making communications work
for everyone

Overview of Perceptual Hashing Technology

Publication date: 22 November 2022

Contents

Section

Disclaimer:
The paper is written from a technical perspective and is intended to outline our current understanding of the uses and limitations of perceptual hashing technology, and highlight areas where further research may be required to inform policy approaches as part of Ofcom's forthcoming online safety functions under the Online Safety Bill[1]. It does not make recommendations as to what those policy approaches should be and does not necessarily represent the concluded position of Ofcom on any matters discussed in the paper.

[1] The Online Safety Bill was introduced into Parliament in March 2022. A further version of the Bill, as amended at Committee stage in the House of Commons, was published on 28 June 2022.

1. Executive summary

Images and videos are central components of most online communications platforms. Academic research has shown that people often find it easier to recall knowledge gained from looking at images than from reading text, a phenomenon that is extensively utilised in everyday communication.

Hashing is an umbrella term for techniques to create fingerprints of files on a computer system. An algorithm known as a hash function is used to compute a fingerprint, known as a hash, from a file. Comparing such a hash with another hash stored in a database is called hash matching. In the context of online safety, hash matching can be a primary means for the detection of known illegal or otherwise harmful[2] images and videos.

Cryptographic hashing can be used to identify exact matches, while perceptual hashing can be used to determine whether images (or videos) are very similar (i.e. a match). Importantly, perceptual hashing assesses the similarity of the images, not of the content depicted in the images: for example, very similar images of different items could be determined to be a match, while very different images of the same items would not.

Hash matching requires hash databases of previously identified content. Several hash databases exist for the sharing of hashes related to child sexual abuse material (CSAM) to assist online platforms with the detection of such content – subject to contractual agreement with the database provider. There is also one such database for terrorism content and another such database for intimate image abuse material. Shared hash databases for the detection of other potentially harmful content, e.g. fraudulent advertisements or imagery promoting illegal drug use, do not currently exist but are theoretically feasible.

Limitations of perceptual hashing

Research has demonstrated that applications of perceptual hashing can be limited due to several inherent properties, such as non-negligible inaccuracy (i.e. false positives and false negatives) and the need to trust in the integrity of databases of images and videos that may be difficult to scrutinise.

Perceptual hashing can furthermore be subject to adversarial exploits and cyberattacks, including the precise perturbation of images and videos to prevent the detection of harmful content (false negatives) or to cause the misclassification of benign content (false positives).

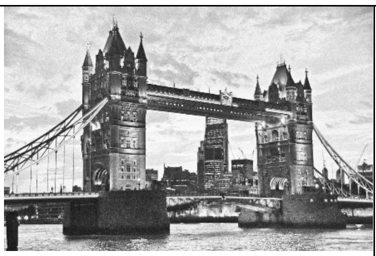
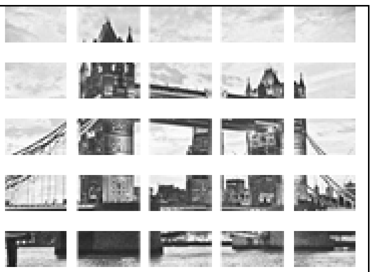
[2] As noted above, this document is a technical report intended to outline Ofcom's current understanding of perceptual hashing technology, rather than a policy document to inform Ofcom's approach to its forthcoming online safety duties. For this reason, references to illegal and harmful content are not necessarily intended to be aligned with the definitions of these terms in the current version of the Online Safety Bill (and we tend to refer to illegal and harmful content interchangeably throughout this document).

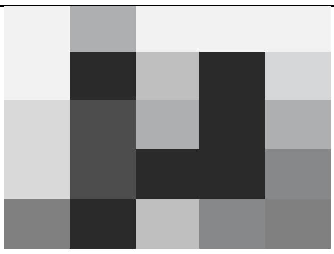
Examples of perceptual hash functions

Most perceptual hash functions operate technically by first converting images to greyscale and downscaling them to a lower resolution to speed up overall processing. Perceptual hash functions can be grouped into three categories based on their underlying design principles: (1) dividing images into squares, (2) transforming images into waves, and (3) using machine learning models.

The first group of hash functions divides images into squares and then calculates a number which represents the content of each square. Example methods are based on: (1) whether the square is brighter or darker than the average greyness of all squares, (2) whether the square is lighter or darker than its neighbouring squares, and (3) the approximate intensity of gradients (i.e. edges) in each block. The concatenation of the numbers representing each block creates the perceptual hash of the image. Example hash functions in this group are *aHash*, *dHash*, and Microsoft's *PhotoDNA*.

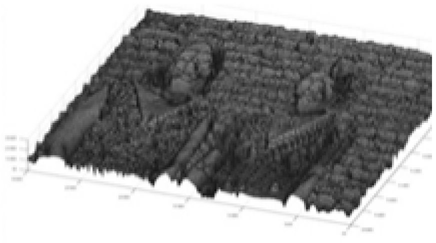
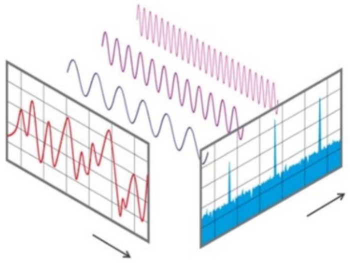
Example 1: A simplified perceptual hash function based on whether parts of the image are brighter or darker than the average.

	0 1 0 0 0
	0 1 0 1 0
	0 1 0 1 0
	0 1 1 1 1
	1 1 0 1 1
		The digits are concatenated, and the resulting hash (in this example) is
		01000 01010 01010 01111 11011

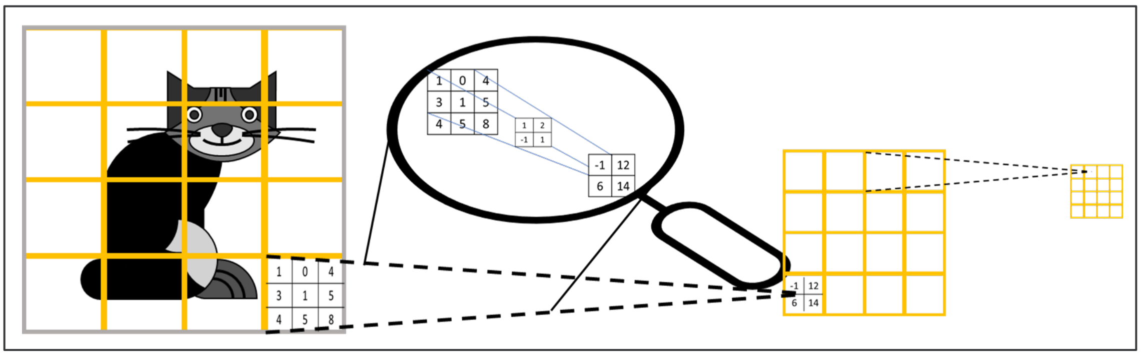
The second group of hash functions decomposes the image into a set of waves using a Fourier-related transformation, whereby the darkness and brightness of pixels in the image are represented by the peaks and valleys of these waves. The hash value can be calculated from the parameters of these waves. Example hash functions in this group are *pHash*, *wHash*, and Meta's *PDQ*.

Example 2: A simplified perceptual hash function based on a Fourier-related transformation.

