The Impact of Image Processing on Perceptual Hash Values

Matej Ferenčak*, Petra Grd† Igor Tomičić†
* Erste&Steiermärkische Bank, Zagreb, Croatia
† Faculty of Organization and Informatics, Varazdin, Croatia
mferencak@foi.unizg.hr; petra.grd@foi.unizg.hr: igor.tomicic@foi.unizg.hr

***Abstract*—In the last few decades image processing has become the focus of research in different fields. It can be used to extract information from images, improve their quality and make it easier for computers to understand them. Some of the challenges that pose a great problem for working with images are detecting modified images or detecting similar images. In order to address those challenges, image hashing can be used. Hashing is calculating a digest value from images and perceptual hash algorithms are a type of hash algorithms with the main idea that similar data has similar hash values, which means that the hash values remain approximately the same if the content is not significantly modified. This paper will compare different perceptual hash algorithms, apply them to differently modified images and analyse the impact of those modifications on perceptual hash values provided by different algorithms. The obtained values will be compared and advantages and disadvantages of different algorithms will be discussed.**

Keywords*—*image processing, perceptual hash, A-Hash, P-Hash, D-Hash, W-Hash

I. INTRODUCTION

Nowadays, more than ever, due to the widespread usage of image processing applications and availability of image editing software, establishing the legitimacy of image data has become an important challenge and has emerged as a major research topic. Recognizing modified photographs and locating similar images are two obstacles that, when combined, pose a substantial barrier to advancement when working with images. There are different techniques to overcome these obstacles, but these approaches fall mostly into two categories [1]: (1) image watermarking techniques and (2) image hashing techniques. In some applications, using an image watermark is not suitable due to capacity, security aspects and transparency [2], so image hashing has become a prevalent technique. Application areas of perceptual hashing vary across different fields, some of them are [3]–[5]: (1) biometrics, (2) security and (3) forensics. In biometrics, perceptual hashing can be used to find similar biometric traits, such as fingerprints of the same person. In security, perceptual hashing can be used for spam filtering, where hash values of an email digest can be used to determine whether an email is a spam, or for digital rights management where similar video, audio or image files can be found based on their perceptual hashes. Perceptual hash in forensics can be used for detection and removal of known terrorist content or child sexual abuse material shared online.

Hashing is the process of calculating a digest value from images, and perceptual hash algorithms are a type of hash algorithms. Whereas cryptographic hash functions operate on the byte-level and small changes in an image lead to significant changes in cryptographic hash, perceptual image hashing functions are based on the extraction of certain robust or invariant features from the image to produce a hash with the property that two completely different images produce uncorrelated hashes, whereas two visually similar images produce highly correlated hashes [3]. This suggests that the hash values remain relatively unchanged if the content is not substantially altered. This is especially important in the image processing domain where minor changes like compression or scaling should not impact the hash value as much.

There are different properties of perceptual hash algorithms and this study will analyse perceptual robustness to examine the similarities and differences between various perceptual hash algorithms, apply those algorithms to images that have been altered in a variety of ways, and analyze how those image modifications affect the perceptual hash values produced by different algorithms. The image modifications can be categorized into content changing and content preserving. Dittmann et al. [6] and Du et al. [7] state that content-preserving manipulations are: (1) transmission errors/noise/data storage errors, (2) compression, quantization/brightness reduction, (3) resolution reduction/scaling, (4) color conversions (from one color space to another, where the goal is to have the colors in different color spaces be as similar as possible), (5) y-distortion and (6) hue and saturation changes; and content modifying changes are: (1) removing image objects, (2) changing image elements positions, (3) adding new objects, (4) changes of image characteristics: color, textures, structure, impression, (5) changes of the image background and (6) changes of light conditions (shadow manipulations). In this study we will focus mostly on content-preserving changes.

The remainder of this study is organized as follows. Section II will describe a state of the art research effort within the field. Section III will describe the dataset used for the analysis and experiments conducted, after which, the obtained results will be discussed. In the end, the conclusion and future research directions will be given.

II. RELATED WORK

In recent years a number of different perceptual hash algorithms have been developed. This section will focus on the state of the art algorithms and comparison of robustness of those algorithms to image manipulations.

