symmetry

Article

Intelligent Visual Similarity-Based Phishing Websites Detection

Jiann-Liang Chen, Yi-Wei Ma * and Kuan-Lung Huang

Department of Electrical Engineering, National Taiwan University of Science and Technology,
Taipei City 106335, Taiwan; lchen@mail.ntust.edu.tw (J.-L.C.); et82122@gmail.com (K.-L.H.)
* Correspondence: ywma@mail.ntust.edu.tw

Received: 24 August 2020; Accepted: 6 October 2020; Published: 14 October 2020

Abstract: This work proposes an intelligent visual technique for detecting phishing websites. The phishing websites are classified into three categories: very similar, local similar, and non-imitating. For cases of 'very similar', this study uses the wavelet Hashing (wHash) mechanism with a color histogram to evaluate the similarity. In cases of 'local similarity', this study uses the Scale-Invariant Feature Transform (SIFT) technique to evaluate the similarity. This work concerns 'very similar' and 'local similar' cases to detect phishing websites. The results of the experiments reveal that the wHash mechanism with a color histogram is more accurate than the currently used perceptual Hashing (pHash) mechanism. The accuracies of SIFT technique are 97.93%, 98.61%, and 99.95% related to Microsoft, Dropbox, and Bank of America data, respectively.

Keywords: phishing website; wavelet Hashing (wHash); Scale-Invariant Feature Transform (SIFT); visualization; image similarity

1. Introduction

With the rapid development of information technology, many Internet applications are attracting increasing attention and becoming part of everyday life. Therefore, it brings all kinds of online scams. Phishing is a scam that uses social engineering to induce users to disclose personal and private information [1]. A phishing website imitates the website of a well-known brand to obtain the trust of the user, who is therefore willing to provide sensitive information, such as an account name, password, financial information, and more. As of 2017, according to the Anti-Phishing Working Group (APWG) report, the average number of phishing websites that can be detected per month is as high as 55,232 [2]. Hence, this work proposes an approach to detecting phishing websites that is based on visual similarity. The rest of the paper is organized as follows. Section 2 presents the background of this study. Section 3 introduces the proposed phishing detection mechanism. Section 4 presents the performance analysis of this study. Finally, conclusions are drawn in Section 5.

2. Related Studies

To detect phishing websites effectively, many studies have used intelligent approaches with features analysis. Jain et al. proposed a two-level authentication approach for declaring a webpage as phishing [3]. Tan et al. proposed a phishing detection technique [4]. Bahnsen et al. proposed use of URLs to classify phishing attacks, the authors claim that recurrent neural networks provide a good accuracy rate than other approaches [5]. Buber et al. proposed use of Natural Language Processing (NLP) to extract features for detecting phishing attacks [6]. Hu et al. proposed use of machine learning with online credibility [7]. Zuhair et al. proposed a hybrid feature for detecting phishing attacks, the authors claim that they use 58 hybrid features to analyze the prediction susceptibility [8]. The features include HTTPs-based, TLD (Top-Level Domain)-based, WHOIS-based, and content-based features.

Feature-based mechanisms can perform well, but the features will gradually become less effective over time or the mechanism will induce phishing attackers to generate means of circumvention [9].

Many studies have addressed this problem. Mao et al. proposed using the CSS (Cascading Style Sheets) webpage style code to test the similarity of two websites, but some phishing websites use page screenshots as a background to avoid textual verification, making this method ineffective [10]. White et al. proposed a perceptual Hashing (pHash) technique for comparing webpages [11]. This method can be effective but only in cases of very high similarity. Rao et al. proposed Speeded Up Robust Features (SURF) for comparing webpages [12]. This method is effective with very similar and partially similar pages, but cannot identify small similarities. Asudeh proposed eight box-shaped filters to detect a logo image and used the Histogram of Gradient (HoG) method to solve problems of noise and rotation, but the method cannot identify phishing sites that use a page screenshot as its background [13]. Wang et al. proposed the SIFT technique with WHOIS feature to detect the form page [14].

3. Materials and Methods

Some phishing websites use screenshots of other websites as the background, adding only input tags. Some features, like logos, are only partially similar. Since a phishing website may use the same home page at various URLs to scam, the phishing webpage is not entirely like the copied webpage. Phishing pages have three types of appearance: very similar to, locally similar to, and non-imitating of a real website. Highly similar pages are detected using a Locality-Sensitive Hashing (LSH) technique with image and color histograms. Locally similar pages are detected using the Scale-Invariant Feature Transform (SIFT) technique.

Although SIFT performs well, its form detection may fail. Some phishing websites have two stages. First, the user selects the account to be logged into, and then is sent to another page with a form. Phishing pages of the cloud service Dropbox and Docusign often use the approach.

3.1. Cases of Very High Similarity—'Very Similar' Cases

In cases of very high similarity, the screenshots of whole webpages must be compared. To improve the rate of comparison without loss of accuracy, an LSH method wavelet Hashing (wHash) mechanism with a color histogram is proposed herein. Data matching often takes a long time, especially when high-dimensional data are involved, as with images. If the amount of data is large, then the time complexity is large [15,16]. The pixels in an image are represented in RGB color space. RGB are the three primary colors of light—red, green and blue, respectively. The intensity of these three colors is given by one of 256 values. The values for three colors from 0 to 255 yield a RGB color histogram. Accordingly, the color characteristics of an image can be represented and used to compare the colors of the images. Directly comparing the colors in two images produces errors. The overall color similarity of two images may be close without those images looking similar. Therefore, the images must be split to improve the accuracy of the color comparison based on local color similarities [17].

3.2. Cases of Local Similarity—'Local Similar' Cases

The best way to avoid the problem of CSS solution is to treat the logo as a feature for detection [10]. The logo is a symbolic visual element, which uses graphics or text to represent the brand of a company. References [13,14] used a logo comparison method, but it can cause some problems. Since a page may have many pictures, a phishing website may use screenshots to form its background and avoid detection. This study develops the SIFT image object detection technique to detect whether a logo has been extracted from the screenshot of a whole webpage.

A. SIFT Technique

A SIFT technique is based on the feature of local appearance at a point of interest on an object. Such features of an image are independent of its size and rotation. They are also robust against light, noise, and small viewing angle changes. Therefore, they are highly visible and more easily extract data. A large feature database facilitates the identification of objects with few mismatches. SIFT feature descriptors can be used to detect partial objects with quite high accuracy. Three of SIFT's key points suffice to calculate the position and orientation. The Difference of Gaussiane (DoG) algorithm is used to find these feature points. Similar features can be found in spaces of different scales [18].

B. Feature Descriptor Generator

Create an 8 × 8 window (the unit is a pixel of the scale) that is centering on a key point, calculate the gradient direction and magnitude of each pixel, and then calculate the gradient histogram for eight directions in every group of 4 × 4 pixels [19]. A 128-dimensional vector is thus obtained.

C. Matcher

Feature matching mainly uses the K-NN (K nearest neighbor) algorithm with KD-tree as the key points index to accelerate the matching process [20].

4. Proposed Phishing Detection Mechanism

Figure 1 presents the proposed system architecture. The system has three parts, which are cache, webpage screenshot matcher, and logo finder. The cache is used mainly to record historical data to accelerate the detection process. The webpage screenshot matcher is used to detect 'very similar' cases, and cases of 'local similar' are analyzed using the logo finder.

