Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 185 (2021) 203–212

Procedia Computer Science

www.elsevier.com/locate/procedia

Complex Adaptive Systems Conference Theme: Big Data, IoT, and AI for a Smarter Future
Malvern, Pennsylvania, June 16-18, 2021

Analysis of Perceptual Hashing Algorithms in Image Manipulation Detection

Priyanka Samanta[a,] *, Shweta Jain[a,b]

[a]*The Graduate Center of CUNY, 365 5th Ave, New York, NY 10016, USA*
[b]*John Jay College of Criminal justice CUNY, 524 W 59th street, New York, NY 10019, USA*

Abstract

Perceptual image hashing is a family of algorithms that generate content-based image hashes. Unlike cryptographic hashes, perceptual hashes are designed to not change much when an image undergoes minor modifications such as compression, color-correction, and brightness. Therefore, these algorithms have been found useful in detecting redundant images, perform reverse image search, and flagging/filtering inappropriate imagery by comparing image hashes with a dataset of known perceptual hashes. While these applications have been served well, there is a new threat to visual information in the form of misinformation through content changing image manipulation using techniques such as object addition/removal and copy-move. While perceptual hashing algorithms can very accurately identify copies of similar images and their legitimately modified versions, their effectiveness in detecting content changing manipulations is not studied well. Existing research and advanced image libraries propose various image hashing algorithms, and several datasets of manipulated images are available in the public domain. In this work, a review and performance analysis of several perceptual hashing algorithms against two such datasets is presented. Results show less than ideal performance in distinguishing maliciously manipulated images from legitimate ones. Based on the results, limitations of perceptual hashing algorithms are identified, and future research directions are proposed.

© 2021 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the Complex Adaptive Systems Conference, June 2021.

Keywords: Perceptual hashing; Image manipulation detection

* Corresponding author.
E-mail address: psamanta@gradcenter.cuny.edu

1877-0509 © 2021 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the Complex Adaptive Systems Conference, June 2021.
10.1016/j.procs.2021.05.021

1. Introduction

Digital images are essential source of information and expression on all online platforms ranging from news to social media and blogs. Most credible online platforms care about the provenance and legitimacy of images on their platforms, even though up until recently, social media platforms have stayed away from moderating any information, including manipulated and copyrighted images. General content moderation by social networks has become a partisan issue but banning manipulated media that spread misinformation is more objective and should not be debatable.

Stylizing and modifications of images for artistic effect, entertainment, and advertisements date back to the time when images were produced on photographic plates. Stylizing included brightness adjustment, rotation, cropping, airbrushing, color correction, red-eye correction, and more. The fashion and advertising industry has always manipulated images to achieve flawless looks for their models. However, image manipulation to spread misinformation is much more recent. The technology is so far advanced that it becomes a challenge to determine whether a copy of an image was simply stylized (legitimate modification) or its content was modified to depict something that was not in the original i.e., content changing (CC) modifications. Specifically, modified versions of genuine images are often widely shared on social media right after a major event transpires. Much before the validity of the image is verified, the misinformation can potentially reach a worldwide audience. Due to the large volume of media uploaded daily, online platforms struggle to find an automated solution to this problem.

While automation using Perceptual Hashing is currently used to filter known banned, duplicate and similar images, whether the same technology can be used to detect content changing (CC) image manipulations is yet to be determined. In this work, a review and performance analysis of several perceptual hashing algorithms using two publicly available datasets of manipulated images, the CASIA2.0 [18] and Photoshop Battle [19] datasets are presented. Results show less than ideal performance in distinguishing images with CC manipulations from content preserving (CP) modifications. Thus, limitations of these algorithms are identified, and future research is proposed.

2. Problem statement

The problem statement for this study is: *Given an input image, a limited size string (hash) and a database of either original images or their hashes, how accurately can the algorithm determine whether the input image has undergone content changing manipulations.*

In this statement, the definition of content preserving modifications are JPEG compression, filtering (colour changing), blurring, and Gamma correction (brightness adjustment) performed for practical, aesthetic and artistic reasons. Content changing manipulations are copy-move, face swap, image splicing, adding or removing objects, and background change which are often done to change the visual information. These modifications are often used for false advertising or in offensive memes. Fig. 1 shows some images with CC manipulations. Note that significantly cropped images are considered to be a different image.

Fig. 1. Examples of content changing manipulation (a) Face swap;(b) Cropped;(c) Logo/text addition;(d) Object added;(e) Copy-move;(f) Background change.

