A comparative study of perceptual hashing algorithms: Application on fingerprint images

Maamar HAMADOUCHE[1], Khalil ZEBBICHE[2], Mohamed GUERROUMI[1], Hanane TEBBI[1] and Youcef ZAFOUNE[1]

[1] *University of Science and Technology Houari-Boumediene, Algiers, Algeria*

[2] *Military Polytechnic School, Algiers, Algeria*

Abstract

Perceptual image hashing has been broadly used in the literature to authenticate images or to identify similar contents for image copy detection. It can be used to improve the security of fingerprint-based identification systems, especially to guarantee the authenticity of fingerprint images. In this paper, a comparative study of the most used techniques in the field of perceptual hashing is provided, aiming at evaluating their performance when applied to fingerprint images. The study includes five techniques, namely: A-Hash, D-Hash, P-Hash, W-Hash and SVD-Hash. The performance has been assessed in terms of perceptual robustness, discrimination capability and authentication characteristics, through extensive experiments. The obtained results are promising and show that overall both the A-Hash and P-Hash performed well when compared to other evaluated techniques.

Keywords

perceptual hashing, fingerprint images, robust hashing

1. Introduction

With the widespread utilization of fingerprint-based identification systems, establishing the authenticity of fingerprint data itself has emerged as an important research issue. Indeed, it is, in many cases, imperative that the authenticity of the transmitted fingerprint images, for example from intelligence agencies to a central database, must be first verified before it is processed by the identification modules. Recently, perceptual image hashing, which is one of the possible techniques that may be used along with digital watermarking, is becoming one of the most widespread research area and has emerged as an efficient way to check the authenticity of multimedia data (i.e. images and videos) [1].

Perceptual image hashing functions are based on extracting certain robust or invariant features from the image to produce a hash (or a fingerprint) with the property that two completely different images provide uncorrelated hashes while two visually similar images (i.e. perceived as similar by the human eye) generate highly correlated hashes. In this case, an efficient perceptual hashing technique should be able to detect that an image has been derived from another one in a way to remain perceptually similar, even their corresponding files are substantially different [2].

ICCSA'2021 : the 2nd International Conference on Computer Science's Complex Systems and their Applications, May 25–26, 2021, Oum El Bouaghi, Algeria

mhamadouche@usthb.dz (M. HAMADOUCHE); khalil.zebbiche@gmail.com (K. ZEBBICHE); mguerroumi@usthb.dz (M. GUERROUMI); htebbi@usthb.dz (H. TEBBI); yzafoune@usthb.dz (Y. ZAFOUNE)

It is meant here by two visually similar images that one image is derived from another via the commonly used content-preserving image manipulations.

In general, an authentication perceptual image hashing system consists of three main phases: the pre-processing stage, the hashing generation stage, and the decision making stage. The main objective of the pre-processing phase is to improve the robustness of features by reducing the effects of possible distortions, by applying image processing operations such as resizing, filtering, color space dimension reduction, etc. In the next phase, the reference hashes are generated and stored in a dataset. In the context of image authentication, the same perceptual hash process is applied to the image to be authenticated to generate a test hash. After that, and at the decision-making phase, the test hash is compared with the reference hashes to check the authenticity of the test image, based on a selected metric such as the Euclidean distance, the Hamming distance, the Normalized Hamming distance, etc. [3]. In the context of authentication applications, the major part of perceptual hashing algorithms can be broadly classified into the following classes : (i) invariant feature transform-based hashing, (ii) local feature points-based hashing, (iii) dimension reduction-based hashing, (iv) statistical features-based hashing, and (v) leaning-based hashing[3]. More details can be found in [4, 5, 6].

The objective of this paper is to analyze and evaluate the performance of five of the commonly used perceptual hashing techniques when applied to fingerprint images in the context of verifying their authenticity. The evaluated techniques are two spatial domain techniques, namely: the Average hashing (A-Hash) and the Difference hashing (D-Hash), and three frequency domain techniques, including: the DCT-based hashing (P-Hash), the Wavelet-based hashing (W-Hash) and the SVD-based hashing (SVD-Hash). Extensive experiments have been conducted on real fingerprint images, in which we have evaluated the perceptual robustness, the discrimination capability and authentication property of each evaluated technique. Carrying out this study is motivated by the fact that, and for non expert persons, they perceive fingerprint images as an alternation of curved dark lines, representing the ridges, along with white lines, which represent the valleys and they almost have the same shape. In other word, visually fingerprint images may, in many cases, look the same. Therefore, it is important to analyze how perceptual hashing techniques deal with such kind of images.