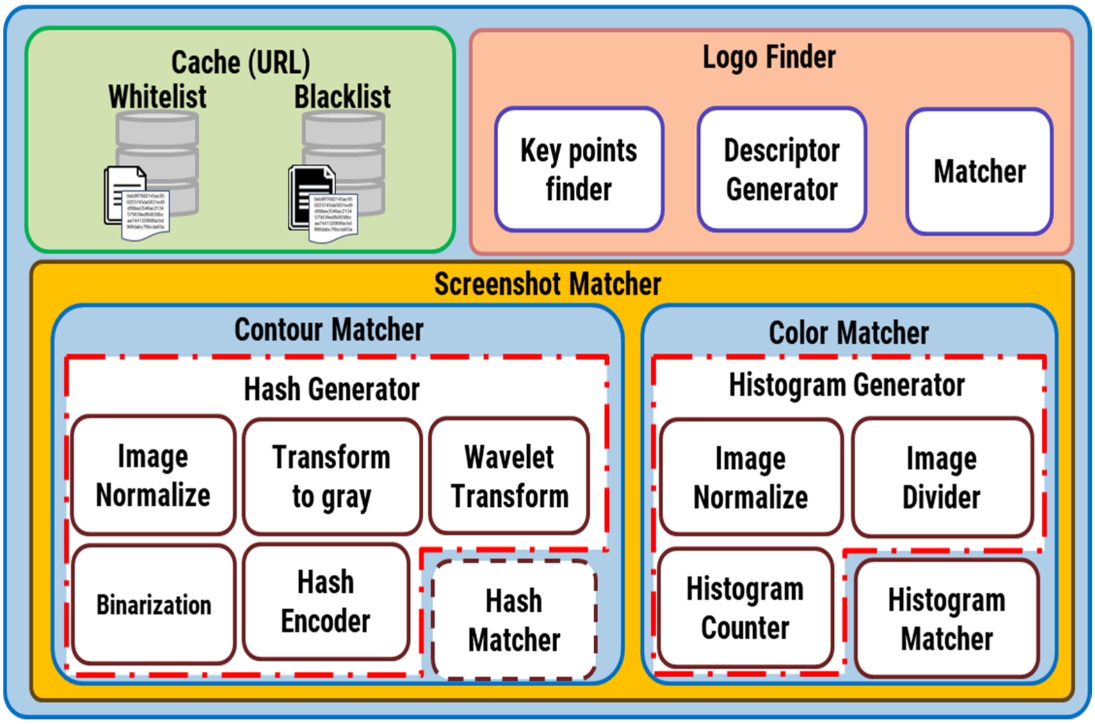

Figure 1. Proposed system architecture.

4.1. *Detection Mechanism*

The proposed mechanism comprises an offline phase and an online phase.

4.1.1. Offline Phase

The image is converted into a hash, and its contours are compared with those of the target webpage; the contour similarity is thus calculated. The color histogram and color similarity are also calculated.

The key points in the screenshot image are detected in the logo finder. The logo finder obtains feature descriptors at key points, and then tries to match to the target's logo, yielding a matching number. The system has three similarity indices, which are contour similarity, color similarity, and number of matched key points of the logo. Subsequently, the known tags with these three indices are used to calculate the performance. Figure 2 presents the operations of the contour, color matcher, and logo finder operations.

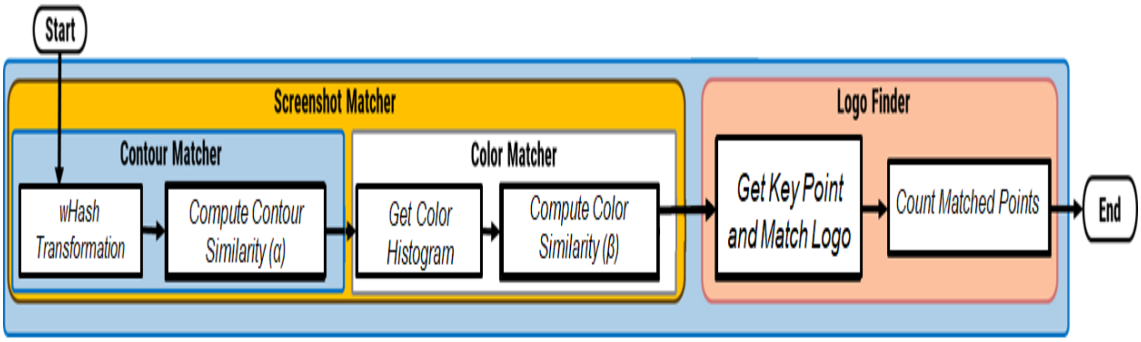

Figure 2. Contour, color matcher, and logo finder operations in offline phase.

4.1.2. Online Phase

The URL of the unknown website is checked at the beginning of the online phase to determine whether it has been detected before. If it has been detected, it is skipped directly. Otherwise, it is entered into the contour matcher to compare its contours. The operations are shown in Figure 3.

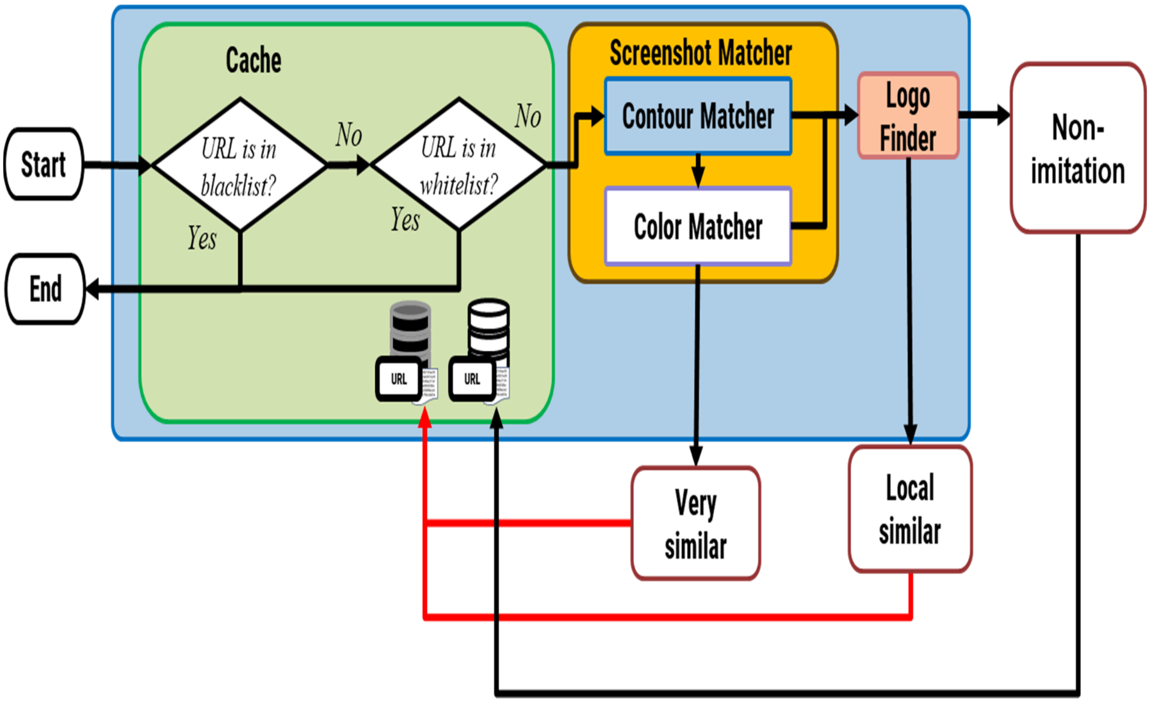

Figure 3. Operations in online phase.

