

iHashDNA

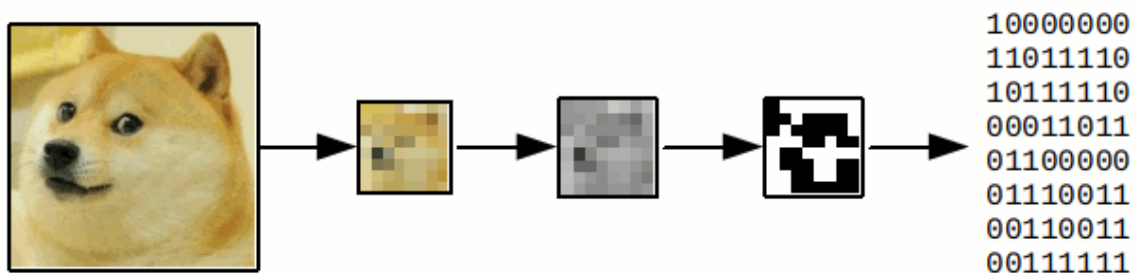
Perceptual hashing library in python (with redis), a wannabe [PhotoDNA](#)

What is Perceptual Hashing

Perceptual hashing is the use of an [algorithm](#) that produces a snippet or [fingerprint](#) of various forms of [multimedia](#).^{[1][2]} Perceptual [hash functions](#) are analogous if [features](#) of the multimedia are similar, whereas [cryptographic hashing](#) relies on the [avalanche effect](#) of a small change in input value creating a drastic change in output value. Perceptual hash functions are widely used in finding cases of online [copyright infringement](#) as well as in [digital forensics](#) because of the ability to have a correlation between hashes so similar data can be found (for instance with a differing [watermark](#)). Based on research at [Northumbria University](#),^[3] it can also be applied to simultaneously identify similar contents for [video copy detection](#) and detect malicious manipulations for video authentication. The system proposed performs better than current video hashing techniques in terms of both identification and authentication.

[Wikipedia, Perceptual Hashing](#)

TLDR: How Perceptual Hashing works



Pic Source: [Why we created 'Imageid' and saved 47% of the moderation effort | by Diego Essaya | Taringa! | Medium](#)

Perceptual hashing converts an image, by degrading it and turning it into "pixels", into a binary (or hexadecimal) sequence. **Unlike cryptographic hashing**, perceptual hashing **lacks of [avalanche effect](#)**, making any change in the image easily perceivable in the hash.

How it works

It uses [phash](#) and [whash](#) by checking initially phash, then whash.

By combining these two with a db (redis), you get this library.

You can:

1. **Ban images:** Add the hash of the image to the DB (and checks if already in it). This includes rotations (90 degrees left right 180 up down) of the pictures.
2. **Unban images:** Remove the hash and all the similar hashes from DB;
3. **Whitelist images:** Ignore a picture hash.

Practical examples

Perceptual hashing is a good way to recognize *two similar images*. If you need to:

- *Fast indexing similar images;*
- *Check for prohibited content without saving it into your DB (child pornography, pornography, porn, gore...);*
- *Check for watermarked original copyrighted content.*

and **more...**

The library can **easily detect an edited photo** if it has:

- *Color changes;*
- *Random garbage over it (watermarks, stickers....);*
- *slight cropping.*

Issues and limitations

Remember that this is not ML-Based.

It can be easily bypassed by cropping the image.

This library is a wannabe [PhotoDNA](#).

How to use it

Requirements

1. Install redis
2. Start redis
3. `git clone https://github.com/matteounitn/ihashdna.git`
4. (Optional) create a venv:
`python3 -m venv venv && source venv/bin/activate`
5. `pip3 install -r requirements.txt`

Then you are good to go!

Example

Checkout [this example](#).