3. Perceptual hash generation steps

Perceptual hashing algorithms use perceptual features of images to generate their hashes. The primary goal is to generate hashes that remain unchanged or change slightly when content preserving modifications are made to the image. Given two images I and I′, $h_I = H(I)$ and $h_{I'} = H(I')$ their corresponding perceptual hashes, and $D(h_I,h_{I'})$ is a similarity metric, and τ is an empirically determined threshold, $D(h_I,h_{I'}) < \tau$ indicates that I and I` are copies of the same image with minor content preserving modifications. The main three steps involved in perceptual hashing algorithms are image pre-processing, perceptual feature extraction, and quantization or compression to generate the final hash string. There are various perceptual hashing algorithms that vary from each other in the way they extract perceptual features from the image [21].

3.1. Pre-processing

In the pre-processing phase, an input image is prepared for feature extraction. This step reduces the size of the data that needs to be processed at a later stage, reducing the overall processing time. Some pre-processing steps include resizing, colour transformation, normalization, filtration, and histogram equalization. For example, the input image might be converted into one of the common colour models such as YCbCr (Luminance [Y], Chrominance-blue [Cb], and Chrominance-red [Cr]), HSV (hue, saturation, value) or Grayscale. Similarly, all digital images contain random noise due to external factors at the time of image capture such as sensor heat and the camera's ability to capture light. Therefore, the pre-processing step might remove noise by applying Gaussian noise filters. Histogram equalization might be used to improve the contrast in an image by spreading out the most frequent intensity values. Images might be resized to evaluate all inputs at a common size.

3.2. Feature extraction

In perceptual hashing, the features extracted from an image should be invariant to content preserving manipulation. There are mainly two types of techniques used for general feature extraction, (a) frequency domain transformation such as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Fourier-Mellin transform and (b) dimensionality reduction techniques such as Principal Component Analysis (PCA), Non-Negative Matrix Factorization (NMF) and Singular Value Decomposition (SVD).

3.3. Quantization and hash generation

Numerical values that represent features of an image can be quantized, generating a fixed sized hash which is a compact and somewhat unique representation of an image. Statistical properties of features are often used for quantization. For example, in DCT algorithms, the DCT coefficients can be quantized by comparing each coefficient with the median or mean of the coefficients.

3.3. Similarity Metrics

In order to use perceptual hashing algorithms to identify similar or duplicate images, a similarity metric is used to compare the hashes computed from images. Some commonly used similarity metrics are Hamming Distance (HD) or Normalized Hamming Distance (NHD), Euclidean distance, Bit Error Rate (BER), and Peak of Cross-Correlation (PCC). In this paper Normalized Hamming Distance (NHD) has been used for all performance analysis.

Hamming Distance (HD) is the minimum number of bits flips that will change one hash into the other [20]. Thus, the Hamming Distance between two hash strings $H_1 = \{ h_1(0), h_1(1), \ldots, h_1(n-1) \}$ and $H_2 = \{h_2(0), h_2(1), \ldots h_2(n-1)\}$ is computed as:

$$HD(H_1, H_2) = \sum_{i=0}^{n-1} h_1(i) \; XOR \; h_2(i) \tag{1}$$

The Hamming Distance from Eq. 1 is divided by the hash length to produce a Normalized Hamming Distance (NHD). This allows for comparison between algorithms that produce hashes of different lengths.

Since similar images produce hashes that have higher similarity, the Hamming Distance between them is small compared to images that are different from each other. Therefore, when comparing two images I and I', if NHD(H(I), H(I')) < T, where T is an empirically determined threshold, then the pair (I, I') are likely similar or have undergone content preserving modifications. If the distance is greater than T, the images are likely different, or one has undergone content changing modifications. Based on the similarity measure, the accuracy, precision and recall of various algorithms can be compared using datasets of original and manipulated images.

4. Related work

There is a rich set of past work in differentiating content preserving manipulations from distinct images. Some of them are capable of detecting content changing manipulations. In this section, these algorithms are categorized by their feature extraction techniques and then their effectiveness in detecting changing manipulations is discussed.

DCT is one of the most well researched techniques [1-7]. It is used for JPEG image compression and was originally proposed by Nasir Ahmed in 1972. The DCT of an image is computed by projecting the image onto a fixed coefficient matrix in the frequency domain by expressing each pixel as the sum of cosine components of their Fourier Transform. As the DCT has a strong "energy compactness" property, most of the image information concentrates on low-frequency DCT coefficients. These low frequency components are mostly stable under any content preserving manipulation but sensitive to changes in perceptual features such as adding or removing objects.

