Forensic Science International: Digital Investigation 44 (2023) 301509

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS 2023 EU - Selected papers of the Tenth Annual DFRWS Europe Conference

Hamming distributions of popular perceptual hashing techniques

Sean McKeown*, William J. Buchanan

School of Computing, Engineering, and the Built Environment, Edinburgh Napier University, Edinburgh, UK

ARTICLE INFO

Article history:

Keywords:
Perceptual Hashing
Fuzzy Hashing
Hash Matching
CSAM
Image Forensics

ABSTRACT

Content-based file matching has been widely deployed for decades, largely for the detection of sources of copyright infringement, extremist materials, and abusive sexual media. Perceptual hashes, such as Microsoft's PhotoDNA, are one automated mechanism for facilitating detection, allowing for machines to approximately match visual features of an image or video in a robust manner. However, there does not appear to be much public evaluation of such approaches, particularly when it comes to how effective they are against content-preserving modifications to media files. In this paper we present a million-image scale evaluation of several perceptual hashing archetypes for popular algorithms (including Facebook's PDQ, Apple's Neuralhash, and the popular pHash library) against seven image variants. The focal point is the distribution of Hamming distance scores between both unrelated images and image variants to better understand the problems faced by each approach.

1. Introduction

High-speed broadband has facilitated a transition from a text-based Internet to one with a media-heavy landscape, which is capitalised upon by platforms such as Facebook, YouTube, Instagram, TikTok, and others. Unfortunately, as with any technology, multimedia can be used for illegal and abusive purposes, necessitating that law enforcement agencies, social media platforms, cloud providers, and so on, engage in some form of content moderation. At Web scale the manual evaluation of multimedia becomes infeasible, resulting in the use of automatic detection technologies. Content-based detection is implemented for a variety of reasons, such as for the detection of extremist and violent material (Microsoft Corporate Blog, 2019), copyright infringing material (Saadatpanah et al., 2020), and the detection of Child Sexual Abuse Material (CSAM) (Lee et al., 2020). The CSAM use-case is particularly important, as the Internet Watch Foundation reports a sharp increase in the detection and distribution of CSAM media - taking action against 250,000 websites in 2021 alone (The Internet Watch Foundation, 2021). Content matching technologies have already been deployed for many cloud-based services, backed by Microsoft's PhotoDNA (Krawetz) and Facebook's PDQ (Facebook), with Apple looking to introduce client-side device scanning (Apple, 2022) to further combat the growing problem.

For large-scale deployment, it is imperative that these detection technologies are well understood, both in terms of their strengths and weaknesses, in order to avoid improper configuration/deployment and incorrect actionable intelligence. The primary contribution of this work is an analysis of contemporary algorithms in still-image content-based perceptual hash matching, with analysis of their robustness in the face of content-preserving attacks, with experiments being conducted at the million-image scale.

2. Background and related work

Content-based file matching can take a variety of forms, with the approach archetypes corresponding to: i) Exact matching with Cryptographic hashes; ii) Approximate similarity binary-level matching; and iii) Semantic approximate matching of file contents (Breitinger et al., 2013).

In each case, a hash digest is created which serves as a fingerprint/signature for the file, which can then be compared to the hashes of other files to detect identical or similar file content. Cryptographic approaches are commonly used in CSAM detection in digital forensics analysis, however, they are easily defeated by modifying a single-bit in a file. Approximate matching at the binary level is suitable for some file types, but is generally a poor choice for media files as simply changing encoding parameters will result in completely different binary content (Farid, 2021). We therefore

* Corresponding author.
E-mail addresses: S.McKeown@napier.ac.uk (S. McKeown), B.Buchanan@napier.ac.uk (W.J. Buchanan).

https://doi.org/10.1016/j.fsidi.2023.301509

choose to focus on the semantic domain, which in this case is Perceptual Image Matching.

2.1. Perceptual Image Matching

Perceptual hashing (Farid, 2021; Hadmi et al., 2012) approaches, which are inspired by the domain of Content-Based Image Retrieval (CBIR) (Rafiee et al., 2010), seek to allow visual similarity and identity between images to be quantified and measured. These techniques are ideally resistant to content-preserving modifications, such as adding noise, or cropping.

A wide array of features have been used as the basis for generating robust image signatures. Histograms and statistical information about the entire image may be used, providing a high-level representation which is insensitive to small, localised, changes in the image. This can take the form of colour histograms (Swain and Ballard, 1991), texture and edge histograms (Manjunath et al., 2001), or frequency domain statistics (Venkatesan et al., 2000). Properties of human vision may also be exploited, such as insensitivity to high-frequency changes in an image over a small area - a property which is exploited by JPEG compression (Wallace, 1992). Low frequency properties of an image may be used to derive a perceptual hash (Fridrich, 1999; Fridrich and Goljan, 2000), which provides robustness to compression artefacts and other content-preserving modifications. In a similar vein, coarse image representations may be used, such as low-resolution versions of the image, or the average colour value of sub-blocks in the image (Steinebach, 2011). Alternatively, invariant relationships in the image may be exploited, such as those found on radial lines projected out from the centre of the image (Standaert et al., 2005), which is particularly effective against re-scaled images.

These hashes are then compared using a variety of similarity metrics (Hamming distance, Euclidean distance, Earthmover distance, L2 distance, and so on), typically normalised to a value between zero and unity, with unity representing complete similarity.

