13_3, 541–564.
doi: 10.13052/jcsm2245-1439.1339

content, depicting the sexual abuse or exploitation of children. It constitutes highly illegal and morally reprehensible material, contributing to the victimization and harm of minors.

The process of robust hashing involves creating a distinct digital fingerprint, or hash, for an image or video. This hash serves as a unique identifier, enabling the comparison of content with known instances of CSAM. In practice, robust hashing aids in the detection and removal of CSAM from online platforms by identifying already-recognized images or videos and preventing their further sharing or distribution.

By employing robust hashing, online platforms gain the ability to proactively detect and eliminate CSAM without solely relying on user reports. This proactive approach contributes significantly to curbing the spread of such harmful content and safeguarding children from potential harm.

Furthermore, robust hashing proves instrumental in assisting law enforcement efforts to identify and locate individuals involved in the production or dissemination of CSAM. The unique hash associated with the content allows law enforcement to trace the distribution of CSAM across platforms, potentially leading to the identification of those responsible for creating and sharing such illicit material.

1.1 Importance of Error Rates

Understanding the false positive and false negative rates of robust hashing is crucial due to their significant impact on the effectiveness and efficiency of detecting Child Sexual Abuse Material (CSAM) and other illicit content online.

False positive rates indicate the percentage of non-CSAM images or videos incorrectly identified as CSAM by the hashing algorithm. Conversely, false negative rates represent the percentage of CSAM content mistakenly classified as non-CSAM. A high false positive rate may lead to innocent users being wrongly accused of sharing CSAM, resulting in serious legal and reputational consequences. Moreover, elevated false positive rates may result in the excessive removal of non-CSAM content, potentially chilling free speech.

Conversely, a high false negative rate means that CSAM content may go undetected and continue to circulate online. This not only perpetuates the distribution of harmful material but also hampers law enforcement's ability to identify and prosecute those responsible for creating and disseminating CSAM.

Therefore, it is crucial to strike a balance between minimizing false positives and false negatives to effectively detect and remove CSAM while mitigating the impact on innocent users and free speech. By comprehending and monitoring the false positive and false negative rates of robust hashing algorithms, online platforms can continuously enhance their detection and removal processes, better safeguarding children and upholding their community standards.

1.2 PhotoDNA

PhotoDNA is one of the most widely used robust hashing algorithms for detecting CSAM (child sexual abuse material) and other illegal online content. It was developed by Microsoft and is now widely used by many online platforms and law enforcement agencies [12]. It is integrated in forensic tools like Griffeye[1] making it an readlily available tool for forensic practitioners.

PhotoDNA works by creating a robust hash (sometimes referred to in the media as a "digital signature") of an image that can be compared to other hashes to identify matches of known CSAM images. The algorithm is said to be resistant to common image manipulations such as cropping, resizing, or color adjustments.

Several studies have been conducted to evaluate the performance of PhotoDNA (e.g. [19]). Most of these studies have been conducted by end users and have not been published. In general, PhotoDNA has been described as highly effective in detecting known CSAM content, with low false positive and false negative rates. For example, a study conducted by the National Center for Missing and Exploited Children (NCMEC) claims that PhotoDNA had a false positive rate of less than 1 in 1 trillion and a false negative rate of less than 2%. In the absence of more detailed information, this needs to be considered carefully, and may be a mixture of evaluation and mathematical assumptions due to the very (perhaps unrealistically) low false positive rate.

However, it is important to note that no algorithm is perfect, and there are still limitations to the effectiveness of PhotoDNA and other robust hashing algorithms. For example, these Robust hashes cannot detect new or novel CSAM content that has not yet been identified and added to the database of known hashes. In general, they can also be circumvented by manipulations such as cropping and rotating, as well as obfuscation or addition of objects.

[1] https://www.griffeye.com/wp-content/uploads/2015/08/Griffeye-ProductSheet-AnalyzeDI-3-8-15.pdf

There are concepts to counteract these shortcomings, but they make the hashing algorithms more complex [23]. Despite these limitations, PhotoDNA and other robust hashing algorithms remain an important tool in the fight against CSAM and other illegal content on the Internet.

The purpose of this paper is to evaluate the performance of PhotoDNA under normal usage conditions. Therefore, security attacks on the algorithm are not considered. We assume that it will most often be used to monitor large amounts of data, either stored on a hard drive or transmitted via messengers or social networks. Here, the main challenge is to avoid high false positive rates. This can be achieved by comparing the robustness against standard operations that the hash needs to be robust against with the likelihood of collisions with the hash of another image. Ideally, a threshold is found that helps distinguish between copies of a work (here: an image) from different works. The actual threshold will not be the subject of this work, as it is a decision to be made by the end users. High thresholds will lead to low false positive rates, but may skip some copies after image operations, low thresholds will lead to the opposite behavior.

1.3 Structure

After motivating and introducing PhotoDNA and the importance of error rates, we briefly review common robust hashing challenges and the state of the art. This is more for completeness of the paper and is much better covered in survey papers. We then summarize what we know about the PhotoDNA algorithm. We then analyze the behavior of the hash based on a 160,000 image test set. We then evaluate the performance on two test sets, the small Galaxy test set and a selection of the huge CoCo set. We then discuss our findings and conclude the paper with a summary and some future work.

2 Challenges

There are several typical challenges to robust hashing that must be addressed in order to effectively combat CSAM and other illegal online content: If an image or video is altered in any way, such as by resizing, cropping, or adding noise, the resulting hash may be different from the original, making it more difficult to identify as CSAM. This is basically the robustness in “robust hashing”. Encryption can be used to hide CSAM content, making it difficult or impossible to detect by hashing algorithms. This is being addressed in the EU discussion on chat control. As new CSAM content is created and shared, it is not yet known to the hashing algorithms. AI-based classification must be used here [13].

The scale of the problem can be overwhelming, with billions of images and videos being shared online every day. It can be challenging to process this volume of data in real time using robust hashing algorithms. This requires efficient algorithms for generation and matching. It also means that low error rates, especially low false positives, are critical. Otherwise, manual verification of illegal content would become overwhelming for involved parties.

To address these challenges, continual development and enhancement of detection and removal processes are imperative. This involves a combination of robust hashing algorithms, machine learning, and human review. Collaborative efforts and information sharing among online platforms, law enforcement agencies, and other stakeholders are also essential to improve the overall effectiveness of these endeavors.