In general, perceptual hash methods include the following steps (Figure 1): preprocessing, feature extraction, feature quantification and hash production. Preprocessing removes superfluous information from an image, making it easier to extract features later. Feature extraction step finds the primary visual features using an extraction method and is the most challenging step of perceptual hashing algorithms. Feature compression blurs the extracted features to improve their robustness and hash creation makes quantitative qualities more abstract [8].

The authors of [9], one of the earliest studies on image hashing utilizing image features, suggested an image authentication system that is tolerant to image compression with loss. They used visually prominent image characteristics and Mexican-Hat wavelets to extract these features from photos. The strategy they proposed was able to determine when portions of an image were eliminated, but they also highlighted areas for improvement in order to apply their method to other image modifications.

The paper by Venkatesan et al. [10] presented a novel image-hashing technique that employs a wavelet representation of images and new random processing strategies for hashing. Error-correcting code-based constructions lower the length of the hash value while maintaining a low collision probability. With this approach, two images can be compared by comparing two bit strings for exact equality, rather than attempting to compare changeable image data, which is a significantly more complex challenge. In their experiments, image hashes proved resistant to a variety of attacks, such as ordinary image processing and malicious distortions.

In [11] a novel and robust hashing paradigm that uses iterative geometric techniques and relies on observations that main geometric features within an image would approximately stay invariant under small perturbations was proposed. The authors use Discrete Wavelet Transformation (DWT) to capture significant image features via time and frequency localization and select relevant regions using a threshold. Experiments revealed robustness to Stirmark attacks, with the exception of extreme rotation and cropping, and that the distances between the hash values of perceptually comparable images are distinct from the distances between images.

Monga and Evans [12] propose an iterative feature detector that uses the first derivative of Gaussian and Morlet wavelets to extract relevant geometry-preserving feature points. To increase perceptual resilience, the authors apply probabilistic quantization to the obtained features. The suggested hash algorithm is resistant to standard benchmark attacks, such as compression, geometric distortions of scaling and small-angle rotation, as well as common signal processing operations. Moreover, content-changing alterations of image data are accurately recognized.

The authors of [13] offer a new pseudo-random (PR) signal representation technique for images, in which they regard images as a succession of linear operator representatives (i.e. matrices). The authors employ Singular Value Decomposition (SVD) to extract images' semi-global, robust PR features. In contrast to DCT/DWT-type fixed basis transforms, SVD selects the optimal basis vectors, and their approach is robust against extreme geometric changes.

Swaminathan et al. [14] created a novel approach for producing an image hash using Fourier-Mellin transform features and controlled randomization. The suggested algorithm is resistant to modest levels of filtering, compression, and basic geometric operations up to 10°rotation and 20% cropping. The proposed hashing system is also capable of identifying malicious operations, such as copy-and-paste editing.

Lv and Wang [15] offer a novel shape-contexts-based approach to image hashing utilizing robust local feature points. Using the robust SIFT-Harris detector, they chose the most stable SIFT keypoints under various content-preserving distortions and generated compact and resilient image hashes by embedding the detected local features into shape-contexts-based descriptors. The suggested technique delivers improved identification performance against geometric attacks and comparable performance against classical distortions.

The previously described algorithms all have one notable problem and that is that they are not robust to rotation changes. In [16], an image hashing algorithm with a ring partition and a Nonnegative Matrix Factorization (NMF) that is both rotation-robust and discriminative was designed. The most significant contribution is a novel construction of rotation-invariant secondary image, which is employed for the first time in image hashing and contributes to making image hash rotation-resistant. In addition, content-preserving modifications modify NMF coefficients almost linearly in order to evaluate hash similarity with correlation coefficient. Experiments have demonstrated that the suggested hashing technique outperforms all other hashing algorithms in terms of robustness and discrimination. The continuation of this research can be found in [17], where authors add ring partition and invariant vector distance to image hashing algorithms to improve rotation robustness and discriminative capabilities. As ring partition is rotation-invariant, this approach to hashing is resistant to image rotation at any angle.