It should also be noted that there is a distinction between hashes which are generated using a *shallow*, heuristic, approach and those derived from a deep learning, data-driven, approach (Dolhansky and Ferrer, 2020). In the former case, hashes are generated in a deterministic manner (such as from statistical features of the image), while the learning approaches will generate different hashes based on the dataset used to train the corresponding model.

2.2. Evaluating Perceptual Image Matching against attack

Despite the popularity and wide spread use of perceptual hashing, the robustness of these algorithms is not well understood for malicious and abusive content detection (Hao et al., 2021), as adversaries can produce image variants that may hinder detection and identification.

One of the most popular, and easily available, perceptual hashing algorithms is pHash (DCT transform based) (Zauner, 2010). This hashing method is available in the Python ImageHash library (Buchner), and is often evaluated together with the other algorithms in the library: ahash (average block colour); dhash (adjacent block colour); and whash (DWT-based) (Dolhansky and Ferrer, 2020; Hao et al., 2021; Hamadouche et al., 2021; Drmic et al., 2017; Jain et al., 2022).

Hamadouche et al. (2021) studied the ImageHash algorithms together with SVD-Hash (Singular Value Decomposition) against a variety of image-wide filters, noise, and scaling for a small dataset of 800 images. While ahash typically produced the smallest distances between original and modification, pHash and dhash were the only algorithms to produce normally distributed Hamming distances - a property which will be discussed further in Section 3.3. Jain et al. (2022) performed more sophisticated perturbation attacks against ImageHash algorithms and Facebook's DCT-based PDQ on over one million images. The authors noted that by using black-box attacks, it was possible to manipulate modified image distances to the extent that the False Positive Rate would become unacceptably large in all cases. Similar work by Hao et al. (2021) for ImageHash and Blockhash (average block colour) allowed for large distances to be achieved for each algorithm using noise, cropping, rotation and scaling attacks. Dmric et al. (Drmic et al., 2017) evaluated the ImageHash library, with 1,480 images, against user-level modifications, such as resizing, rotation, adding borders, etc., together with social media post-processing tests. pHash generated the best F1 score (weighted combination of precision/recall) when aggregated across attacks, with social media manipulations disturbing performance less than the more direct manipulations.

McKeown and Russell (McKeown et al., 2019) explored the use of pHash and Blockhash for matching originals to thumbnail cache entries in Windows Vista, 7, and 10, noting that neither algorithm could achieve an acceptable False Positive Rate to False Negative Rate trade-off for forensics purposes. However, the combination of both algorithms may be sufficient. The authors also noted particular weaknesses in the algorithms, such as fractal/patterned images for pHash and solid colour background/smooth gradients for Blockhash.

Looking more widely, Zauner (2010) and Breitinger et al. (2013) compared DCT, Block Mean Based, Radial (e.g. Radon projection) and Marr-Hildreth (MH) operator hashes. Particular attention was paid to the strengths and weaknesses of each algorithm for each attack. JPEG compression did not have much effect on any tested algorithm, while resizing only particularly affected MH (most algorithms downscale images in pre-processing). Mirroring, rotating, and cropping all have much larger effects on performance across attacks, except when algorithms build in specific handling (such as for rotation in rHash (Steinebach, 2011)).

Machine learning approaches have also been shown to be susceptible to trivial content-preserving manipulations, with Struppek et al. (2022a) demonstrating that Apple's Neuralhash is not robust to gradient and familiar transformation-based attacks, with hash collision attacks also being possible. Dohlansky et al. (Dolhansky and Ferrer, 2020) found similar results for AlexNet, ResNet, and EfficientNet, although the authors noted that cross-attacks between shallow and deep approaches do not work well, such that combining algorithms from each class may lead to additional robustness.

3. Methodology

While there is a body of existing work in the evaluation of perceptual hashing, there is generally a focus on aggregating statistics across multiple attack types, or in generating specific attack scenarios, rather than understanding the behavioural properties of the approaches against common user-level attacks. With a similar approach to Hamadouche et al. (2021), we set out to understand the Hamming distance distribution of perceptual hashing algorithms in order to better quantify behaviour for not only the aggregate/average case, but also the best and worst case scenarios. To achieve this, we make use of the Flickr 1 Million dataset (MIRFLICKR Download), removing SHA256 hash duplicates, which allows us to scale experiments to large numbers of natural images. The original hashes, Hamming scores, and additional statistical data used in this paper are available online with DOI: 10.5281/zenodo.7426035.

Table 1
Perceptual Hashes used in experiments.

Blockhash	An implementation of the Block Mean Based approach (Yang et al., 2006). It can be found, written in C, on the Commons Machinery Github (Commonsmachinery, 2018). Generates 256-bit hashes.
ColourHash	Part of the Python ImageHash Library (Buchner). Images are matched based on colour distributions. Generates 44-bit hashes.
NeuralHash (Apple)	Apple's CSAM hashing scheme. This is a machine-learned approached, with the neuralhash_128x96_seed1.dat model being extracted from an iPhone14,5_15.3.1_19D52 IPSW firmware image. Steps taken folllow those on Github (Ygvar, 2022). Generates 96-bit hashes.
PDQ (Facebook)	Facebook's improved version of the pHash (DCT) algorithm, with optimisations for downsampling, and larger hash output size by default (for better Web-scale accuracy). The Python implementation - available on Facebook's Threat Exchange Github (Facebook) - was used. Generates 256-bit hashes.
pHash	DCT-based, as implemented in the ImageHash Python library (Buchner). Generates 64-bit hashes.
Wavehash	A hash based on the Discrete Wavelet Transform (DWT) (as opposed to the DCT based approaches above). Also part of the Python ImageHash library (Buchner). Generates 64-bit hashes.