In phash [1, 2], DCT is applied on the entire image and then the coefficients are quantized to form a binary hash vector. phash is designed for duplicate and content preserving manipulation detection. Zeng et al [3] proposed colour histogram along with DCT as feature extraction technique, then compressed the hash using Principal Component Analysis (PCA). The proposed algorithm can detect logo insertion as demonstrated with 12 images. Wang et al [4] applied DCT to create perceptual hash vectors to specifically detect only copy-move forgery.

Tang et al [5] used DCT coefficients of non-overlapping blocks of an image which is invariant to content preserving manipulation and achieved 98.5% accuracy. Wang et al [6] used Watson's DCT visual model on image blocks and showed invariance to content preserving manipulation and sensitivity to content changing manipulation. However, the authors tested their algorithm on 1000 tampered images and the performance metric is described in tampering rate and detection rate. They showed their algorithm achieved similar detection rate as tampering rate increased and that varying threshold can affect the sensitivity of robustness to CP and detection of CC manipulations. Lin et al [7] proposed an algorithm based on the positional relationship of DCT coefficients of different image blocks to identify content changing modifications. However, performance was evaluated only on a few Lena images.

Discrete Wavelet transform can detect edges in images effectively. The output of a wavelet decomposition of an image are approximation coefficient matrix (LL) and three detailed coefficient matrices which capture horizontal (LH), vertical (HL) and diagonal (HH) edge information. Several researchers have used DWT to create perpetual hashes. Their work differs in processing of coefficients and quantization to produce the hash.

Hu et al [8] used the detailed coefficients (LH, HL, HH) from 2-level DWT decomposition to distinguish between content preserving and content changing edits. Only 44 images were used to test detection of content preserving manipulation and 2 images for content changing manipulation. Fawad et al [9] used 4-level DWT decomposition of non-overlapping image blocks and used LL, LH, LL, and HL to compute the hash to detect content changing manipulation from content preserving edits. Authors have shown their algorithm can detect both low contrast and high contrast small tampering but the technique was tested only on 2 such images. Karsh et al [10] applied 4-level DWT and spectral residual model to extract global features and then saliency map to detect local features. The final hash is calculated by combining 2 intermediate hash values. The algorithm was tested on content preserving images as well as 800 content changing images manipulations, showing 100% detection of content changing modifications. Sing et al [11] created hash from DWT and SVD which is invariant to content preserving modifications.

Non-negative matrix factorization (NMF) decomposes V, a M x N matrix into B, a M x K matrix and C, a K x N matrix, such that all elements in the three matrices are non-negative, V=B x C and K < min (M, N). NMF is a dimensionality reduction technique and due to the non-negative constraint, it can be used on images.

Tang et al [13] used NMF for perceptual hashing by dividing an image into concentric rings and performing NMF of matrices generated from each ring. They also used NMF twice to generate the hash [14] to enable detection of content changing manipulation. However, the experiments showed results with only 2 tampered images. Monga et al [15] computed image hashes by first dividing an image into blocks and then performing NMF of the factors generated from the NMF of matrices representing each block. In this paper, author showed in an example image that NMF is

sensitive to local tampering but the experiments were not performed on large scale datasets.

Singular value decomposition (SVD) is a dimensionality reduction technique that can preserve the magnitude and locations of the most important features of an image with the help of the N largest left singular vectors and N largest right singular vectors. Ricardo et al [16] used SVD decomposition on input images and quantized the first left and right singular vectors using order-3 Reed Muller decoder to generate the hash which is robust against rotation, scaling and flipping. Kozat et al [17] proposed three different combinations: SVD-SVD, DCT-SVD and DWT-SVD to compute perceptual hashes of images. This algorithm is robust against JPEG compression, rotation ($1\text{-}20^0$) and cropping (10-60%).

Lamdan et al [25] proposed a geometric hashing scheme to recognize objects in images by comparing the object hashes in a database. This technique can be used in perceptual hashing by comparing objects and their coordinates in a pair of images. Lv et al [26] proposed a SIFT-Harris detector based hashing scheme which can extract features based on shape context and their coordinates. This algorithm is able to detect content changing modifications and is invariant to content preserving manipulations.

Du et al [27] surveyed the most recent perceptual hashing algorithms and their performance for content preserving manipulation detection. Content changing manipulations was not explored. This work mainly focused on frequency transformation-based techniques (DCT, DWT) and matrix factorization techniques (NMF, SVD) for perceptual hashing.

