Detection of Tampered Images Using Blockchain Technology

Nour Jnoub, Wolfgang Klas

Multimedia Information System
Faculty of Computer Science
University of Vienna
Währinger Str. 29, 1090 Vienna, Austria
firstname.lastname@univie.ac.at

***Abstract*—Multimedia data, especially images are increasing dramatically. Images published on the Web are more often facing the risk of being tampered or manipulated since their content is easily mutable. Using blockchain technology provides advantages and, at the same time, challenges when dealing with this issue due to the following reasons: (a) data in a blockchain are well saved and immutable and (b) adding the data directly to a blockchain may consume much time which makes it computationally and economically expensive. Thus, we propose a blockchain-based solution which considers two key aspects: First, using a blockchain to register information about ownership and copyrights for authors, as well as descriptive information of an image, used to detect copyright violations. Second, avoiding insertion of raw image data into the blockchain, but storing only unique descriptive metadata about the images, allowing for a more efficient implementation of the system. This work considers different well-known image matching approaches to validate the power of the proposed approach, which allows for an efficient checking of violations of copyrights for a given image.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Nowadays, the World Wide Web is the biggest popular repository to disseminate data. Consequently, a massive number of images are generated and shared throughout different applications and Media platforms like news agencies and photography web pages. Due to that, people are facing the problems being overloaded by a huge number of images, becoming aware of manipulated images, or wrongly presented images. Indexing the entire available information about images and detecting the tampered ones is a significant effort and maybe also very complicated. Thus, the need for a safe way of publishing images, e.g., avoiding misuse of images like Facebook profile pictures, is getting more important than ever before, especially, when users can easily tamper images.

Different content-based image retrieval systems have been developed [1], [2]. They are in general either metadata-based or content-based approaches. Such systems help to retrieve similar images even with respect to semantic concepts [3], but those systems do not help to monitor violations of copyrights of images. Making usage of distributed blockchain technology is an effective and efficient way for monitoring ownership and copyrights of images by detecting tampered images based on their unique descriptive metadata.

978-1-7281-1328-9/19/$31.00 ©2019 IEEE

Fig. 1. Hash key generation example

In this paper, we introduce our Factcheck Blockchain (FCBC) system which helps to preserve ownership and to detect copyright violations of images. Its application aims at:

1) raising the authenticity for images since unique descriptive image information will be maintained by the system,
2) providing transparency when dealing with published images since the modified ones can be detected,
3) proofing the applicability of the approach based on different examples of images.

Our initial experiments show quite promising results.

II. Approach

We are illustrating our approach along two use cases. Use case 1 covers the registration of images in a blockchain, i.e., we register the ID of the owner (used to identify copyright information stored elsewhere) of an image and identifying hash keys on the image as well as on the features of the image.

Use case 2 covers the checking of copyrights. In particular it is checked whether an image is identical to an image previously already registered in the blockchain, or the image is a *I-derivation* from an image previously registered, or it cannot be compared to any previously registered image. We call an image D *I-derived* from another image O, if the image D is the result of a specific so called SURF-invariant manipulation of type rotation, intensity change, fish eye distortion, salt and pepper noise, and JPEG compression, acccording to [13, 18].

The feature extraction for the system we propose in this paper is performed using Speed Up Robust Feature (SURF) [4].

A. Use Case 1 - Registration of images in a blockchain

In order to be able to proof ownership of and/or copyrights on an image, we store unique information about the images in a blockchain. We use hash functions to compute the unique information to be stored in the blockchain. One hash key is generated from the original image representation (e.g., hash key of an RGB image). A second hash key is generated from extracted image key points and descriptors of the image according to SURF. Furthermore, we store an ID as a hash key of information on the owner (e.g., name, id, credentials, copyright information). All these unique hash values are stored as one record representing an image in the blockchain. Algorithm 1 illustrates these steps.