Table 2
Image modifications applied to each image in the dataset.

Border (30px)	Add a fixed size 30-pixel black border to the outside of the image, extending it, rather than overwriting any of the existing content.
Compression (Q30)	Reduce the JPEG image quality to 30%, scaling down the default JPEG quantization tables. Other modifications pass-through the quantisation tables (approx 96% quality for this dataset).
Crop (5%)	Remove 5% of the image from the top, left, right, and bottom of the image, in effect reducing the overall pixel count to 81% of the original ($0.9height \times 0.9width$).
Mirror (x-axis)	Flip the image on its x-axis, preserving viewability, but generating significant pixel/binary level changes.
Scale (1.5x)	Scale the image up to 150% the size of the original, using the PIL (S. L. AB (PythonWare) Image.resize function.
Thumbs96 (Windows)	Generate legitimate Windows 10 96×96 pixel thumbnails as per the methodology in McKeown and Russel (McKeown et al., 2019), which generate a larger Hamming distance in this cited work for pHash and Blockhash than their 256×256 thumbnail counterpart.
Watermark	Add a watermark to the bottom right of the image. The watermark consists of a logo, text and a URL. The watermark was scaled to 10% of the image height, with a minimum of 40 pixels.

3.1. Selection of hashing algorithms

Our perceptual hash selection takes into account a wide range of feature extraction techniques, but also focuses on popular algorithms. The chosen hash algorithms are listed in Table 1. The list covers: frequency transforms; colour histograms; block mean colour; and deep learning-derived approaches. Aside from additional deep learning approaches, one notable omission of the shallow approach to hashing in this testing is Microsoft's PhotoDNA (Krawetz) as it is a controlled technology and not readily available. While such evaluations may already be available to those with access, it would be beneficial for future work to make evaluations open-source, particularly as it is the de-facto standard for the perceptual detection of CSAM (Farid, 2018). The details of its implementation are not widely detailed, however it appears that PhotoDNA uses an edge-based method (Prokos et al., 2021), such that it is unclear what particular weaknesses it may share with the algorithms presented in this paper. Another possible algorithm class to include in future testing is the Scale Invariant Feature Transform (SIFT), a feature-point based approach, which is more commonly used for image forgery detection (Bourouis et al., 2020). ForBild/rHash (Steinebach, 2011) was also considered, though it was excluded early on as the sourced implementation generated many hash collisions due to black backgrounds, or similar colour compositions.

3.2. Image modifications

A set of six image modifications were chosen as *attacks* against the perceptual hashing algorithms. Each modification was implemented using the Python PIL library (S. L. AB (PythonWare). This means that all images were modified using the same tool, introducing some limitations in terms of the diversity of the images produced, but we do not expect this to have a significant effect on the outcomes.

To reduce the total processing time across all hash and modification permutations, a random subset of 250,000 images was selected for the creation of modifications. Table 2 describes the chosen modifications, which are intended to reflect the low-barrier to entry, unsophisticated, content-preserving black-box attacks that a typical user may employ which do not compromise viewability (such as for evading copyright detection (Jabade, 2016)), with the addition of Windows 10 generated thumbnails as a realistic downscaling exemplar. Examples of the visually distinct modifications are provided in Fig. 1.

3.3. Evaluation process and criteria

There are several things to consider when evaluating perceptual hashing systems. The first is the relative discrimination behaviour between unrelated images (inter-image). When measuring the distance between two hashes (usually via normalised Hamming distance), unrelated images should be essentially tossing a coin for each bit, resulting in a distance of around 0.5, on average (mean and median). Additionally, we should expect that very close hashes, and particularly exact matches, should not occur in large groups, and the occurrences should be reasonable and expected, that is, highly similar images.

In order to characterise **inter-score** behaviour for the original (unmodified) images, each image in the Flickr 1 Million dataset was compared to a random selection of 50 other images, for a total of approximately 50 million comparisons[1] to generate statistics from. To better understand the impact of the various image modifications on the behaviour of each hash, inter-scores were also analysed for each of the seven modifications, comparing to 50 random images in each 250,000 set, resulting in 12.5 million additional samples per modification for each hashing algorithm. Results for these comparisons (via normalised Hamming distances) are reported in Section 4.1.

The second main characteristic is the performance of a perceptual hashing algorithm when comparing an image to its

[1] There are a possible total of 500 billion comparisons ($\frac{1million \times 1million}{2}$) for 1 million images, which is unnecessary to acquire the understanding we need here.

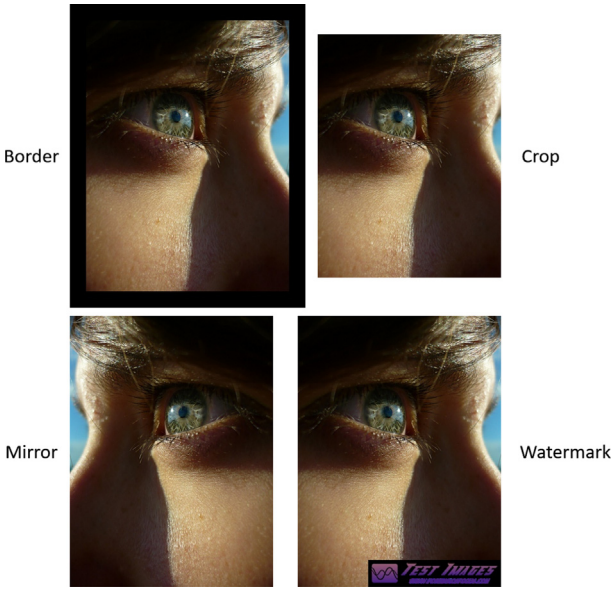

Fig. 1. Visibly distinct modification examples for 50.jpg in the dataset.