5. Evaluation

While several feature extraction techniques have been proposed in research literature, the setup in which they are applied are different. Therefore, it is hard to determine the relative efficacy of these techniques in distinguishing between content preserving and content changing image modifications. For a fair comparison, each feature extraction technique should be studied under the same baseline. This paper addresses the research gap by evaluating several feature extraction techniques to a common setup which is based on the phash algorithm [2]. Therefore, the image is first converted to Gray scale, then resized to 32x32 using area interpolation and then a 7x7 average filter kernel is applied. Fig. 2 shows how an image looks like after the pre-processing steps. Next each feature extraction technique is applied to evaluate how they perform relative to the phash algorithm. As the object detection and shape context-based hashing are very different approaches and might be computationally expensive, those techniques are not evaluated in this work. Instead, this work focuses on four feature extraction processes: DCT, DWT, NMF and SVD.

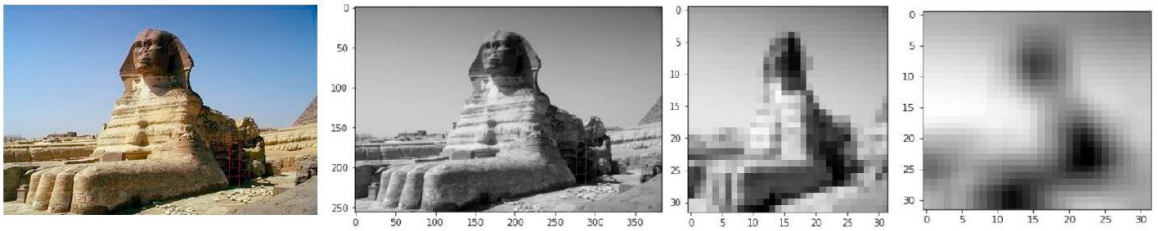

Fig. 2. Different pre-processing stages; a) original image, b) Gray-scaled image, c) 32 x 32 resized image, d) after applying 7x7 kernel

5.1. Setup for baseline algorithms

phash: The core feature extraction technique for phash algorithm is DCT. After pre-processing, DCT of the 32x32 image is computed which results in 32x32 coefficients matrix where each coefficient is denoted by $C_{ij,}$ (i=0…31, j=0…31). Then the top left 8x8 lower frequency coefficients C_{ij} (i=0…7, j=0…7) are selected for the final hash calculation. Each coefficient is quantized by comparing with the median (C_m) of all 8x8=64 coefficients. Therefore $C_{ij} = 0$ if $C_{ij} <= C_m$ and 1 otherwise. The final hash is a 64-bit string.

whash: The core feature extraction technique for whash algorithm is DWT. After pre-processing, 1 level DWT of the 32x32 image is computer using Haar wavelet. An approximation coefficient matrix is used here for the hash generation. Like the phash algorithm, each approximation coefficient is quantized as 0 or 1 based on its value being greater or less/equal to the median. The final hash value is 256 bits.

NMF: As the name suggests, the core feature extraction technique for this baseline algorithm is NMF. In this

paper, 2 versions of NMF baselines have been studied. For both the versions, at first rank 5 NMF of 32x32 images are computed. This results in 2 matrices of shape 32x5 and 5x32. For version 1, the wider matrix (5x32) is considered for the hash generation. Hash is computed by following the quantization technique suggested in [13]. Each coefficient entry of the r^{th} row and n^{th} column, $C_{r,n}=0$ if $C_{r,n}<=C_{r,n+1}$,$C_{r,n}=1$ otherwise. The final hash value is 160 bit long for version 1. For version 2, both the taller (32x5) and wider (5x32) matrices are used for the hash generation. The same quantization technique is applied for both matrices and finally they are concatenated to form 320-bit long hash.

SVD: SVD is the core feature extraction technique for this baseline. Similar to NMF, 2 versions of SVD baselines have been studied in this paper. After pre-processing, the 32x32 image is factorized in the form $\mathbf{U\Sigma V^*}$ where $\mathbf{\Sigma}$ is a 32x32 diagonal matrix with non-negative real numbers in the diagonal and **U and V*** are two 32x32 orthogonal matrices. **U** and **V** are called the left and right matrices and their columns are called left and right singular vectors respectively. In version 1 of the SVD based feature extraction technique, first 5 right singular vectors are used to create a hash. In version 2, first 5 left and first 5 right singular vectors are used to create hash. The quantization technique is the same as the NMF baseline. SVD version 1 produces 160-bit long hash while version 2 produces a 320-bit hash.

5.2. Description of the Datasets and Data Cleaning

In CASIA2 [18] dataset, before pre-processing there are 12,614 colour images, 7,491 original images and 5,123 images with content changing modifications such as face swap, object addition, object removal, background colour change and copy move forgery. The image sizes are between 240x160 to 900x600 and image formats are TIFF and JPEG compressed images. In the data cleaning step, original images which do not have any content changing derivatives are discarded. After data cleaning, the CASIA2 dataset has 1935 original images and 5069 content changing derivatives of the original.