Ding et al. [8] suggested a new perceptual hash technique for multispectral (MS) remote sensing image authentication. In order to compactly represent the perceptual characteristics of the multispectral image, the authors implemented an affinity propagation method to classify the MS images into several clusters based on the information of the bands of these images. By dividing each band into a grid, the features of the grid cell at the same place within

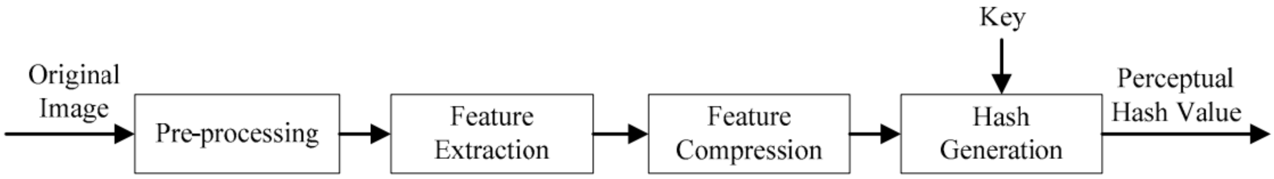

Fig. 1: Steps of the perceptual hash algorithm [8]

the cluster are retrieved and fused using DWT, while PCA-based data compression on the fused feature reduces the effect of noise.

In [18] a scheme is described for efficiently producing image hashing by combining local texture and color angle features. Weber's law is utilized to extract features, and the Weber Local Binary Pattern (WLBP) value of each low-frequency wavelet coefficient is generated prior to a histogram count and PCA dimensionality reduction operation. For color feature extraction, the color angular pattern is applied to each pixel in order to determine the correlations between the various color channels. The color angle values are then dimensionally reduced using a DCT and significant coefficients are chosen prior to cascading the two features to produce an integrated hash.

Qi and Zhao [19] presented a Color Opponent Component (COC) and Quadtree-based perceptual image hashing algorithm. Combining color features with structural features improves the image classification performance of the suggested technique. The input image is subjected to image normalization and Gaussian low-pass filtering to generate a secondary image from which COC is extracted. The color change information is derived from the COC as a color feature, and Quadtree decomposition (QD) is applied to the intensity image of the secondary image to get Quadtree structure features.

Singh et al. [20] created an efficient image hashing technique for authenticating image content based on local and global characteristics. Local information is derived using KAZE features that employ a non-linear diffusion filter, while global features are estimated using a reference image formulation.

Taking into account all of the mentioned papers, there is only a small number of papers dealing with the impact of image processing and manipulation on perceptual hash values and those that analyse the impact, do so on a small set of image data or on images of a particular subjects. This study will analyse a large set of images of different categories with different image manipulations and different perceptual hash algorithms to reach the general conclusion.

III. EXPERIMENTS

In this section different experiments have been conducted to analyse the impact of image processing on most often used perceptual hash algorithms in practice, as identified in literature [3]. Perceptual hash values of original images obtained with Average Hash (A-Hash), Perceptive Hash (P-Hash), Difference Hash (D-Hash) and Wavelet Hash (W-Hash) algorithms are compared with perceptual hash values of images processed with image scaling, rotation, images converted to grayscale, brightness modification, image sharpening, gaussian blurring, median filter and noise adding.

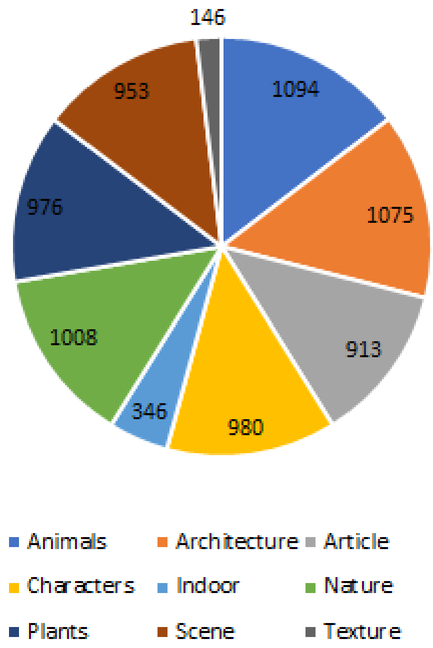

Fig. 2: Distribution of images per class in CASIA 2.0 authentic dataset

A. Dataset

In order to conduct the experiments and achieve the objective of this study, authentic images from CASIA 2.0 image tampering detection evaluation dataset [21] are used. The authentic part of the dataset consists of 7 491 images of animals, architecture, articles, characters, indoor images, nature, plants, scene and texture. The dataset was selected because of the many different categories of images which was important for reaching general conclusions. All images have the dimensions of 256x384 px or 384x256 px and the number of images in each category can be seen in Figure 2. The dataset is slightly imbalanced, with texture and indoor class having less images (146 and 346 respectively) than the other classes.