variant. This second image could be a pixel perfect copy, a re-encoded version of the image into a different file format, a thumbnail, cropped manipulation, etc. Ideally the perceptual hashing algorithm generates the same, or similar, hash for each of these images, such that the Hamming distance is zero, or very small. The distance distributions of inter- and intra-scores for each algorithm then allow for a wider understanding of potential false positive and false negative rates for a given threshold distance, as discussed in prior work (Hao et al., 2021; McKeown et al., 2019).

We explore the **intra-score** characteristics of the various algorithms and image modifications in Section 4.2. Each modified image in the dataset was hashed and compared to its original version, generating 250,000 Hamming distance comparisons for each hash algorithm/modification pair. Results present not only measures of distribution (range, mean, median, and standard deviation), but also exact hash match percentages to demonstrate particularly strong matching performance.

4. Findings

4.1. Inter-score (different images)

The findings for inter-image distributions (i.e., between unrelated images) are presented first, with Section 4.1.1 describing each algorithm's behaviour for the original Flickr 1 Million image set. This essentially acts as a baseline for each algorithm's behaviour when images are not expected to match. Section 4.1.2 compares images in each modification class (e.g., cropped to cropped) in order to determine if there is any bias introduced by the modifications themselves, even if the images are still assumed to be unrelated. This is followed by Section 4.2 which moves on to examine the impact that the modifications have when comparing the same image to versions of itself, providing further insight into difficulties and biases.

4.1.1. Original to original

Inter-score comparisons between original images in the dataset, for each hash, are presented in Table 3, with distributions depicted in Fig. 2.

While almost all of the algorithms do well to centre inter-image scores around 0.5, which is clearly a design goal (best achieved by PDQ), Colourhash behaves quite differently. Colourhash has much lower scores for unrelated images, meaning that unrelated images may match with a higher probability than is desirable. This is particularly clear when analysing equivalence classes (distance 0), where three hashes accounted for over 60,000 images in the dataset, making it functionally indiscriminate. Wavehash, despite its reasonable mean/median scores, produced a single class of 5,527 items, and several classes in the hundreds, again making it a poor discriminator at scale. Of the remaining algorithms, only Blockhash produced a class size greater than 5, with two hashes corresponding to over 100 images.

In terms of their distributions, the algorithms which use lower hash sizes by default generate sparse distributions, with some distance intervals being unused, essentially quantizing the output across the Hamming space. This is particularly clear for Colourhash, pHash and Wavehash, where the sparseness creates a large mismatch with the Normal Probability Density Function (NPDF) in Fig. 2. NeuralHash has an intermediate hash length, but fits the NPDF much better, despite a notable gap in the centre of the distribution. The spikes produced by Blockhash and PDQ, which both use large hashes, are therefore likely algorithmic, making certain hash codes and distances more likely to occur by chance.

Of the hashes which conform to 0.5 as the median/mean, Wavehash has the widest distribution (highest standard deviation), with tails making it reasonably likely that it generates a zero distance match by chance at any reasonable scale, as was evidence from the equivalence class of 5,527 items. PDQ, on the other hand, has a very tight distribution, likely by design, allowing it to scale upwards without concern for accidental matches.

Table 3
Inter-score normalised Hamming distances between random images in the Flickr 1 Million dataset. Each image was compared to 50 random images. The ideal value is 0.5 distance to make best use of the Hamming space.

hash alg.	mean	median	stdev	range
blockhash	0.4923	0.4922	0.0785	0.0078–1.0000
colourhash	0.1601	0.1591	0.0547	0.0000–0.3636
neuralhash	0.4973	0.5000	0.0600	0.0521–0.7917
pdq	0.5000	0.5000	0.0321	0.0391–0.6953
phash	0.4904	0.5000	0.0649	0.0938–0.8438
wavehash	0.4854	0.5000	0.1241	0.0000–1.0000

4.1.2. Modified to modified

From the equivalence class analysis alone, the Colourhash and Wavehash algorithms should be discounted as potential solutions. However, Wavehash remains in the following tables to provide insight into how a non-DCT based frequency transform is impacted by the modifications.

Table 4 represents inter-score distances for each algorithm where the images in each modification class are compared to each other (e.g. cropped-to-cropped). The perceptual hashes largely remain stable across the modified image comparisons, though adding in repeated content in the form of a border has had a larger than expected impact for some hashes. Blockhash, pHash and Wavehash all have sizable reductions in mean distance between unrelated images as a result of the shared border. This means that they are more likely to match images with borders to each other, regardless of the content residing within the border. NeuralHash appears to be the only algorithm which is not particularly affected by the addition of a border. The watermark has a similar effect, though it is not nearly as pronounced.