While this paper partially addresses these challenges, it is crucial to acknowledge them, particularly in the context of the ongoing discussion about chat control[2] [1, 36]. The limited scope of alterations in the tests conducted here, treating them more as different versions of a given photo, underscores the need for further exploration. Additionally, the paper does not address handling of false positive analysis, encryption, and the classification of unknown content.

3 State of the Art

Hash-based algorithms are used in various application areas, such as image search, duplicate or near-duplicate detection, or image authentication [5, 6, 16, 27]. In this paper, we assume that the difference between cryptographic and robust hashing is known. Briefly, cryptographic hashing is not robust and will generate hashes with no similarity between versions of an image after lossy compression or scaling. Many robust hashing algorithms use perceptual features of images [31–34]. With advances in deep learning, neural network-based approaches have also been explored for robust hashing. These approaches use deep neural networks to learn feature representations that capture image content and generate compact hash codes for similarity comparison [3, 4, 18].

[2] https://www.bundestag.de/resource/blob/949084/2c559a363cdaf9c5f1d44cae218c6e76/Stellungnahme-Steinebach-ENG-data.pdf

3.1 Security vs. Robustness

It is often overlooked that content recognition methods are often not designed to be secure. The task of robust hashing methods and classifiers is to recognize or classify content. It is not assumed that an attacker will directly target the methods to prevent this recognition. In the field of multimedia security, a distinction is made between robustness and security. Robustness addresses changes to content that are caused by processing that is normally expected, such as scaling or lossy compression. Robust hashing methods are resistant to this, and classifiers should not exhibit any serious drops in performance here either. Security, on the other hand, comes from an attacker deliberately targeting the algorithms that use the methods [11]. For example, robust hashing methods can be used to make local changes to the image that cause the hash to change significantly, even though the image itself is not or only slightly disturbed.

This also applies to modern hashing methods based on machine learning, such as NeuralHash from Apple [30]. These attacks can potentially be carried out in both directions: The hash of an image is changed so that it is no longer recognized. Or the hash of another image is changed in such a way that it is mistakenly considered to be stored in a database.

The term “robustness” is also used for attacks against classification by machine learning but does not have the same meaning as for robust hashing methods. So-called “adversarial robustness attacks” attempt to modify classification results by making slight changes to the image. This means that nudity can no longer be recognized in the given context or the age of people is incorrectly estimated. The best-known example here is the photo of a panda that is classified as a gibbon as a result of an attack [9].

3.2 Attacks on PhotoDNA

In [17], preimage attacks on PhotoDNA (as well as facebook PDQ) are shown. By accepting a certain amount of noise, it is possible to generate image pairs with matching hashes.

In [15] the privacy of the hashes was verified. It was argued that it is not possible to derive the original images from their hashes using machine learning. Recent experiments, however, show that images can be reconstructed from the hashes.[3] The question is whether the re-created images rely more on the hashes or on the training data of the re-creation system.

[3] https://www.anishathalye.com/2021/12/20/inverting-photodna/

3.3 Own Previous Work

An alternative robust hash of ours is the ForBild block hash presented in [20, 26]. It is the result of an evaluation of image hashing methods [39]. Based on this hash, we have added segmentation countermeasures based on face detection [28], watershed image segmentation [27] and machine learning [23, 24]. Beyond image recognition, we also addressed the possibility of combining privacy and robust hashing in [2,10,29]. As an alternative to robust hashing, we also evaluated feature-based montage detection using SIFT and SURF in [25].

4 The PhotoDNA Algorithm

The full mechanisms of PhotoDNA have not been disclosed beyond some basic papers by the creators [7] and a presentation by Microsoft. Nevertheless, there have been some attempts to recreate the algorithms from the known facts.[4] PhotoDNA is included in forensic tool sets.

From the available information, we assume the following algorithm:

1. Normalization: Convert to grayscale and downscale to 26x26 pixels. Note: Both operations can affect the hash result due to their handling of edges and textures, so reimplementations may produce values different from the leaked library.
2. Segmentation: The 26x26 pixels are divided into 6x6 quadrants with an overlap of 2 pixels. There are 6 quadrants per row, starting at 1, 5, 9, 13, 17, and 21. Figure 1 illustrates the quadrants. There are 36 quadrants
3. Gradients: Sobel gradients are computed for each quadrant. This results in four values representing horizontal and vertical positive and negative sums. The range of values is 0 to 255. In some papers it is mentioned that the value 255 means “255 or more”.
4. comparison: There are several ways to compare two hashes. The most common seems to be the Euclidean distance [8]. As far as we know, there are no official thresholds for deciding whether two images are identical or not. The choice of threshold will control the likelihood of false positives or false negatives.

The Figure 2 illustrates the hash calculation with an example. The full hash of the image is shown in Table 1. Note that the hash in the figure has

[4] https://www.hackerfactor.com/blog/index.php?archives/931-PhotoDNA-and-Limitations.html

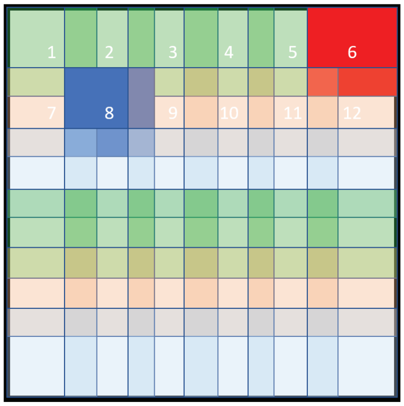
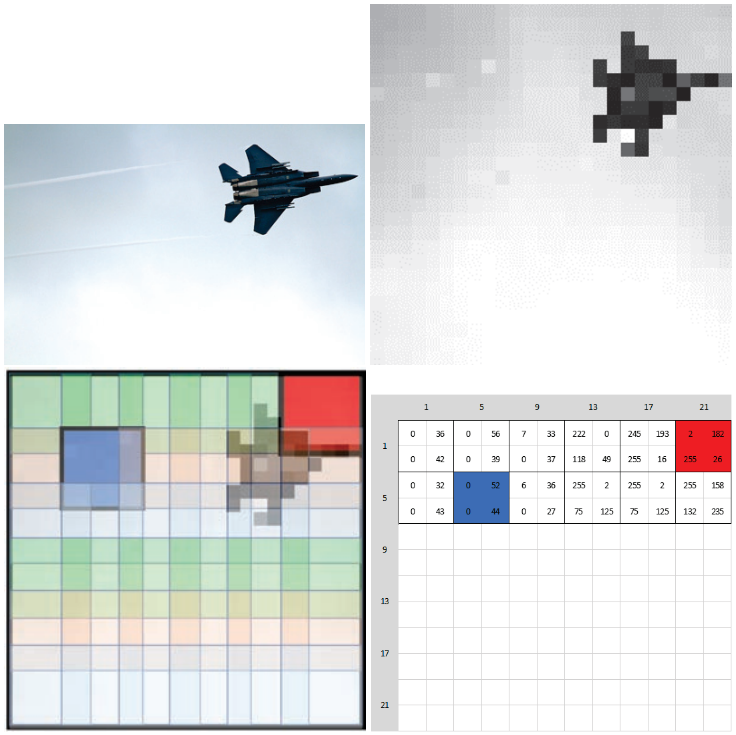


Figure 1 Sections of PhotoDNA. The numbers indicate the positions of the first 12 6x6 quadrants. The quadrants overlap by 2 pixels. The red quadrant 6 and the blue quadrant 8 illustrate the horizontal and vertical overlap.