All modifications analysed in this study were applied to all 7 491 images, which resulted with the final dataset of 59 928 images.

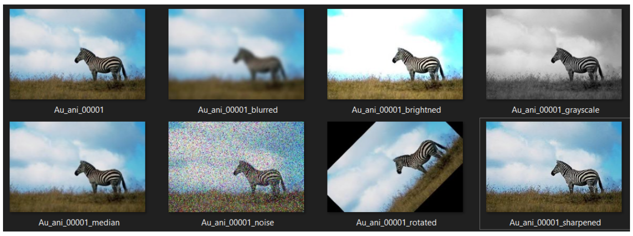

Fig. 3: Example of image modifications

B. Testing and Results

The goal of this study, as mentioned earlier, is to analyse how different image modifications impact the perceptual hash values of different algorithms. To this end, we applied eight modifications to each image from the CASIA 2.0 authentic dataset. Table I shows the description and parameters for each image modification and an example of an image and its modification can be seen in Figure 3.

TABLE I: Image processing parameters

Modification	Parameter
Image Scaling	Factor=0.75
Image Rotation	Angle=45°
Grayscale Conversion	Intensity range=[0,255]
Brightness Modification	Factor=1.5
Image Sharpening	Factor=4
Gaussian Blurring	Radius=3
Median Filter	Filter size=3
Salt & Pepper Noise	Amount=0.3

In order to make necessary image modifications, we created a Python script using Python Imaging Library (PIL) which resulted with a new dataset consisting of 59 928 images with different modifications and original images.

Next step was to calculate perceptual hash values for the original image and each modified image. The values were calculated with A-Hash, P-Hash, D-Hash and W-Hash algorithms of the ImageHash Python library. The first step in all of these methods is converting an image into grayscale and scaling an image into an 8x8 image. After that, the computations for each of these 64 pixels are done depending on the algorithm used, and it gives each one of them a binary value of either 1 or 0. A-Hash calculates the average value of pixels and gives an output of 1 if the pixel is more than or equal to the average, otherwise the output is 0. P-Hash performs the same operation as A-Hash, but it performs a Discrete Cosine Transformation (DCT) and operates in the frequency domain first. D-Hash is a gradient hash and calculates the difference between every pixel and compares it to the average difference. W-Hash is wavelet hashing, which operates in the frequency domain like P-Hash, but employs Discrete Wavelet Transform (DWT) rather than DCT [22].

After we obtained all hash values, the differences between those values need to be calculated. Different distance measures between hash values can be utilized, such as Hamming distance, Euclidean distance, Manhattan distance and Cosine distance [23], but most papers use Hamming distance or Normalised Hamming distance to compare hash values. Hamming distance is the number of symbols or positions of two strings at which their corresponding characters are different (Eq. 1) [24]. Normalized Hamming distance is the ratio of the Hamming distance to the length of the string, or in this case 16.

$$D_H = \sum_{i=1}^{k} |x_i - y_i|, \qquad \begin{matrix} x = y \Rightarrow D = 0 \\ x \neq y \Rightarrow D = 1 \end{matrix} \tag{1}$$

In this study, we will use Normalized Hamming distance metric to evaluate the similarity between hash values. The distance was calculated between each hash value (A-Hash, P-Hash, D-Hash, W-Hash) of each original image and all of its modifications.