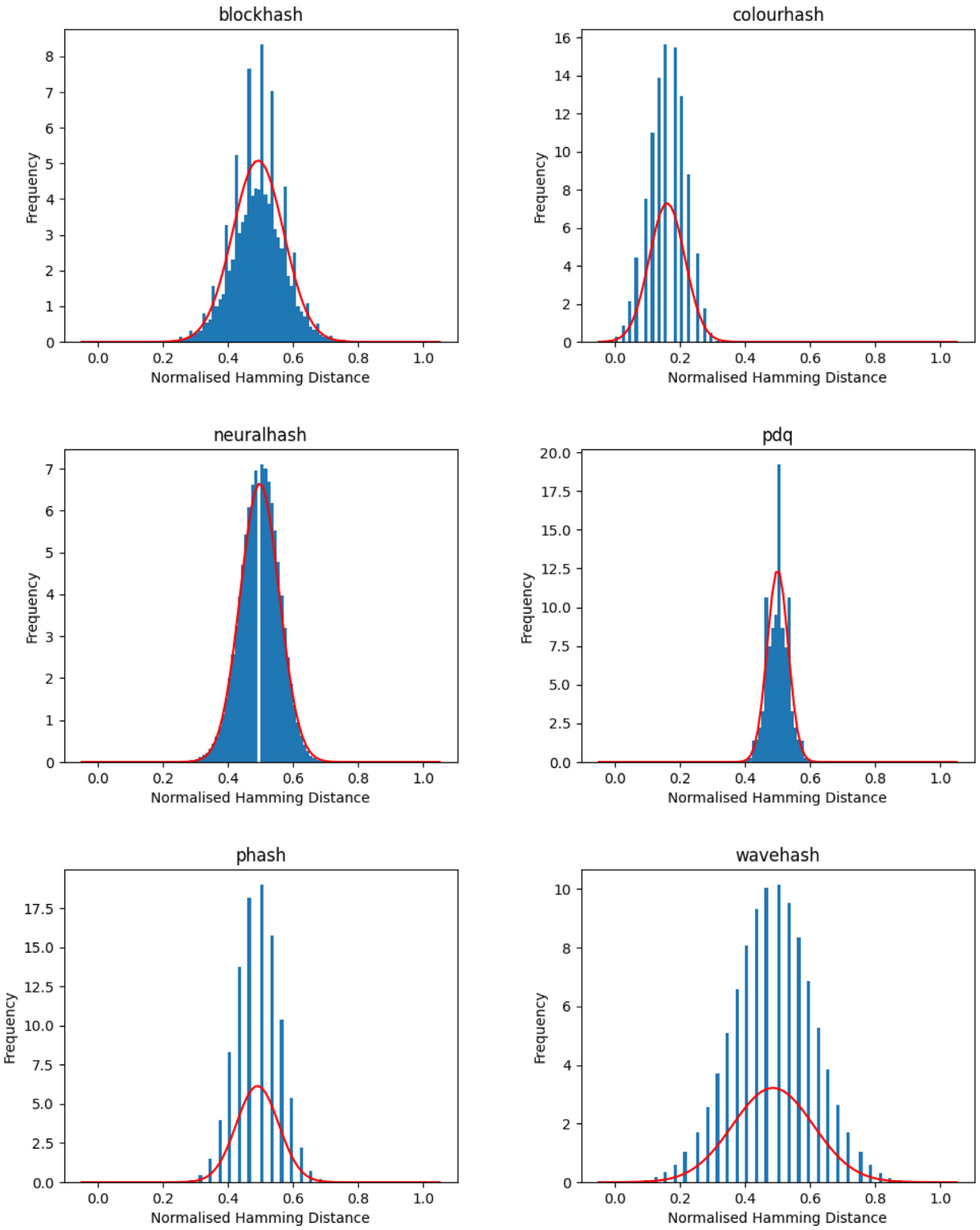

Fig. 2. Normalised Hamming **inter-scores** with Normal Probability Density Function (NPDF) (50 million pair-wise samples) for the original Flickr 1 Million dataset. a) Blockhash, b) Colourhash, C) NeuralHash, d) PDQ, e) pHash, f) Wavehash

Table 4
Inter-score normalised Hamming distances between random images within each modification category for the Flickr 1 Million dataset. Each image was compared to 50 random images (e.g. each cropped image was compared to 50 random cropped images for the same algorithm). The ideal mean/median is a distance of 0.5, demonstrating that the modification does not bias the algorithm.