The comparison is performed with the hash value of the target website and the hash value of the blacklist. If the similarity exceeds the predefined threshold, then a color similarity comparison is conducted to ensure that the screenshots are similar. Otherwise, the system will perform a logo comparison. After the contours have been compared, the color histogram is calculated. The color histogram of the unknown website is compared with that of the target and the websites on the blacklist. If the similarity exceeds the predefined threshold, then the URL, hash and color histogram of the unknown website will be saved to the blacklist. Otherwise, the system performs a logo comparison. Figure 4 shown the operations in online phase.

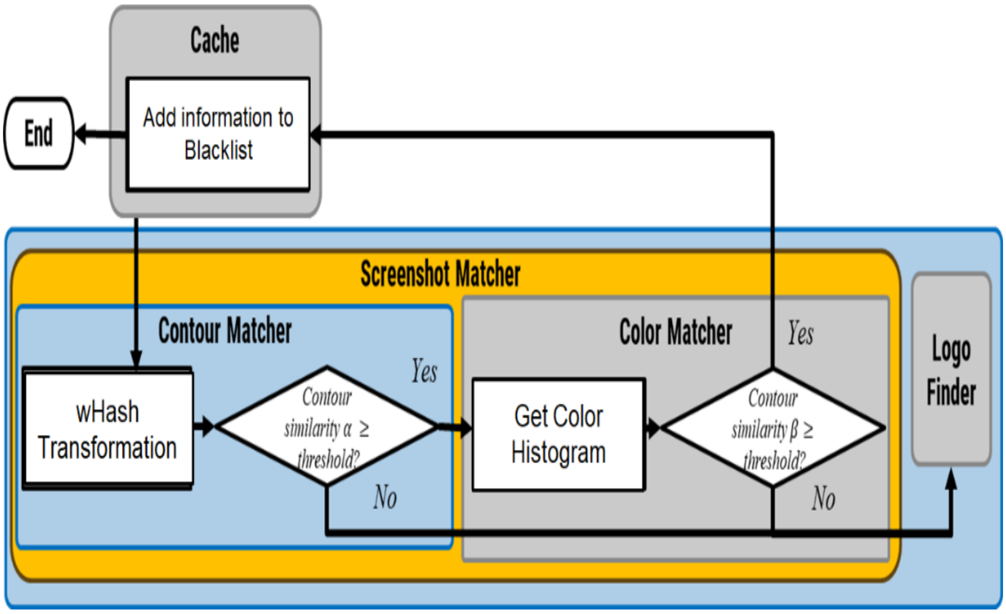

Figure 4. Contour and color matcher operations in online phase.

Figure 5 presents the operations that are performed at the key points of the screenshot of the webpage and the generation of a feature descriptor. If the number of matched key points exceeds the predefined threshold, then the URL, hash and color histogram of the unknown website is saved in the blacklist. Otherwise, the URL, hash and color histogram of the unknown website is saved in the whitelist.

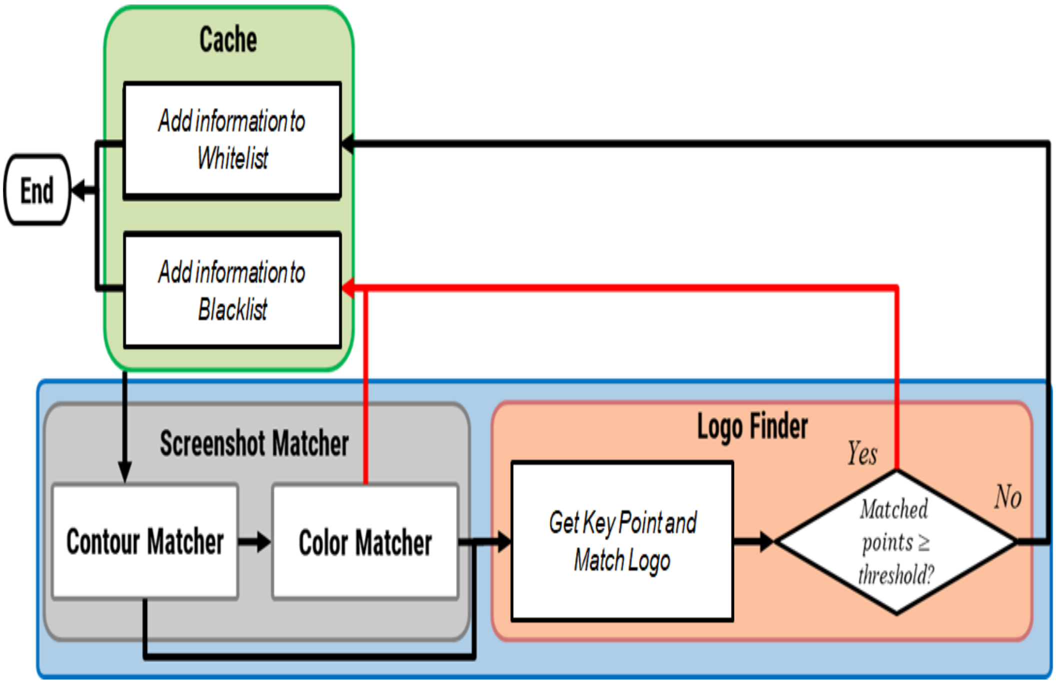
Cache
Add information to Whitelist
Add information to Blacklist
End
Screenshot Matcher
Contour Matcher
Color Matcher
Logo Finder
Get Key Point and Match Logo
Matched points ≥ threshold?
Yes
No

Figure 5. Logo finder in online phase.

4.2. System Modules

Three modules cache, screenshot matcher, and logo finder are presented.

4.2.1. Cache

Cache contains a blacklist and a whitelist. It stores URLs, hashes, webpage screenshot data, and feature descriptors, among other information. A URL can be used to determine whether the webpage has been previously detected, and other data are used to accelerate the detection of the webpage if it has not already been detected.

4.2.2. Screenshot Matcher

The screenshot matcher is mainly used to compare the webpage screenshots. Its two modules are contour matcher and color matcher, as shown in Figure 6.

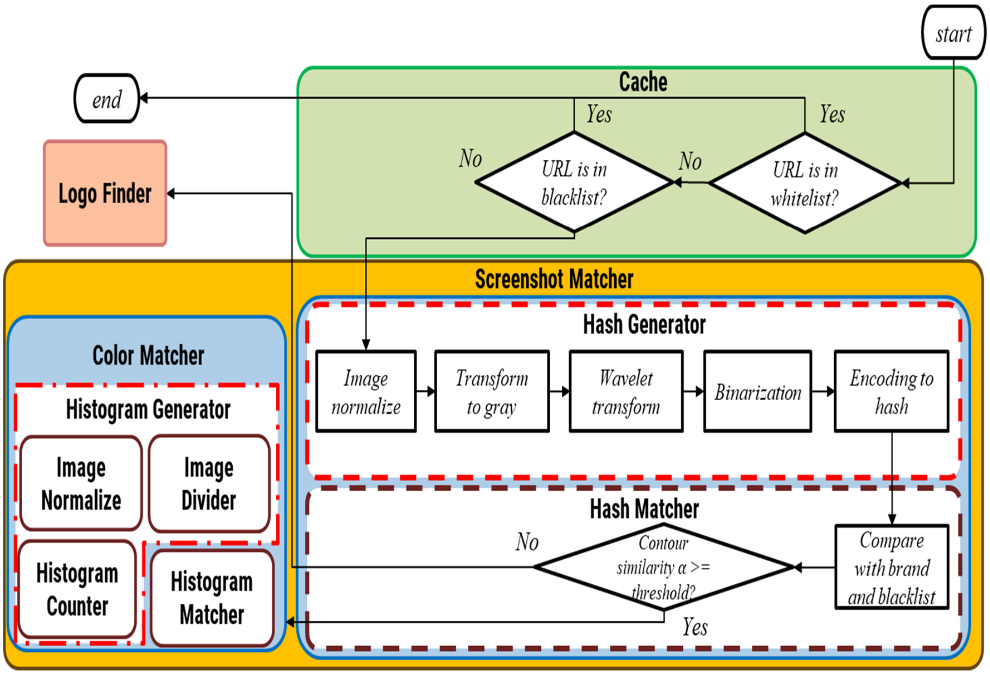

Figure 6. Cache and screenshot matcher.

