Evaluating Perceptual Hashing Algorithms in Detecting Image Manipulation Over Social Media Platforms

Article · June 2022

CITATIONS
6

READS
106

3 authors, including:

Mohammed Alkhowaiter
University of Central Florida
9 PUBLICATIONS **58** CITATIONS

SEE PROFILE

Evaluating Perceptual Hashing Algorithms in Detecting Image Manipulation Over Social Media Platforms

Mohammed Alkhowaiter
Electrical and Computer Engineering
University of Central Florida
Orlando, United States
mok11@knights.ucf.edu

Khalid Almubarak
College of Computer Engineering and Science
Prince Sattam University
Al-Kharj, Saudi Arabia
k.almubarak@psau.edu.sa

Cliff Zou
Department of Computer Science
University of Central Florida
Orlando, United States
changchun.zou@ucf.edu

***Abstract*—Perceptual hash is a fingerprint of features of multimedia content. Compared with crypto hash, perceptual hash shows many advantages when defending against image-based fake news attacks in terms of detecting deliberate image manipulation while still tolerating normal format or resolution changes conducted on user-uploaded images by content-hosting providers such as social media platforms. Previous research into perceptual hash has studied general image manipulation without considering legitimate image transformation by social media platforms. This paper evaluates and analyzes six state-of-the-art perceptual hash algorithms for detecting image manipulation over two major social media platforms: Facebook and Twitter. Our real-world image evaluation shows differences in the two platforms' image processing and how the six algorithms perform in detecting image manipulation over these platforms. We also present a new approach to finding the optimal detection threshold for each perceptual hash algorithm in distinguishing the platform's standard image processing from deliberate image manipulation.**

***Index Terms*—Perceptual hashing, Computer security, Social media, Fake news, Digital forensics**

I. Introduction

Social media platforms—Facebook, Twitter, etc.—speedup the spread of information over the well-connected cyber world. However, at the same time these platforms help to forge misinformation that can quickly reach a massive number of people. Propagation of fake information and news can lead to deception, emotional distress, and influenced public opinions and actions. An investigation into the truth of news on Twitter from 2006 to 2017 showed that falsehood diffuses faster and deeper than truth [1]. The risk of misinformation increases during great events. A study on fake images on Twitter during Hurricane Sandy (2012) [2] showed that around 90 percentage of retweets were from tweets of fake images. These fake images not only mislead users but also can contain malicious URLs. Figure 1 shows an example of a fake image that spread on social media for President Reagan. The fake image (b) is a real-life sample that was collected by [3].

Nevertheless, most social media platform users are not aware of the risk of re-sharing information from unknown sources. Hence, it is impossible to prevent the spread of disinformation without an automation technique, and a solution that decreases Internet misinformation is urgently needed. Today, most platforms have taken steps to reduce misinformation by verifying their user accounts and adding colored verified badges next to authentic accounts. For instance, during the US 2020 presidential election and COVID-19, social platforms [4], [5] labeled the misinformation posts.

(a) (b)

Fig. 1. (a) President Reagan addresses the nation on his tax legislation. (b) Faked image of President Reagan pointing on a nuclear explosion.

Many researchers have studied image similarity measuring systems over past years, and they have introduced them for different purposes such as image duplication, image retrieval, image forgery detection, digital forensics, and image search engines such as TinEye [6]. These systems cannot rely on the traditional cryptographic hashing because a single bit change in an image will void its crypto hash. On the other hand, perceptual hashing can tolerate these effects and other processing such as compression, scaling, blurring, rotation, etc. It generates a fingerprint of an image by analyzing and extracting features of the image that can be invariant under various attacks. These features after that are taken to finalize the hash value. This value is compared with the tested image hash value to decide whether the tested image is similar, tampered with, or different.

Social media platforms automatically alter images upon sharing for many reasons i.e. images are re-scaled and compressed to save room on the servers, and each platform shares its preferences of image sizes [7]. This means that upon sharing your image, the social media platform will resize it

978-1-6654-9952-1/22/$31.00 ©2022 IEEE

to fit its preference dimension. For instance, an image of the size 3000x2000 pixels will be down-scaled by Facebook into 1875x1250, by Instagram into 1080x720, and by Twitter into 680x453. This scaling diversity is one of the image attacks that distinguished systems suffer from while authenticating the images.

Many academic researchers [8], [9] work on image authentication using a perceptual hashing approach and reached a high value of robustness in preventing one or more image attacks. In the survey paper [10], Du et al. classified the images attacks into two branches. First, content-preserving manipulations that do not change an image's content, such as compression, brightness reduction, and scaling. Second, content-changing manipulations that change an image's content i.e., removing image objects (persons, objects, etc.), moving image elements, changing their positions, adding new objects, etc. Therefore, many researches set up benchmarks that test the robustness of perceptual hashing models against these two attacks such as CASIA [11], USC-SIPI [12], PS-Battles [3], IMD2020 [13]. Each dataset was generated with a different scale of alteration and purpose, and none of these datasets were generated concerning tracing the impact of social platforms on images.