 *Source: Fourier transforms of Images, by Rachel Thomas. Fourier transforms of images | plus.maths.org. © University of Cambridge*	 *Source: Phonical, CC BY-SA 4.0 via Wikimedia Commons*

The third group of hash functions uses a machine learning architecture called Convolutional Neural Networks (CNN) to repeatedly convolute (i.e. merge) neighbouring pixels. An example hash function in this group is Apple's *NeuralHash*.

Example 3: A simplified perceptual hash function based on a Convolutional Neural Network.



Source: Ofcom

2. Introduction

Preface

There is a famous saying that a picture is worth a thousand words. This has never been truer than in the modern age of electronic communications, where the use of visual media in everyday communication plays an ever more important role in people's lives. Pictures, videos, infographics, memes, GIFs, etc. are ubiquitous in online communications. They are used to augment, or can be augmented by, textual and auditory communication, or can stand on their own. Most online platforms have recognised that providing the best possible support for this style of communication can be a competitive advantage and have integrated related functionalities prominently in their interfaces. Other online platforms were built entirely from scratch to be centred around the use of visual media for communication.

People may prefer visual or visually enhanced communication since this supports superior knowledge retention and recall.[3] Others may use visual components in their communication due to an inherent ambiguity of the message or to comply with social norms and trends.[4]

Just as they are used very widely in benign communication, visual components are also likely to be implicated in most categories of online harms — although to varying degrees. Illegal harms such as dissemination of CSAM and intimate image abuse are image-based offences.[5] Other harms, such as those related to terrorism and fraud, manifest themselves in significant proportions both with and without visual components, including auditory and textual elements.

The superior retention of knowledge acquired from visual communication, the extensive use of visual media items in everyday communication, and the risk of exposure to online harms from visual media – as outlined above – are a potent combination. Harmful images and videos may, therefore, have particularly significant or long-lasting impacts. Content moderation of online communications will therefore likely need to deal with visual components such as images, video recordings and live streams, at scale.

There are two main challenges to content moderation: identification of new, unknown harmful content, and detection of known harmful content. Some platforms approach identification of new, previously unknown harmful material by using human moderators alone, although it is often augmented by automated means. Most online platforms automate at least some content

[3] Paivio, A., Rogers, T. B., & Smythe, P. C. (1968). Why are pictures easier to recall than words?. Psychonomic Science, 11(4), 137-138.
Madigan, S. (2014). Picture memory. Imagery, memory and cognition, 65-89.
[4] Highfield, T., & Leaver, T. (2016). Instagrammatics and digital methods: Studying visual social media, from selfies and GIFs to memes and emoji. Communication research and practice, 2(1).
[5] However, the availability of audio recordings and textual descriptions of child sexual abuse (CSA) is also likely to cause distress to victims of CSA and others exposed to them. Currently, there is very little research on the prevalence of audio recordings and textual descriptions of CSA online. There are also other harms relating to CSA that take place online which are not image-based (for instance, grooming).

moderation when it comes to detecting known harmful content, given the volume of content concerned would be challenging to deal with by human moderation alone.

In this document we explain how hashing technologies can be a primary means for the detection of known harmful visual media items. We also discuss some of the issues that may arise where hashing technology is used, including its reported limitations and the potential implications should hashing be deployed inappropriately and without considering the risks from adversarial exploitation.

What is hashing?

Hashing is an umbrella term for techniques to create a short identifier for files on a computer system. Such files can be images, videos, music, Word documents, executables, or any other file on a computer system. The identifiers are called *hashes,* or *hash* in singular. Several established, distinct algorithms to compute such hashes exist and will be discussed in this report.

An algorithm to compute a file's hash is called a *hash function*. The computer file is the *input,* and the hash is the *output* of the hash function. Any two hash functions will (most likely) output a different hash for the same input. The same hash function with the same input will always create the same output, i.e. hash functions are deterministic.

A catalogue of hashes stored in a file system is called a hash table or *hash database* (we use the latter term throughout this document). Hash databases can be useful for a few reasons: hashes are shorter than the files from which they were calculated, and a hash database can therefore be used as a space-efficient index of files held elsewhere in the system (comparable to the register of books in a library). It is faster to search the content of a hash database to establish the existence and location of a file on a computer system instead of searching in the actual data storage. A hash database can also be used as an index of files which possess certain properties. This makes it possible to compare other files against such an index to establish whether other files also possess those properties without the need to retain the original files.

Hashing to establish similarity between files (e.g. images) is an imperfect technology, i.e. there will always be cases in which it fails to recognise files to be similar when they are and cases in which it falsely recognises files as similar when they are not. Only in the case of establishing whether two files are identical can it be assumed that the error rate is negligible, by using a specific type of hash function called cryptographic hash functions.

What hashing cannot do

Hashing can be used to fingerprint digital files in their entirety but not individual components within them. Therefore, it is not suitable to recognise whether the same component (e.g. object or person in an image) may be included in two different files unless the entirety of both files is very similar (e.g. depicting the same building at very similar angles, with very similar lighting, same environment/background, etc.).

For example, two very different pictures of the same hamburger should not be classified as similar with hashing technology, but two very similar pictures of different hamburgers may be classified as similar.

3. Perceptual hashing

Perceptual vs. cryptographic hashing

Hashing functions belong to either of two categories: cryptographic or perceptual hashing. Cryptographic hash functions create “random-looking” identifiers. They utilise the so-called “avalanche effect”, meaning that the smallest change in input data will result in an immensely different cryptographic hash. For example, when changing one character in a piece of text or one pixel in an image we expect that the “new” cryptographic hash will have no similarity to the “previous” cryptographic hash.

If two cryptographic hashes differ even just a little, we can rely on the assumption that they were calculated from two different input files. But if two cryptographic hashes are the same, then we can largely rely on the assumption that they were calculated from the same input file[6].

Perceptual hash functions, on the other hand, aim to create very similar hashes for very similar input files. The similarity between any two hashes is defined through a *distance metric* (e.g. the Euclidean distance[7]), and different perceptual hash functions may require the use of different distance metrics. The goal is to approximate the level of similarity between input files perceived by humans through the distance between / similarity of the perceptual hashes.

Perceptual hashes have a non-negligible risk of false positives and false negatives, i.e. dissimilar input files may have identical or similar hashes while similar input files may have dissimilar hashes. We discuss this in more detail in the section Limitations and Considerations on page 13.

Perceptual hashing is also sometimes called fuzzy hashing, robust hashing or locality-sensitive hashing. Although these terms do describe very similar processes and are often used as synonyms on a conceptual level, they are technically distinct from each other.

Any mention of hashes, hash functions or hash databases in this document refers to perceptual hashes, perceptual hash functions and perceptual hash databases unless explicitly stated otherwise. This document does not reference fuzzy hashing, robust hashing or locality-sensitive hashing outside this section.

Mathematical definition of perceptual hashing

A perceptual hash function preserves perceptual similarity between media items as locality in the hash space, e.g. the Euclidean space, with robustness against slight alterations such as resizing an image, editing a file’s metadata or changing the file format. Using a perceptual hash function, a

[6] Some cryptographic hash functions are collision-prone or vulnerable to preimage attacks. Examples of secure cryptographic hash functions can be found in the NIST publications FIPS 180-4 and FIPS 202.

