Video Forgery Detection Using Hashing

Jayant Yadav
Maharaja Agrasen Institute of Technology
Delhi,India

Abstract— **There is a huge surge in the short video clips with introduction of many applications, and this has also increased the issue of video tampering. Also, there are many tools easily accessible for video editing. This raises the need to providing a. method for proving the authenticity of any provided video. This paper proposes the method to detect the tampered frames and regions of a given video.**

Keywords—Image Forgery, Image Tampering, Hashing.

I. Introduction

With easy-to-use video editing tools available in hands of a user it is easier to tamper any video. Also, with the use of machine learning models the attacker can add new frames or objects in the video that can be harder for a human to decide. So, developing a robust method to detect the tampered region of the video can be a major use during copyright investigations.

This paper proposes the methods to implement the hashing algorithm on each frame of a video and to detect the forged frames in the video and highlighting the tampered region in the frame as well. As for every different value a unique hash can be generated this algorithm can provides robustness and accuracy for the given task.

II. Related Work

Every mobile or any digital device user come across video in any medium and for an organization dealing with copyrights or digital investigation it is an important part of the process to detect the tampering in a video. In past years many methods have been proposed for the same. These methods can be divided into two categories: frame-based hashing and spatio-temporal based hashing respectively. In [1] the authors have proposed the algorithm which used the centroid of gradient orientations (CGO) where the video is loosely compressed which makes it robust to noise but it is limited in certain aspects. In the other work submitted by [2] used a method to extract a rotation-invariant hash based on Radial hASHing (RASH) algorithm. The latter method is more robust and can generate a better feature vector for tampering detection.

III. Propsed Work

The Figure 1 shows the process to detect the tampered part to the image and the flow of logic on a higher level.

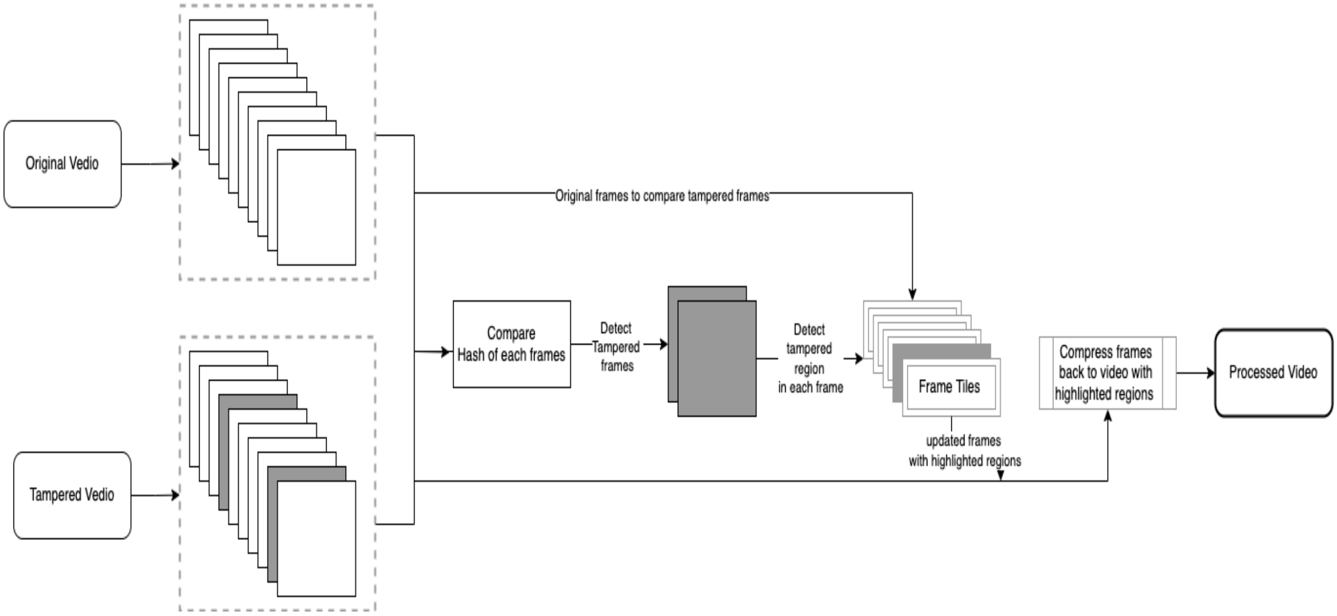

Original Vedio
Tampered Vedio
Original frames to compare tampered frames
Compare Hash of each frames
Detect Tampered frames
Detect tampered region in each frame
Frame Tiles
updated frames with highlighted regions
Compress frames back to video with highlighted regions
Processed Video

Figure 1: Proposed Algorithm

The proposed method can be further divided into sub-parts; Generate frames, Create hash of frames and detect tampered frame, evaluate tampered frame and highlight forged region, compress frames back to video respectively.