The rest of the paper is organized as follows. The evaluated techniques are described in Section 2. Evaluation results and analysis are provided in Section 3. Conclusions are drawn in Section 4.

2. Perceptual Image Hashing Techniques

In this section, five of the most used perceptual hashing techniques are described. It is worth mentioning that, in this study, we focus on perceptual image hashing algorithms that can produce fast image hashes, while still preserving image identity.

2.1. A-Hash

The average hashing technique, referred to as A-Hash, is one of the simplest and basic methods used to generate perceptual hashes of images [7, 8, 9]. This technique produces the hash value of the image based on its low frequencies, which represent the image structure, and eliminates the higher frequencies, corresponding to the image details[10]. For this purpose, the A-Hash uses a number of pre-processing operations including blurring, re-sizing, colors reduction and normalization [8]. It is worth noting that the primary goal of the A-Hash is to find the average color of all the image matrix values by calculating the mean of the matrix. In this study, we followed the same steps cited in [8] to implement this algorithm. These steps are summarized as follows : (i) the input image is resized to a size of 8×8 pixels; (ii) a color space conversion from RGB color space to gray-scale (YCbCr) color space is conducted; (iii) the mean value of all luminance values of the precedent image matrix is calculated; (iv) a comparison of each element of the image matrix and the calculated mean value is done, and a new binary matrix with 64 elements is obtained, where 1 indicates that the intensity of the element is greater than the average and 0 otherwise; (v) construct the vector from the resulting binary matrix, starting from the top left and going to the bottom right, to obtain a 64-bit long hash. The resulting hash can be later compared with other images hashes to retrieve the "similarity score" based on the distance metric between the two hashes.

2.2. D-Hash

The difference hash technique, also known as D-Hash, is an alternative method and similar to the A-Hash one [11]. Like the A-Hash, D-Hash focuses on the image structure, which is achieved by reducing the image size, i.e. by eliminating the higher frequencies from the image. The main difference between the two techniques is that, the D-Hash generates the hashes by computing the difference hash similarity of the image based on the change of color gradient between adjacent pixels in the image matrix[12, 13]. As for the A-Hash algorithm, we followed the same steps cited in [8] to implement the D-Hash algorithm. These steps are summarized as follows : (i) the image is reduced to a 9×8 block size to produce a total of 72 pixels; (ii) the image is converted to a gray-scale space color; (iii) for each row, we perform a comparison of the difference between each two adjacent pixels, to obtain a total of 8 differences per row; (iv) the 64 differences are computed for each image and then used to build the image hash, so that if the difference value is negative then the hash bit is set to 0, otherwise it is set to 0. At the end, a 64-bit hash is obtained.

2.3. P-Hash

The perceptive hash, denoted as P-Hash, is a technique that extends the A-Hash method by using the Discrete Cosine Transform (DCT) to obtain the most sensitive information of the human vision system (HVS) [9]. This technique uses the same approach like the A-Hash: finding the mean values and compare [8], but instead of using image intensities to perform the hash generation process, it uses a range of low frequencies obtained after applying the DCT technique [14]. The implementation of P-Hash includes the following steps:(i) the image is resized to a 32×32 pixels matrix; (ii) the obtained image is then converted to the gray-scale space color; (ii) a 32×32 DCT is performed on the gray-scale image to obtain a 32×32 DCT coefficient matrix, where the energy of the image will be gathered into a few low-frequency DCT coefficients; (iv) a vector of length 64 is constructed by concatenating the DCT coefficients from (1,1), corresponding the upper left corner of the 64 size matrix, to the coefficient (8,8), representing the lower right corner; (v) the mean of the resulting coefficients array is computed; (vi) a comparison of the 64 DCT coefficients with the mean value is performed, in a way that the hash bit is set to 1 if the coefficient is greater than the mean value, and 0 otherwise; (vii) finally, a 64 bits binary hash is obtained.

2.4. W-Hash

The wavelet hash technique, referred to as W-Hash, is a frequency domain hashing technique that uses the Discrete Wavelet Transform (DWT) to generate perceptual hashes. It is based on analyzing the image in the wavelet domain, while retaining temporal information [15]. Note that this transform is often used to remove redundancy in a data with highly correlated neighboring values, such as pixels in images [10]. The original W-Hash was introduced by Venkatesan *et al.* [16], who described the main steps to implement this technique as follows: (i) a randomly tilling of each sub-band of the image is calculated in a given number of level wavelet decomposition using the Haar wavelet [17]; (ii) the resulting statistics vector is then quantized using a randomized quantizer; (iii) the calculated quantized statistics vector is decoded using a first-order Reed-Muller error-correcting decoder, to produce a binary hash value with a length n; (iv) finally, the resulting intermediate hash value is passed by another decoding stage of a linear code with random parameters, to transform it into an even shorter hash code.