Algorithm 1 Registration of image and copyright info

Global variables
BC, the blockchain storing info on images
end Global variables
Input:
$Image, Owner\_ID$
Output:
Registration of image in blockchain,
returning hash-keys

1: **procedure** INSERT(Image, Owner_ID)
2: $Features \leftarrow get\_SURF\_Features(Image)$
3: $HF \leftarrow get\_hashkey(Features)$
4: $HI \leftarrow get\_hashkey(Image)$
5: $entry \leftarrow [HF, HI, Features, Owner\_ID]$
6: $Add\_To\_Blockchain(BC, entry)$
7: **return** HI, HF
8: **end procedure**

B. Use Case 2 - matching of images

In a first step we want to check whether an image has been registered before. We have to distinguish two cases:

(i) The image to be tested is identical to an image already registered in the blockchain. If this case is detected, one can conclude that if some legal entity has some ownership rights or copyrights on the original image, then it also has rights on the image tested. One can detect this case by testing whether the hash key of the test image features and the hash key of the test image are already stored as a record in the blockchain (see line 2-10 in Algorithm 2). In this case the returned answer of our algorithm indicates, that the test image does exist identically in the blockchain (see line 11-12 in Algorithm 2).

Algorithm 2 Checking Image - A

Global variables
BC, the blockchain storing info on images
end Global variables
Input:
$Image$
Output:
$Image$ exists as an identical image in the blockchain or it seems to be a derived image, or $Image$ has not been found in BC

1: **function** CHECKIMAGE_A(Image)
2: $ImFeatures \leftarrow get\_SURF\_Features(Image)$
3: $hF \leftarrow hashKey(ImFeatures)$
4: **if** $[hF, \_, \_, \_]$ contained in BC **then**
5: *// ImFeatures denotes the features of the original image that is represented by the hashkey stored in the blockchain, in other words hashKey(ImFeatures) = hashKey(original Image Features), i.e., hash keys will be identical.*
6: *// Note: test image needs not to be identical but can still be different (invariant modifications) from original, hence need to check the follwing:*
7: $hI \leftarrow hashkey(Image)$
8:
9: **if** $[hF, hI, \_, \_]$ contained in BC **then**
10: *// Conclusion: case A: test image is identical to original. If legal entity has rights on original image, then it also has rights on test image*
11: $result \leftarrow [FALSE, hI, ImFeatures]$
12: **return** $[\text{"}Identical\text{"}, result]$
13: **else**
14: *// hashkey of orig. image != hashkey hI*
15: *// Conclusion: case B: test image changed by means of SURF invariant modifications (rotation, etc.). Test image is derived from original one. If legal entity has rights on original image, then there is indication that it also may have rights on test image*
16: $result \leftarrow checkImage\_B(Image)$
17: **return** $[\text{"}Derived\text{"}, result]$
18: **end if**
19: **else**
20: **return** $[\text{"}NotFound, NULL]$
21: **end if**
22: **end function**

(ii) If the hash key of the image to be tested is not equal to any hash key of images contained in the blockchain, it still can be assumed that the test image may be an *I-derivation* of any image registered in the blockchain. I.e., the test image may have been modified by means of SURF invariant modifications like some rotation of the image. In order to detect the derivation of images we call the algorithm 3 ($checkImage\_B$).

Algorithm 3 first computes the SURF-based image features (key points and descriptors, see line 2). Then these extracted

features are compared with the image features of all images registered on the blockchain (see lines 3-15). The auxiliary function $NextBlockchain$ (see algorithm 4) is used to retrieve the records from the blockchain. The features of the test image are matched with features of a registered image by using K-Nearest Neighbor (KNN), where $k = 2$ denotes the number of closest neighbors to be considered (see line 8). Then a matching ratio $matchRatio$ is calculated by $calcMatchRatio$ (see line 9) accordinhg to equation 1.

$$Ratio = \frac{\#\ of\ matched\ keypoints(original, modified)}{\#\ of\ key\ points\ in\ the\ original\ image} \quad (1)$$

As soon as the $matchRatio$ calculated reaches some threshold (see line 10), there is clear indication, that the image to be tested is similar enough to the corresponding original image in the blockchain and the algorithm acknowledges the *I-derivation* by setting $Derived = True$ (see line 13) and returning also the matching image and its feature description from the blockchain (see line 16).