	1		5		9		13		17		21	
1	0	36	0	56	7	33	222	0	245	193	2	182
	0	42	0	39	0	37	118	49	255	16	255	26
5	0	32	0	52	6	36	255	2	255	2	255	158
	0	43	0	44	0	27	75	125	75	125	132	235
9												
13												
17												
21												

Figure 2 Hash example of one image of the coco dataset. Top left: original image, top right: gray scale and resize to 26x26 pixel, bottom left: assignment to quadrants, bottom right: values in the of the hash of the first 12 quadrants.

Table 1 Hash of image in Figure 2

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
0	0	36	0	42	0	56	0	39	7	33	0	37
12	0	118	49	245	193	255	16	2	181	255	26	0
24	0	32	0	43	0	52	0	44	6	36	0	27
36	255	2	75	125	255	158	132	235	10	255	164	255
48	0	45	0	40	0	44	0	52	3	12	0	36
60	166	3	13	144	224	203	9	255	18	136	0	150
72	0	49	0	44	0	38	0	45	4	7	0	32
84	12	0	0	39	25	6	0	86	42	2	0	43
96	0	41	0	36	2	20	0	26	3	3	0	16
108	0	0	0	0	0	0	0	0	0	0	0	0
120	0	26	1	5	1	12	0	2	1	1	0	5
132	0	0	0	3	1	0	0	4	13	0	0	13

been restructured to match the meaning of the values and follows the quadrant concept, while in the table the hash is given as a sequence of values.

5 Analysis

Our first goal is to analyze the behavior of PhotoDNA in terms of its distribution of values. To do this, we generated a large number of hashes from the Microsoft coco dataset [14]. We used 40,000 images from "2017 test images" and 120,000 images from "2017 unlabeled images" together as a 160,000 image test set.

5.1 Hash Value Distribution

We first computed the hashes of all images, resulting in a matrix of 144x160,00 values ranging from 0 to 255. Then we analyzed the behavior of the values. The overall mean of all values is 63.732465. This is far from the theoretically expected 127.5 if there were an even distribution between 0 and 255. Image 3 shows the mean values of the individual 144 elements for the 160,000 hashes. As you can see, there are significant differences between the elements. The middle elements tend to have higher values than the edge elements.

In Figure 4 we show the distribution of values 0 to 255. One can see that low values occur more often than high ones, but there is a local peak at value 255.

PhotoDNA calculates hashes by subgroups and four directions. In Figure 5 we show that the likelihood for value 255 for direction c seems

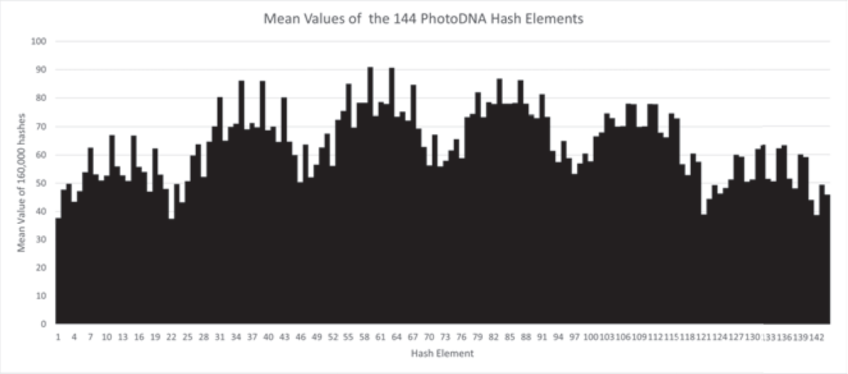


Figure 3 Mean of 144 hash elements.

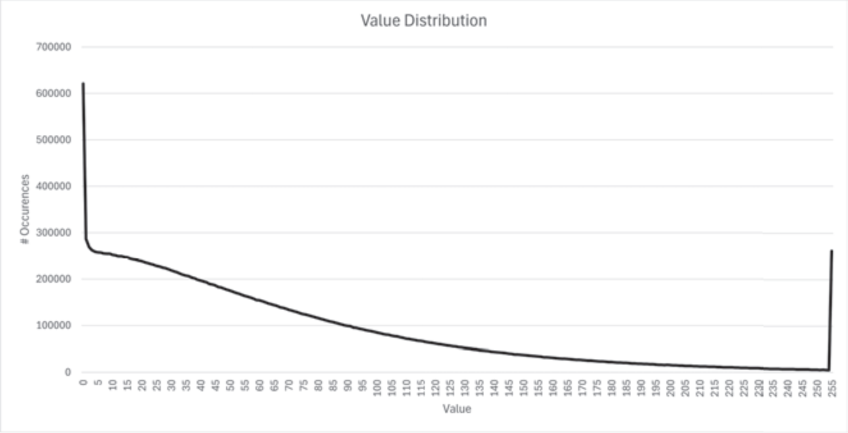


Figure 4 Value distribution.

to be higher than for the other three directions. We sorted the 144 elements in a new way, first arranging all 36 values of direction one, then all 36 values of direction two, and so on. This makes all values of one direction (identified by a to d in the figure) neighbors. For value 0 on the other hand no such clear distribution can be observed.