To discuss the results and draw the conclusions, min, max and mean values of perceptual hash differences have been calculated and are shown in Table II. It is important to note that the closer the Normalized Hamming distance is to zero, the more similar the images are. The results obtained in the experiments show that the A-Hash method achieves the lowest mean Normalized Hamming distances between reference hashes and their respective adjusted counterparts in all cases except for brightness modification and grayscale conversion where W-Hash has the lowest mean Normalized Hamming distance and outperforms the A-Hash. In most cases (Median Filtering, Salt & Pepper noise, Image Scaling, Image Rotation and Image Sharpening), W-Hash has been found to be in the second-place position. In total, D-Hash has been shown to be the least robust to changes in most cases. Also, while image rotation could be classified as a content modifying manipulation, it was interesting to see how different algorithm hash values behave during image rotation. The results have shown that P-Hash has the highest mean Normalized Hamming distance between original image and rotated image.

Given that the dataset contains nine distinct image categories, it was intriguing to determine whether algorithmic behavior varies depending on image category. If we take a look at the hash values per image category (Table III), we can conclude that there are no significant differences in hash value changes depending on image subject. However, in many categories and image manipulations, A-Hash and W-Hash have the same performance.

IV. Conclusion

A robust perceptual hashing technique should be one that yields the same or similar hashes for visually identical photos even if their digital representations are no longer the same. The objective of this study was to analyse the changes in perceptual hash values during different image manipulations. We carried out a series of experiments to evaluate the perceptual resilience of the four algorithms and for each of the algorithms, we calculated the hashes from the original 7 491 images and their corresponding changed versions according to the eight modifications

TABLE II: Normalized Hamming distances for different image manipulations

	A-Hash			P-Hash			D-Hash			W-Hash		
	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
Brightness Modification	0.000	0.875	0.112	0.000	0.875	0.198	0.000	0.938	0.142	0.000	1.000	**0.085**
Gaussian Blurring	0.000	0.250	**0.016**	0.000	0.375	0.022	0.000	0.313	0.044	0.000	0.500	0.043
Grayscale Conversion	0.000	0.250	**0.004**	0.000	0.313	0.010	0.000	0.438	0.014	0.000	0.625	0.032
Median Filtering	0.000	0.625	**0.019**	0.000	0.625	0.032	0.000	0.438	0.037	0.000	0.750	0.021
Salt & Pepper noise	0.000	0.750	**0.051**	0.000	0.750	0.120	0.000	0.813	0.169	0.000	0.813	0.076
Image Scaling	0.000	0.250	**0.006**	0.000	0.438	0.012	0.000	0.500	0.017	0.000	0.438	0.006
Image Rotation	0.188	1.000	**0.818**	0.563	1.000	0.914	0.375	1.000	0.904	0.125	1.000	0.828
Image Sharpening	0.000	0.625	**0.036**	0.000	0.563	0.062	0.000	0.688	0.055	0.000	0.625	0.045

TABLE III: Mean Normalized Hamming distances for each modification per image category

	Animals				Architecture				Art			
	aHash	pHash	dHash	wHash	aHash	pHash	dHash	wHash	aHash	pHash	dHash	wHash
Brightness Modification	0.107	0.180	0.135	**0.082**	0.103	0.202	0.155	**0.083**	0.127	0.211	0.158	**0.099**
Gaussian Blurring	**0.016**	0.021	0.041	0.041	**0.015**	0.022	0.051	0.044	**0.015**	0.026	0.042	0.045
Grayscale Conversion	0.005	0.010	0.014	**0.004**	**0.004**	0.009	0.020	**0.004**	0.004	0.009	0.013	**0.003**
Median Filtering	**0.018**	0.029	0.031	**0.018**	**0.019**	0.041	0.046	0.023	**0.020**	0.031	0.040	0.021
Salt & Pepper noise	**0.060**	0.119	0.169	0.090	**0.045**	0.124	0.198	0.069	**0.051**	0.121	0.162	0.074
Image Scaling	**0.005**	0.012	0.016	0.006	**0.005**	0.012	0.019	0.007	**0.006**	0.013	0.015	0.007
Image Rotation	**0.830**	0.908	0.897	0.836	**0.807**	0.920	0.903	0.823	**0.804**	0.907	0.897	0.810
Image Sharpening	**0.034**	0.060	0.049	0.044	**0.038**	0.074	0.067	0.050	**0.036**	0.065	0.057	0.042