2.5. SVD-Hash

The Singular Value Decomposition hash, denoted by SVD-Hash, was first introduced by Kozat *et al.* [18]. The general mechanism of this technique is to derive a secondary image, from the original one using a pseudo-randomly (PR) extracting features that approximately capture semi-global geometric characteristics. Then, the final features are extracted and further suitably quantized to form the final hash value. The SVD-Hash algorithm implementation steps are summarized as follows: (i) from the input image matrix of size $n \times n$, form p possibly overlapping blocks, so that each of them has the size of $m \times m$; (ii) for each resulting block, generate the corresponding feature vector using the SVD transformation ; (iii) generate a secondary image via the PR combination of all intermediate feature vectors; (iv) apply the same steps 1 and 2 to the new resulting image; (v) finally, combine the generated feature vectors from the second image to build the final hash vector.

3. Experimental Results

In this section, intensive experiments have been carried out to evaluate the performance of the techniques described in Section 2. In all experiments, real fingerprint images from the 'FVC 2000, DB3_a' database [19] have been used. This database contains 800 fingerprint images of size 448×478. Note that these images have been slightly resized to have a size of 448×480 (i.e., having a height and width divisible by 8). Three aspects are considered in our experiments: (i) perceptual robustness, (ii) discrimination capability, and (iii) authentication. Moreover, and in order to make the comparison as fair as possible between the evaluated techniques, the Normalized Hamming distance has been used as a metric to evaluate the similarity between extracted hashes. To perform perceptual robustness and authentication testing, eleven kinds of commonly used content-preserving image manipulations were utilized to produce visually similar images. The details of image manipulations and the corresponding parameter settings are given in Table 1.

Table 1
Content-preserving manipulations and parameters setting

Manipulation	Parameters setting
Gaussian noise (GN)	variance = 0.07
Average filtering (AF)	filter size = 9
Salt & pepper noise (SP)	density = 0.20
Gaussian blurring (GB)	standard deviation = 7
Gamma correction (GC)	gamma = 0.8
Motion blurring (MB)	len = 40, theta = 55
JPEG compression (JC)	quality factor = 40
Median filtering (MF)	filter size = 7
Wiener filtering (WF)	filter size = 7
Image sharpening (SH)	alpha = 0.49
Image scaling (SC)	factor = 0.5

3.1. Perceptual Robustness

An efficient perceptual hashing technique should be robust and resist content-preserving manipulations with moderate strength. In other words, for visually similar images, it should produce the same or similar hashes even their digital representations are no longer the same. This characteristic can be measured by evaluating the perceptual robustness. In this work, we conducted a set of experiments to assess the perceptual robustness of the five evaluated techniques. Hence, for every evaluated technique, we extracted the original (reference) hashes from the original 800 fingerprint images and their corresponding manipulated versions under the eleven content-preserving manipulations listed in Table 1. Then, the Normalized Hamming distance is calculated between each original hash and its corresponding manipulated hash. The minimum, maximum, and mean values of the resulting distances after each operation are presented in Table 2. Note that, to make the comparison as fair as possible, we have normalized the obtained results by mapping them to the same interval.

The obtained results clearly show that, in most cases, the A-Hash technique provides the lowest mean values of the Normalized Hamming distances, computed between reference hashes and their manipulated ones. The P-Hash comes in the second place in most cases. Moreover, the D-Hash shows an acceptable level of robustness against the most manipulations. The least robust technique is the SVD-Hash, where the results clearly show that the applied manipulations affects the Hamming distances significantly. It is worth mentioning that in the absence of manipulations, the Normalized Hamming distances should be equal to 0.

Table 2
Hamming distances under different content-preserving manipulations. Best values are in bold.