5.2 Structure Display

To analyze how PhotoDNA represents image structures, we computed the hashes of simple structures as shown in Figure 6. As we can see in the Tables 2, the inverted structures of images 1 and 2 the lead to high values

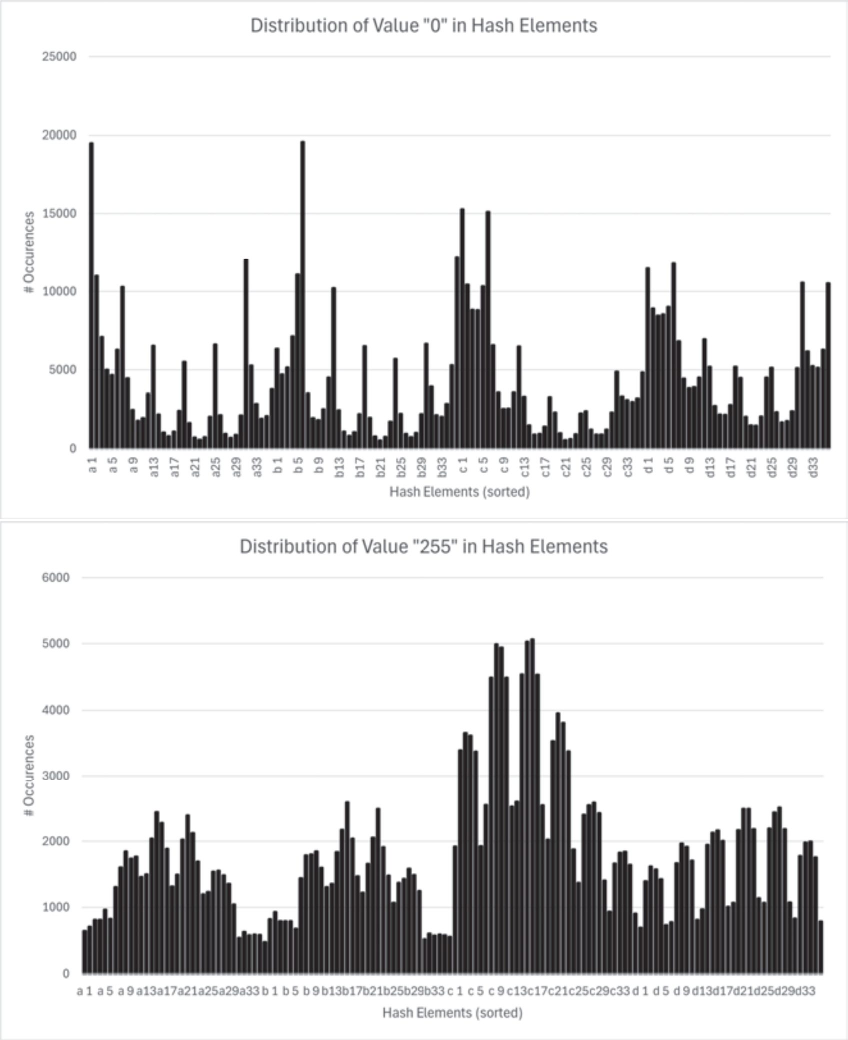


Figure 5 Value occurrences in directions a to d for values 0 (top) and 255 (bottom).

in the third or fourth position in the hash quadrants. The hashes of images 1 and 3 are identical. This could either mean that there is an additional normalization step in the hash generation to ensure maximum contrast, or that the edge values are so high that the lighter version has already reached the value “255 or higher”. The same is true for images 5 and 6. With the addition of lines like in images 4,7,8 and 9 more hash entropy is generated.

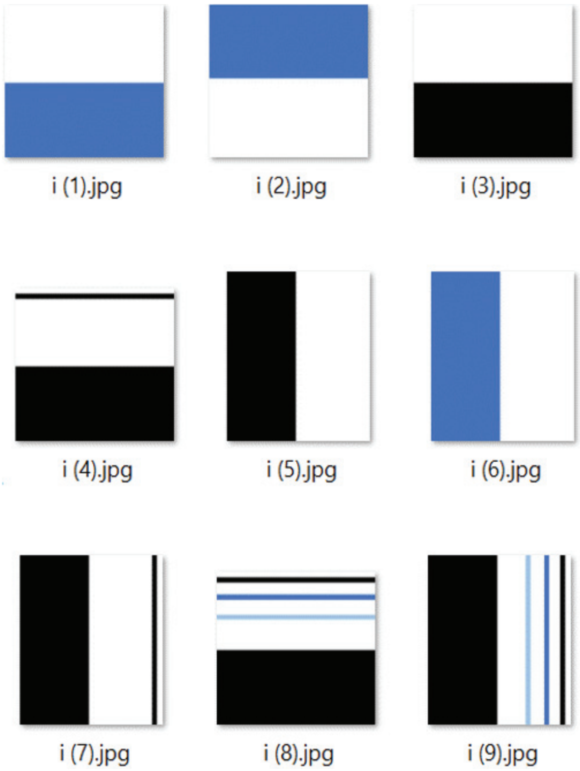


Figure 6 Nine test images with simple structures.

6 Evaluation Galaxy

To evaluate the false positive and false negative rates of PhotoDNA, we use the Galaxy test set, which shows a group of cheerleaders in similar poses. It has been used to evaluate robust hash algorithms [20, 26].

2,000 images were randomly selected and resized to a longest side length of 400 pixels. The following attacks were applied by Irfanview[5]:

- 80: convert=jpg80
- 70: convert=jpg70
- 60: convert=jpg60
- 30: convert=jpg30
- c2: crop=(2,2,1000,1000,0) and jpgq=80

[5] https://www.irfanview.com/, version 4.60

Table 2 Hash values of the nine example structures. Side numbers identify image 1 to 9, top numbers identify the hash positions 1 to 144

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	187	0	0	0	214	0	0	0	214	0	0	0	214	0	0	0	214	0	0	0	187
5	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	189	0	0	0
8	0	0	73	175	0	0	83	201	0	0	83	201	0	0	83	201	0	0	83	201	0	0	73	175
9	0	0	0	0	0	0	0	0	0	255	0	0	20	209	0	0	84	42	0	0	179	74	0	0

	25	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	216	0	0	0
8	0	0	40	81	0	0	46	93	0	0	46	93	0	0	46	93	0	0	46	93	0	0	40	81
9	0	0	0	0	0	0	0	0	0	255	0	0	23	238	0	0	96	49	0	0	204	84	0	0

	49	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**	**64**	**65**	**66**	**67**	**68**	**69**	**70**	**71**	**72**
1	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0
2	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255
3	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0
4	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0
5	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	216	0	0	0
8	0	0	213	19	0	0	244	22	0	0	244	22	0	0	244	22	0	0	244	22	0	0	213	19
9	0	0	0	0	0	0	0	0	0	255	0	0	23	238	0	0	96	49	0	0	204	84	0	0