[7] The Euclidean distance between two points in Euclidean space is the length of a straight line between those points. It is calculated using the Pythagorean theorem, which states that the square of a hypotenuse equals the sum of the squares of its legs, i.e. given an n-dimensional Euclidean space: $\mathrm{c}^2 = \mathrm{leg}_1^2 + \mathrm{leg}_2^2 + \cdots + \mathrm{leg}_\mathrm{n}^2$. Each character in a hash is mapped to a dimension in the n-dimensional space to give the hash’s location in Euclidean space.

string distance metric and a similarity acceptance threshold, two media items can be determined as perceptually similar or not.

A *k*-bit perceptual hash function $PHF_k(\cdot)$ is a deterministic algorithm to generate a perceptual hash $h = PHF_k(s) \in \{0,1\}^k$ from string s. A string distance metric $d(\cdot,\cdot) = [0,1]$ with similarity acceptance threshold $\varepsilon \leq 1$ is a measure to quantify the similarity of two hashes with

$$d\left(h1, h2\right) = \begin{cases} 0 & : h1 \text{ } and \text{ } h2 \text{ } are \text{ } identical \\ Pr[0 < d \leq \varepsilon] \geq \delta_1 & : h1 \text{ } and \text{ } h2 \text{ } are \text{ } similar \\ Pr[d > \varepsilon] \geq \delta_2 & : h1 \text{ } and \text{ } h2 \text{ } are \text{ } dissimilar \end{cases},$$

whereby δ_1 and δ_2 are accepted probability thresholds for accurate similarity measures of h_1 and h_2 and $1 - \delta_1$ and $1 - \delta_2$ are the probabilities of a false negative and false positive similarity measure, respectively. If δ_1 and δ_2 are 100%, we call $PHF_k(\cdot)$ a *perfect* or *flawless* perceptual hash function (although no perceptual hash function with such a property is known today).

Let $\mathcal{D} \subseteq \{0,1\}^k$ be a database of hashes of known media items. A media item s is classified as *identical to* or *perceptually similar to* one of the known media items in $\mathcal{D}$ if $PHF_k(s) \in \mathcal{D}$ or $\exists h' \in \mathcal{D}$ such that $d = (PHF_k(s), h') \leq \varepsilon$.

It is important to note that what may be considered an appropriate similarity acceptance threshold may vary dependent on context – for example, there may be situations where the benefit of reducing the prevalence of false positives to a lower level is deemed to outweigh the risk of increasing the prevalence of false negatives, or vice versa. Perceptual hash functions can therefore be adjusted by platforms, to reflect the level of similarity that is considered acceptable given the particular use case.

4. Example calculation of an image hash

This section provides a *simplified* visual example of how one may calculate a hash from an image. The process depicted here is inspired by a hash function called aHash, which will be explained later in this document.

Figure 1. This is the original, unedited image used for this example.

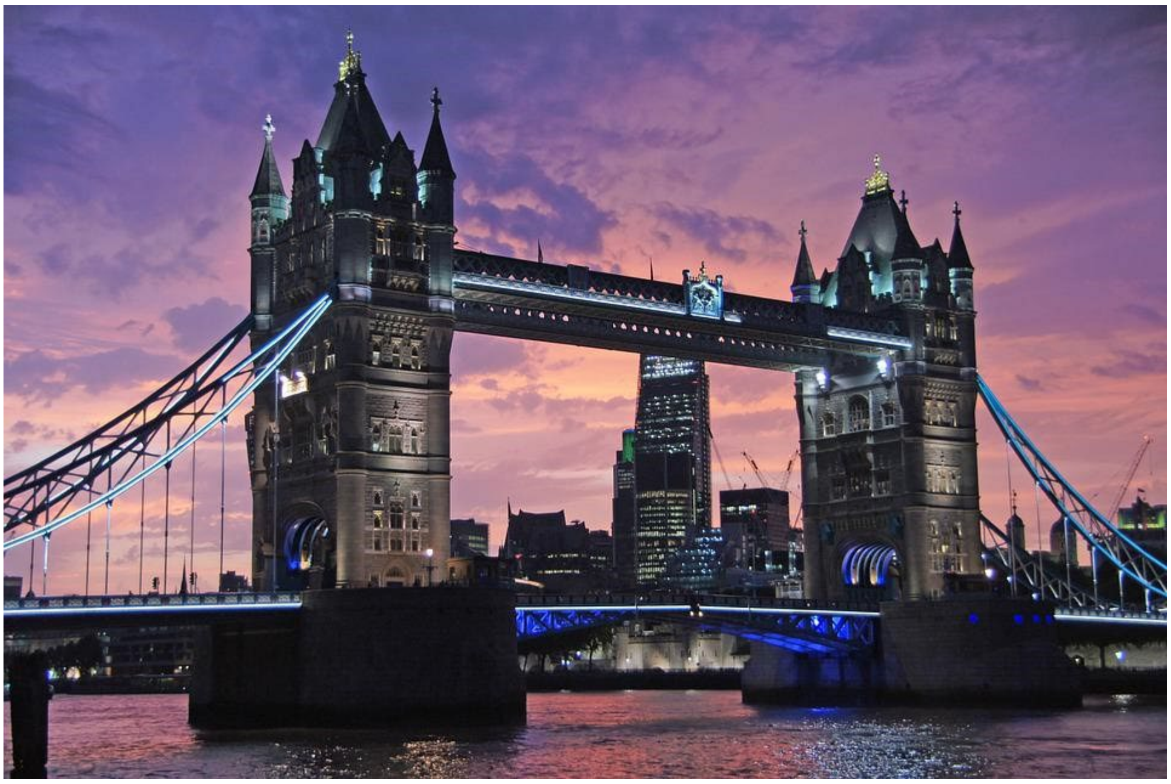


Figure 2. The image is transformed to grayscale and reduced in size.

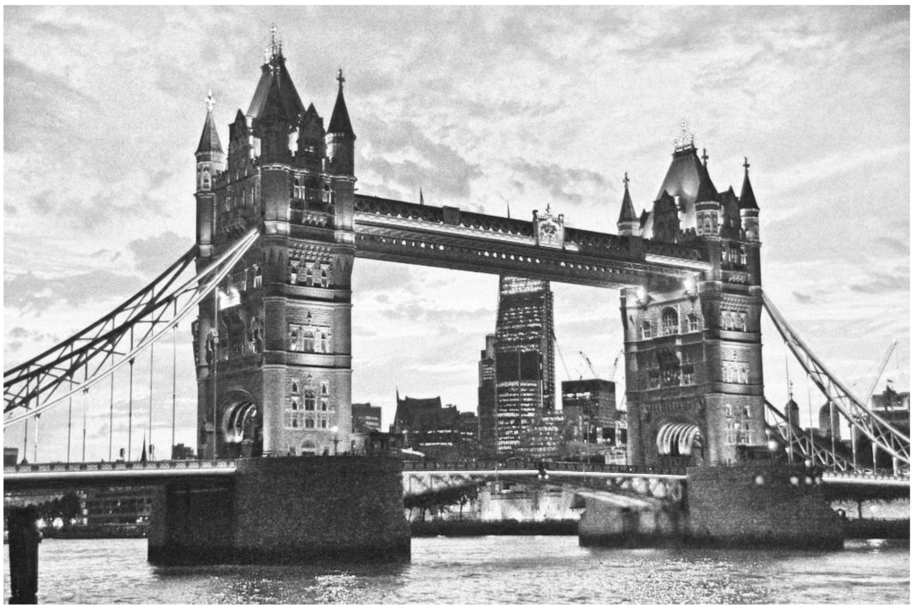


Figure 3. The image is divided into separate images in a 5x5 grid.