In Photoshop Battle dataset [19], it has 11,142 original images and 99,886 images with content changing modifications. The images are of high resolution with size 960x640. Each original image has up to 67 copies of images with content changing modifications. At the data cleaning stage, after removing corrupt image files, the dataset left with 11,071 original images and 87,898 images with content changing modifications.

Amongst other image datasets such as Realistic Tampering Dataset (RTD) [22] which has only 220 original vs tampered images, McGill Calibrated Color Image Database [23] which has no tampered images, only has original and scaled images, RAISE [24] has 8156 raw images but no tampered counterpart. Photoshop Battle and CASIA2 are the most recent comprehensive datasets which provides a wide variety of original and content changing manipulated images. CASIA2 has mixed of low- and high-resolution images where Photoshop Battle dataset has all high-resolution images. Hence in this work these two datasets have been used covering the wide spectrum of image manipulation.

5.3. Experiment setup

For both the datasets, the content preserving manipulations are created by applying the techniques listed in Table 1. As a result, CASIA2 dataset contents 25,155 copies of content preserving copies of 1935 original images and Photoshop battle dataset contents 143,923 copies of content preserving copies of 11,071 original images. Finally, each one of the 6 algorithms are used on CASI2 and Photoshop battle datasets to compute corresponding image hashes. The Normalized Hamming Distances between original images and all their content preserving and content changing copies are calculated for each algorithm. The distribution of these Hamming Distances is plotted to determine the optimum threshold for classification.

Table 1. Types of content preserving (CP) modifications.

Content preserving edits	Parameters
JPEG compression	QF= 90%, 65%, 40%
Gamma correction	γ=0.8,0.9,1.1,1.2,1.3
Additive Gaussian noise	SD=0.1,0.25,0.5,1.5
Blurring	3X3 median filter

5.4. Results

Fig. 3 shows the boxplots of Normalized Hamming Distances for the content changing manipulations using different algorithms, while Fig. 4 shows results for content preserving modifications. For content changing manipulation detection SVD has higher mean and maxima compared to other algorithms in both datasets, which shows better performance in detecting content changing manipulations. On the other hand, phash and whash shows better results for content preserving manipulation detection. For both datasets, phash has fewer outliers than whash, hence misclassification probability is less.

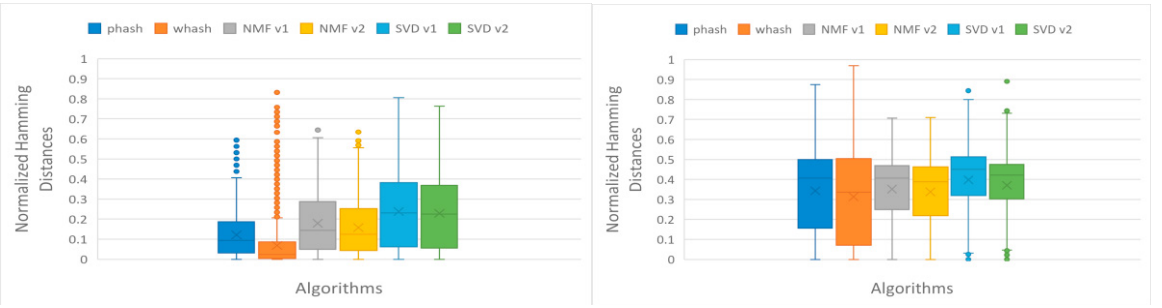

Fig.. 3. Normalized Hamming Distance comparison for content changing manipulations on CASIA2 and PB datasets

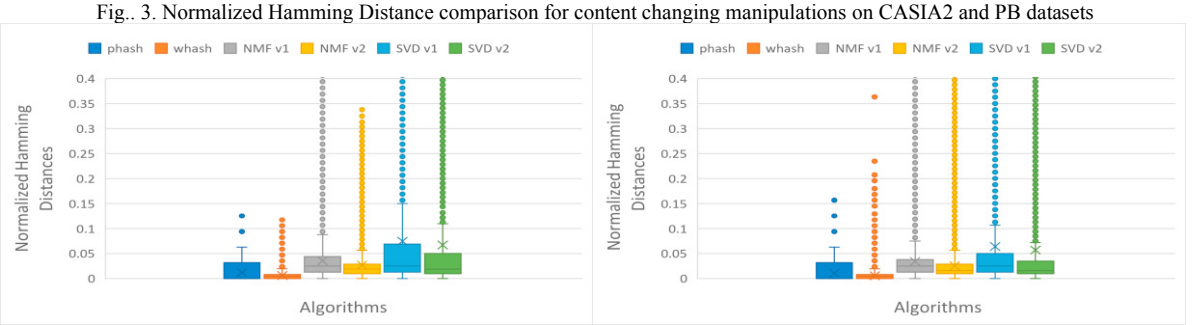

Fig. 4. Normalized Hamming Distance comparison for content preserving manipulations on CASIA2 and PB datasets (note different y-axis scale than Fig. 4)