	73	**74**	**75**	**76**	**77**	**78**	**79**	**80**	**81**	**82**	**83**	**84**	**85**	**86**	**87**	**88**	**89**	**90**	**91**	**92**	**93**	**94**	**95**	**96**
1	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0
2	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255
3	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0
4	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0
5	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	216	0	0	0
8	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0	0	0	255	0
9	0	0	0	0	0	0	0	0	0	255	0	0	23	238	0	0	96	49	0	0	204	84	0	0

	97	**98**	**99**	**100**	**101**	**102**	**103**	**104**	**105**	**106**	**107**	**108**	**109**	**110**	**111**	**112**	**113**	**114**	**115**	**116**	**117**	**118**	**119**	**120**
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	216	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	255	0	0	23	238	0	0	96	49	0	0	204	84	0	0

	121	**122**	**123**	**124**	**125**	**126**	**127**	**128**	**129**	**130**	**131**	**132**	**133**	**134**	**135**	**136**	**137**	**138**	**139**	**140**	**141**	**142**	**143**	**144**
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	255	0	0	0	255	0	0	0	0	0	0	189	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	255	0	0	20	209	0	0	84	42	0	0	179	74	0	0

- c5: crop=(5,5,1000,1000,0) and jpgq=80
- c10: crop=(10,10,1000,1000,0) and jpgq=80
- c20: crop=(20,20,1000,1000,0) and jpgq=80
- 200: resize_long=200 and jpgq=80
- 300: resize_long=300 and jpgq=80
- 500: resize_long=500 and jpgq=80

Table 3 Robustness in percent for 2,000 test images against attacks

Threshold	**80**	**70**	**60**	**30**	**c2**	**c5**	**c10**	**c20**	**300**	**200**	**500**
0	0	0	0	0	0	0	0	0	0	0	0
25	4,8	4,4	4,55	4,2	0	0	0	0	2,65	1,95	3,45
50	47,55	47,35	47,35	47,15	22,75	0	0	0	44,85	42,5	46,35
75	75,8	75,45	75,35	75,3	65,65	3,2	0	0	74,4	73,2	75,15
100	90,6	90,35	90,25	90,45	87,45	46,3	0,05	0	89,75	89,75	90,35
125	97,25	97,25	97,2	97,2	96,5	84	1	0	97,2	97,1	97,2
150	98,8	98,7	98,75	98,7	98,65	96,25	16,45	0	98,7	98,7	98,65
175	99,55	99,55	99,55	99,55	99,5	98,6	55,95	0	99,55	99,55	99,55
200	99,8	99,8	99,8	99,8	99,75	99,7	85,5	0,05	99,8	99,8	99,8
225	99,9	99,9	99,9	99,9	99,85	99,8	96,15	0,75	99,9	99,85	99,9
250	100	100	100	100	100	99,95	98,85	5,4	100	100	100
275	100	100	100	100	100	100	99,55	17,8	100	100	100
300	100	100	100	100	100	100	99,9	38,35	100	100	100

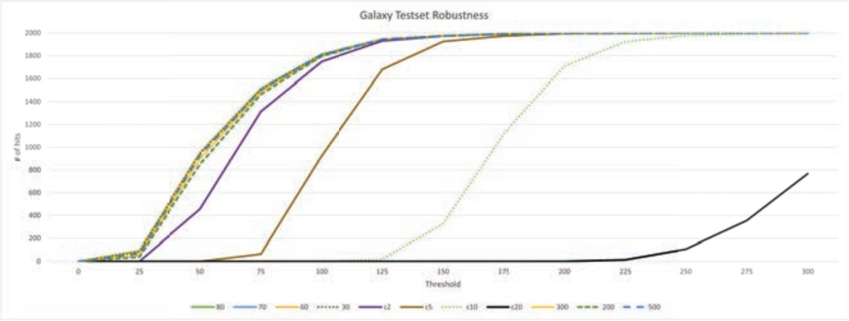


Figure 7 Robustness galaxy testset.

Cropping is done by setting the start point of the crop area to the top left. Crop(2,2,....) means that the image with a maximum length of 400 pixels has been cropped at position (2,2) by removing one pixel row and column from the top left. Crop(20,20,...) was the maximum crop, removing 19 rows or about 2.5% of the long side.

The Table 3 shows the robustness of the thresholds 0 to 300 in steps of 25. The overall robustness is very high, especially against jpeg compression. Robustness against one cropping line is achieved between thresholds 125 and 150. For nine cropping lines, the threshold is between 225 and 250. Figure 7 illustrates the results.

The Table 4 shows the collisions or false positives of the hash set. We calculated the hashes of all 2000 images and their distances to each other. Then we counted how many images had a hash distance below a given threshold. You can see that for the threshold of 225 we had a false positive rate of 0.3%.

Table 4 Thresholds and False Positives (in percent) for 2,000 images. Columns 1 to 5 mean the number of collisions. For example, with a threshold of 375 there is a 0,1% chance for three collisions

	# collisions				
Threshold	**1**	**2**	**3**	**4**	**5**
0	0	0	0	0	0
25	0	0	0	0	0
50	0	0	0	0	0
75	0	0	0	0	0
100	0	0	0	0	0
125	0	0	0	0	0
150	0	0	0	0	0
175	0	0	0	0	0
200	0	0	0	0	0
225	0,3	0	0	0	0
250	0,3	0	0	0	0
275	0,4	0	0	0	0
300	0,6	0	0	0	0
325	1,05	0,05	0	0	0
350	1,55	0,25	0	0	0
375	2,05	0,5	0,1	0,05	0
400	3,3	0,8	0,45	0,2	0,2
425	4,75	1,1	0,75	0,5	0,25
450	7	2,4	1,2	0,8	0,6
475	11	4,4	2,15	1,5	1,1
500	19,45	8,85	5,05	3,25	2,25

This means that the claimed error rates of FNR 2% and FPR 1 in 1 trillion would only allow a limited number of attacks, if any. For attacks like cropping, these failure rates seem unrealistic.

7 Coco Evaluation

We also evaluated the likelihood of collision for the larger coco set of 160,000 images, and looked at robustness against attacks using a subset of those images.