Contour Matcher Hash Generator

Since one web screenshot is compared with another, the involved data are numerous and high-dimensional, so a hash generator converts image to a hash value to reduce the time complexity. The hash generator module will convert an image to a hash value by wHash mechanism. The process comprises the following five steps:

1. Normalize image: the image is compressed such that one of its sides has a specific length. The image size that is used in the comparison may not be the same length. Thus, it must be converted to ensure that the hash values are the same length.
2. Transform to gray: the image is transformed to grayscale to reduce the amount of information involved.
3. Wavelet transform: the low-frequency element of the image is extracted using by a discrete wavelet transformation.
4. Binarization: transformation of the image by DWT makes it rough. In the next step, the average value of the grayscale pixels of the image is calculated, and then compared to that of other pixels. If the value is greater than the average value, then it is set to 1; otherwise, it is 0.

5. Encode to hash: the pixels in the image are arranged in a certain order onto a hash.

Contour Matcher Hash Matcher

After the hash value has been obtained, the hash matcher compares hashes using the Hamming distance. The Hamming distance is expressed as Δ_H, which is the sum of bits after exclusive or operations, as in Equation (1), in which n is the length of the hash; a_i is the hash value of the ith bit in image a, and b_i is the hash value of the ith bit in image b.

$$\Delta_H = \sum_{i=1}^{n} a_i \oplus b_i \tag{1}$$

After the Hamming distance has been obtained, n and Δ_H are used to yield the similarity α, as in Equation (2).

$$\alpha = \frac{n - \Delta_H}{n} \tag{2}$$

After the comparison has been made, if the similarity α exceeds the predefined threshold, then next step will involve the color matcher, which will compare the colors; otherwise, it will involve the logo finder and determine whether the image includes a logo.

Color Matcher

If the contour similarity reaches the predefined threshold, then it must be confirmed, so a color histogram of the images is compared. The operations are shown in Figure 7.

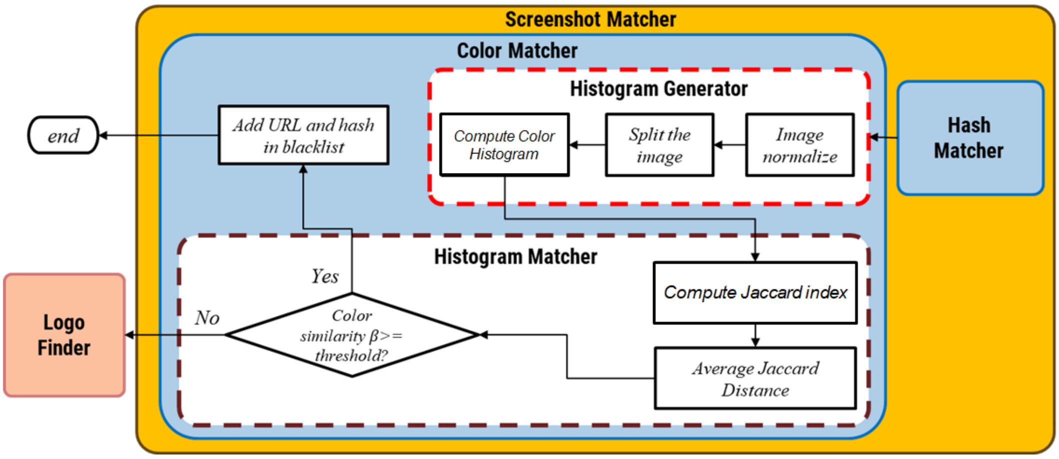

Figure 7. Color matcher operations.

Color Matcher Histogram Generator

wHash mechanism compresses image information, yielding a rough image, which is converted into a hash. However, using only contour similarity may result in a mismatch, so the color histograms are used to confirm the similarity of colors to increase accuracy. The color histogram is generated in the following three steps:

1. Normalize image: this step compresses the image to one with a specific side of a particular length. The image size may not be the same length, but the number of pixels is fixed, so all compared images must be normalized.
2. Split the image: the compared images are split into nxn blocks, which are then compared, yielding greater accuracy than a comparison of whole images, because the color differences between corresponding parts may vary.

3. Count the color histogram of each part.

Color Matcher Histogram Matcher

Histogram matcher uses the Jaccard index, which is obtained from images *a* and *b*, as in Equation (3):

$$\frac{a \cap b}{a \cup b} = J \tag{3}$$

Calculate the Jaccard index, J_i for each pixel color intensity, as in Equation (4):

$$\frac{a_i \cap b_i}{a_i \cup b_i} = \frac{Min(a_i,\ b_i)}{Max(a_i,\ b_i)} = J_i \tag{4}$$

Calculate the Jaccard index, J_k, for each block, as in Equation (5):

$$\frac{\sum_{i=1}^{n} J_i}{n} = J_k,\ n = 768 \tag{5}$$

Calculate the Jaccard index of the entire image, which is the color similarity β, as in Equation (6):

$$\frac{\sum_{k=1}^{m} J_k}{m} = \beta \tag{6}$$

If color similarity β is lower than the predefined threshold, the logo finder makes a further evaluation.

4.2.3. Logo Finder

After the image has been loaded, the feature descriptor is extracted, and a KD-tree is then constructed to accelerate feature matching. The feature descriptor is matched by the K-NN algorithm. Figure 8 shown the logo finder operation.

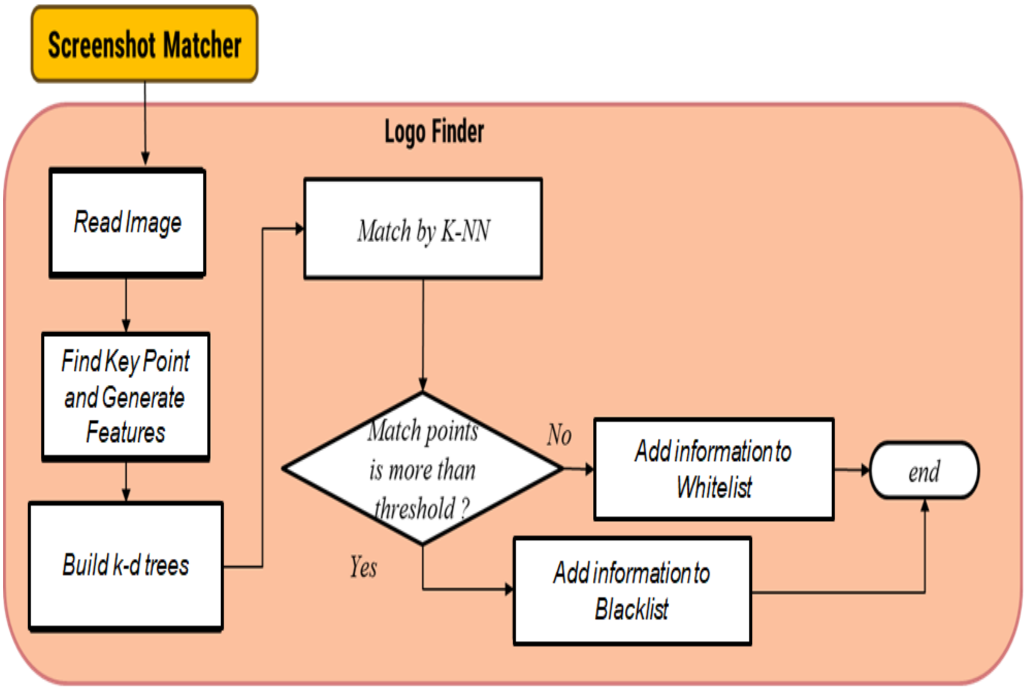
Screenshot Matcher
Logo Finder
Read Image
Find Key Point and Generate Features
Build k-d trees
Match by K-NN
Match points is more than threshold ?
No
Yes
Add information to Whitelist
Add information to Blacklist
end

Figure 8. Logo finder operation.