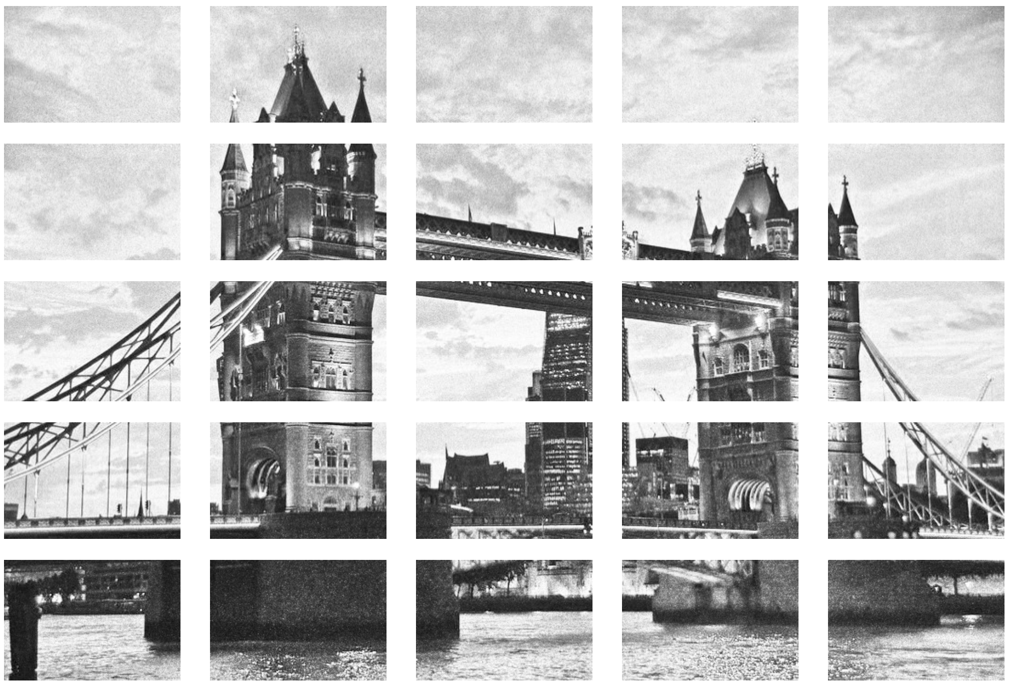


Figure 4. The colour values of all pixels in each sub-image of the 5x5 grid are averaged.

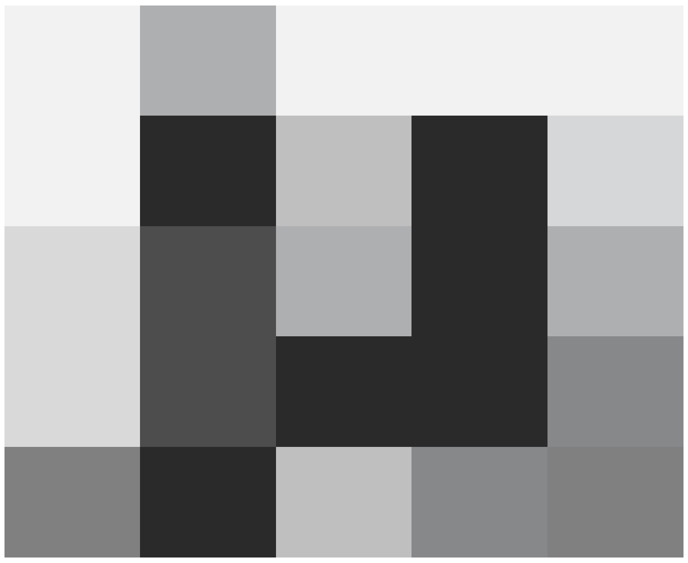


Figure 5. Each sub-image is labelled 1 or 0 based on whether its colour is brighter or darker than the average of all sub-images.

0	1	0	0	0
0	1	0	1	0
0	1	0	1	0
0	1	1	1	1
1	1	0	1	1

The digits are concatenated, and the resulting hash is 01000 01010 01010 01111 11011.

5. Commercial use cases

Image search

Hashing is used by search engines specialised on image search, such as Google Image Search and TinEye. The search engine has access to large databases of perceptual hashes of images found on websites which have previously been visited and indexed by search engine operators. The databases usually also contain metadata such as the date when and URL where the image was added to the database, available licensing information, etc.

Users may upload an image or enter the URL of an image as (part of) their search query. The server calculates the perceptual hash of the user's input and searches its database for images with the same or similar perceptual hash. These images are then served to the user as a result. Due to the imperfection of perceptual hash functions, not all images in the search results are usually perceptually similar to the input image (due to so-called hash collisions, discussed in more detail in section Limitations on page 14). It is noteworthy that most such search services use additional methods to find similar images beyond perceptual hashing, e.g. utilising metadata to identify famous landmarks or people and then including images of the same landmark or person which are not perceptually similar to the search input in the search results.

Data deduplication

Deduplication of data stored on computer systems, most frequently in databases, is a common and important process. It reduces the amount of space needed for storage and improves the time needed to retrieve items from the database. It can also contribute to ensuring data integrity, e.g. to ensure when editing or deleting a file that no identical or near-identical copies of the old (version of the) file remain in the database.

Deduplication can be done with data at rest (e.g. periodically checking databases for duplicates) and data in transit (e.g. checking for duplicates before adding new data to a database). The choice between cryptographic and perceptual hash functions depends on the type of data and application for the data, i.e. whether one cares about deduplication of identical data only or also near-identical data.

6. Online safety use cases

This section introduces three use cases of hashing technology in the context of online safety. None of these is discussed exhaustively, and in particular the report is not intended to take a position on whether (or in what circumstances) hashing *should* be used in any of these use cases. The purpose of this section is to illustrate that perceptual hashing is content agnostic and as such may be applied to various types of harmful online content which manifest with visual components.

The content presented in these use cases may be emotionally challenging. If you wish to skip this section, please move forward to section Limitations on page 14.

Terrorism

There is no single, globally recognised definition of terrorism or terrorist content, and the criminal law in this area will vary between jurisdictions. General definitions of "terrorism" will necessarily be broad and relatively imprecise and are likely to depend on the purpose behind a particular action (or threat of action), although an example of a legal definition can be found in section 1 of the Terrorism Act 2000[8]. Violent extremism is another concept often considered alongside terrorism.

Countermeasures against terrorist communication often focus on two domains: strategic communication, such as counter messaging; and content moderation, such as detection and removal of relevant content, including images or videos of terrorist activity[9]. The Global Internet Forum to Counter Terrorism (GIFCT) hash-sharing database supports the detection and removal of known terrorist content and is discussed further below.

Child sexual abuse material (CSAM)

CSAM refers to images or videos depicting the sexual abuse and exploitation of children.

Individuals who search for or possess CSAM may also be active in the distribution of such material. There is also an overlap between offenders who produce and/or distribute CSAM and those who engage in direct sexual offences against children.[10] The identification of CSAM shared online can therefore not only stop ongoing abuse but also prevent future abuse.