For the collision test, we computed the hashes of all images and randomly selected hashes and computed the minimum distances to the rest of the 159,999 images. Since collisions mean a hash distance below a defined threshold, we also analyzed the impact of the attack on a subset of 40,000 of the 1600,000 images 5. From the table, we can define a threshold depending on the attacks we need to be robust against. A threshold of 250 will result in a false negative rate of about one in a thousand with a JPEG quality factor of 70. Note that compared to the Galaxy set, the images are slightly larger,

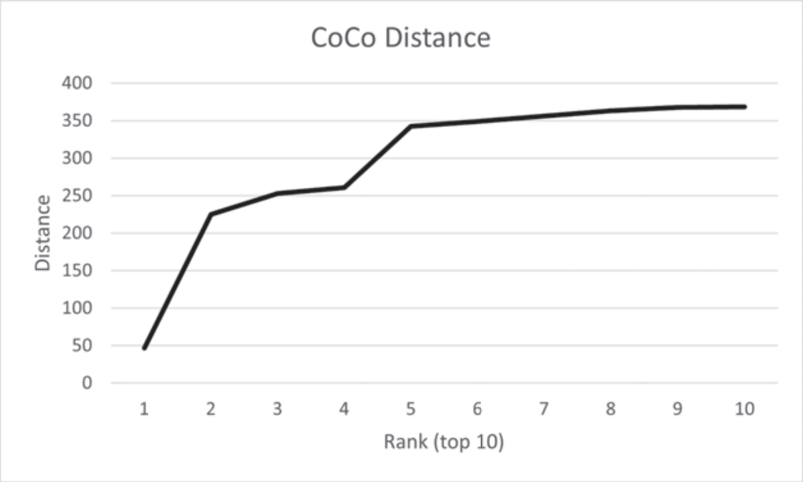


Figure 8 Top 10 low distances, 600 randomly selected images, 160.000 images in hash data base.

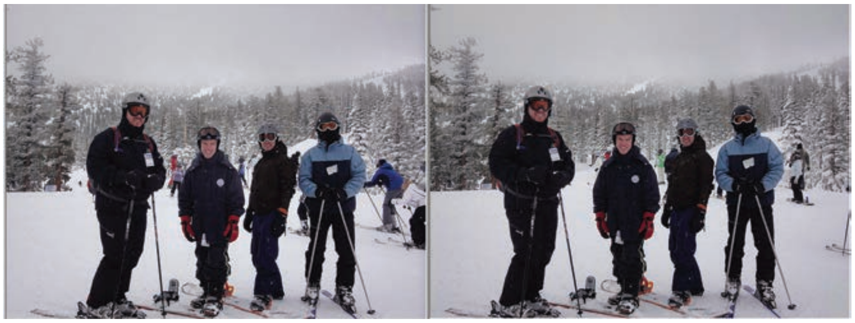


Figure 9 Example of image pair with small distance of 221.

and most have a long side of 640 pixels, so the impact of the fixed amount of cropping should be less.

Figure 8 shows the results of the collision test. There was one image pair in the coco set with a very low hash distance of 50. When we examined the pair, we found that the images were a color and a black-and-white version of the same photo. Figure 9 shows the second image pair in the ranking. Here we can see that PhotoDNA returns similar hashes for similar images, as expected. The two photos appear to be part of a series of images taken in a short period of time. The next set of images (not provided in this paper) in the ranking show giraffes in the same scene with moving heads from photo to photo.

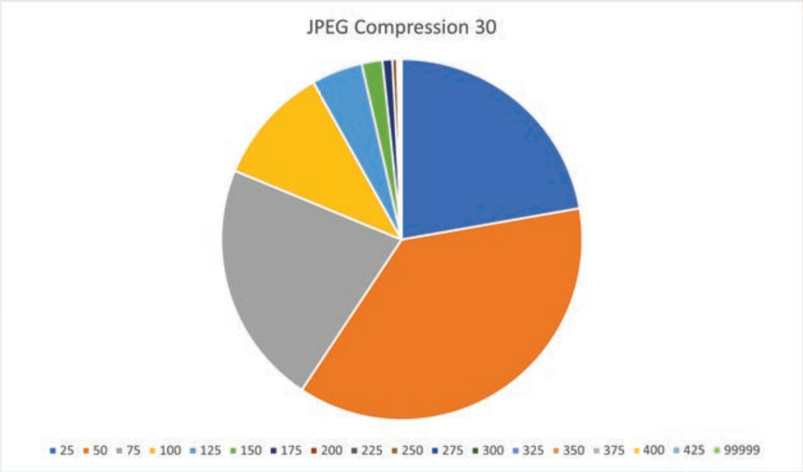


Figure 10 Diagram of JPEG compression with qf 30, legend shows the max values of the range columns from Table 5.

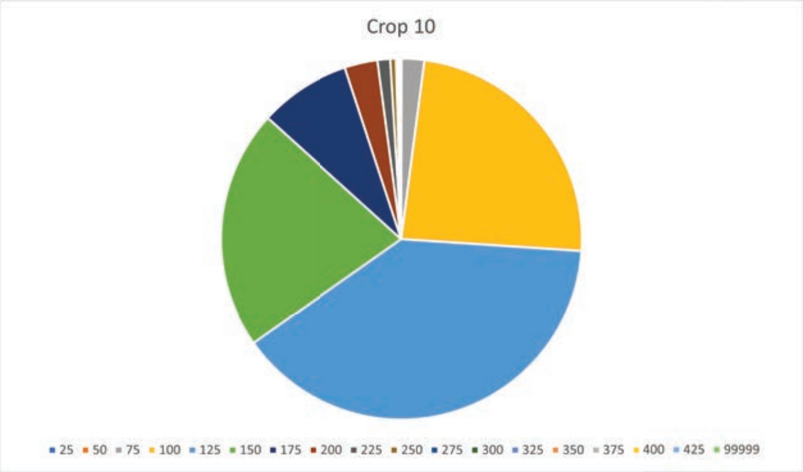


Figure 11 Diagram of cropping at position (10,10), legend shows the max values of the range columns from Table 5.

Figures 10 and 11 illustrate some details of Table 5. One can see, that cropping increases the hash distance between original image and modified images much stronger than JPEG compression, even at the low quality of QF 30. Bins 25 and 50 are almost empty in for cropping while being dominant for JPEG 30. For cropping, the largest diagram segments are 100, 125 and 150.