	A-Hash			D-Hash			P-Hash			W-Hash			SVD-Hash		
	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
GN	0.000	0.214	**0.035**	0.000	0.375	0.135	0.000	0.273	0.110	0.043	0.385	0.138	0.037	0.567	0.134
AF	0.000	0.156	0.018	0.000	0.119	0.019	0.000	0.097	0.025	0.006	0.035	**0.015**	0.010	0.200	0.058
SP	0.000	0.117	**0.013**	0.000	0.217	0.041	0.000	0.136	0.033	0.022	0.165	0.055	0.021	0.338	0.066
GB	0.000	0.097	**0.008**	0.000	0.127	0.079	0.000	0.097	0.012	0.000	0.029	0.010	0.012	0.214	0.070
GC	0.000	0.097	**0.011**	0.000	0.098	0.016	0.000	0.117	0.020	0.024	0.159	0.071	0.003	0.163	0.032
MB	0.000	0.136	**0.020**	0.000	0.177	0.046	0.000	0.195	0.064	0.000	0.035	0.014	0.014	0.221	0.073
JC	0.000	0.078	**0.001**	0.000	0.039	0.003	0.000	0.039	0.005	0.000	0.010	0.002	0.001	0.012	0.003
MF	0.000	0.253	**0.036**	0.000	0.217	0.048	0.000	0.195	0.055	0.011	0.189	0.043	0.014	0.329	0.066
WF	0.000	0.078	**0.007**	0.000	0.138	0.015	0.000	0.078	0.009	0.000	0.043	0.015	0.009	0.106	0.030
SH	0.000	0.078	0.007	0.000	0.079	0.007	0.000	0.058	**0.003**	0.000	0.030	0.008	0.006	0.083	0.015
SC	0.000	0.078	**0.007**	0.000	0.059	0.010	0.000	0.058	0.008	0.000	0.011	0.001	0.026	0.944	0.235