hash algorithm	modification	mean	median	stdev	range
blockhash	border	0.3284	0.3281	0.0549	0.0391–0.8047
blockhash	compression	0.4922	0.4922	0.0782	0.0391–0.9805
blockhash	crop	0.4941	0.4922	0.0761	0.0234–0.9766
blockhash	mirror	0.4921	0.4922	0.0784	0.0312–0.9609
blockhash	scale	0.4922	0.4922	0.0783	0.0391–0.9492
blockhash	thumbs96	0.4921	0.4922	0.0787	0.0234–0.9609
blockhash	watermark	0.4789	0.4766	0.0758	0.0391–0.9531
neuralhash	border	0.4969	0.5000	0.0601	0.0000–0.7812
neuralhash	compression	0.4974	0.5000	0.0600	0.0729–0.7812
neuralhash	crop	0.4976	0.5000	0.0599	0.0833–0.7812
neuralhash	mirror	0.4974	0.5000	0.0600	0.0521–0.8021
neuralhash	scale	0.4973	0.5000	0.0600	0.0417–0.7812
neuralhash	thumbs96	0.4969	0.5000	0.0602	0.0312–0.7812
neuralhash	watermark	0.4969	0.5000	0.0602	0.0521–0.7812
pdq	border	0.4794	0.4766	0.0363	0.2109–0.6719
pdq	compression	0.5000	0.5000	0.0321	0.2500–0.6797
pdq	crop	0.5000	0.5000	0.0318	0.2891–0.7109
pdq	mirror	0.5000	0.5000	0.0321	0.2969–0.6797
pdq	scale	0.5000	0.5000	0.0321	0.2812–0.7188
pdq	thumbs96	0.5000	0.5000	0.0321	0.2812–0.7031
pdq	watermark	0.4941	0.4922	0.0333	0.2031–0.7344
phash	border	0.4283	0.4375	0.0727	0.0625–0.8438
phash	compression	0.4904	0.5000	0.0649	0.1562–0.8438
phash	crop	0.4907	0.5000	0.0641	0.0625–0.8125
phash	mirror	0.4904	0.5000	0.0649	0.1562–0.8438
phash	scale	0.4904	0.5000	0.0649	0.1250–0.8438
phash	thumbs96	0.4904	0.5000	0.0650	0.1250–0.8125
phash	watermark	0.4704	0.4688	0.0674	0.0000–0.8125
wavehash	border	0.3013	0.2812	0.1167	0.0000–0.9688
wavehash	compression	0.4854	0.5000	0.1240	0.0000–1.0000
wavehash	crop	0.4869	0.5000	0.1212	0.0000–1.0000
wavehash	mirror	0.4854	0.5000	0.1239	0.0000–1.0000
wavehash	scale	0.4854	0.5000	0.1240	0.0000–1.0000
wavehash	thumbs96	0.4854	0.5000	0.1241	0.0000–1.0000
wavehash	watermark	0.4631	0.4688	0.1174	0.0000–1.0000

4.2. Intra-score (versions of the same image)

Intra-score values, comparing the original image to its modifications, are depicted in Table 5. These scores give us an opportunity to explore how well the various techniques can deal with attempts to circumvent content-based detection.

All algorithms stumble into very high Hamming distances at some point, represented by the high maximum distance values, which simply suggests that there is at least one image in the dataset that troubles each algorithm for almost every modification. These outliers are often hidden when considering aggregate behaviour, and they do not always occur, but particularly troublesome ranges are present for *Blockhash-Mirror*, *PDQ-Border*, *PDQ-Mirror*, and *pHash-Mirror*.

Most algorithms coped well with image scaling, likely because one of the main steps in generating perceptual hashes is to downscale the image to a manageable size/complexity. Most algorithms have a very high exact match percentage for scaling,[2] peaking at 97.9% with Wavehash, meaning that only 2.1% of upscaled images would not have an exact hash match. Interestingly, while thumbnail performance is often almost as good, it does not produce anywhere near the same percentage of exact hash matches. All algorithms also coped very well with poor quality JPEGs.

NeuralHash produces consistently good results, though it has a very wide distribution (almost normal around 0.3) for mirroring attacks, when ideally the distribution should be very long tailed with most distances around zero. On paper, Wavehash also falls into this category, but the false positive rate is likely too high for most use cases due to the inter-score distribution. Blockhash, pHash and PDQ all perform well on average, however, each of them have cases where the distribution is very wide due to a high standard deviation (border, crop and mirror for all three, and watermark for PDQ and pHash). Generally, mirroring an image seems to cause the most disturbance, with many algorithms essentially distributing scores as if it was an unrelated image. Borders appear to be second most impactful due to the introduction of common content.

5. Conclusion and future work

In this work, we have explored several popular and widely used perceptual hashing algorithms in order to understand the distributions of their Hamming distances when discriminating between unrelated images, and versions of the same image. Of the tested algorithms, all but ColourHash and Wavehash were shown to have reasonable inter-score distributions, resulting in few false positives by complete chance. PDQ is particularly strong in this regard. Surprisingly, adding a border or watermark is enough to substantially throw off the distances between unrelated images for most algorithms, simply due to a small portion of shared content being introduced.

When comparing the images to various content-preserving modifications, mirroring the image on the x-axis was found to be particularly destructive across the board, with the addition of a border being a distant second. While distances should be clustered around 0 in these intra-score cases, distributions for unfavourable modifications often resulted in relatively wide spreads, sometimes centred around 0.4 and 0.5. Additionally, most techniques had at least one poorly handled image for each modification. Overall, NeuralHash appears to be the least affected by the modifications, in addition to having a desirable inter-score distribution, suggesting that deep, learned, approaches may be strong against naive user-level attack.

The understanding generated above allows for further insights to be mined in terms of setting appropriate thresholds (given some level of false positive/false negative trade-off) (McKeown et al., 2019), but also to allow us to consider black-box mitigations against common problems in matching. Pre-processing approaches for mirrored and bordered images would help reduce negative impacts on scoring. This could be handled in a similar manner to how ForBild mitigates its algorithmic weakness to rotated images, by rotating the image such that the darkest corner is always in the same position (Steinebach, 2011).

The analysis above is also largely content-independent. While prior work has acknowledged that there are types of image which cause issues with certain algorithms (McKeown et al., 2019; Struppek et al., 2022b), these particular weaknesses could be better documented at the archetype/algorithm level. This is particularly important for the impact it may have on varying corpora, such as when comparing animated images, or exported video frames using still image hashing formats.

[2] Curiously, the more nuanced downsampling approach taken by PDQ seems to hinder it in this case.