Hence, it is not immediately apparent to which extent a perceptual hashing algorithm effective when images receive unknown attacks with many different social media platforms. In this paper, we explore a hypothesis of whether perceptual hashing algorithms can effectively authenticate images on popular social media platforms, despite the normal image processing conducted by these social media platforms on user-uploaded images. We create a dataset and evaluate state-of-art perceptual hashing algorithms over two popular social media platforms: Facebook and Twitter. We choose these two platforms for study because other platforms put tight restrictions on automatic image uploading/downloading, which prevents us from conducting large-scale image testing and experiments. For example, Instagram needs to upload and download the individual image manually, while Facebook, on the other hand, allows us to share a group of 100 images at one time. Twitter API allows developers to post 100 images per hour automatically.

This evaluation will re-implement six algorithms [8], [9], [14]–[17], and pass a set of images through these systems to check the robustness of their authentication based on a metric algorithm we introduce in this paper. The testbed of this evaluation will use real images from the platforms that we generated. This assessment will help develop an effective image authentication platform for social media images in future work.

In summary, the contributions of this paper are:

- Introduce a new evaluation dataset for image authentication over social media platforms named as (SMPI[1]).
- Present and apply the state-of-art perceptual hash algorithms on social media platforms' user-uploaded images for evaluation.
- Evaluate and compare the performance of six state-of-art perceptual hash algorithms on two major social media platforms for the first time.
- Present a new approach to finding the optimal detection threshold for any perceptual hash algorithm to detect image manipulation on social media platforms.

[1]https://github.com/mohammedkw11/SMPI

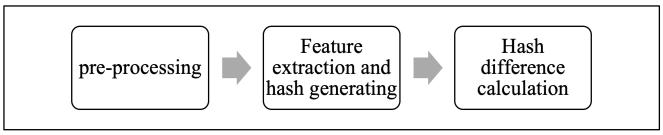

Fig. 2. Stages of perceptual hash image authentication.

The structure of the paper is as follows: section II discusses our methodologies in implementing perceptual hashing algorithms. Next section III, we discuss our proposed evaluating method. In section IV, we show the experimental results using different metrics. Finally, in section V, we provide a summary of our work and conclude with future enhancement in image authentication.

II. METHODOLOGIES

Many researchers developed image hashing systems that defend against one or more image attacks. Different approaches and algorithms are applied in their systems to reach the best image hashing, i.e., robust against content-preserving attacks. [10] classified the techniques of image hashing based on five approaches proposed in the publications: (1) Invariant feature transform methods that extract image features from transform domains and then make use of the coefficients to create the hash; (2) Local feature points such as corners, edges, salient regions, etc.; (3) Dimension reduction method where robust features are extracted from embedding the low-level features of the high dimensional space into a lower dimension, (4) Statistical feature-based approach where calculation of image statistics, such as histogram and mean, are the feature from the image; and (5) Learning-based method that applies machine learning for image feature extracting and authentication.

Based on reviewing perceptual hash algorithms and their papers, any perceptual hash algorithm can be represented by three stages in image authentication as illustrated in Figure 2, which will be explained in detail below.

A. Preparing the images: (pre-processing) stage

All six approaches pass images through different enhancement lines to normalize the images in the best representation and for robust features. The common enhancement is resizing the image into fixed and small sizes to speed up the operation. Usually, it is resized into square $M \times M$, i.e., 128×128 or 224×224 pixels using the bilinear interpolation method. Another enhancement is converting the color space domain from one form to another, such as from RGB to CIELAB ($L * a * b*$) [17], and $L*$ is only used for feature extraction because it matches the human perception of lightness and is more stable. The grey-scale conversion is desirable for most developments due to its simplicity when dealing with $1D$ instead of $3D$ channels in RGB. Filters sometimes are applied,

such as Gaussian and bilateral filters [9] to remove regular noises.

B. Feature extraction and encoding

Choosing among the best state-of-art perceptual hash algorithm modules [8], [9], [14]–[17], we focus on reviewing and re-implementing these six modules that cover state-of-art algorithms DCT, Wavelet, RPIVD, QFT, SimCLR. Each algorithm is adaptive in an image hashing scheme design to generate the hash value that will be used at the next stage. Their authors select the size of the hash in each algorithm as optimal representation, and respectfully short definitions are provided below.

1) DCT: Discrete Cosine Transform (DCT) [8] is one of the popular algorithms that was well implemented for image compression and hashing in the last two decades and used by [18]–[20]. This method is invariant feature transform-based, i.e., it can represent the image in uniqueness with small data. Basically, DCT operates on a function at a finite number of discrete data points. These data were evaluated in terms of the sum of cosine functions with different frequencies to convert it from the spatial domain to the frequency domain. The hash size is eight vectors of byte-sized integers or 64 bits.

2) Wavelet: the introduction of the DCT led to the development of wavelet coding DWT [15], which also takes a large volume of research. The wavelet is also a frequency-based technique but uses temporal details to overcome the drawbacks of DCT. The hash size is eight vectors of byte-sized integers or 64 bits.