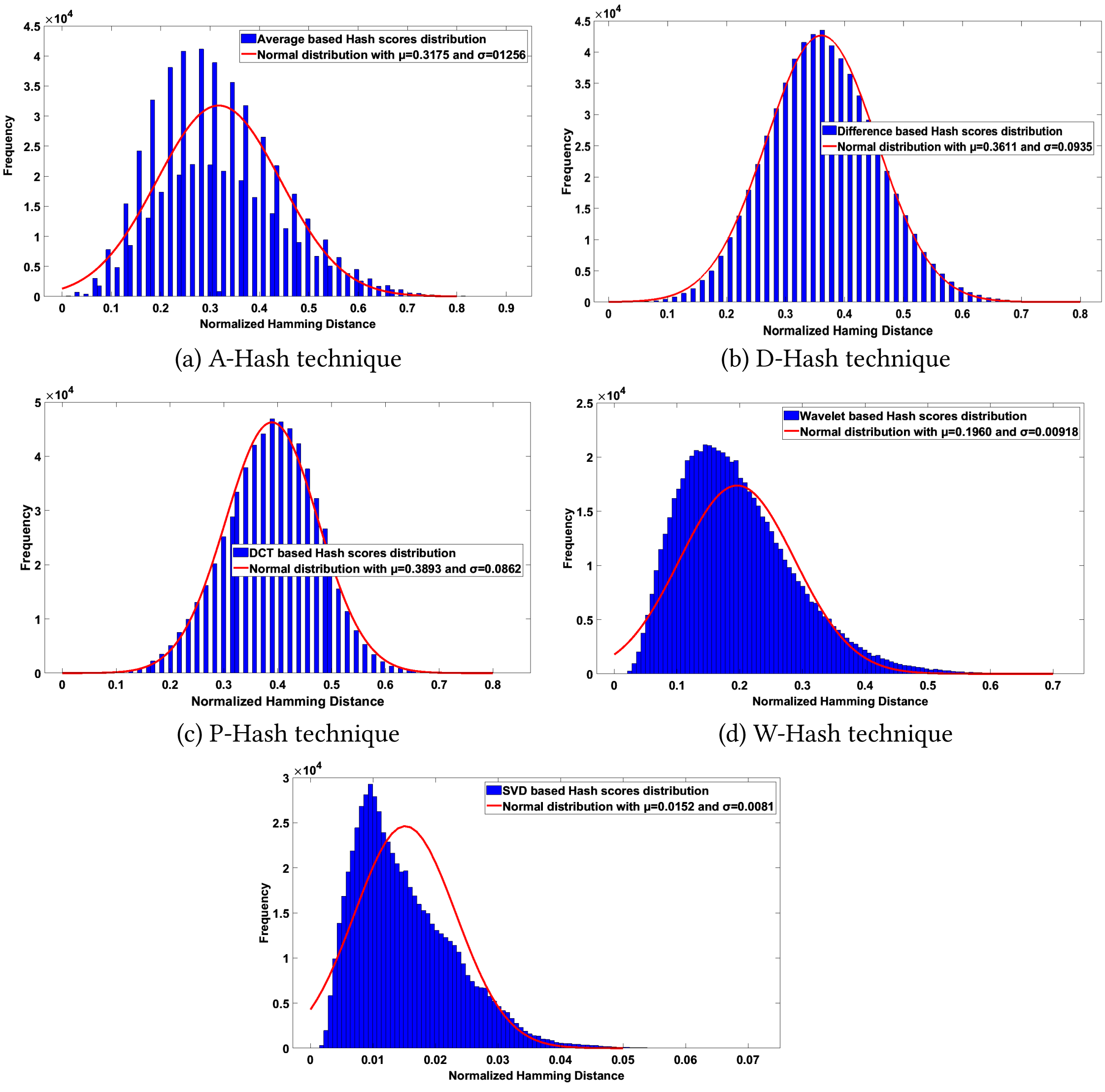
3.2. Discrimination capability

The discrimination capability of a hashing algorithm, also known as anti-collision capability [20], can be defined as its ability to generate significantly different hashes for visually distinct images. This means that an algorithm with high discrimination capability has a very low probability to generate similar hashes for two perceptually different images [3]. In general, the discrimination capability is evaluated by computing the collision probability of two hashes for two visually distinct images, which is in our case equals to the probability that the Normalized Hamming distance is smaller than the predetermined threshold. According to [21, 20, 22], the collision probability P_c of the hashes for two visually distinct images is given by

$$\begin{aligned} P_c(T) &= \frac{1}{\sqrt{2}\delta} \int_{-\infty}^{T} exp\left[-\frac{(x-\mu)^2}{2\delta^2}\right] dx \\ &= \frac{1}{2} erfc\left(-\frac{T-\mu}{\sqrt{2}\delta}\right) \end{aligned} \tag{1}$$

where $erfc(.)$ is the complementary error function, T is the predetermined threshold, μ is the mean value, and δ is the standard deviation.

In order to compute P_c, we first need to estimate the parameters μ and δ corresponding to the Normalized Hamming distance values computed from a large set of visually distinct images. To do so, and for each evaluated technique, we first extracted the reference hashes for all the 800 fingerprint images and then calculated the Normalized Hashing distance for each hash with the other 799 hashes. Consequently, we obtained $800 \times (800-1)/2 = 319600$ Normalized Hashing distances. Then, the values of μ and δ of the obtained Normalized hashing distances are computed by assuming that they all follow a normal distribution. The distribution of these 319 600 distances between hashing pairs, along with the obtained values of μ and δ, for the five evaluated techniques are shown in Fig. 1, where the abscissa is the Normalized Hamming distances and the ordinate represents their frequency. Note that this assumption is widely adopted in the literature [20, 21, 3].



(a) A-Hash technique

(b) D-Hash technique

(c) P-Hash technique

(d) W-Hash technique

(e) SVD-Hash technique

Figure 1: Distribution of Normalized Hamming distances corresponding to the five tested techniques.

As can be noticed, only the D-Hash and the P-Hash algorithms generate Normalized Hamming distances that follow a normal distribution. The other algorithms are not really approximated by this distribution, and this fact, may lead to a less accurate evaluation of the discrimination capability by the collision probability measure. After obtaining the parameters μ and δ, we applied Eq. 1 to compute the collision probability P_c for different values of thresholds T. Note that the values of T have been empirically calculated as described in [21]. The obtained results are given in Table 3. From these results, one can observe that, the smaller the threshold T is set, the smaller the collision probability is. Furthermore, the collision probability values are close to each other, with a slight superiority of the P-Hash technique.

Table 3
Collision Probability of the five evaluated algorithms with different thresholds T

Technique	Threshold T	Collision probability
A-Hash	0.02	0.89×10^{-2}
	0.04	01.36×10^{-2}
	0.06	02.02×10^{-2}
	0.08	02.93×10^{-2}
	0.10	04.17×10^{-2}
	0.12	05.79×10^{-2}
D-Hash	0.04	0.03×10^{-2}
	0.08	0.13×10^{-2}
	0.12	0.50×10^{-2}
	0.16	01.57×10^{-2}
	0.20	04.24×10^{-2}
	0.24	09.76×10^{-2}
P-Hash	0.04	0.00×10^{-2}
	0.08	0.02×10^{-2}
	0.12	0.09×10^{-2}
	0.16	0.39×10^{-2}
	0.20	01.40×10^{-2}
	0.24	04.16×10^{-2}
W-Hash	0.004	01.82×10^{-2}
	0.008	02.03×10^{-2}
	0.012	02.25×10^{-2}
	0.016	02.50×10^{-2}
	0.020	02.76×10^{-2}
	0.024	03.05×10^{-2}
SVD-Hash	0.0008	03.77×10^{-2}
	0.0010	03.98×10^{-2}
	0.0012	04.20×10^{-2}
	0.0014	04.42×10^{-2}
	0.0016	04.66×10^{-2}
	0.0018	04.90×10^{-2}

3.3. Authentication results

Since perceptual robustness and discrimination are contradictory with each other, we evaluate, in this section, the overall performance of the five evaluated techniques in terms of tampering detection accuracy. This evaluation has been carried out by computing three metrics, namely: the Recall, Precision and F1-measure. The precision expresses the proportion of the images that the hashing algorithm identifies as visually similar and they actually are. Whereas, The Recall is the proportion of similar images that the algorithm identifies as similar. Therefore, Precision measures the accuracy of the algorithm to detect visually similar images, while Recall measures

Table 4
Quantitative comparisons using the Recall, Precision and F1-measure metrics. Best results are in bold.