Algorithm 3 Checking Image - B

Global variables:
BC, the blockchain storing info on images
end Global variables:
Input:
$Image$
Output:
$Derived, O\_imageHK, O\_Features$ for copyright checking

1: **function** CHECKIMAGE_B(Image)
2: $ImFeatures \leftarrow get\_SURF\_Features(Image)$
3: $I\_orig \leftarrow NextBlockchain(BC, start)$
4: $Derived \leftarrow False$
5: **repeat**
6: $O\_Features \leftarrow I\_orig.features$
7: $O\_imageHK \leftarrow I\_orig.HI$
8: $Pts \leftarrow knnMatch(O\_Features, ImFeatures, k \leftarrow 2)$
9: $matchRatio \leftarrow calcMatchRatio(Pts, numberOfPoints(O\_Features))$
10: **if** $matchRatio < Threshold$ **then**
11: $I\_orig \leftarrow NextBlockchain(BC, prev)$
12: **else**
13: $Derived \leftarrow True$
14: **end if**
15: **until** $Derived$ OR I_orig = $NULL$
16: **return** $Derived, O\_imageHK, O\_Features$
17: **end function**

C. Matching features between two images

After extracting SURF features from both images, e.g., the original image and the modified image, Features in the original image should be matched with the corresponding feature in the modified image. The goal of matching is to find the closest matching feature. For this purpose, we use the well-known MATLAB[1] function $matchFeatures$[2] which returns indices of the matching features in the original and modified image using a pairwise distance metric.The $matchFeatures$ function has been used with respect to its default settings.

The aim of matching features is due to the fact that changes of the original image may lead to changes regarding key points and descriptors compared to the original image [5]. Thus, if descriptors are generated from the modified image, it will help to detect any simple changes in the original image.

Algorithm 4 Utility function to read on blockchain

Input:
bc ... the blockchain
$status$ ... indicates how to read the blockchain
Output:
entry [HF, HI, Features, Owner_ID] from the blockchain

1: **function** NEXTBLOCKCHAIN(bc, status)
2: **if** $status$ = $start$ **then**
3: // get most recent entry from blockchain
4: $r \leftarrow MostRecentEntryFromBC(bc)$
5: **return** r
6: **end if**
7: **if** $status$ = $prev$ **then**
8: // get next previous entry from blockchain
9: $r \leftarrow PreviousEntryFromBC(bc)$
10: **return** r
11: **end if**
12: **end function**

III. RESULTS

In our work, we investigated the performance of Speed Up Robust Feature Approach (SURF) using different images with different modifications: JPEG compression, level: 0.05, 45-degree rotation, 135-degree rotation, fisheye distortion, salt and pepper noise level: 0.05 and intensity change (using histogram equalization).

We rely on a benchmark of performance comparison for distorted images of SIFT, SURF, BRIEF, and ORB. We have chosen SURF for our approach due to its performance with respect to accuracy and computation complexity [5].

In our tests, we evaluated in total 132 test images that come from different fields, e.g., sports, politics, animals, humans, and movies.

The check of a test image has been performed with respect to different types of images. This includes some modified versions (*I-derived*) of the same tracked image which is already in the blockchain as well as images with same content but slightly different perspective to check if the system matches or denies them correctly.

[1]MATLAB 2015, The MathWorks, Inc., Natick, Massachusetts, United States.

[2]https://de.mathworks.com/help/vision/ref/matchfeatures.htmlbsvbhh1-1-MatchThreshold

Mod. Type	Rotation 45	Intensity change	Fish eye distorted	Salt and pepper	JPEG	Rotation 135
SURF (avg ratio %)	31	33	16	64	32	30
Detection accuracy % ($threshold = 30$)	81	63	36	100	72	40
Detection accuracy % ($threshold = 20$)	100	63	66	100	72	90

TABLE I
EXPERIMENTAL RESULTS FOR MATCHING IMAGES USING DIFFERENT TYPES OF IMAGE MODIFICATIONS

Table I denotes the ratio of matching for the considered images in our tests. The numbers in the table show the average matched ratio and detection accuracy when the $threshold$ is set to 30% and 20% respectively. The calculation is based on averaging (equation 2) the match ratios (equation 1), where N is the number of images in the test collection.

$$\frac{\sum_{n=1}^{Images=N} Ratio(n)}{N} \tag{2}$$

IV. DISCUSSION

Multimedia data are usually huge and saving such data to the Blockchain is not an easy doable task and may consume much time, since hashing such source data is computationally expensive, especially if a system is interacting with a huge amount of data, the whole system might stop.