The next step is to determine a threshold Hamming Distance that denotes the boundary between the two categories of image modification. This is achieved through a data driven approach by plotting the Probability Density Functions (PDF) of the Normalized Hamming Distance distributions of (a) original images and their content preserving copies, (b) originals and copies with content changing modifications and (c) pairs of distinct original images. Fig. 5 shows the PDF distribution for phash algorithm on CASIA2 dataset. The exact same method has applied for all the algorithms. In Fig. 5, there is an overlap between distributions for content changing and content preserving modifications. All the algorithms show same pattern for both the datasets. The intersection point "A" is the Equal Error Rate (EER). On the left of A, images have higher than 0.5 probability of being content preserving copies while on the right of A, images have higher probability of being content changing copies. Therefore, point A is selected as the threshold $\boldsymbol{\tau}$ for the image classification task. The threshold $\boldsymbol{\tau}$ may differ for an algorithm on different datasets. Table 2 shows the thresholds determined through the probability density function calculation for each algorithm. In this work precision, recall, F1 score and accuracy for both the classes are used as performance metrics.

Algorithms	CASIA2	Photoshop Battle
phash (DCT based)	0.04	0.04
whash (DWT based)	0.02	0.02
NMF v1	0.06	0.08
NMF v2	0.05	0.07
SVD v1	0.05	0.07
SVD v2	0.04	0.06

Table 2. Threshold for different algorithms for CASIA2 and PB datasets.

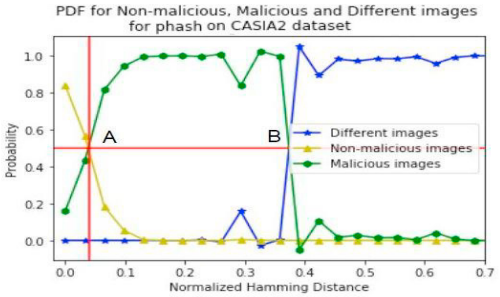

Fig. 5. Probability distribution of Hamming Distances and threshold determination of phash algorithm on CASIA2

Precision: Precision for CP is a measure of proportion of content preserving images predicted correctly out of total predicted content preserving images. Precision for CC is a measure of proportion of content changing images predicted correctly out of total predicted content changing images.

Recall: Recall for CP is a measure of the proportion of content preserving images predicted correctly out of total content preserving images. Recall for CC is a measure of the proportion of content changing images predicted correctly out of total content changing images.

F1 score: It is a measure that combines precision and recall for each class. *F1 score= 2(precision X recall)/(precision + recall).* The F1-Macro score is the average F1 score for CC and CP classification. F1 macro score shows how good an algorithm is doing in terms of misclassification.

Accuracy: Accuracy is measured as the fraction of correct predictions$accuracy = (TP + TN)/(TP + FN + TN + FP)$. Accuracy shows how good an algorithm is going in terms of true class prediction.

Table 3. Precision, Recall, F1 score and accuracy comparisons on CASIA2 and PB datasets.

Algorithms	CASIA2			Photoshop Battle		
	CP	CC		CP	CC	
	Precision, Recall, F1 score (%)	Precision, Recall, F1 score (%)	Accuracy, F1-macro(%)	Precision, Recall, F1 score (%)	Precision, Recall, F1 score (%)	Accuracy F1 -macro (%)
phash (DCT)	**94.14, 97.02, 95.5**	82.61, 70.04, **75.80**	**92.50, 85.65**	**94.08, 97.87, 95.93**	96.26, 89.93, **92.98**	**94.86, 94.55**
whash (DWT)	91.22, 93.45, 92.32	63.06, 55.42 58.99,	87.07, 75.65	92.26, 95.48 , 93.84	90.59, 87.02, 88.76	91.65, 91.30
NMF v1	94.03, 87.92, 90.86	54.70, 72.33, 66.59	85.31, 78.72	93.47, 92.90, 93.18	54.70, 89.38, 67.86	91.56, 80.52
NMF v2	94.33, 89.93, 90.12	59.46, 73.22, 65.62	87.13, 77.87	93.25, 94.49, 93.86	90.79, 88.80, 89.78	92.34, 91.82
SVD v1	94.34, 69.09, 79.76	**34.13, 79.4, 47.73**	70.83, 63.74	93.57, 79.42, 85.91	**73.00 ,91.07, 81.07**	83.84, 83.49
SVD v2	94.67, 72.09, 82.84	**36.58, 79.89, 50.18**	73.39, 66.51	93.84, 80.40, 86.60	**74.01, 91.36, 81.77**	84.56, 84.18