	A-Hash			**D-Hash**			**P-Hash**			**W-Hash**			**SVD-Hash**		
	Prec.	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
GN	0.456	0.722	0.559	0.818	0.101	0.180	0.886	0.494	**0.651**	NaN	0.000	NaN	NaN	0.000	NaN
AF	0.461	0.899	0.609	0.912	0.971	0.941	0.878	1.000	0.935	0.984	0.977	**0.981**	0.240	0.159	0.191
SP	0.474	0.947	0.632	0.895	0.827	0.860	0.916	0.994	**0.953**	0.750	0.007	0.015	1.000	0.145	0.253
GB	0.472	0.976	0.636	0.894	0.992	0.941	0.886	1.000	0.939	0.979	0.997	**0.988**	0.087	0.100	0.093
GC	0.477	0.967	0.639	0.911	0.985	**0.946**	0.877	0.999	0.934	0.428	0.004	0.007	0.752	0.589	0.660
MB	0.441	0.889	0.590	0.906	0.784	0.840	0.944	0.862	0.901	0.979	0.979	**0.979**	0.064	0.072	0.068
JC	0.441	0.996	0.640	0.909	1.000	0.952	0.884	1.000	0.938	0.979	1.000	**0.989**	0.930	1.000	0.964
MF	0.427	0.707	0.532	0.901	0.766	0.828	0.872	0.914	**0.892**	0.935	0.162	0.277	0.342	0.242	0.284
WF	0.467	0.986	0.634	0.902	0.991	0.944	0.881	1.000	0.937	0.979	0.920	**0.948**	0.724	0.554	0.627
SH	0.471	0.989	0.638	0.909	0.996	0.950	0.889	1.000	0.941	0.977	0.994	**0.985**	0.956	0.877	0.915
SC	0.482	0.987	0.648	0.911	1.000	0.953	0.882	1.000	0.937	0.978	1.000	**0.989**	0.052	0.002	0.005
Mean	0.461	0.915	0.614	0.897	0.856	0.849	0.890	0.933	**0.905**	0.815	0.640	0.651	0.468	0.340	0.369

the ability of the algorithm to find all the similar images among the dataset [23]. These two metrics are calculated by

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN} \tag{2}$$

where TP is the true positive value which represents the images that have been labeled as similar by the algorithm and they actually are. FP is the false positive value, corresponding to the images that have been labeled as similar by the algorithm, but they are actually not; and FN is the false negative value, representing the images that have been labeled as different by the algorithm, but they are actually similar.

The F1-score is the weighted harmonic average of the Precision and the Recall and it can be calculated as follows [23]

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

Note that higher values indicate better performance for these three metrics.

The same set of tests described in Section 3.1 has been performed to compute the three aforementioned metrics and the predetermined thresholds are chosen to provide the best results and are set as follows : A-Hash: $T = 0.04$, D-Hash : $T = 0.06$, P-Hash: $T = 0.08$, W-Hash: $T = 0.018$ and SVD-Hash: $T = 0.0018$. The obtained results are given in Table 4.

As it can be observed, and by evaluating the performance of each technique against a specific content-preserving operation, the results obtained from Table 4 can be described as follows:

- For the Gaussian noise manipulation, the P-Hash is the most powerful technique with a F1-score value of 0.651, whereas the W-Hash and SVD-Hash have not resisted to this attack and were unable to calculate the F1-score with Recall and Precision values of zero.