---

[8] *Terrorism Act 2000. This defines terrorism as the use or threat of action that involves serious violence against a person, involves serious damage to property, endangers another person's life, creates a serious risk to the health or safety of the public (or a section of the public), or is designed seriously to interfere with or seriously to disrupt an electronic system for the purpose of advancing a political, religious, racial or ideological cause. Except where the use or threat of action involves the use of firearms or explosives, to fall within the definition of terrorism the use or threat must also be designed to influence the government, an international governmental organisation or to intimidate the public (or a section of the public).[9] Ganesh, B., & Bright, J. (2020). Countering extremists on social media: Challenges for strategic communication and content moderation. Policy & Internet, 12(1), 6-19.*

[9] *Ganesh, B., & Bright, J. (2020). Countering extremists on social media: Challenges for strategic communication and content moderation. Policy & Internet, 12(1), 6-19.*

[10] *Cale, J., Holt, T., Leclerc, B., Singh, S., & Drew, J. (2021). Crime commission processes in child sexual abuse material production and distribution: A systematic review. Trends and issues in crime and criminal justice, (617), 1-22.*

CSAM is distributed via numerous communication channels, including peer-to-peer networks, the dark web, the indexed or surface web, using mobile devices and applications, and through social media, e-mail, instant messaging, newsgroups, bulletin boards, and chat rooms[11]. Large virtual communities of offenders have been observed to share millions of items of CSAM indirectly via hyperlinks.[1][12]

Prior research indicates that two of the most effective means for the detection, and subsequent deletion, of CSAM online are perceptual hashing and metadata analysis[13], such as file and folder names[14]. Several hash databases to support the detection of known CSAM exist, e.g. the National Center for Missing and Exploited Children (NCMEC) hash database, the Internet Watch Foundation (IWF) hash list and the International Child Sexual Exploitation (ICSE) hash database (discussed in section VII below).

Intimate image abuse

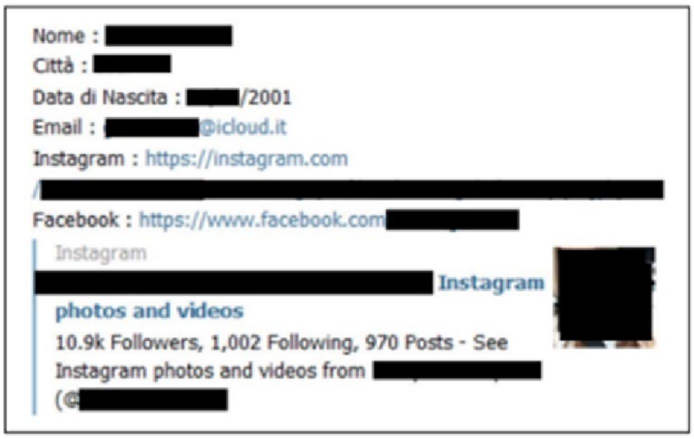
Intimate image abuse is generally considered to be the sharing of, or threatening to share, intimate images of another person without their consent. This may be a criminal offence in England and Wales (where, among other things, there is intent to cause distress) and in Scotland (where, among other things, there is intent or recklessness as to whether fear, alarm, or distress are caused).

Different social media and messaging platforms have reportedly been utilised in the non-consensual distribution of intimate images, intimate image abuse, and a related phenomenon called sextortion.[15] Some online platforms reportedly host communities dedicated to the non-consensual sharing of intimate images, with functionalities to collect, categorise, archive, search and retrieve intimate images and associated personal data via chat bots (see Figure 6)[16], while other online communities reportedly operate a for-profit model in which members pay money to get invited to more "exclusive" private groups.[17]

Hashing technology can be used to detect previously reported illegal images and videos, remove them from a platform or service at scale and prevent them from being reuploaded. The StopNCII bank is a hash database of material provided by victims of intimate image abuse for this purpose.

[11] *Lee, H. E., Ermakova, T., Ververis, V., & Fabian, B. (2020). Detecting child sexual abuse material: A comprehensive survey. Forensic Science International: Digital Investigation, 34, 301022.*
[12] *Westlake, B. G., & Bouchard, M. (2016). Liking and hyperlinking: Community detection in online child sexual exploitation networks. Social science research, 59, 23-36.*
[13] *Guerra, E., & Westlake, B. G. (2021). Detecting child sexual abuse images: traits of child sexual exploitation hosting and displaying websites. Child Abuse & Neglect, 122, 105336.*
[14] What is Metadata?. And why you need to start leveraging… | by Louise de Leyritz | Towards Data Science
[15] *Telegram: Where women's nudes are shared without consent - BBC News*
Why reports of 'sextortion' are soaring across the UK – and how to stay safe
Intimate Image Abuse: An Evolving Landscape | SWGfL
[16] *Semenzin, S., & Bainotti, L. (2020). The use of telegram for non-consensual dissemination of intimate images: gendered affordances and the construction of masculinities. Social Media+ Society, 6(4).*
[17] *Korea shocked by Telegram chat room sexual abuse scandal*

Figure 6. A chat bot advertising intimate images of a women alongside her city, date of birth, email address, Instagram and Facebook accounts in Italian.



Nome : [redacted]
Città : [redacted]
Data di Nascita : [redacted]/2001
Email : [redacted]@icloud.it
Instagram : https://instagram.com
/[redacted]
Facebook : https://www.facebook.com[redacted]

Instagram
[redacted] Instagram
photos and videos
10.9k Followers, 1,002 Following, 970 Posts - See Instagram photos and videos from [redacted]
(@[redacted]

[redacted]

Source: Semenzin, S., & Bainotti, L. (2020). The use of telegram for non-consensual dissemination of intimate images: gendered affordances and the construction of masculinities. Social Media+ Society, 6(4).

7. Limitations and other considerations

This section briefly sets out some of the issues to which the use of perceptual hashing technology can give rise. Available evidence, for example, shows that perceptual hash functions are imperfect and subject to some inherent limitations. Researchers have also demonstrated the potential for perceptual hash functions or their associated systems to be intentionally exploited and considered potential mitigation techniques which could be employed. While certain of the limitations we describe here – such as considerations relating to database integrity and security – are broadly generalisable to a range of technologies, others are specific to perceptual hashing.

It should be noted that while the intentional exploits outlined below have been demonstrated within a research environment, there is a lack of evidence on the extent to which any of these approaches would effectively scale to real-world implementations.

Imperfection of the hash function

Perceptual hash functions are generally considered to excel at the detection of non-adversarial edits that may naturally occur during day-to-day handling of media files on computers and during transit between computers, e.g. conversion from one file format to another, minor file size/quality reduction, addition or removal of small logos or watermarks, etc.

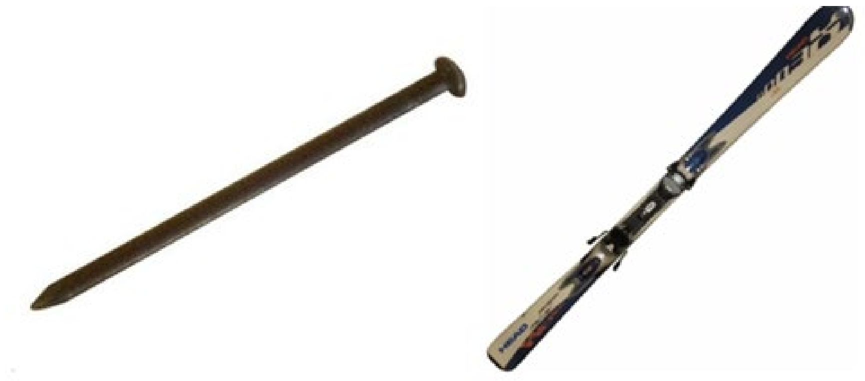
However, image editing software is capable of manipulating an image such that it retains visual similarity to the observer but may not be recognised as similar by currently known hash functions. Experts in this area acknowledge that development of more resistant hash functions remains a problem where more work is needed.