	Characters				Indoor				Nature			
	aHash	pHash	dHash	wHash	aHash	pHash	dHash	wHash	aHash	pHash	dHash	wHash
Brightness Modification	0.127	0.207	0.151	**0.107**	0.109	0.192	0.113	**0.075**	0.110	0.239	0.163	**0.058**
Gaussian Blurring	**0.012**	0.018	0.040	0.039	0.019	**0.018**	0.036	0.041	**0.013**	0.022	0.049	0.033
Grayscale Conversion	0.003	0.006	0.010	**0.002**	0.004	0.008	0.011	**0.003**	0.004	0.013	0.018	**0.003**
Median Filtering	**0.014**	0.025	0.028	0.016	**0.021**	0.031	0.029	0.022	**0.017**	0.032	0.039	0.019
Salt & Pepper noise	**0.042**	0.094	0.136	0.068	**0.050**	0.118	0.137	0.072	**0.039**	0.142	0.209	0.069
Image Scaling	**0.004**	0.008	0.013	0.005	**0.006**	0.009	0.012	**0.006**	**0.004**	0.014	0.021	0.006
Image Rotation	**0.828**	0.920	0.914	0.836	**0.803**	0.907	0.899	0.809	**0.808**	0.919	0.896	0.825
Image Sharpening	**0.029**	0.046	0.042	0.036	**0.038**	0.062	0.048	0.041	**0.031**	0.068	0.058	0.041

	Plants				Scene				Texture			
	aHash	pHash	dHash	wHash	aHash	pHash	dHash	wHash	aHash	pHash	dHash	wHash
Brightness Modification	0.096	0.161	0.105	**0.073**	0.116	0.199	0.139	**0.091**	0.117	0.143	**0.115**	0.117
Gaussian Blurring	**0.019**	0.020	0.042	0.047	**0.016**	0.022	0.047	0.044	**0.028**	0.048	0.039	0.083
Grayscale Conversion	0.005	0.008	0.011	**0.003**	0.004	0.009	0.013	**0.002**	0.012	0.020	0.017	**0.009**
Median Filtering	**0.021**	0.026	0.032	0.023	**0.020**	0.037	0.041	0.022	**0.045**	0.057	0.058	0.055
Salt & Pepper noise	**0.062**	0.119	0.144	0.088	**0.046**	0.113	0.168	0.067	**0.116**	0.188	0.191	0.141
Image Scaling	0.008	0.010	0.015	**0.007**	0.005	0.010	0.017	**0.004**	**0.017**	0.024	0.030	**0.017**
Image Rotation	**0.810**	0.907	0.909	0.821	**0.836**	0.918	0.911	0.845	**0.855**	0.916	0.910	0.860
Image Sharpening	**0.040**	0.052	0.048	0.046	**0.040**	0.068	0.061	0.048	**0.078**	0.092	0.086	0.089

described in Table I. After that, the Normalized Hamming distance between each original hash and its associated altered hash is calculated. The experiment results have shown that A-Hash method achieves the lowest mean Normalized Hamming distances between reference hashes and their respective adjusted counterparts in all cases except for brightness modification and grayscale conversion where W-Hash has the lowest mean Normalized Hamming distance. The future work will focus on the analysis and comparison of additional algorithms and additional manipulations, as well as the comparison of hash values for different content modifying manipulations.

ACKNOWLEDGMENT

This research was funded by the project "Development of CSTI platform for retrieval and analysis of structured and unstructured data". The project received funding from the European Regional Development Fund through OP Competitiveness and Cohesion 2014–2020 within the Call for Proposals "Development of the products and services arising from research and development activities—Phase II", under grant number KK.01.2.1.02.0310.

REFERENCES

[1] S. P. Singh, G. Bhatnagar, and A. K. Singh, "A new robust reference image hashing system," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 4, pp. 2211–2225, 2022.

[2] R. Chandramouli, N. Memon, and M. Rabbani, "Digital watermarking," *Encyclopedia of Imaging Science and Technology*, vol. 10, p. 0471443395, 2002.

[3] M. Hamadouche, K. Zebbiche, M. Guerroumi, H. Tebbi, and Y. Zafoune, "A comparative study of perceptual hashing algorithms: Application on fingerprint images," 2021.