Table 5
Intra-score normalised Hamming distances between the original image and various modifications of the original for the Flickr 1 Million dataset. The ideal distance value is 0, which is reflected in the '% matches' column. Bolded items are very good performances, while underlined items are as bad, or worse, than unrelated images.

hash algorithm	modification	mean	median	stdev	range	% exact matches
blockhash	border	0.2783	0.2734	0.0851	0.0000–0.7188	0.0000
blockhash	compression	**0.0095**	**0.0078**	0.0142	0.0000–0.4492	36.4310
blockhash	crop	0.1668	0.1641	0.0610	0.0000–0.7344	0.0180
blockhash	mirror	0.4450	0.4531	0.1410	0.0000–1.0000	0.0460
blockhash	scale	**0.0013**	**0.0000**	0.0040	0.0000–0.1719	**85.4360**
blockhash	thumbs96	**0.0254**	**0.0234**	0.0179	0.0000–0.4961	5.5890
blockhash	watermark	**0.0504**	**0.0469**	0.0305	0.0000–0.3672	2.9450
neuralhash	border	0.0763	0.0729	0.0381	0.0000–0.5312	0.3670
neuralhash	compression	**0.0082**	0.0104	**0.0105**	0.0000–0.3333	49.1630
neuralhash	crop	**0.0605**	**0.0521**	0.0332	0.0000–0.4688	1.1540
neuralhash	mirror	0.2823	0.2812	0.1251	0.0000–0.6979	0.0960
neuralhash	scale	**0.0036**	**0.0000**	0.0069	0.0000–0.1458	**73.6020**
neuralhash	thumbs96	0.0809	0.0729	0.0445	0.0000–0.4792	0.9690
neuralhash	watermark	**0.0551**	**0.0417**	0.0424	0.0000–0.5729	3.6560
pdq	border	0.3949	0.3984	0.0599	0.0547–0.7422	0.0000
pdq	compression	**0.0094**	**0.0078**	0.0091	0.0000–0.4453	23.4350
pdq	crop	0.3255	0.3281	0.0564	0.0234–0.6719	0.0000
pdq	mirror	0.4975	0.5000	0.0226	0.0000–0.9844	0.0020
pdq	scale	**0.0237**	**0.0234**	0.0132	0.0000–0.4609	1.6320
pdq	thumbs96	0.0721	0.0703	0.0281	0.0000–0.4844	0.0060
pdq	watermark	0.1029	0.0938	0.0489	0.0000–0.5469	0.0020
phash	border	0.2656	0.2500	0.0745	0.0000–0.6562	0.0000
phash	compression	**0.0053**	**0.0000**	0.0138	0.0000–0.4688	**83.9040**
phash	crop	0.1686	0.1562	0.0586	0.0000–0.6562	0.0430
phash	mirror	0.4904	0.5000	0.0339	0.1562–0.6875	0.0000
phash	scale	**0.0020**	**0.0000**	0.0091	0.0000–0.4688	**94.0050**
phash	thumbs96	**0.0245**	**0.0312**	0.0250	0.0000–0.5938	39.0830
phash	watermark	0.1227	0.1250	0.0824	0.0000–0.6875	4.5350
wavehash	border	0.2744	0.2500	0.1333	0.0000–0.8750	0.2570
wavehash	compression	**0.0029**	**0.0000**	0.0106	0.0000–0.5312	**91.4070**
wavehash	crop	0.1049	0.0938	0.0649	0.0000–0.8750	3.5740
wavehash	mirror	0.3474	0.3125	0.1698	0.0000–1.0000	1.2670
wavehash	scale	**0.0007**	**0.0000**	0.0052	0.0000–0.8438	**97.9830**
wavehash	thumbs96	**0.0158**	**0.0000**	0.0214	0.0000–0.5938	57.3560
wavehash	watermark	**0.0451**	**0.0312**	0.0483	0.0000–0.4688	39.6030

References

Apple, 2022. Csam detection technical summary [Online. https://www.apple.com/child-safety/pdf/CSAM_Detection_Technical_Summary.pdf. (Accessed 2 February 2022).

Bourouis, S., Alroobaea, R., Alharbi, A.M., Andejany, M., Rubaiee, S., 2020. Recent advances in digital multimedia tampering detection for forensics analysis. Symmetry 12 (11), 1811.

Breitinger, F., Liu, H., Winter, C., Baier, H., Rybalchenko, A., Steinebach, M., 2013. Towards a process model for hash functions in digital forensics. In: *International Conference on Digital Forensics and Cyber Crime*. Springer, pp. 170–186.

Buchner, J.. Image hash [Online]. Available: https://github.com/JohannesBuchner/imagehash.

Commonsmachinery, 2018. Contribute to blockhash development by creating an account on Github. original-date: 2014-09-02T17:46:34Z. [Online]. Available: https://github.com/commonsmachinery/blockhash.

Dolhansky, B., Ferrer, C.C., 2020. Adversarial collision attacks on image hashing functions. arXiv:2011.09473 [cs]. [Online]. Available: http://arxiv.org/abs/2011.09473.

Drmic, A., Silic, M., Delac, G., Vladimir, K., Kurdija, A.S., 2017. Evaluating robustness of perceptual image hashing algorithms. In: 2017 *40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, pp. 995–1000. Opatija, Croatia.

[Online]. Available: Facebook. Pdq https://github.com/facebook/ThreatExchange/tree/main/pdq/python.

Farid, H., 2018. Reining in online abuses. Technol. Innovat. 19 (3), 593–599.

Farid, H., 2021. An overview of perceptual hashing. J. Online.Trust.Saf. 1 (1).

Fridrich, J., 1999. Robust bit extraction from images. In: Multimedia Computing and Systems, 1999. IEEE International Conference on, vol. 2, pp. 536–540 vol. 2.