Naturally occurring hash collisions

One property of perceptual hash functions is that, unlike secure cryptographic hash functions[18], one may find two perceptually different input files having the same hash, also called a *hash collision*. This is possible because the hash functions' desired ability to calculate similar hashes from similar input may assume two inputs are similar based on features (e.g. pixels or pixel-regions of an image) that differ from those a human would consider. Figure 7 shows an example of such a naturally occurring hash collision.

[18] A secure cryptographic hash function is a hash function for which it is computationally infeasible to find a message that corresponds to a given hash value, or to find two different messages that produce the same hash value. The security strength can be quantified in terms of collision resistance, preimage resistance and second preimage resistance. Examples of secure cryptographic hash functions can be found in the NIST publications FIPS 180-4 and FIPS 202.

Figure 7. Example of two perceptually dissimilar images with the same hash. Neither image was manipulated, i.e. it is a case of a naturally occurring hash collision.



Source: ImageNet contains naturally occurring NeuralHash collisions

Trust in the hash database

Hashing enables the comparison of the hash of a media item with those stored in a hash database. Implementations of perceptual hashing are therefore very much reliant on the underlying hash database. Creating and maintaining a hash database, and ensuring its integrity (particularly where it is made available to others), can require a considerable amount of effort. The operator usually needs to maintain the highest level of trust and adhere to the highest standards of information security. A common approach is to operate a joint database which many entities may query for hash comparisons, but which only particularly trustworthy entities – for example, specialist non-governmental organisations or agencies -may edit, e.g. to add new hashes to the database.

Where entities rely on hash databases, they might make themselves vulnerable to criticism where such a database becomes (temporarily) unavailable, may be incomplete or may include incorrect entries.

Inscrutability of the hash database

Once a hash database has been created, it can be difficult or impossible to scrutinise it, particularly in cases where the hashes relate to media files which it would be illegal to possess. This means it may be difficult or impossible to determine why hashes have been added to it, i.e. it can be difficult or impossible to validate that no hashes have been added to the database in error (or for nefarious reasons, which is discussed in the sub-section on Intentional Exploitations of Perceptual Hash Functions on page 17).

Furthermore, it is unknown how one could migrate a hash database from one hash function to another in case one wanted to phase out the formerly used hash function (e.g. if a currently-used hash function would be deemed ineffective, inefficient or insecure). This means one would likely need to create a new hash database from scratch when migrating to a different hash function, which could mean losing the accumulated knowledge represented in that original database. Further, where the original files are not retained or available to moderators it may be difficult or impossible to verify whether the content identified is, in fact, a match. These limitations can be mitigated if one were to retain the original files alongside their hashes, but such an endeavour can be costly and, in some cases, potentially illegal.

Restricted availability of hash databases

Most hash databases are operated by private organisations and are not a public service. As such, it may be difficult for some companies to gain access to these resources – for instance, where there is a cost associated; specific qualifying conditions need to be met; or if those private organisations would not want to be associated with the company in question.

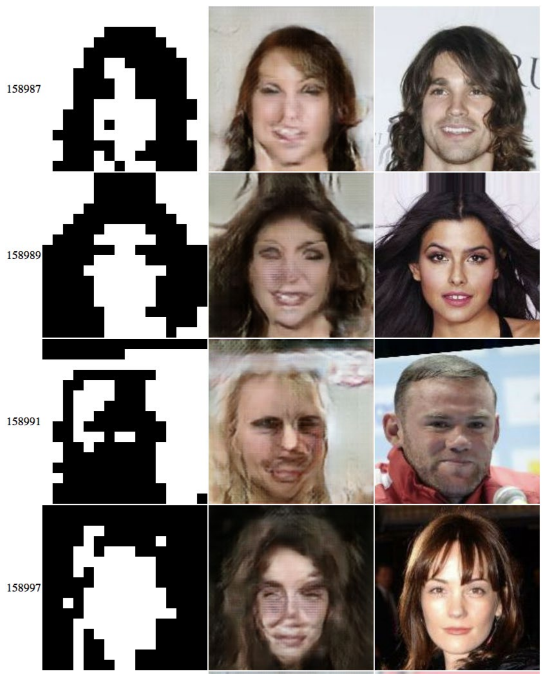
Invertibility of hashes

Researchers have demonstrated the potential to *invert* a given perceptual hash. This means that, while it is not possible to entirely recover the data from the original input file, it may be feasible to find a range of candidate files which could have been used as an input when that hash was originally calculated[19][20]. The expected effort required to invert hashes depends on the hash function and usually scales with the number of input candidates one may want to find.

These input candidates constitute a hash collision which may depict unrelated content or be visually similar to the original input file. See Figure 8 for examples of the latter. Thus, given a perceptual hash, there is a non-negligible risk of leakage of privacy and confidentiality of the input file - for example, this could mean that to some extent it may be possible for the contents of the original file to be discerned.

[19] Black-Box Attacks on Perceptual Image Hashes with GANs | by Nick Locascio | Towards Data Science
[20] Inverting PhotoDNA

Figure 8. Inversion of perceptual hashes. Images in the middle are inverted hashes calculated by a ML model, images on the right are the original images.



Source: *Black-Box Attacks on Perceptual Image Hashes with GANs | by Nick Locascio | Towards Data Science*

Hashes of personal data are personal data too

Where a hash of a media item would be classified as personal data and subject to data protection legal requirements, this has significant implications for both data storage and data sharing

Intentional exploitation of perceptual hash functions

Perceptual hashing-based systems benefit if their main components, including the (parameters of the) hash function and the hash database, are kept secret in order to reduce their attack surface.[21]

Hash databases are an integral component of most hashing systems, particularly those that compare media items against a database of previously categorised content. Therefore, as noted in previous sections, the integrity of the hash database is essential. Intentional integrity violations could occur in several ways:

- Content in the database could be deleted.
- Inappropriate content could be added to the database[22].
- Appropriate content could be carefully perturbed before being added to the database to create a hash collision with some other, targeted media item that does not possess the

[21] *The attack surface comprises those elements of a system where an attacker can try to enter, cause an effect on, or extract data from the system.*

[22] *OnlyFans accused of conspiring to blacklist rivals - BBC News*

characteristics necessary for inclusion in the database[23].

Furthermore, perceptual hash functions could be exploited with careful manipulation of content that ordinarily would match with hashes in a hash database but does not match after the manipulation, yet the manipulated content could be perceptually indistinguishable from the original[24].

Many of the attacks and exploits demonstrated by researchers require access to (a copy of) the hash function and/or knowledge of at least one hash in the database (or the corresponding image)[25]. As such, these exploits could successfully be mitigated by ensuring that details of the hash function and the system's internal parameters (e.g. the bit-length of hashes or how the similarity between two hashes is calculated) are effectively restricted.