Table 3 shows the performance comparison of all the algorithms. For both the datasets, DCT based phash algorithm has shown the highest F1 macro score and highest accuracy for both the classes. This signifies that the DCT based phash technique offers least misclassification as well as highest prediction accuracy. While the recall for content changing manipulation of SVD based algorithms are comparatively better, at the same time the precision is low, signifying a higher misclassification of content preserving modifications as content changing manipulations, which is not desirable. The phash algorithm turns out to be the best at detecting CP manipulation, but also better than others at detecting CC modifications.

However, none of the feature extraction techniques are satisfactory in detecting CC modification in practice. While prior research has shown good performance for NMF and SVD based techniques in exemplar cases of CC modifications, their results were probably also an artefact of the way images were processed and partitioned before the features were extracted.

Having evaluated the above feature extraction techniques quantitatively, manual investigation of the patterns behind failure in detecting CC manipulation was performed. Two patterns immediately became evident, first relates to the size of the manipulation and the second relates to the contrast. Most feature extraction techniques missed CC modifications if the object removal or addition affected a relatively small area in the original image as shown in Fig.. 6. To validate this hypothesis, a controlled experiment on 5 images was performed.

For each image, such as the one in Fig. 6, the small, modified area was cropped, and the hashes of the cropped region were computed using each technique. Results show improved performance compared to when the entire image was hashed. Similarly, when objects with similar contrast as the background on the original images are added or removed, the hash computed by the phash algorithm does not change much. Therefore, such changes remain undetected by phash. On the other hand, the NMF technique is quite good at recognizing several low contrasts changes that phash could not detect but fails to detect high contrast changes.

Fig. 6. Images with small manipulation areas

Therefore, if images are first classified using their phash and then using NMF, the recall for the sequential application of the techniques leads to 79% recall for CC modifications in the CASIA2 dataset. Furthermore, by sequencing phash, NMFv1 and NMFv2 for image manipulation detection, the recall for CC modifications is 81% for the CASIA2 dataset. Sequencing phash with SVDv1 gives 83.01% recall and phash with SVDv1 and SVDv2 gives 84.11% recall. The highest recall, 85.91%, was from sequencing phash with NMFv1 and SVDv1. Similar trend is observed in photoshop battle dataset and the highest recall 95.5% achieved by sequencing phash, NMFv1 and SVDv1.

However, sequential applications can come with high computational cost. Therefore, it is important to understand the speed of execution of various feature extractions techniques. Table 4 shows the time it takes to run the program to calculate the hash of each image. These times are the average time taken to compute hashes of 100 images on an Intel Core i7-7500 CPU 16.0 GB RAM machine. Thus, all algorithms are quite fast and given the speed, it is feasible to compute hashes using several techniques and then using a decision tree-based classifier to detect CC and CP modifications more accurately.

Table 4. Running time comparison for different algorithms.

Algorithms	Run time
phash (DCT based)	0.007 sec
whash (DWT based)	0.019 sec
NMF v1	0.013 sec
NMF v2	0.015 sec
SVD v1	0.004 sec
SVD v2	0.009 sec

6. Conclusion

In this paper the effectiveness in detecting content changing image manipulation of existing perceptual hashing algorithms have been studied. This paper compares different state of the art perceptual feature extraction techniques to generate perceptual hashes for a better understanding of different mechanisms. Overall DCT based hashing shows promising results for the classification of content preserving modification from content changing manipulation. SVD algorithm shows better results in manipulation detection but needs improvement for content preserving manipulation.

However, none of these algorithms are perfect. Future work will include using multiple hashing algorithms and using a Decision Tree to understand if a combination of algorithms can result in better performance. Further work is also needed to understand how quantization, which is necessary for compact representation of the resulting hash, affects performance in the dimensionality reduction techniques. Given the advantage of DCT in detecting both CC and CP modifications, perhaps a better quantization technique might be to compute DCT of the image generated after dimensionality reduction. A Hash computed from the resulting DCT coefficients might yield better results as it combines advantages of the two feature extraction processes.

References

[1] phash: The open source perceptual hash library. URL: https://www.phash.org/

[2] Zauner, Christoph. (2010) "Implementation and benchmarking of perceptual image hash functions."

[3] Jie, Zeng. (2013) "A novel block-DCT and PCA based image perceptual hashing algorithm." arXiv preprint arXiv:1306.4079.