Table 5 Histogram of 40,000 images from coco test 2017. Range shows the range of hash distances, the cells the number of images with that distance under the given attack

Range		**80**	**70**	**30**	**C2**	**C10**	**R300**	**R500**
0	**25**	9101	9098	8889	2906	0	6809	8228
26	**50**	14709	14703	14842	18123	19	15972	15217
51	**75**	8726	8739	8757	10571	808	9385	8960
76	**100**	4231	4229	4253	4829	9593	4469	4329
101	**125**	1827	1823	1838	2032	15672	1905	1845
126	**150**	737	738	740	825	8592	762	738
151	**175**	341	345	353	365	3295	356	352
176	**200**	161	157	162	181	1180	175	164
201	**225**	94	96	89	92	456	93	93
226	**250**	30	29	32	30	201	31	31
251	**275**	19	18	19	21	84	17	19
276	**300**	11	12	13	12	40	13	11
301	**325**	7	7	7	7	25	7	7
326	**350**	3	3	3	3	16	3	3
351	**375**	0	0	0	0	5	0	0
376	**400**	0	1	0	1	6	0	0
401	**425**	2	1	2	1	4	2	2
426	**99999**	1	1	1	1	4	1	1

8 Discussion

From our experiments, we make a number of observations that should be relevant to future applications:

The hash demonstrates remarkable resilience to lossy compression, and scaling poses no issues, as anticipated, given that the image undergoes scaling down to 26 by 26 pixels during the hashing process. However, the touted advantage of PhotoDNA—its robustness against cropping—does have limitations. The robustness diminishes significantly once more than a few columns and rows are removed. To achieve high robustness, the threshold must be set so high that a substantial number of false positives become inevitable. Our experiments revealed this occurrence within the range of 9 to 19 deleted rows/columns, with the results becoming notably poor at 19, rendering it inappropriate to classify the hash as robust.

Regarding collision probability, indicating instances where the hashes of images are close to each other, PhotoDNA exhibits commendable performance. Different versions of an image exhibit significantly smaller distances from each other than random unrelated images. Nevertheless, it was observed that images from a series can indeed cause collisions with each other. This observation is noteworthy as it introduces a nuance to the conventional definition of "hash," typically reserved for methods that detect versions of a given image. With PhotoDNA, one might consider it a hash with an additional capability for recognizing similar content. Notably, there are forensic applications of PhotoDNA that align with this concept, utilizing the hash to search for images with similarities.

Looking at the results of the Galaxy and CoCo experiments, a threshold between 150 and 175 is a good compromise between true positives and false positives. No collisions were observed and the robustness is already around 99 percent for many image operations. However, it should be noted that the experiments can be seen as minimal compared to practical use. It is to be expected that in a scenario like chat control, billions of images will have to be checked every day, and the comparison will be made with a larger database than in the tests (2,000 or 160,000). For example, the UK's Internet Watch Foundation talks about 300,000 images,[6] safer.io states 29 million hashes,[7] but mixes cryptographic, image and video hashes together. Therefore, in practice it may be necessary to set the threshold lower.

The analysis of the basic structure of the hash in the "Analysis" section reveals a notable lack of robustness against mirroring or rotation, since there is no inherent normalization or alignment. Consequently, users of the hash must devise a solution to increase its robustness against such operations. One approach is to rotate the hash itself, which is particularly promising for simpler structures, as shown in Table 2. Alternatively, users may choose to rotate or mirror the images before re-hashing them. A note[8] from Microsoft dated June 2023 states that an update to PhotoDNA is available that deals with mirroring and rotation.

The hash, comprising 144 bytes, is relatively large compared to a block hash [38], which typically has only 32 bytes or 256 bits. This places it in the realm reminiscent of SIFT or SURF [35]. Additionally, the computation involving roots and squares is of high complexity compared to the straightforward computation of a Hamming distance. Given the importance of efficiency, especially for large hash collections, such as those cited above [37], it may be worthwhile to explore more efficient comparison methods.

[6] https://en.wikipedia.org/wiki/PhotoDNA

[7] https://safer.io/how-it-works/

[8] https://www.linkedin.com/pulse/update-photodna-adrian-chandley/

The distribution of hash values also raises noteworthy observations. In the extensive test involving 160,000 images, the mean value was approximately 64, representing 25% of the value range from 0 to 255. Notably, one edge direction produced more occurrences of value 255 higher values than the remaining directions, as depicted in Figure 5. Further investigation is warranted to determine if this behavior compromises resistance to attacks and contributes to an increased incidence of false positives. This is particularly relevant since the actual range of values used appears smaller than the available range.

9 Summary, Conclusion and Future Work

The aim of this paper is to give a first overview of the basic behavior of PhotoDNA with respect to robustness and its false positives. It can be stated that the hash provides very good results, however, it also does not provide the extreme performances that are sometimes mentioned in the public discussion. Some properties, such as the value distribution of the 144 hash elements, can be considered in more detail in future work. Likewise, it would be worthwhile to find a more efficient solution for the hash comparison.

As a more high level conclusion, PhotoDNA is a solid robust hash algorithm with potential of improvement. It needs to be stressed that the algorithm is not specifically desgined to deal with CSAM but will work with any other content as well. Infrastructures for content identification established for identifying CSAM therefore can be misused for other purposes by simply replacing the hash data base for comparison.

The detection of CSAM is one core issue in the EU discussion of rules to prevent and combat child sexual abuse.[9] Therefore understanding the performance of tools like PhotoDNA is vital to understand the impact of regulations utilizing client-side-scanning [22].

Acknowledgment

This research work has been funded by BMBF and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

[9] https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52022PC0209

References

[1] Ross Anderson. Chat control or child protection? *arXiv preprint arXiv:2210.08958*, 2022.

[2] Uwe Breidenbach, Martin Steinebach, and Huajian Liu. Privacy-enhanced robust image hashing with bloom filters. In Melanie Volkamer and Christian Wressnegger, editors, *ARES 2020: The 15th International Conference on Availability, Reliability and Security, Virtual Event, Ireland, August 25–28, 2020*, pages 56:1–56:10. ACM, 2020.

[3] Olena Buchko et al. Classification of confidential images using neural hash. *NaUKMA Research Papers Computer Science*, 5:68–71, 2022.

[4] Veena Desai and DH Rao. Image hash using neural networks. *International Journal of Computer Applications*, 63(22), 2013.

[5] Andrea Drmic, Marin Silic, Goran Delac, Klemo Vladimir, and Adrian S. Kurdija. Evaluating robustness of perceptual image hashing algorithms. In *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 995–1000. IEEE, 2017.

[6] Ling Du, Anthony T.S. Ho, and Runmin Cong. Perceptual hashing for image authentication: A survey. *Signal Processing: Image Communication*, 81:115713, 2020.