5. Performance Analysis

The performance of the proposed phishing websites detection system was analyzed using cases of high similarity 'very similar', local similarity 'local similar', and a complete test. In the 'very similar' cases, the main considerations are the setting of the best threshold and the comparison of efficiencies of the pHash and wHash mechanisms. In the 'local similar' cases, the main considerations are the selection of the best number of matching key points and the performance comparison of the SIFT and SURF mechanisms. Finally, the performance that is related to the classification effect was evaluated by performing the complete test.

Table 1 presents the studied data. The source of the phishing website used in the experiment was found from PhishTank, some invalid websites were deducted from the more than 20,000 websites found in PhishTank. The three most imitated sites are Microsoft, Dropbox, and Bank of America. Additionally, 1171 legal websites were collected from Internet. The websites of Microsoft, Dropbox, and Bank of America, for use in that involve 'very similar' and 'local similar'. 'Very similar' cases involve whether the unknown webpage is similar to a legal webpage or the webpages on the blacklist. 'Local similar' cases involve whether the webpages contain a logo. 'Non-imitating' cases involve the webpages that are normal. The complete test determines whether a website is a phishing website by identifying imitation. The balanced data set in Table 2 was used to analyze phishing websites to ensure that the results of the performance evaluation were not biased by the amount of data. The phishing website that was used in this work was taken from PhishTank. Additionally, 1267 legal websites collected from the Internet were used.

Table 1. Imbalanced data set.

Website Owner	Microsoft	Dropbox	Bank of America	Others
Very similar webs	276	94	70	1267
Local similar webs	394	207	152	

Table 2. Balanced data set.

Website Owner	Microsoft/Dropbox/Bank of America	Others
Number webs	150	150

5.1. 'Very Similar' Cases

Of the 1937 websites, 276 were 'very similar' to Microsoft's websites and 1661 were non-similar websites. The blue line in Figure 9 represents contour similarity. The orange line indicates the type of site; type 0 is not similar to a Microsoft website, whereas type 1 is 'very similar'. Sites are sorted from small to large similarity. A site with greater similarity is more likely to be a phishing website. Only two websites with a contour similarity α that exceeds 0.85 are mismatches. The good performance in the studied data is at similarity $\beta \geq 0.78$, as shown in Figure 10. The existing pHash mechanism is effective for classification when the threshold exceeds 0.65, but it still results in several misclassifications. The analysis result is shown in Figure 11.

The proposed screenshot matcher can accurately classify 'very similar' cases, as shown in Figure 12. The detection results reveal that color similarity compensates for a lack of contour similarity, and the predicted results (blue line) and the labeled values (orange line) all then overlap.

The analysis result with Dropbox data is shown in Figure 13. The good performance of the studied data is at similarity $\beta \geq 0.94$. The existing pHash mechanism is effective for classification when the threshold exceeds 0.78, and can be completely classified into two classes. The analysis result is shown in Figure 14.

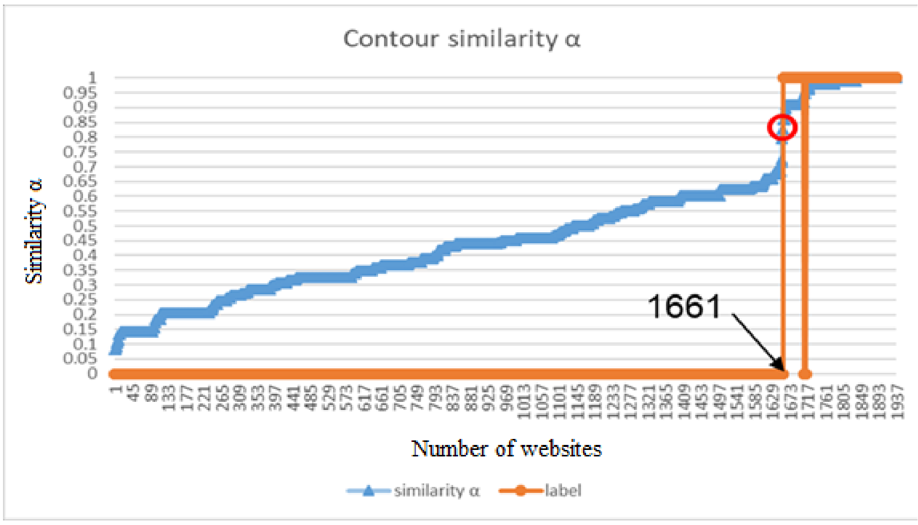

Figure 9. Similarity $\alpha \geq 0.85$ for Microsoft's data (wHash mechanism).

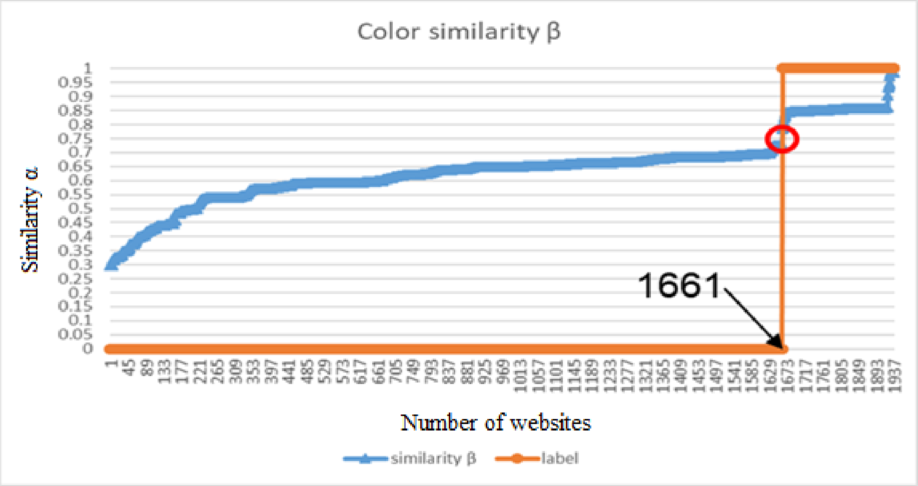

Figure 10. Similarity $\beta \geq 0.78$ for Microsoft's data (wHash mechanism).