3) Visual Model-Based: In this paper [16], Wang proposed a perceptual image hash method for content authentication. They combine a statistical feature-based approach with visual perception using Watson's visual model theory. Watson's visual model is used in order to preserve sensitive features that are important for humans perceiving image content processing. On the other hand, key-point-based features and image-block-based features are used to generate the intermediate hash by extracting key-point-based features using the input image to SIFT algorithm. To achieve this, the proposed method comprises two main stages: 1) Hash Generation Algorithm and 2) Tampering Detection and Tampering Localization. The module is against different image attacks, including geometric attacks. The hash size is 50 vectors of floating or 1600 bits.

4) RPIVD: Tang [17] designed a robust image hashing using Ring Partition and Invariant Vector Distance (RPIVD) that is considered a statistical feature-based approach. Their module is processed in three stages: (1) preparing the image by bilinear interpolation resizing into $M \times M$, low pass filtering, and converting the color space to CIE $L * a * b*$ in order to take $L*$, (2) partitioning the image into equal rings which they choose 512 rings, (3) applying four statistical measures (mean, variance, skewness, and kurtosis) to each ring. This paper provides robust image hashing against several attacks but most strongly against rotation. The hash size selected for this evaluation is the optimal representation from [16], which is 40 vectors with 11 bits representation of each vector or 440 bits in total.

5) QFT: Yan in the work [9] introduced another perceptual hashing approach that used Quaternion Fourier Transform (QFT) to construct feature hash, and QFMT to construct geometric hash. QFT is considered an invariant feature transform-based method that extracts image features from color and structural information to form a quaternion image to produce the hash. QFT can defend against common image attacks and performs well at detecting and locating various types of attacks. The hash size is 40 vectors of floating or 1280 bits.

6) SimCLR: Chen et al. [14] proposed a self-supervised learning framework using contrastive learning to learn better visual representation. The model consists of three main components. The model uses data augmentation as a critical part for the model to map similar images to proper embedding representation. To improve the learned representation, they introduced fully connected non-linear layers between the representation and contrastive loss. Moreover, large batches and more training steps are essential parts. Moreover, SimCLR uses contrastive loss in which the model tries to maximize positive examples, which are two images augmented from the same image and share similar contents. At the same time, contrastive loss minimizes negative examples, which are the reset of the images in the batch to prevent the model from mapping all the images to a constant value. Therefore, SimCLR naturally fits as a prominent model in image authentication tasks.

C. Hashes difference: Similarity metric

After the generation of perceptual hashing, images can be authenticated based on the different values. There are multiple metrics for perceptual hashing comparison; however, most of the approaches as mentioned earlier follow one of two metrics for measuring: Hamming distance [8], [15] and Euclidean distance [9], [16], [17]. We integrated a Locality Sensitivity Hashing (LSH) [21] and Hamming distance in [14] to work on Neural Hash framework as in [22]. Further details will be provided in the IV section. The threshold T is set by each algorithm to distinguish whether the distance value is less or equal to the threshold value in order to decide whether the images are similar, tampered with, or different. Smaller T is better and more secure, especially for image authentication and collision probability reduction. Therefore, the best system algorithms can authenticate the received image with a small T and give the tampered images a high distance value.

III. PROPOSED IMAGE EVALUATION

Social network platforms, Facebook and Twitter, apply image processing and enhancements such as scaling, compression, brightness reduction, and contrast. Each platform has different image effects upon posting for storage preference and transmission. To analyze the image processing operation by each platform, we upload an image on the platform, and then we download the image again. From these steps, we found out

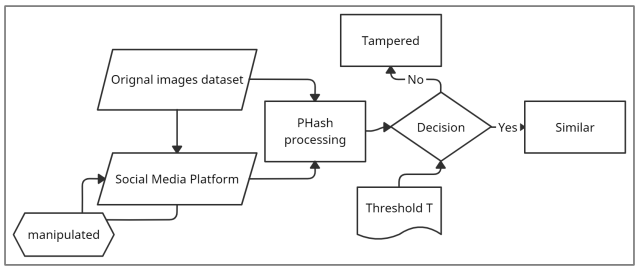

Fig. 3. Methodology used by our proposed authentication platform for social media images.

that for different social media platforms, each downloaded image has different sizing, different scale, different smoothness, and different quality.

The scheme of the evaluation approach that we follow is shown in Figure 3, and six perceptual hash approaches are evaluated for the performance comparison. To prepare for evaluation, the first step is to generate the dataset SMPI. Creating the SMPI dataset consists of four stages. First, we randomly select 6497 images from Holopix50k [23] dataset that was originally collected from users of the Holopix mobile social platform due to the large and variant scales of the images. Secondly, each image is first shared on Facebook and Twitter and then downloaded. Third, the downloaded image is manipulated randomly by one of these attacks: copy-move, splicing, or removal. Finally, the manipulated image is shared again to each social media platform and then downloaded again. After this four-stage operation, we have three copies of each image: the original image (from the first step), the shared image of the original (after the second step), and the altered/shared image (after the fourth step). In total, we collect over 19K images, where each category consisting of 6497 samples.

The three image manipulation attacks mentioned above are further explained in the following. *Splicing* is designed to cut a random part of a different image and paste it randomly onto the target image. *Copy-move* copies a part of an image randomly and pasts it on a different location randomly of the same image. Finally, *removal* selects a part of the image randomly and applies a 5x5 kernel simple blurring filter 50 times on the same location. The dimension of all three types of alterations is randomly chosen where the length and width have to be larger than 4x4 pixels and smaller than the size of the entire image that receives the alteration. Figure 4 represents samples of the three categories of images in our dataset and of the three different image alterations.