Fridrich, J., Goljan, M., 2000. Robust hash functions for digital watermarking. In: Information Technology: Coding and Computing, 2000. Proceedings. International Conference on. IEEE, pp. 178–183.

Hadmi, A., Puech, W., Said, B.A.E., Ouahman, A.A., 2012. Perceptual image hashing. In: *Watermarking-Volume 2. INTECH*. Open Access Publisher.

Hamadouche, M., Zebbiche, K., Guerroumi, M., Tebbi, H., Zafoune, Y., 2021. A comparative study of perceptual hashing algorithms: application on fingerprint images. In: 2nd International Conference on Computer Science's Complex Systems and their Application, p. 12. Algeria.

Hao, Q., Luo, L., Jan, S.T., Wang, G., 2021. 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, pp. 69–85. Virtual Event Republic of Korea: ACM.

Jabade, V., 2016. Modelling of geometric attacks for digital image watermarking. Int. J. Innovat.Eng. Res.Technol. 3, 8.

Jain, S., Crețu, A.-M., de Montjoye, Y.-A., 2022. Adversarial Detection Avoidance Attacks: evaluating the robustness of perceptual hashing-based client-side scanning. In: 31st USENIX Security Symposium, vol. 22. USENIX Security, pp. 2317–2334.

Krawetz, N.. Photodna and limitations. Not known [Online]. Available: https://www.hackerfactor.com/blog/index.php?archives/931-PhotoDNA-and-Limitations.html.

Lee, H.-E., Ermakova, T., Ververis, V., Fabian, B., 2020. Detecting child sexual abuse material: a comprehensive survey. Forensic Sci. Int.: Digit. Invest. 34, 301022.

Manjunath, B.S., Ohm, J.-R., Vasudevan, V.V., Yamada, A., 2001. Color and texture descriptors. IEEE Trans. Circ. Syst. Video Technol. 11 (6), 703–715.

McKeown, S., Russell, G., Leimich, P., 2019. Fast forensic triage using centralised thumbnail caches on Windows operating systems. Journal of Digital Forensics, Security and Law 14 (3).

Microsoft Corporate Blog, 2019. The Christchurch Call and steps to tackle terrorist and violent extremist content [Online]. Available: https://blogs.microsoft.com/on-the-issues/2019/05/15/the-christchurch-call-and-steps-to-tackle-terrorist-and-violent-extremist-content/.

MIRFLICKR Download [Online]. Available: http://press.liacs.nl/mirflickr/mirdownload.html.

Prokos, J., Jois, T.M., Fendley, N., Schuster, R., Green, M., Tromer, E., Cao, Y., 2021. Squint hard enough: evaluating perceptual hashing with machine learning. Paper 2021/1531 Cryptology ePrint Archive [Online]. Available: https://eprint.iacr.org/2021/1531.

Rafiee, G., Dlay, S.S., Woo, W.L., 2010. A review of content-based image retrieval. In: Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on. IEEE, pp. 775–779.

S. L. AB (PythonWare. PIL: Python imaging library [Online]. Available: http://www.

pythonware.com/products/pil.
Saadatpanah, P., Shafahi, A., Goldstein, T., 2020. Adversarial attacks on copyright detection systems. In: *International Conference on Machine Learning*. PMLR, pp. 8307–8315.
Standaert, F.-X., Lefebvre, E., Rouvroy, G., Macq, B., Quisquater, J.-J., Legat, J.-D., 2005. Practical evaluation of a radial soft hash algorithm. In: International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II, vol. 2. IEEE, pp. 89–94.
Steinebach, M., 2011. Robust hashing for efficient forensic analysis of image sets. In: *International Conference on Digital Forensics and Cyber Crime*. Springer, pp. 180–187.
Struppek, L., Hintersdorf, D., Neider, D., Kersting, K., 2022a. Learning to break deep perceptual hashing: the use case NeuralHash. In: 2022 ACM Conference on Fairness, Accountability, and Transparency, pp. 58–69.
Struppek, L., Hintersdorf, D., Neider, D., Kersting, K., 2022b. Learning to Break Deep Perceptual Hashing: The Use Case NeuralHash. In: 2022 ACM Conference on Fairness, Accountability, and Transparency, pp. 58–69.
Swain, M.J., Ballard, D.H., 1991. Color indexing. Int. J. Comput. Vis. 7 (1), 11–32.
The Internet Watch Foundation, 2021. IWF: the annual report 2021 [Online]. Available: https://annualreport2021.iwf.org.uk/pdf/IWF-Annual-Report-2021.pdf.
Venkatesan, R., Koon, S.-M., Jakubowski, M.H., Moulin, P., 2000. Robust image hashing. In: Image Processing, 2000. Proceedings. 2000 International Conference on, vol. 3. IEEE, pp. 664–666.
Wallace, G.K., 1992. The JPEG still picture compression standard. Consum. Electron.IEEE Trans. 38 (1) xviii-xxxiv.
Yang, B., Gu, F., Niu, X., 2006. Block mean value based image perceptual hashing. In: Intelligent Information Hiding and Multimedia Signal Processing, 2006. *IIH-MSP'06. International Conference on*. IEEE, pp. 167–172.
Ygvar, A., 2022. AppleNeuralHash2ONNX original-date: 2021-08-15T19:52:47Z. [Online]. Available: https://github.com/AsuharietYgvar/AppleNeuralHash2ONNX.
Zauner, C., 2010. Implementation and Benchmarking of Perceptual Image Hash Functions.