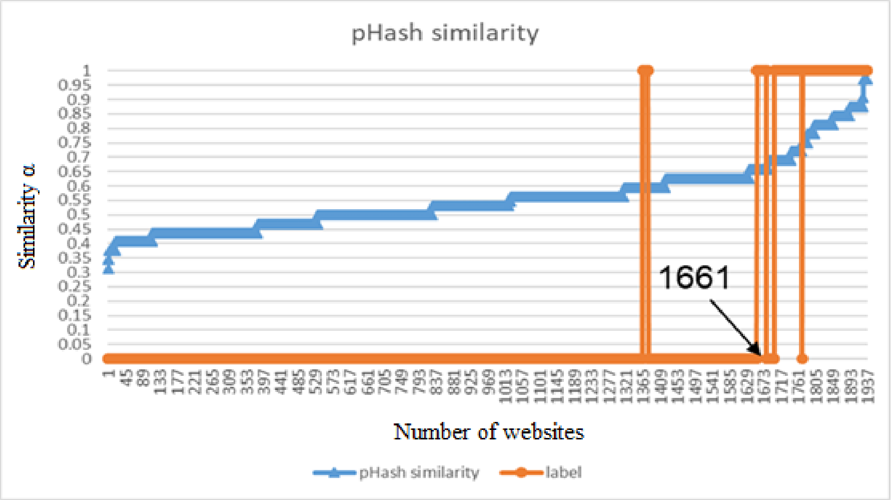

Figure 11. Similarity ≥ 0.65 for Microsoft's data (pHash mechanism).

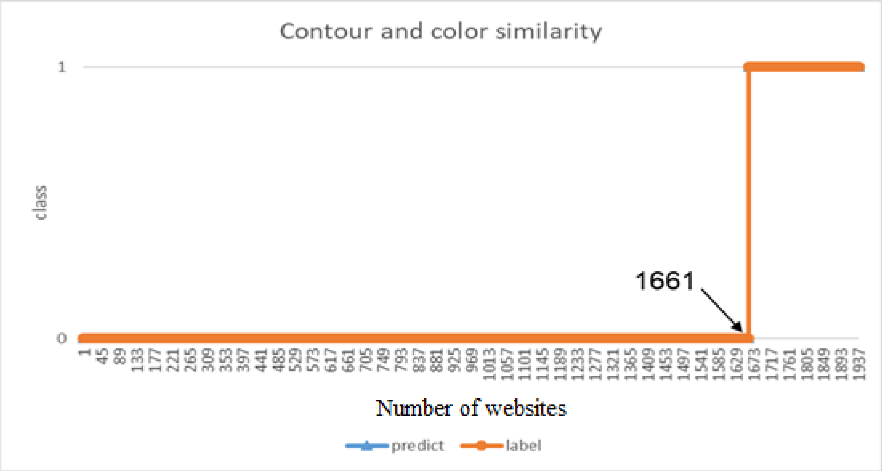

Figure 12. Screenshot matcher for Microsoft's data.

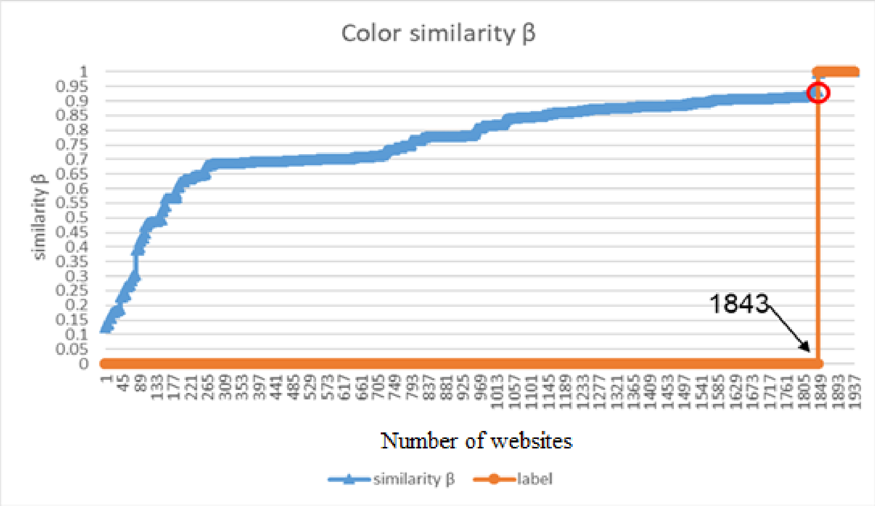

Figure 13. Similarity $\beta \geq 0.94$ for Dropbox's data (wHash mechanism).

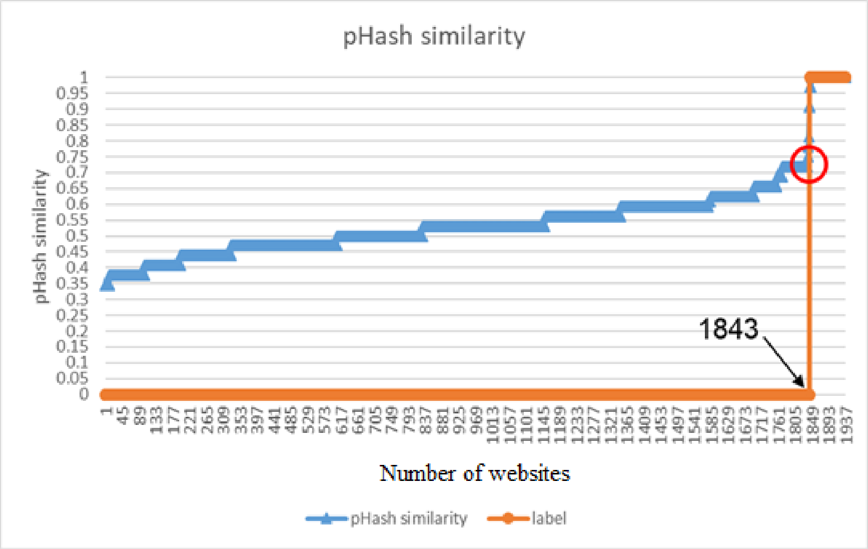

Figure 14. Similarity ≥ 0.78 for Dropbox's data (pHash mechanism).

The proposed screenshot matcher can accurately classify cases of 'very similar', as shown in Figure 15. The detection results reveal that color similarity compensates for a lack of contour similarity, and the predicted results (blue line) and labeled values (orange line) then all overlap.

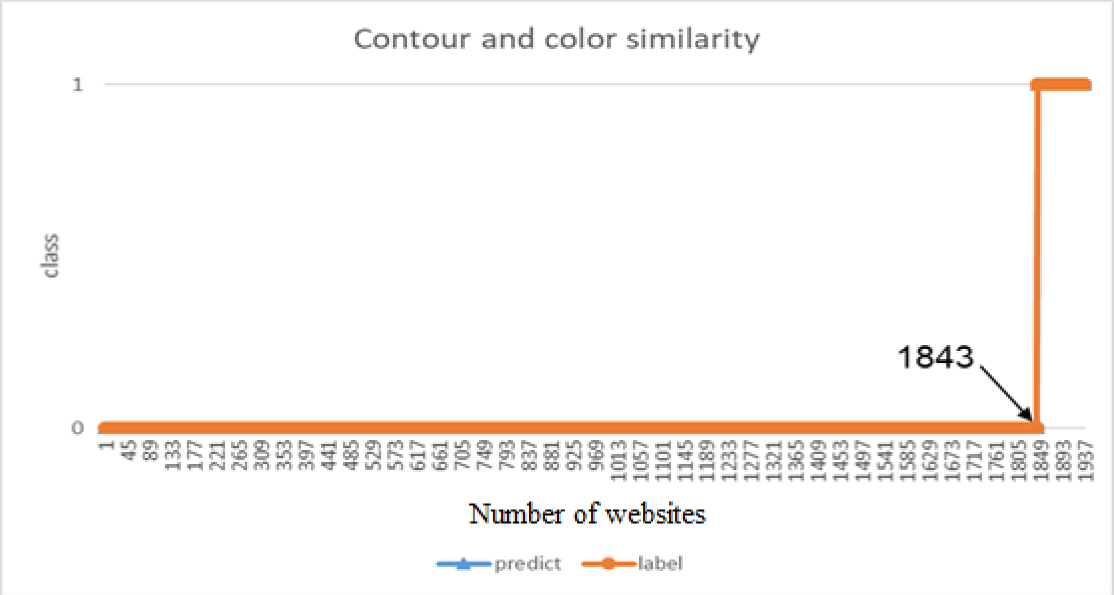

Figure 15. Screenshot matcher for Dropbox's data.