- For the Average filtering, the three techniques W-Hash, D-Hash and P-Hash performed very well and provide F1-score values of 0.981, 0.941 and 0.935 respectively. The worst technique is the SVD-Hash which generates a poor value of F1-score, equals to 0.191.
- For Salt & pepper noise addition, the P-Hash technique is in the first place with a F1-score value of 0.953, followed by the D-Hash technique with a F1-score value of 0.860, while the the SVD-Hash and W-Hash are the last-rank techniques with F1-score values of 0.253 and 0.015, respectively.
- For the Gaussian blurring, the most powerful techniques are W-Hash, D-Hash and P-Hash with F1-score values of 0.988, 0.941 and 0.939, respectively. The worst technique is SVD-Hash with a F1-score value of 0.093.
- For the Gamma correction manipulation, both D-Hash and P-Hash techniques performed well with F1-score values of 0.946 and 0.934, respectively. The W-Hash technique is the worst one with a F1-score value of 0.007.
- For the Motion blurring operation, the most successful techniques are the W-Hash, the P-Hash and the D-Hash with F1 score values of 0.979, 0.901 and 0.840, respectively. The worst performance is obtained by the SVD-Hash algorithm, which produces a F1-score value of 0.068.
- For the JPEG compression, all techniques performed well and provide good F1-score values, particularly, the W-Hash and the SVD-Hash, which have the highest F1-score values of 0.989 and 0.964, respectively. The A-Hash has generated the lowest F1-score value of 0.640.
- For the Median filtering, the best techniques are the P-Hash and the D-Hash with F1-score values of 0.892 and 0.828, respectively. The worst techniques are the SVD-Hash and the W-Hash with F1-score values of 284 and 0.277, respectively.
- For the Wiener filtering, almost all techniques performed well and the most successful ones are the W-Hash, the D-Hash and the P-Hash with F1-score values of 0.948, 0.944 and 0.937, respectively. The worst technique is the SVD-Hash which generates F1-score value of 0.627.
- For Image sharpening operation, except the A-Hash technique which yields a low F1-score value of 0.638, the other techniques were robust to this operation, and they all produce F1-score values over 0.951.
- For Image scaling operation, almost all techniques have resisted to this operation, except for the SVD-Hash algorithm which performed very poorly and provides a F1-score value of 0.005. The most robust algorithms are the W-Hash, the D-Hash and the P-Hash with F1-score values of 0.989, 0.953 and 0.937, respectively.

Overall, the most successful technique is P-hash, which achieves a F1-score value of 0.905 at a Precision value of 0.890 and a Recall value of 0.933. The second one is the D-hash, which reaches a F1-score value of 0.849 at a Precision value of 0.892 and a Recall value of 0.856. The next successful technique is the W-hash, which yields a F1-score value of 0.651 at a Precision value of 0.815 and a Recall value of 0.640. The forth place was assigned to the A-Hash algorithm, with a F1-score value of 0.614 at a Precision value of 0.461 and a Recall value of 0.915. Finally, the worst performance is obtained by the SVD-hash algorithm, which generates a F1-score value of 0.369 at a Precision value of 0.468 and a Recall value of 0.340.

4. Conclusion

In this work, we studied and analyzed the performance of five of the commonly used and fastest perceptual hashing techniques when considering fingerprint images, which visually have almost the same shape (i.e. alternation of ridges and valleys). This study involves the following techniques: the A-Hash, the D-Hash, the P-Hash, the W-Hash and the SVD-Hash. It has been performed through a set of extensive experiments applied to real fingerprint images and has focused on assessing three major aspects, namely: perceptual robustness, discrimination capability and authentication capacity. The obtained results are very interesting. Indeed, for the perceptual robustness property, the A-Hash technique clearly outperforms the other techniques and shows more robustness against the major part of the applied manipulations. For the discrimination capability, the evaluated techniques provide close performance in terms of the probability of collision, with a slight superiority for the P-Hash technique. In regard to the authentication property, which represents the overall performance, the P-Hash has provided the best results, when considering the whole set of applied manipulations.

Although these results are promising, they show that there is no best technique and the choice of a perceptual hashing technique will depend on the context in which it is used. As future works, this study can be extended to include more sophisticated hashing techniques and apply them to other fingerprint databases with different visual aspects, such as the background color, the shapes (flat or rolled), sensor types, etc.

References

[1] F. Khelifi, A. Bouridane, Perceptual video hashing for content identification and authentication, IEEE Trans. Circuits Syst. Video Technol. 29 (2019) 50 – 67. doi:10.1109/TCSVT.2017.2776159.

[2] R. Gennaro, D. Hadaller, T. Jafarikhah, Z. Liu, W. E. Skeith, A. Timashova, Publicly evaluatable perceptual hashing, in: M. Conti, J. Zhou, E. Casalicchio, A. Spognardi (Eds.), Applied Cryptography and Network Security., volume 12147 of *Lecture Notes in Computer Science*, Springer, Cham, 2020, pp. 436–455. doi:10.1007/978-3-030-57878-7_21.

[3] L. Du, Z. He, Y. Wang, X. Wang, A. T. S. Ho, An image hashing algorithm for authentication with multi-attack reference generation and adaptive thresholding, Algorithms 13 (2020) 227. doi:10.3390/a13090227.

[4] A. Hadmi, W. Puech, B. A. E. Said, A. A. Ouahman, Perceptual image hashing, in: M. D. Gupta (Ed.), Watermarking, 2nd. ed., InTechOpen, 2012, p. 17–42.