Our assumption for the authentication system is that we have an authentication platform for social media images. The first publisher can post and generate the original perceptual hash of the image before posting it to social platforms. The platform's users could check the validation of the image by checking the image with the authentication platform by posting the image there. The outputs for the checking will represent a message of authentic or tampered. Dissimilar images for this evaluation are ignored because all the authentication

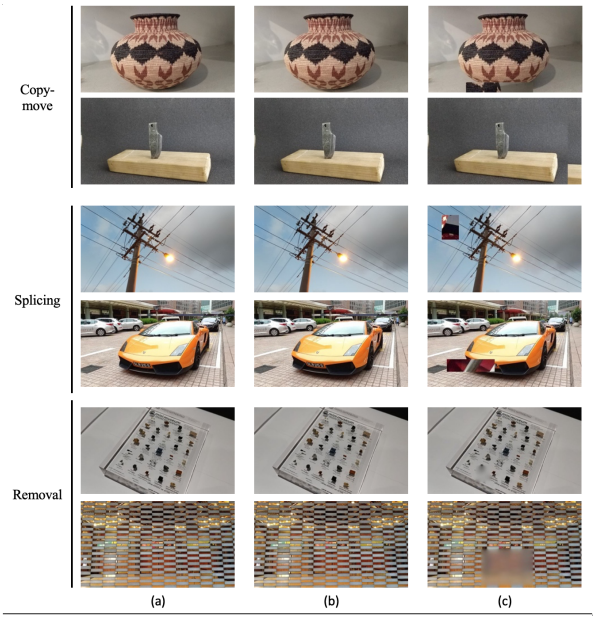

Fig. 4. Sample of images in the dataset for evaluation: (a) original images; (b) images posted and then downloaded as is; (c) images tampered, posted, and then downloaded.

algorithms are simply successful in the detection.

The second step of the approach is preparing the algorithms to generate the perceptual hash for images. We choose the six modules based on state-of-art approaches [8], [9], [14]–[17], the most recognized algorithms for the last decade and the technique that the algorithms follow among five approaches that [10] classified. The majority of approaches are reverse-engineered [9], [14], [16], [17] and [8], [15] are provided as open sources [24], [25]. The decision-making is chosen based on the thresholds set by the authors of [8], [9], [15]–[17] which are 0.25, 0.004, 5.0, 200, and 0.8, respectively, except SimCLR model that has never been used in image authentication tasks. Therefore, we selected T = 0.04 based on the best result we obtained, as it is described in part B of the IV section.

Afterward, the evaluation of the proposed methods was measured by four metrics approaches false-negative rate (FNR) and false-positive rate (FPR), gap difference $diff$, accuracy, and average time processing.

1) FNR-FPR: Equation 1 and Equation 2 represent the false-negative rate (FNR) and false positive rate (FPR) from [26] respectfully. FN is calculated by dividing FN over the summation of FN and true positive (TP). In contrast, FP is calculated by dividing FP over the summation of FP and true negative (TN). The lower rate of either FN or FP indicates the better robustness of the authentication capability of a perceptual hash algorithm.

$$FNR = FN/(FN + TP) \tag{1}$$

$$FPR = FP/(FP + TN) \tag{2}$$

$$PTA_{similar} = \frac{\text{no. of true positive results}}{\text{no. of similar image tests}} \tag{3}$$

2) $diff$: the division of average altered images hashing over average similar images hashing as shown by Equation 4:

$$diff = avg(ph_{alter})/avg(ph_{similar}) \quad (4)$$

The perceptual hash distance, $ph_{similar}$ in Equation 4 refers to the Hamming distance or Euclidean distance value between the original image (Figure 4a) and social media downloaded and unaltered image (Figure 4b). The smaller value of $ph_{similar}$ is better, which means the perceptual hash algorithm can detect that the downloaded image from the social media platform is unaltered from the original image beside the image processing that social media platforms add upon posting such as compression and resizing. On the other hand, the perceptual hash distance, ph_{alter} refers to the Hamming distance or Euclidean distance value between the original image (Figure 4a) and social media downloaded and altered image (Figure 4c). The larger value of ph_{alter} is better, which means the perceptual hash algorithm can detect an altered image used in the social media platform. The avg in the equation refers to the average of all images' values in similar and alteration tests.

Furthermore, the $diff$ value calculated in Equation 4 reflects how well the perceptual hash algorithm can detect image alteration when an image is used on a social media platform. A good perceptual hash algorithm should have a larger $diff$ value for better decision-making in detecting image alteration in social media platforms. On the other hand, a smaller $diff$ value will make it harder to choose the detection threshold, hence, increasing the false-negative and false-positive rates.

3) Accuracy: Equation 5 calculates the accuracy of each algorithm based on the values of TP, TN, FN, and FP.

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (5)$$

4) average processing time: the average time that each run of generating perceptual hashing takes. This metric is affected by the processor resource and the environment we use during evaluation. The processor is a 2.7 GHz Dual-Core Intel Core i5, the memory is 8 GB 1867 MHz DDR3, and the operating system is macOS Monterey.