[4] M. Rickard. Perceptual hashing. [Online]. Available: https://matt-rickard.com/perceptual-hashing

[5] Ofcom. Perceptual hashing technology. [Online]. Available: https://www.ofcom.org.uk/__data/assets/pdf_file/0036/247977/Perceptual-hashing-technology.pdf

[6] J. Dittmann, A. Steinmetz, and R. Steinmetz, "Content-based digital signature for motion pictures authentication and content-fragile watermarking," in *Proceedings IEEE International Conference on Multimedia Computing and Systems*, vol. 2, 1999, pp. 209–213 vol.2.

[7] L. Du, A. T. Ho, and R. Cong, “Perceptual hashing for image authentication: A survey,” *Signal Processing: Image Communication*, vol. 81, p. 115713, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0923596519301286
[8] K. Ding, S. Chen, and F. Meng, “A novel perceptual hash algorithm for multispectral image authentication,” *Algorithms*, vol. 11, no. 1, 2018.
[9] S. Bhattacharjee and M. Kutter, “Compression tolerant image authentication,” in *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*, vol. 1, 1998, pp. 435–439 vol.1.
[10] R. Venkatesan, S.-M. Koon, M. Jakubowski, and P. Moulin, “Robust image hashing,” in *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, vol. 3, 2000, pp. 664–666 vol.3.
[11] M. K. Mıhçak and R. Venkatesan, “New iterative geometric methods for robust perceptual image hashing,” in *Security and Privacy in Digital Rights Management*, T. Sander, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 13–21.
[12] V. Monga and B. Evans, “Robust perceptual image hashing using feature points,” in *2004 International Conference on Image Processing, 2004. ICIP ’04.*, vol. 1, 2004, pp. 677–680 Vol. 1.
[13] S. Kozat, R. Venkatesan, and M. Mihcak, “Robust perceptual image hashing via matrix invariants,” in *2004 International Conference on Image Processing, 2004. ICIP ’04.*, vol. 5, 2004, pp. 3443–3446 Vol. 5.
[14] A. Swaminathan, Y. Mao, and M. Wu, “Robust and secure image hashing,” *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 215–230, 2006.
[15] X. Lv and Z. J. Wang, “Perceptual image hashing based on shape contexts and local feature points,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1081–1093, 2012.
[16] Z. Tang, X. Zhang, and S. Zhang, “Robust perceptual image hashing based on ring partition and nmf,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 711–724, 2014.
[17] Z. Tang, X. Zhang, X. Li, and S. Zhang, “Robust image hashing with ring partition and invariant vector distance,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 200–214, 2016.
[18] C. Qin, Y. Hu, H. Yao, X. Duan, and L. Gao, “Perceptual image hashing based on weber local binary pattern and color angle representation,” *IEEE Access*, vol. 7, pp. 45 460–45 471, 2019.
[19] Q. Shen and Y. Zhao, “Perceptual hashing for color image based on color opponent component and quadtree structure,” *Signal Processing*, vol. 166, p. 107244, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168419302907
[20] S. P. Singh, G. Bhatnagar, and A. K. Singh, “A new robust reference image hashing system,” *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 4, pp. 2211–2225, 2022.
[21] J. Dong, W. Wang, and T. Tan, “Casia image tampering detection evaluation database,” in *2013 IEEE China Summit and International Conference on Signal and Information Processing*, 2013, pp. 422–426.
[22] D. Petrov. Wavelet image hash in python. [Online]. Available: https://fullstackml.com/wavelet-image-hash-in-python-3504fdd282b5
[23] S. Huang, J. Qin, T. Tao, L. Yang, and X. Liu, “A novel method to perceive self-vehicle state based on vehicle video by image similarity calculation,” *IEEE Open Journal of Instrumentation and Measurement*, vol. 1, pp. 1–11, 2022.
[24] X.-S. Yang, “Chapter 6 - genetic algorithms,” in *Nature-Inspired Optimization Algorithms (Second Edition)*, second edition ed., X.-S. Yang, Ed. Academic Press, 2021, pp. 91–100. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128219867000135