For the Bank of America website, when the contour similarity α exceeds 0.85, the good performance of the studied data is at similarity $\beta \geq 0.78$. The existing pHash mechanism is effective for classification when the threshold exceeds 0.75, and can be completely classified into two classes. The proposed screenshot matcher can accurately classify the cases of 'very similar', as shown in Figure 16. The detection results reveal that color similarity compensates for a lack of contour similarity, and the predicted results (blue line) and labeled values (orange line) then all overlap.

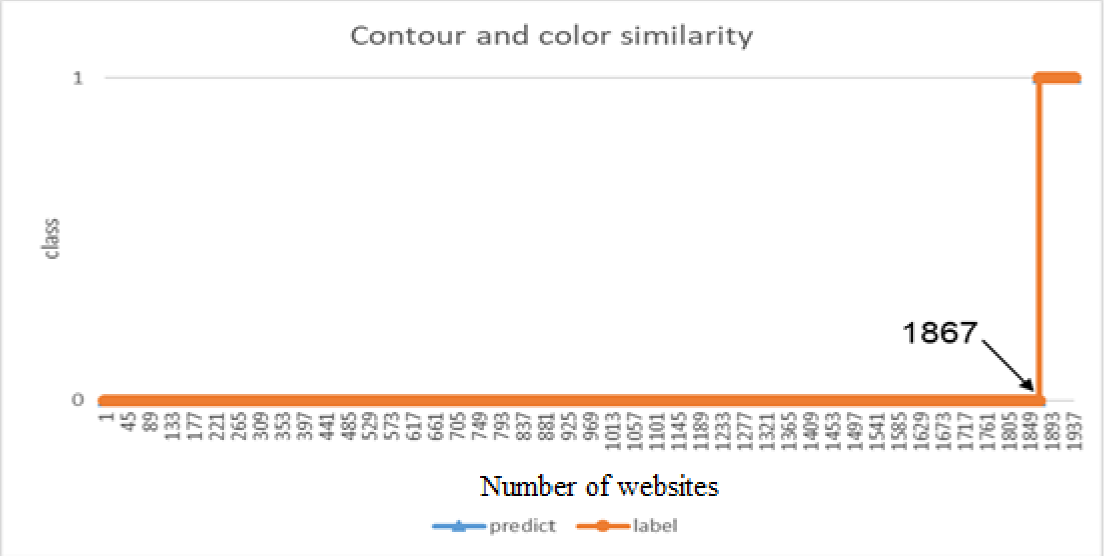

Figure 16. Screenshot matcher for Bank of America's data.

Table 3 presents the computation time. Although the time taken by the wHash mechanism slightly exceeds that by pHash mechanism, the analysis results reveals that the former is more adaptable to various images. The overall performance of the three cases is shown in Table 4.

Table 3. Time of wHash and pHash mechanisms for Microsoft's data.

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	Avg.
wHash (s)	484	478	486	472	471	472	478	477	478	477	477.3
pHash (s)	389	392	387	399	398	390	392	392	395	399	393.3

Table 4. Overall performance.

Target Webpage	Similar Webpages	Detected Webpages	Accuracy	Precision	Recall	F1-Score
Microsoft	276	276	100.00%	100.00%	100.00%	100.00%
Dropbox	94	94	100.00%	100.00%	100.00%	100.00%
Bank of America	70	70	100.00%	100.00%	100.00%	100.00%

5.2. 'Local Similar' Cases

The logos were studied, as shown in Figure 17. The best number of key matches for classification was found, as shown in Figure 18, the best match between Microsoft and Dropbox data is three key points. As the number of key points increases, the F1-score drops. Bank of America's data did not work well at first, but as the number of matched key points gradually increased, and the best number of points was found to be 11–17. The number of key points that are required for images that are smaller than 100 × 100 is about 3–4.

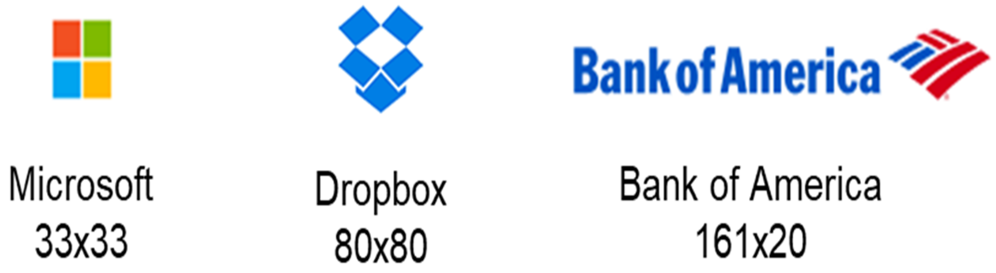

Figure 17. Studied logos.

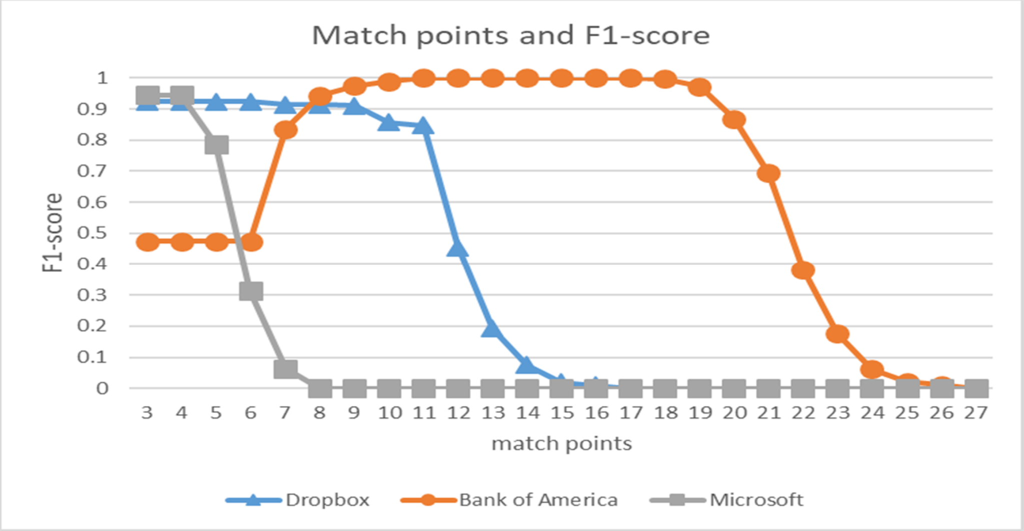

Figure 18. Number of key points.

Although SURF technique has a higher processing speed, it is susceptible to noise interference and found fewer key points than the SIFT technique. Table 5 compares the computation times of SIFT and SURF methods with Microsoft's data. The average time taken by SIFT method is about 2980.5 s. The average time taken by SURF method is about 1270.1 s. The speed of SURF method is about 2.3 times that of SIFT method, but its accuracy is lower. Table 6 presents the overall performance of the logo finder. The number of false positives is always 0.

Table 5. Time taken by SIFT and SURF for Microsoft's data.