However, parameter inference may still be possible where an adversary is able to observe the input and output of the hash function, and the risk may be increased where the would-be attacker can choose the input files.

Research suggests that, while it is often possible to detect such attacks with server-side deployments of hash functions and take appropriate countermeasures, for client-side deployments, counteraction may be considerably more difficult. For example, researchers have previously demonstrated that in the case of client-side deployments an attacker could use a method called *reverse engineering* to extract (parts of) the hash function or hash database from the source code of the application or operating system. [26]

It is also noteworthy to point out the conundrum that perceptual hashing is most robust against adversarial attacks if all components are kept secret (also called "security through obscurity"), yet public scrutiny of hash functions could lead to improvements that minimise the risk of naturally occurring false positives and false negatives while also likely having a positive impact on public acceptance of content moderation due to the increased transparency.

---

[23] *Dolhansky, B., & Ferrer, C. C. (2020). Adversarial collision attacks on image hashing functions. arXiv preprint arXiv:2011.09473.*
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. Cryptology ePrint Archive.
Struppek, L., Hintersdorf, D., Neider, D., & Kersting, K. (2022, June). Learning to break deep perceptual hashing: The use case neuralhash. In 2022 ACM Conference on Fairness, Accountability, and Transparency (pp. 58-69).

[24] *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. Cryptology ePrint Archive.*
Struppek, L., Hintersdorf, D., Neider, D., & Kersting, K. (2022, June). Learning to break deep perceptual hashing: The use case neuralhash. In 2022 ACM Conference on Fairness, Accountability, and Transparency (pp. 58-69).
Jain, S., Cretu, A. M., & de Montjoye, Y. A. (2022, August). Adversarial Detection Avoidance Attacks: Evaluating the robustness of perceptual hashing-based client-side scanning. In 31st USENIX Security Symposium (USENIX Security 22).

[25] *Dolhansky, B., & Ferrer, C. C. (2020). Adversarial collision attacks on image hashing functions. arXiv preprint arXiv:2011.09473.*
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. Cryptology ePrint Archive.
Struppek, L., Hintersdorf, D., Neider, D., & Kersting, K. (2022, June). Learning to break deep perceptual hashing: The use case neuralhash. In 2022 ACM Conference on Fairness, Accountability, and Transparency (pp. 58-69).

[26] *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. Cryptology ePrint Archive.*
Struppek, L., Hintersdorf, D., Neider, D., & Kersting, K. (2022, June). Learning to break deep perceptual hashing: The use case neuralhash. In 2022 ACM Conference on Fairness, Accountability, and Transparency (pp. 58-69).

8. Common hash functions and databases

The list of hash functions discussed in this section is provided as an overview and is not intended to be comprehensive.

Proprietary (and formerly proprietary) hash functions

PhotoDNA

PhotoDNA is a hash function developed by Microsoft and Dartmouth College for the detection of CSAM[27]. It can be licensed by eligible entities, including technology companies and law enforcement organisations. Developed in 2009, it is likely one of the oldest perceptual hash functions – at least among the proprietary technologies.

PhotoDNA creates hashes of images but an extension to videos is often labelled with the same umbrella term: Released in 2018, PhotoDNA for video[28] resamples a video into a number of individual frames (i.e. images) and calculates the PhotoDNA hash for each frame.

CSAI Match

CSAI Match is a tool originally developed in 2014 by engineers at YouTube, which utilises a perceptual hash function for the detection of CSAM videos. Google have since made CSAI Match available as a cloud service to technology companies and NGOs[29].

NeuralHash

NeuralHash is a perceptual hash function developed by Apple for the detection of CSAM[30]. The hashes are computed using an embedding network to produce image descriptors and then converting those descriptors to integers using a Hyperplane LSH (Locality Sensitivity Hashing) process.

PDQ

The PDQ hash function was developed by Meta for the detection of CSAM. Development began as early as 2015 and it was made available to the public under an open-source license in 2019[31]. A quality metric is calculated alongside each image hash to flag images that are relatively featureless (e.g. images with large sections in a monotonous background colour, such as Figure 9 on page 14) and therefore more likely subject to naturally occurring hash collisions.

[27] PhotoDNA | Microsoft
[28] How PhotoDNA for Video is being used to fight online child exploitation – On the Issues
[29] YouTube CSAI Match
[30] CSAM Detection - Technical Summary
[31] Open-Sourcing Photo- and Video-Matching Technology to Make the Internet Safer | Meta

TMK+PDQF

TMK+PDQF is the combination of the PDQ hash function and a technique called TMK, both developed by Meta, to create hashes of videos. TMK denotes a method to create perceptual hashes of videos using any hashing technique that creates a floating-point vector. PDQF denotes that PDQ is used without the final binarization of the floating-point vector. TMK was released by Meta under open-source licence in 2019.

Hash functions without proprietary background

The hash functions presented in this section describe generalised versions of the corresponding methods. Different implementations may feature small alterations. The technical aspects are provided as complementary reading.

Average hashing (aHash)

Average hashing, also sometimes call *mean hashing*, is a simplistic hash function based on the mean colour of different parts of an image. It excels at speed of computation but lacks resilience against many types of image perturbations.

Technical aspects

The first step is to downscale the image to 8x8 resolution (pixels) and convert it to grayscale. Then compute the mean values of the 64 pixels. The hash is a 64-bit string where each bit is one or zero based on the corresponding pixel in the downscaled 64-pixel image being above (i.e. brighter) the mean value or not (i.e. equal greyness or darker). Similarity between two hashes is determined based on their Hamming distance.

Difference hashing (dHash)

Difference hashing can be considered an extension of average hashing but instead of focusing on the average pixel colour, dHash focuses on the difference between neighbouring pixels, i.e. the gradient. It is more resilient to image perturbations than aHash but also slower to compute.

Technical aspects

First, the image is downscaled to a 9x8 resolution and converted to grayscale. Then, for each pair of horizontally neighbouring pixels, determine whether the left pixel is brighter than the right pixel. If yes, allocate value 1 to the corresponding bit in the hash and allocate 0 otherwise. With 8 pairs of neighbouring pixels per row of 9 pixels and 8 rows in total, the resulting hash is 64 bits long. Similarity between two hashes is determined based on their Hamming distance.

Perceptual hashing (pHash)

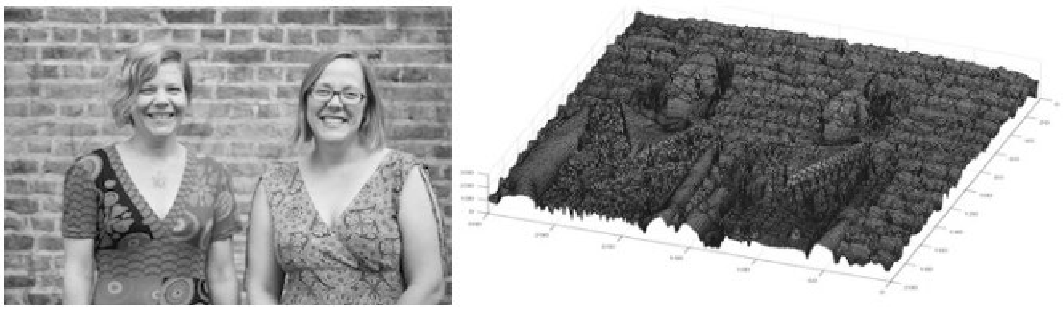
pHash is likely the most well-known perceptual hash function without a commercial background. Its key component is the Discrete Cosine Transform, which separates an image into a set of (weighted) cosine waves of increasing frequency. It is more resilient to image perturbations than dHash but also slower to compute.