IV. EXPERIMENTAL RESULTS

This section uses our proposed evaluation scheme in the previous section to evaluate those seven perceptual hash algorithms. We implemented and tested the scheme using Python (3.8.5) and some open source libraries such as OpenCV [25] to build [8] and [15]. In addition, we implemented a Neural Hash scheme which consists of a neural network that extracts image features and maps them into a vector with a fixed length. Then, the vector is fed to LSH, which maps each vector to a specific bucket where similar images have similar vectors mapped to the same bucket. The last step is to convert these matrix-vector buckets to binary hash using hamming distance. We used a SimCLR pre-trained on Imagenet [27] as in [14] to map the images into a vector of 128 length. Then this vector is mapped to 1024 bits. All our implementation for SimCLR-LSH used Pytorch framework [28]. For the remaining algorithms, we re-implemented them as they are described in these publications [9], [16], [17].

A. Evaluation based on algorithms' original decision thresholds set by their authors

Following the scheme in Figure 3, we generated 155,928 tests equally distributed by the six modules. These tests are calculated by using the original images from the SMPI dataset. Each image calculated twice: one to calculate the $ph_{similar}$ (Figure 4a Vs Figure 4b) and the second for ph_{alter} (Figure 4a Vs Figure 4c) which creates 12994 tests for each platform and in total of 25988 tests for the two platforms. Afterward, these outputs are summarized and shown at Table I. The table's minimum score (min) means the lowest perceptual hashes value among the 6497 tests at each approach and platform. The average (avg) indicates the average of 6497 tests under each algorithm and platform, and finally, the maximum (max) refers to the highest perceptual hash among the 6497 evaluations at each algorithm and platform.

Table I shows the summary of the perceptual hashing distance of these six algorithm modules in terms of the evaluation of $ph_{similar}$ of images group (a&b) in Figure 4. In addition, the perceptual hashing difference values for the altered images using groups (a&c) in Figure 4 are represented too. The threshold T is directly brought from the original papers presenting those perceptual hash algorithms [8], [9], [15]–[17] except [14] that we generated. For a perceptual hash algorithm, if the perceptual hash value exceeds the threshold, an FNR is generated since the image is supposed to be authentic, but the algorithm detects otherwise. Whereas an FPR is generated when the image is supposed to be unauthentic, but the algorithm decides otherwise. Table II represents the outcomes of FNR and FPR for Facebook and Twitter platforms.

Looking at the average scores on Facebook evaluation at Table I, it shows that all the results of tampered a&c are higher than similar a&b tests, which indicates that these algorithms recognize alteration. The selected T by the authors remains in between on all algorithms except Visual Model-Based (Vsul M-B) [16] that the average on a&b and a&c exceeded the T. Twitter, on the other hand, shows identical results on measuring (a&b) on all evaluations. These results show that the Twitter platform applies manipulation upon sharing in small factors value if the image meets the recommended dimension [7], which our selected dataset follows. For example, we upload an image to the Twitter platform with a size of 233KB and a dimension of 1280×720. After downloading the image back, we receive a new size, 230KB at the same dimension. Another trial was done that exceeded the recommended dimension on an image with a size 262KB and a dimension of 1850×1233. We received a new alteration with 82KB in size and 680×453 in dimension. Hence, it is probable Twitter applies compression and re-scaling with different factors based on the image size and dimension.

TABLE I
PERCEPTUAL HASHING SIMILARITY SCORE BETWEEN THE ORIGINAL IMAGES (FIGURE 4A), THE POSTED IMAGES (FIGURE 4B), AND ALTERED IMAGES (FIGURE 4C) ON SOCIAL MEDIA PLATFORMS.

Algorithm	T	**Facebook**						**Twitter**					
		min		***avg***		***max***		***min***		***avg***		***max***	
		a&b	a&c	a&b	a&c	a&b	a&c	a&b	a&c	a&b	a&c	a&b	a&c
DCT [8]	0.25	0.0	0.0	0.04	0.28	0.5	1	0.0	0.0	0.0	0.27	0.0	1
Wavelet [15]	0.004	0.0	0.0	0.0004	0.02	0.03	0.5	0.0	0.0	0.0	0.025	0.0	0.68
Vsul M-B [16]	5	0.83	1.08	6.14	7.52	13.40	13.85	0	1.0	0	6.99	0.0	14.48
RPIVD [17]	200	1	1	5.46	117.63	85.0	989.0	0.0	1.0	0.0	118.08	0.0	1094.0
QFT [9]	0.8	0.0007	0.002	0.008	0.20	0.16	2.89	0.0	0.002	0.0	0.211	0.0	2.84
SimCLR+LSH [14]	0.04	0.001	0.009	0.035	0.11	0.16	0.47	0.0	0.005	0.0	0.098	0.0	0.55

Figure 5 represents the percentage values of different image hashes a&c from Table I to similar image hashes a&b using Equation 4. If the gap percentage is high, it represents the algorithm's robustness in detection alteration. Otherwise, it is vulnerable for the algorithms to minimize the comparison score for similar tests and maximize the score for the altered tests. From the chart, the gap is suitable enough on the four algorithms: DCT , Wavelet , QFT, and SimCLR, regarding the Facebook metric with values of 7, 50, 25, and 3.14, respectively. The Visual Based algorithm kept the different gap on Facebook at the NT with a value of 6.23, which increases the model's sensitivity to distinguish between similarity and alteration. The other model, RPIVD went too low, even below the suggested T by the author. However, at the same time, NT shows a high gap, 25, and the highest accuracy amongst others. In contrast, on Twitter, as the similarity scores shows zeros at Table I, the gap difference is significant where we solve the division by zero by adding a small fraction that a close to zero.