[5] K. Alice, N. Ramaraj, Combining hashing techniques in image authentication system:a survey, Int. J. Sci. Res. 4 (2015) 528–530.

[6] L. Du, A. T. S. Ho, R. Cong, Perceptual hashing for image authentication: A survey, Signal Process. Image Commun. 81 (2020). doi:10.1016/j.image.2019.115713.

[7] S. F. C. Haviana, D. Kurniadi, Average hashing for perceptual image similarity in mobile phone application, J. Telemat. Inform. 4 (2016). doi:10.12928/JTI.V4I1.

[8] V. Popkov, Possible application of perceptual image hashing, Master's thesis, Tallinn University Of Technology, 2015.

[9] M. Fei, Z. Ju, X. Zhen, J. Li, Real-time visual tracking based on improved perceptual hashing, Multimed. Tools Appl. 76 (2016). doi:10.1007/s11042-016-3723-5.

[10] A. Drmic, M. Silic, G. Delac, K. Vladimir, A. S. Kurdija, Evaluating robustness of perceptual image hashing algorithms, in: Proceedings of the 40th Int. Conv. on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, Opatija Croatia, 2017, pp. 995–1000. doi:10.23919/MIPRO.2017.7973569.

[11] R. Fitas, B. Rocha, V. Costa, A. Sousa, Design and comparison of image hashing methods: A case study on cork stopper unique identification, J. Imaging 7 (2021). doi:10.3390/jimaging7030048.

[12] D. z Wang, J. y. Liang, Research and design of theme image crawler based on difference hash algorithm, in: Proceedings of IOP Conf. Series: Materials Science and Engineering, volume 563, IOP Publishing, 2019, pp. 1–7. doi:10.1088/1757-899X/563/4/042080.

[13] N. Karunanayake, J. Rajasegaran, A. Gunathillake, S. Seneviratne, G. Jourjon, Design and comparison of image hashing methods: A case study on cork stopper unique identification, IEEE Trans. Mob. Comput. (2020). doi:10.1109/TMC.2020.3007260.

[14] F. Vega, J. Medina, D. Mendoza, V. Saquicela, M. Espinoza, A robust video identification framework using perceptual image hashing, in: Proceedings of the 2017 XLIII Latin American Computer Conference (CLEI), IEEE, Cordoba Argentina, 2017, pp. 1–10. doi:10.1109/CLEI.2017.8226396.

[15] W.-C. Huang, F. D. Troia, M. Stamp, Robust hashing for image-based malware classification, in: Proceedings of the 15th Int. Joint Conf. on e-Business and Telecommunications - BASS, volume 1, SciTePress, Porto Portugal, 2018, pp. 451–459. doi:10.5220/0006942204510459.

[16] R. Venkatesan, S. M. Koon, M. H. Jakubowski, P. Moulin, Robust hashing for image-based malware classification, in: Proceedings of the IEEE Int. Conf. on Image Processing (ICIP), volume 3, IEEE, Vancouver Canada, 2000, pp. 664–666. doi:10.1109/ICIP.2000.899541.

[17] E. Aboufadel, S. Schlicker, Wavelets, introduction, in: R. A. Meyers (Ed.), Encyclopedia of Physical Science and Technology, 3rd. ed., Academic Press, New York, NY, 2003, pp. 773–788.

[18] S. S. Kozat, R. Venkatesan, M. K. Mihcak, Robust perceptual image hashing via matrix invariants, in: Proceedings of the 2004 Int. Conf. on Image Processing (ICIP '04), IEEE, Singapore, 2004, p. 3443–3446. doi:10.1109/ICIP.2004.1421855.

[19] Fingerprint verification competition, 2000. URL: http://biometrics.cse.msu.edu/fvc00db/index.html.

[20] C. Qin, X. Chen, J. Dong, X. Zhang, Perceptual image hashing with selective sampling for salient structure features, Displays 45 (2016). doi:10.1016/j.displa.2016.09.003.

[21] L. Du, Z. Chen, A. T. S. Ho, Binary multi-view perceptual hashing for image authentication, Multimed .Tools Appl. (2020). doi:10.1007/s11042-020-08736-6.

[22] Q. Chuana, S. Meihui, C. Chin-Chen, Perceptual hashing for color images based on hybrid extraction of structural features, Signal Process. 124 (2018). doi:10.1016/j.sigpro.2017.07.019.

[23] M. Grandini, E. Bagli, G. Visani, Metrics for multi-class classification: An overview, white paper arXiv:2008.05756v1 [stat.ML] (2020). doi:10.1007/s11042-020-08736-6.