Technical aspects

The image is first converted to greyscale and smoothed using a box filter with 7x7 kernel, i.e. the mean values are calculated from their neighbouring pixels in a 7x7 square. Some implementations of pHash skip the smoothing operations and may therefore not be compatible with those that do.

Next, the image is resized to 32x32 pixels before the Discrete Cosine Transform is calculated. Of the resulting 32x32 matrix, retain the lower frequencies of DCT weightings in an 8x8 square except the very lowest row and column. The resulting 64 DCT weightings are binarized based on whether they are above or below the average weighting. Similarity between two hashes is determined based on their Hamming distance.

Figure 11. Visualisation of a grayscale image being interpreted as a three-dimensional "landscape", whereby the intensity of greyness acts as the third dimension. This visualisation can help understand the application of the Discrete Cosine Transform to images.



Source: Fourier transforms of Images, by Rachel Thomas. [Fourier transforms of images | plus.maths.org](). © University of Cambridge.

Wavelet hashing (wHash)

wHash is a hash function like pHash in that both use transformations to decompose an image into frequencies. wHash uses Discrete Wavelet Transform instead of the Discrete Cosine Transform used by pHash. Due to the nature of wavelets, it can be expected that wHash will perform better on images with fewer but more intense image perturbations (e.g. a non-transparent logo overlaying a significant proportion of the image) while pHash will perform better on images with less intense perturbations covering a larger proportion on the image (e.g. image-wide noise due to image compression or smoothing).

Technical aspects

The implementation of wHash is almost identical to pHash. The differences are: It decomposes the image into scaled and shifted Haar wavelets (analogous to the cosine waves of different frequency when using pHash). The lowest Haar frequency is discarded since it only represents the image contrast. The remaining lowest 8x8 array is used to calculate the hash based on whether they are above or below the average value. Similarity between two hashes is determined based on their Hamming distance.

Hash databases

NCMEC hash database

The US's National Child Victim Identification Program maintains a hash database for the detection of CSAM[32]. The database was launched in 2003 and is maintained by the US' National Center for Missing and Exploited Children (NCMEC), a private non-profit corporation in the US. In 2021, NCMEC added 1.4 million hashes to the list, which contains over 5m hash values of known CSAM.

Thorn Safer database

In 2020, US based NGO Thorn, which develops technology to combat child sexual exploitation, launched a commercially available tool called Safer to assist platforms in identifying and reporting child sexual abuse material33. The Safer hash database reportedly contains over 32 million hashes of known CSAM

IWF hash list

The Internet Watch Foundation (IWF), a not-for-profit charity in the UK, maintains a hash database for CSAM detection[34]. At the end of 2021 the IWF's hash database, also known as the IWF hash list, contained the hashes of over a million known CSAM images and videos[35].

ICSE hash database

Interpol's International Child Sexual Exploitation (ICSE) image and video database is a tool available to specialist investigators in 68 countries[36]. The database contains more than 2.7 million images and videos of known CSAM and utilises the PhotoDNA hash function.

EOKM hash database

The Dutch Expertisebureau Online Kindermisbruik (EOKM) operates a hash database of known CSAM[37] material reported to its Meldpunt Kinderporno reporting system as well as material provided by the Dutch National Police (TBKK). Databases from Interpol and NCMEC are also being used. It was launched in 2019 and an API is available for hosting providers.

[32] CyberTipline Data
[33] The road to Safer: Equipping industry to end CSAM | Thorn
[34] Image Hash List | IWF
[35] Hash Metadata Analysis CSAM Imagery | IWF Annual Report 2021
[36] International Child Sexual Exploitation database
[37] Microsoft Word - Hash Check Service - Introduction to performing hash checks - v2.docx

Project Arachnid

Project Arachnid operates a hash database of CSAM material and other “harmful-abusive material” that does not meet criminal thresholds[38]. It is operated by the NGO the Canadian Centre for Child Protection.

GIFCT hash-sharing database

The Global Internet Forum to Counter Terrorism (GIFCT) is a consortium of technology companies that maintains a hash-sharing database[39]. It features at least 320 thousand hashes of known images and videos produced by entities on the United Nation’s designated terrorist groups list and material related to a small number of selected cases of violent extremism.

StopNCII bank

The StopNCII bank[40] is a hash database launched in December 2021 and operated by the UK Revenge Porn Helpline at the South West Grid for Learning, a not-for-profit charity in the UK. People above 18 can calculate the hashes of their intimate images and videos locally on their devices. The hashes are uploaded to the database but not the images or videos.

The hashes are shared with industry partners, who use these hashes to find corresponding images and videos on their platforms and escalate them for review by moderators. It was developed in close collaboration with Meta. Current industry partners are Facebook and Instagram. The StopNCII bank uses PDQ for hashing of images and MD5 for videos.

Perceptual hash functions for purposes other than online harms

The examples cited below are not intended to represent a comprehensive list but are provided to broaden understanding of the maturity of the technology.

While this document is focused on perceptual hashing technologies for visual content, such as images and videos, similar technologies may also be used for audio files.

Some of the most advanced perceptual hashing techniques for audio are intended for music information retrieval and music recognition. The goal of the former is to recognise music from a full recording and provide metadata related to the recognised song (e.g. names of the song, artists, and album, etc.) while the latter identifies music based on a snippet of the song (or the song’s melody).

The earliest noteworthy methods developed for music identification required the entire song as input and could only handle minor differences in the input file such as different encodings (e.g. wav, mpeg, mp3). More recent techniques can identify music from people humming the part of the

[38] Home – Project Arachnid
[39] Explainers | GIFCT
[40] Stop Non-Consensual Intimate Image Abuse - StopNCII.org

melody or from a short recording of the music even in a crowded environment. *Shazam!* is an example of the latter type.[41]

An example of a non-commercial hash function for audio signals is the Bark Audio Hash. It utilises the *bark scale frequency spectrum* to fingerprint audio signals based only on frequencies to which the human auditory system is most sensitive.

ContentID[42] is a system originally developed by Google to enable content owners to identify and take down or monetise copyright infringements of their content. The algorithm runs directly on YouTube's servers. After a pre-authorisation step, content owners may upload samples of their protected music or videos to the hash database.

Audible Magic[43] is an example of a tool developed by a vendor and licensed to both platforms and content owners to scan the internet for copyright infringements. Audible Magic is used by the platforms Twitch, Dailymotion, Vimeo, Veoh and others.

It is widely assumed that perceptual hashing technology lies at the core of both systems. Researchers[44] have demonstrated successful evasion attacks against both systems.

[41] *Wang, A. (2006). The Shazam music recognition service. Communications of the ACM, 49(8), 44-48.*
[42] How Content ID works - YouTube Help
[43] Audible Magic - Automatic Content Recognition, Content Identification (ID) and Rights Administration
[44] *Saadatpanah, P., Shafahi, A., & Goldstein, T. (2020, November). Adversarial attacks on copyright detection systems. In International Conference on Machine Learning (pp. 8307-8315). PMLR.*