B. Finding optimal decision thresholds for social media platforms

The six perceptual hash algorithms under study were well designed by their authors with carefully-set decision thresholds in detecting image manipulation based on specific datasets [11], [12]. However, their authors determined the decision thresholds based on general modification/manipulation operation on images. In the specific social media platform scenario under study in this paper, we want a perceptual hash algorithm to detect any deliberate image manipulation while at the same time not treating the image resizing/compression operation by the social media platform as image manipulation. Therefore, for each perceptual hash algorithm, there should exist a better decision threshold for the social media platform environment. In this section, we present the method for finding the optimal decision threshold and then verify that the performance will be better in comparison to those original decision thresholds as it appears at Table III.

We recalculate the threshold for each algorithm based on the perceptual hashes outputs through the conducted tests on Table I. Based on [9], the optimal threshold can be determined using the probability of true authentication (PTA), which essentially calculates the probability distribution of $ph_{similar}$ and ph_{alter} results using TP and TN decisions of both social

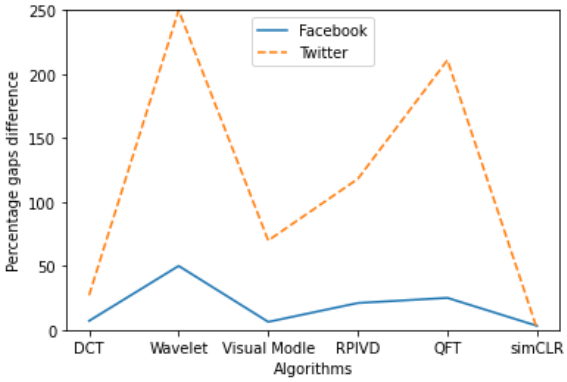

Fig. 5. Perceptual hash gaps (*diff* defined in Equation 1) between similar images and tampered images.

media platforms as it appears in Equation 6 and Equation 7. Based on the threshold value, which begins with zero, the probability of true authentication for similar images, i.e., $PTA_{similar}$, is represented by the blue line at Figure 6 should be zero since all the images recognized as tampered; therefore, tampered images probability, i.e., $PTA_{tampered}$, shown by the orange dashed line should be one.

$$PTA_{similar} = \frac{\text{no. of TP results}}{\text{no. of similar image tests}} \tag{6}$$

$$PTA_{tampered} = \frac{\text{no. of TN results}}{\text{no. of tampered image tests}} \tag{7}$$

With each increase in the threshold value, the rate of true authentication for both states will change until they intersect at a certain point, as shown in Figure 6, e.g., RPIVD with a red dot and value of 8. This point is selected to represent the new threshold (NT) where it balances the performance of FPR and FNR and gives the best accuracy. Table II shows the comparison of FNR and FPR rates for original thresholds (OT) and NT for all six algorithms. The outcomes of NT show significant enhancements on Facebook and Twitter platforms with minor changes at Wavelet and Vsul M-B algorithms. Accuracy comparisons at Table III represents remarkable changes too.

We conducted an experiment on SimCLR for Facebook images that shows a promising approach. We used a Pretrained SimCLR as feature extraction. Then, We replaced both LSH and hamming distance with a Normalized Euclidean Distance as shown in Equation 8. Also, we fed the output of the learned

TABLE II
FNR AND FPR COMPARISON ON SCALE OF ORIGINAL THRESHOLDS (OT) AND NEW THRESHOLDS (NT) FOR FACEBOOK AND TWITTER.

Algorithm	Facebook						Twitter					
	Original Thresholds OT			New Thresholds NT			Original Thresholds OT			New Thresholds OT		
	OT	*FNR*	*FPR*	*NT*	*FNR*	*FPR*	*OT*	*FNR*	*FPR*	*NT*	*FNR*	*FPR*
DCT [8]	0.25	0.009	0.32	0.12	0.29	0.27	0.25	0.0	0.33	0.12	0.0	0.225
Wavelet [15]	0.004	0.013	0.63	0.004	0.013	0.63	0.004	0.0	0.64	0.004	0.0	0.64
Vsul M-B [16]	5	0.75	0.05	6.46	0.66	0.17	5	0.0	0.19	6.46	0.0	0.37
RPIVD [17]	200	0.0	0.79	8	0.099	0.098	200	0.0	0.79	8	0.0	0.15
QFT [9]	0.8	0.0	0.92	0.013	0.16	0.14	0.8	0.0	0.9	0.013	0.0	0.19
SimCLR+LSH [14]	NA	NA	NA	0.04	0.16	0.17	NA	NA	NA	0.04	0.0	0.26

TABLE III
FACEBOOK AND TWITTER ACCURACY COMPARISON BETWEEN ORIGINAL THRESHOLDS (OT) AND NEW THRESHOLDS (NT).