Thus, extracting meaningful image features is mandatory and applying such a system can assert integrity, since once the features are put in the Blockchain, and these are associated with the owners of the images, the data cannot be modified even by the providers of the service, but it is also important to highlight that, (a) transactions imply delays, (b) the number of features stored in the Blockchain matters, (c) programming in Solidity language in its current version $v0.4.24$ is still not favorable regarding its limitations [6] - in case one would like to encode system functionality by means of smart contracts.

Thus, we considered two use cases to check how far we can benefit from the Blockchain characteristics in order to keep those data safe and still be able to check, if any changes or modifications have been done to such data.

Additionally, there are still few research works trying to combine Blockchain technology with the detection of deviation, misuse, or illegal modification. There are still limitations regarding the matching of images due to the fact that, if the system is extended to work with a higher level of semantic representation of inserted images in a formal language description, e.g., an ontology. Such an ontology can be used to map the extracted features with a higher level of semantics and, consequently, the content of images will be better understood which would improve the overall system performance. This work can be considered as one of the first bricks in this research field that may motivate other researchers to investigate more in this research area.

V. CONCLUSION

This work proposes an approach for testing images against a collection of images registered in a blockchain in order to detect if the test image is identical to or derived from a registered image. The use of blockchain technology provides the benefit of an immutable registry for images and their feature descriptions. This supports use cases where image owners may want to proof e.g., potential misuse of images they own or copyright violations. We illustrated the approach by following two use cases: First, registering image and owner information in a blockchain, second, checking a test image against a collection of images. We use selected features and algorithms according to SURF. We did some initial experiments with 132 test images showing promising results, using an Ethereum-based private blockchain. To the best of our knowledge, few works that have considered this problem due to (a) the size of the image data, (b) the costs of operating a Blockchain, (c) the complexity of evaluating the performance of such systems and the delay caused by mining.

In our future work, we will extend the proposed approach to be applied to text, audio, and video content. However, dealing with video data in the frame of Blockchain research is still very challenging due to the very high volume of content that might be faced during the feature extraction process.

ACKNOWLEDGEMENTS

The Computer Science Faculty operates the recently founded BlockchainSci-Lab at University of Vienna. The lab offers students the opportunity to be familiar and work with state-of-the-art systems and platforms to learn about Blockchain technology and to design and implement Blockchain applications by participating in dedicated projects. This is one of the projects of the Lab.

REFERENCES

[1] N. Singhai and S. K. Shandilya, "A survey on: content based image retrieval systems," *International Journal of Computer Applications*, vol. 4, no. 2, pp. 22–26, 2010.

[2] T. Dharani and I. L. Aroquiaraj, "A survey on content based image retrieval," in *Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on*. IEEE, 2013, pp. 485–490.

[3] S. Jabeen, Z. Mehmood, T. Mahmood, T. Saba, A. Rehman, and M. T. Mahmood, "An effective content-based image retrieval technique for image visuals representation based on the bag-of-visual-words model," *PloS one*, vol. 13, no. 4, p. e0194526, 2018.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[5] E. Karami, S. Prasad, and M. Shehata, "Image matching using sift, surf, brief and orb: performance comparison for distorted images," *arXiv preprint arXiv:1710.02726*, 2017.

[6] M. Wohrer and U. Zdun, "Smart contracts: security patterns in the ethereum ecosystem and solidity," in *Blockchain Oriented Software Engineering (IWBOSE), 2018 International Workshop on*. IEEE, 2018, pp. 2–8.