[4] Wang, Huan, and Hongxia Wang. (2018) "Perceptual hashing-based image copy-move forgery detection." *Security and Communication Networks 2018*.

[5] Tang, Zhenjun, et al. (2014) "Robust image hashing with dominant DCT coefficients." *Optik,* **125.18**: 5102-5107.

[6] Wang, Xiaofeng, et al. (2015) "A visual model-based perceptual image hash for content authentication." *IEEE Transactions on Information Forensics and Security*, **10.7**: 1336-1349.

[7] Lin, Ching-Yung, and Shih-Fu Chang. (2001) "A robust image authentication method distinguishing JPEG compression from malicious manipulation." *IEEE Transactions on Circuits and Systems for Video Technology*, **11(2)**, 53–168.

[8] Hu, Yuanyuan, and Xiamu Niu. (2010) "DWT based robust image hashing algorithm." *INC2010: 6th International Conference on Networked Computing*, IEEE.

[9] Ahmed, Fawad, Mohammed Yakoob Siyal, and Vali Uddin Abbas. (2010) "A secure and robust hash-based scheme for image authentication." *Signal Processing*, **90.5**: 1456-1470.

[10] Karsh, Ram Kumar, and Rabul Hussain Laskar. (2017) "Robust image hashing through DWT-SVD and spectral residual method." *EURASIP Journal on Image and Video Processing 2017*, **1**: 31.

[11] Singh, Satendra Pal, and Gaurav Bhatnagar. (2017) "A robust image hashing based on discrete wavelet transform." *2017 IEEE International Conference on Signal and Image Processing Applications* (ICSIPA), IEEE.

[12] M.K. Mihcak, and R. Venkatesan. (2001) "New iterative geometric methods for robust perceptual image hashing." *ACM Workshop Security and Privacy in Digital Rights Management*, Philadelphia, PA, Nov. 2001.

[13] Tang, Zhenjun, Xianquan Zhang, and Shichao Zhang. (2013) "Robust perceptual image hashing based on ring partition and NMF." *IEEE transactions on knowledge and data engineering*, **26.3**: 711-724.

[14] Tang, Zhenjun, et al. (2008) "Robust image hashing for tamper detection using non-negative matrix factorization." *Journal of ubiquitous convergence technology*, **2.1**: 18.

[15] Monga, Vishal, and Mehmet Kivanç Mihçak. (2007) "Robust and Secure Image Hashing via Non-Negative Matrix Factorizations." *IEEE Trans. Information Forensics and Security*, **2.3-1**: 376-390.

[16] Hernandez, Ricardo Antoio Parrao, Mariko Nakano Miyatake, and Brian M. Kurkoski. (2011) "Robust image hashing using image normalization and SVD decomposition." *2011 IEEE 54th International Midwest Symposium on Circuits and Systems* (MWSCAS), IEEE.

[17] Kozat, S., Venkatesan, R., and Mihcak, M. (2004) "Robust perceptual image hashing via matrix invariants." *Image Processing*, ICIP '04, 2004 International Conference on Image Processing, vol. 5, 3443 – 3446.

[18] Casia2.0 Dataset. URL: https://ieee-dataport.org/open-access/modified-casia

[19] Photoshop Battle Dataset. URL: https://github.com/dbisUnibas/PS-Battles

[20] Hamming, Richard W. (1950) "Error detecting and error correcting codes." *The Bell system technical journal,* **29.2**: 147-160.

[21] Du, Ling, Anthony TS Ho, and Runmin Cong. (2020) "Perceptual hashing for image authentication: A survey." *Signal Processing: Image Communication*, **81**: 115713.

[22] Realistic Tampering Dataset (RTD). URL: http://kt.agh.edu.pl/~korus/downloads/dataset-realistic-tampering/

[23] McGill Calibrated Color Image Database. URL: http://tabby.vision.mcgill.ca/html/browsedownload.html

[24] RAISE: URL: http://loki.disi.unitn.it/RAISE/

[25] Lamdan, Yehezkel, and Haim J. Wolfson. (1988) "Geometric hashing: A general and efficient model-based recognition scheme." *1988 Second International Conference on Computer Vision*, IEEE Computer Society.

[26] Lv, Xudong, and Z. Jane Wang. (2012) "Perceptual image hashing based on shape contexts and local feature points." *IEEE Transactions on Information Forensics and Security*, **7.3**: 1081-1093.

[27] Du, Ling, Anthony TS Ho, and Runmin Cong. (2020) "Perceptual hashing for image authentication: A survey." *Signal Processing: Image Communication*, **81**: 115713.