Algorithm	Facebook				Twitter			
	OT	*Accuracy*	*NT*	*Accuracy*	*OT*	*Accuracy*	*NT*	*Accuracy*
DCT [8]	0.25	75.33%	0.12	71.34	0.25	75.24%	0.12	85.47%
Wavelet [15]	0.004	67.50%	0.004	67.50	0.004	67.65%	0.004	67.65%
Vsul M-B [16]	5	59.42%	6.46	58.42	5	87.71%	6.46	81.21%
RPIVD [17]	200	60.33%	8	90.12	200	60.40%	8	92.06%
QFT [9]	0.8	53.50%	0.013	84.21	0.8	53.67%	0.013	90.48%
SimCLR+LSH [14]	NA	NA	0.04	82.87	NA	NA	0.04	86.70%

representation, which is the output of the final convolution layer to the Normalized Euclidean Distance instead of the output of the fully connected layer as suggested in [14]. The accuracy of the model increased to 90%. However, the output of SimCLR representation is a 4069 length vector where each element of the vector is represented by 32-Floating Point bits. This experiment suggests that SimCLR results have the potential to be improved.

$$d(u,v) = \frac{1}{2}\frac{\sigma^2(u-v)}{\sigma^2(u)+\sigma^2(v)+\epsilon} \tag{8}$$

The distance denoted by d measures the normalized euclidean distance between two images vectors u and v. The first term in the numerator calculates the difference between the two image vectors and their variance afterward. The denominator term refers to the summation between the two vectors' variance and ϵ, which is a small number added for numerical stability. The result of the quotient, then, is multiplied by 1/2.

The average performance analyses are shown at Table IV. It is clear that increasing operation for feature extraction also increases overhead. [8] and [15] were the lightest overall on feature processing whereas [14] time consumption is the highest by a significant time. Therefore, the algorithms of images systems should consider different types of environments instead of using highly advanced resources in the developments pipeline.

V. CONCLUSION AND FUTURE WORK

We have evaluated six approaches in image authentication and perceptual hashing. This evaluation used social media platforms, Facebook and Twitter, as real environments to figure out the robustness and weakness of each of the six approaches. Results show that each platform employs different image processing to the shared images and different processing

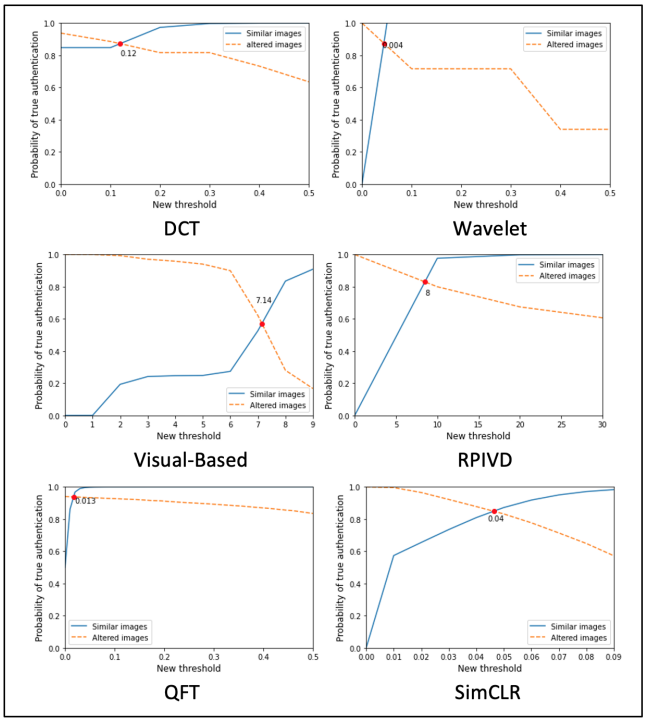

Fig. 6. The new threshold (NT) calculation for each approach.

TABLE IV
AVERAGE PERFORMANCE ANALYSES.

Algorithm	*Processing Time (s)*
DCT [8]	0.023
Wavelet [15]	0.065
Vsul M-B [16]	1.76
RPIVD [17]	0.18
QFT [9]	0.087
SimCLR-LSH [14]	11.76

factors such as compression and resizing. Facebook is the platform that applies the most effects on the image upon sharing, whereas Twitter is the least with zeros distance values on testing similar images on most algorithms. On the other hand, among these approaches, [17] that follows statistical feature-based is the best approach that shows excellent results at all platforms with the highest accuracy. In contrast, [15] represents the worst with a high percentage of FP and lowest accuracy on all platforms.

Image feature preservation plays the primary role in designing a robust authentication system. The six approaches follow different ways of vector preservation and apply one or more operations for that objective, such as RPIVD applies four statistical measures (mean, variance, skewness, and kurtosis). These operations could generate a unique perceptual hash. However, it could increase the performance overhead.