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	Avg.
SIFT (s)	2980	2977	2985	2978	2980	2979	2983	2982	2983	2978	2980.5
SURF (s)	1270	1263	1269	1266	1268	1277	1269	1266	1278	1275	1270.1

Table 6. Overall performance of logo finder (SIFT method).

Target Webpage	Have Logo Webpages	Match Points	Etected Webpages	Accuracy	Precision	Recall	F1-Score
Microsoft	393	3	353	97.93%	100.00%	89.85%	94.65%
Dropbox	207	3	178	98.61%	100.00%	99.71%	94.51%
Bank of America	152	11	151	99.95%	100.00%	99.34%	99.67%

5.3. *Complete Test*

Tables 7 and 8 present the overall performance of the studied data. From the observation of the two tables, the performance difference is not obvious.

Table 7. Overall performance using unbalanced data set.

Target Webpage	Imitation Webpages	Match Points	Detected Webpages	Accuracy	Precision	Recall	F1-Score
Microsoft	393	3	363	98.14%	99.17%	91.60%	95.24%
Dropbox	207	3	180	98.61%	100.00%	99.71%	94.51%
Bank of America	152	11	151	99.95%	100.00%	99.34%	99.67%

Table 8. Overall performance using balanced data set.

Target Webpage	Imitation Webpages	Match Points	Detected Webpages	Accuracy	Precision	Recall	F1-Score
Microsoft	150	3	143	96.33%	98.60%	94.00%	96.25%
Dropbox	150	3	143	97.67%	100.00%	95.33%	97.61%
Bank of America	150	11	150	99.67%	100.00%	99.34%	99.67%

6. Conclusions

This study proposed an effective visual mechanism for detecting 'very similar' and 'local similar' phishing websites. In the case of 'very similar', the wHash mechanism with the color histogram has a higher accuracy than the pHash mechanism, and the former is more stable than the pHash mechanism. In the case of 'local similar', logo detection by SIFT technique is a suitable choice. This study also adds a cache to reduce the detection time, increasing the detection speed up to 4.6 times. In a complete test with imbalanced data, the accuracies of Microsoft, Dropbox, and Bank of America data were 98.14%, 98.61%, and 99.95% separately. However, the performance difference is not obvious in a complete test with balanced data. The threshold setting and processing speed should be discussed in the future.

Author Contributions: J.-L.C.: writing, review & editing; Y.-W.M.: writing, review & editing; K.-L.H.: writing original draft. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ministry of Science and Technology, Taiwan grant number MOST 108-2221-E-011 -068 -MY2.

Acknowledgments: Authors thank for financial supports of the Ministry of Science and Technology, Taiwan under contract MOST 108-2221-E-011 -068 -MY2.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kim, M.; Song, C.; Kim, H.; Park, D.; Kwon, Y.; Namkung, E.; Harris, I.G.; Carlsson, M. Scam detection assistant: Automated protection from scammers. In Proceedings of the 1st International Conference on Societal Automation, Krakow, Poland, 4–6 September 2019; pp. 1–8.
2. APWG's Q3 2017 Phishing Activity Trends Report. Available online: http://docs.apwg.org/reports/apwg_trends_report_q3_2017.pdf (accessed on 27 February 2018).
3. Jain, A.K.; Gupta, B.B. Two-level authentication approach to protect from phishing attacks in real time. *J. Ambient Intell. Humaniz. Comput.* **2018**, *9*, 1783–1796. [CrossRef]
4. Tan, C.L.; Chiew, K.L.; Wong, K.S. PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder. *Decis. Support Syst.* **2017**, *88*, 18–27. [CrossRef]
5. Bahnsen, A.C.; Bohorquez, E.C.; Villegas, S.; Vargas, J.; González, F.A. Classifying phishing URLs using recurrent neural networks. In Proceedings of the APWG Symposium on Electronic Crime Research, Scottsdale, AZ, USA, 25–27 April 2017; pp. 1–8.
6. Buber, E.; Dırı, B.; Sahingoz, O.K. Detecting phishing attacks from URL by using NLP techniques. In Proceedings of the International Conference on Computer Science and Engineering, London, UK, 5–7 July 2017; pp. 337–342.
7. Hu, Z.; Chiong, R.; Pranata, I.; Susilo, W.; Bao, Y. Identifying malicious web domains using machine learning techniques with online credibility and performance data. In Proceedings of the Congress on Evolutionary Computation, Vancouver, BC, Canada, 24–29 July 2016; pp. 5186–5194.
8. Zuhair, H.; Selamat, A.; Salleh, M. New hybrid features for phish website prediction. *Int. J. Adv. Soft Comput. Its Appl.* **2016**, *8*, 28–43.
9. Althobaiti, K.; Rummani, G.; Vaniea, K. A Review of human- and computer-facing URL phishing features. In Proceedings of the IEEE European Symposium on Security and Privacy Workshop, Stockholm, Sweden, 17–19 June 2019.
10. Mao, J.; Tian, W.; Li, P.; Wei, T.; Liang, Z. Phishing-alarm: Robust and efficient phishing detection via page component similarity. *IEEE Access* **2017**, *5*, 17020–17030. [CrossRef]
11. White, J.S.; Matthews, J.N.; Stacy, J.L. A Method for the Automated Detection of Phishing Websites through Both Site Characteristics and Image Analysis. In Proceedings of the SPIE; The International Society for Optical Engineering: Bellingham, WA, USA, 2012; Volume 8408, pp. 1–11.
12. Rao, R.S.; Ali, S.T. A computer vision technique to detect phishing attacks. In Proceedings of the 5th International Conference on Communication Systems and Network Technologies, Gwalior, India, 4–6 April 2015; pp. 596–601.
13. Asudeh, O. A New Real-time Approach for Website Phishing Detection Based on Visual Similarity. Ph.D. Thesis, The University of Texas at Arlington, Arlington, TX, USA, 2016.
14. Wang, G.; Liu, H.; Becerra, S.; Wang, K.; Belongie, S.; Shacham, H.; Savage, S. *Verilogo: Proactive Phishing Detection via Logo Recognition*; University of California: San Diego, CA, USA, 2011.
15. Singh, S.P.; Bhatnagar, G. A Robust Image Hashing Based on Discrete Wavelet Transform. In Proceedings of the IEEE International Conference on Signal and Image Processing Applications, Singapore, 4–6 August 2017; pp. 440–444.
16. Nevriyanto, A.; Sutarno, S.; Siswanti, S.D.; Erwin, E. Image steganography using combine of discrete wavelet transform and singular value decomposition for more robustness and higher peak signal noise ratio. In Proceedings of the International Conference on Electrical Engineering and Computer Science, Mexico City, Mexico, 5–7 September 2018; pp. 147–152.
17. Wang, S.; Qin, H. A study of order-based block color feature image retrieval compared with cumulative color histogram method. In Proceedings of the International Conference on Fuzzy Systems and Knowledge Discovery, Tianjin, China, 14–16 August 2009; pp. 81–84.
18. Jin, X.; Kim, J. ArtWork recognition in 360-degree image using 32-hedron based rectilinear projection and scale invariant feature transform. In Proceedings of the IEEE International Conference on Electronic Information and Communication Technology, Harbin, China, 20–22 August 2016; pp. 356–359.

19. Shin, J.; Kim, D.; Ruland, C. Content based image authentication using HOG feature descriptor. In Proceedings of the IEEE International Conference on Image Processing, Paris, France, 27–30 October 2014; pp. 5292–5296.
20. Pudchuen, N.; Deelertpaiboon, C. Visual odometry based on k-nearest neighbor matching and robust motion estimation. In Proceedings of the International Electrical Engineering Congress, Krabi, Thailand, 7–9 March 2018; pp. 1–4.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.