[7] Hany Farid. Reining in online abuses. *Technology & Innovation*, 19(3):593–599, 2018.

[8] Hany Farid. An overview of perceptual hashing. *Journal of Online Trust and Safety*, 1(1), 2021.

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[10] Marius Leon Hammann, Martin Steinebach, Huajian Liu, and Niklas Bunzel. Predicting positions of flipped bits in robust image hashes. *Electronic Imaging*, 35:375–1, 2023.

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

[12] J Langston. How photodna for video is being used to fight online child exploitation. *Combating child pornography: Steps are needed to ensure that tips to law enforcement are useful and forensic examinations are cost effective*, 2018.

[13] Hee-Eun Lee, Tatiana Ermakova, Vasilis Ververis, and Benjamin Fabian. Detecting child sexual abuse material: A comprehensive survey. *Forensic Science International: Digital Investigation*, 34:301022, 2020.

[14] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

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

[16] Dat Tien Nguyen, Firoj Alam, Ferda Ofli, and Muhammad Imran. Automatic image filtering on social networks using deep learning and perceptual hashing during crises.

[17] Jonathan Prokos, Tushar M. Jois, Neil Fendley, Roei Schuster, Matthew Green, Eran Tromer, and Yinzhi Cao. Squint hard enough: Evaluating perceptual hashing with machine learning. Cryptology ePrint Archive, Paper 2021/1531, 2021. https://eprint.iacr.org/2021/1531.

[18] Chuan Qin, Enli Liu, Guorui Feng, and Xinpeng Zhang. Perceptual image hashing for content authentication based on convolutional neural network with multiple constraints. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(11):4523–4537, 2020.

[19] Aditya Singh, Mayank Vatsa, and Richa Singh. Photo dna. 2020.

[20] Martin Steinebach. Robust hashing for efficient forensic analysis of image sets. In Pavel Gladyshev and Marcus K. Rogers, editors, *Digital Forensics and Cyber Crime*, volume 88 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 180–187. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[21] Martin Steinebach. An analysis of photodna. In *Proceedings of the 18th International Conference on Availability, Reliability and Security, ARES 2023, Benevento, Italy, 29 August 2023- 1 September 2023*, pages 44:1–44:8. ACM, 2023.

[22] Martin Steinebach. Erkennung von kindesmissbrauch in medien: Methoden und ihre herausforderungen. *Datenschutz und Datensicherheit-DuD*, 47(4):225–228, 2023.

[23] Martin Steinebach, Tiberius Berwanger, and Huajian Liu. Towards image hashing robust against cropping and rotation. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–7, 2022.

[24] Martin Steinebach, Tiberius Berwanger, and Huajian Liu. Image hashing robust against cropping and rotation. *Journal of Cyber Security and Mobility*, pages 129–160, 2023.

[25] Martin Steinebach, Karol Gotkowski, and Hujian Liu. Fake news detection by image montage recognition. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–9, New York, NY, USA, 2019. ACM.

[26] Martin Steinebach, Huajian Liu, and York Yannikos. Forbild: Efficient robust image hashing. In *Media Watermarking, Security, and Forensics 2012*, volume 8303, pages 195–202. SPIE, 2012.

[27] Martin Steinebach, Huajian Liu, and York Yannikos. Efficient cropping-resistant robust image hashing. In *2014 Ninth International Conference on Availability, Reliability and Security*, pages 579–585. IEEE, 2014.

[28] Martin Steinebach, Huajian Liu, and York Yannikos. Facehash: Face detection and robust hashing. In Pavel Gladyshev, Andrew Marrington, and Ibrahim Baggili, editors, *Digital Forensics and Cyber Crime*, volume 132 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 102–115. Springer International Publishing, Cham, 2014.

[29] Martin Steinebach, Sebastian Lutz, and Huajian Liu. Privacy and robust hashes: Privacy-preserving forensics for image re-identification. *Journal of Cyber Security and Mobility*, pages 111–140, 2020.

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

[31] Rui Sun and Wenjun Zeng. Secure and robust image hashing via compressive sensing. *Multimedia Tools and Applications*, 70, 06 2012.

[32] Zhenjun Tang, Lv Chen, Xianquan Zhang, and Shichao Zhang. Robust image hashing with tensor decomposition. *IEEE Transactions on Knowledge and Data Engineering*, 31(3):549–560, 2019.

[33] Zhenjun Tang, Fan Yang, Liyan Huang, and Xianquan Zhang. Robust image hashing with dominant dct coefficients. *Optik*, 125(18):5102–5107, 2014.

[34] Zhenjun Tang, Xianquan Zhang, Xuan Dai, Jianzhong Yang, and Tianxiu Wu. Robust image hash function using local color features. *AEU – International Journal of Electronics and Communications*, 67(8):717–722, 2013.

[35] Shaharyar Ahmed Khan Tareen and Zahra Saleem. A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In *2018 International conference on computing, mathematics and engineering technologies (iCoMET)*, pages 1–10. IEEE, 2018.

[36] Rebekka Weiß and Simran Mann. Bitkom on the eu proposal on chat control. *Policy*, 49(30):27576–214, 2022.

[37] Christian Winter, Martin Steinebach, and York Yannikos. Fast indexing strategies for robust image hashes. *Digital Investigation*, 11:S27–S35, 2014.

[38] Bian Yang, Fan Gu, and Xiamu Niu. Block mean value based image perceptual hashing. In *2006 International Conference on Intelligent Information Hiding and Multimedia*, pages 167–172. IEEE, 2006.

[39] Christoph Zauner, Martin Steinebach, and Eckehard Hermann. Rihamark: perceptual image hash benchmarking. In Nasir D. Memon, Jana Dittmann, Adnan M. Alattar, and Edward J. Delp III, editors, *Media Watermarking, Security, and Forensics III*, SPIE Proceedings, page 78800X. SPIE, 2011.

Biography

Martin Steinebach is the manager of the Media Security and IT Forensics division at Fraunhofer SIT. From 2003 to 2007 he managed the Media Security in IT division at Fraunhofer IPSI. He studied computer science at the Technical University of Darmstadt and finished his diploma thesis on copyright protection for digital audio in 1999. In 2003 he received his PhD at the Technical University of Darmstadt for this work on digital audio watermarking. In 2016 he became honorary professor at the TU Darmstadt. He gives lectures on Multimedia Security as well as Civil Security. He is Principle Investigator at ATHENE and represents IT Forensics and AI security. Before he was Principle Investigator at CASED with the topics Multimedia Security and IT Forensics.