Selecting the optimal threshold for making a decision is challenging. The authors of those six evaluated algorithms determine the threshold based on their targeting image alteration methods and the dataset used in training or evaluation. For instance, most of these algorithms target many types of image attacks. However, at the same time, they focus on one alteration, such as [15] for image compression and [17] for image rotation. A study of image authentication under minor alteration could be an area of interest with high security considering lowering the collision rate and minimizing the threshold T. Perhaps a theory of cryptographic hashing in uniqueness and identical could be targeted in perceptual hashing developments.

The hash value size is a significant parameter in finding a solid and efficient perceptual hash algorithm. Increasing hash size can improve decision accuracy, but at the same time, the hashing computational cost and storage requirement will also increase. The future work on this aspect is to find the suitable trade-off between accuracy and efficiency in image authentication for different applications.

We also intend to investigate more about the machine learning approach. The initial results show that SimCLR with LSH is a promising approach to image authentication. However, there is a place for improvement. We can design a custom contrastive learning that serves the image authentication task to enhance the results. Further, we can conduct more tests on SimCLR model with Normalized Euclidean Distance to validate and improve the accuracy and model performance. Also, we can look into other neural network architectures that may work in the image authentication domain.

REFERENCES

[1] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.

[2] A. Gupta, H. Lamba *et al.*, "Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy," in *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume*. International World Wide Web Conferences Steering Committee / ACM, 2013, pp. 729–736.

[3] S. Heller, L. Rossetto, and H. Schuldt, "The PS-Battles Dataset – an Image Collection for Image Manipulation Detection," *CoRR*, vol. abs/1804.04866, 2018.

[4] Meta. (2022) Facebook's Third-Party Fact-Checking Program. [Online]. Available: https://www.facebook.com/journalismproject/programs/third-party-fact-checking

[5] Y. Roth and N. Pickles". (2020) "updating our approach to misleading information". [Online]. Available: "https://blog.twitter.com/en_us/topics/product/2020/updating-our-approach-to-misleading-information"

[6] T. website". (2022) "what is tineye". [Online]. Available: "https://tineye.com"

[7] K. Olafson and T. Tran". (2022) "social media image sizes 2021: Cheat sheet for every network". [Online]. Available: "https://blog.hootsuite.com/social-media-image-sizes-guide"

[8] Y. Tang, Zhenjun *et al.*, "Robust image hashing with dominant dct coefficients," *Optik - International Journal for Light and Electron Optics*, vol. 125, no. 18, pp. 5102–5107, 2014.

[9] C.-P. Yan, C.-M. Pun, and X.-C. Yuan, "Quaternion-based image hashing for adaptive tampering localization," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2664–2677, 2016.

[10] L. Du, A. T. S. Ho, and R. Cong, "Perceptual hashing for image authentication: A survey," *Signal Process. Image Commun.*, vol. 81, 2020.

[11] Y. Zheng". (2021) "casia dataset". [Online]. Available: "https://ieee-dataport.org/open-access/modified-casia#files"

[12] U. University". (1977) "usc-sipi image database". [Online]. Available: "https://sipi.usc.edu/database/"

[13] B. Novozamsky, Adam *et al.*, "Imd2020: A large-scale annotated dataset tailored for detecting manipulated images," in *2020 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, March 2020, pp. 71–80.

[14] M. N. "T. Chen, S. Kornblith and G. Hinton", ""a simple framework for contrastive learning of visual representations"," *"In International conference on machine learning"*, vol. "10", pp. "1597–1607", "2020".

[15] M. H. J. R. Venkatesan, S.-M. Koon and P. Moulin, "Robust image hashing," *Proceedings 2000 International Conference on Image Processing*, vol. 3, pp. 664–666, 2000.

[16] K. Wang, Xiaofeng *et al.*, "A visual model-based perceptual image hash for content authentication," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1336–1349, 2015.

[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, X. Chen, D. Ye, J. Wang, and X. Sun, "A novel image hashing scheme with perceptual robustness using block truncation coding," *Information Sciences*, vol. 361, pp. 84–99, 2016.

[19] Z. Tang, H. Lao, X. Zhang, and K. Liu, "Robust image hashing via dct and lle," *Computers & Security*, vol. 62, pp. 133–148, 2016.

[20] C. Qin, X. Chen, J. Dong, and X. Zhang, "Perceptual image hashing with selective sampling for salient structure features," *Displays*, vol. 45, pp. 26–37, 2016.

[21] K. M. I. Omid Jafari *et al.*, "A survey on locality sensitive hashing algorithms and their applications," *CoRR*, vol. abs/2102.08942, 2021.

[22] A. Inc.". (2021) "csam detection". [Online]. Available: "https://www.apple.com/child-safety/pdf/CSAM_Detection_Technical_Summary.pdf"

[23] K. Hua, Yiwen *et al.*, "Holopix50k: A large-scale in-the-wild stereo image dataset," *arXiv preprint arXiv:2003.11172*, 2020.

[24] C. "Zauner. (2010) "phash.org: Home of phash, the open source perceptual hash library". [Online]. Available: "https://www.phash.org"

[25] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[26] Z. Zhu, Wen *et al.*, "Sensitivity, specificity, accuracy, associated confidence interval and roc analysis with practical sas implementations," *NESUG proceedings: health care and life sciences, Baltimore, Maryland*, vol. 19, p. 67, 2010.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[28] A. Paszke, S. Gross *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.