A. *Generate Frames*

First the original and targeted video have been processed and as per the user's configuration frames are generated for each second. Ideally for this experiment the frames rate per second is kept at 30.

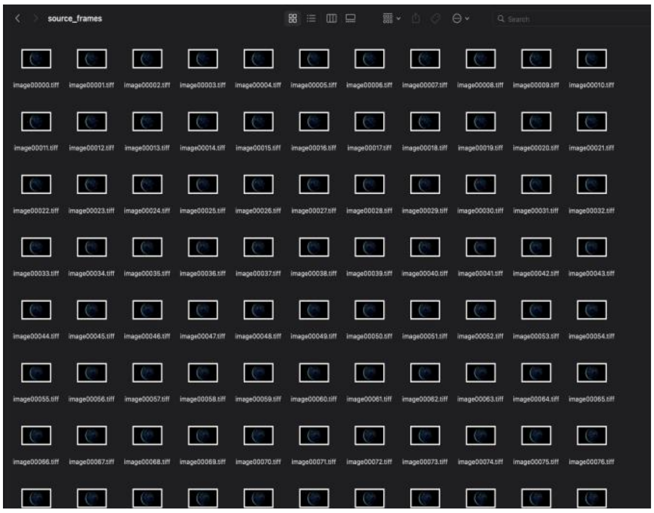
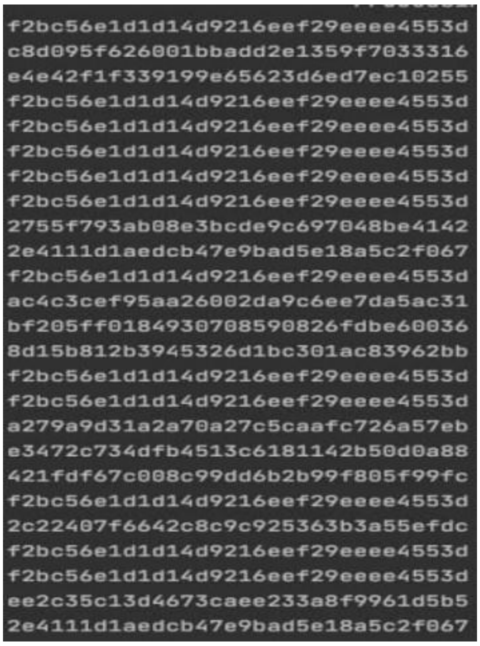


Figure 2- Frames Generated

As per shown in figure-2 each image is generate and stored in ".tiff" format. Here ".jpeg" proved not useful due to use of lossy compression which leads into information loss during generation of remaking video from frames.

B. *Create hashes for frames and detect tampered frame*



```
f2bc56e1d1d14d9216eef29eeee4553d
c8d095f626001bbadd2e1359f7033316
e4e42f1f339199e65623d6ed7ec10255
f2bc56e1d1d14d9216eef29eeee4553d
f2bc56e1d1d14d9216eef29eeee4553d
f2bc56e1d1d14d9216eef29eeee4553d
f2bc56e1d1d14d9216eef29eeee4553d
f2bc56e1d1d14d9216eef29eeee4553d
2755f793ab08e3bcde9c697048be4142
2e4111d1aedcb47e9bad5e18a5c2f067
f2bc56e1d1d14d9216eef29eeee4553d
ac4c3cef95aa26002da9c6ee7da5ac31
bf205ff018493070859082 6fdbe60036
8d15b812b3945326d1bc301ac83962bb
f2bc56e1d1d14d9216eef29eeee4553d
f2bc56e1d1d14d9216eef29eeee4553d
a279a9d31a2a70a27c5caafc726a57eb
e3472c734dfb4513c6181142b50d0a88
421fdf67c008c99dd6b2b99f805f99fc
f2bc56e1d1d14d9216eef29eeee4553d
2c22407f6642c8c9c925363b3a55efdc
f2bc56e1d1d14d9216eef29eeee4553d
f2bc56e1d1d14d9216eef29eeee4553d
ee2c35c13d4673caee233a8f9961d5b5
2e4111d1aedcb47e9bad5e18a5c2f067
```

Figure 3 Hash values for the tiles

So, when for both source and target video frames are generated and then taken to generate hash values of each frame. MD5 algorithm is used for generating a 128-bit hash values of each frame. These hash values are further mapped with their counter parts (i.e., source-frame mapped with target-frame). By comparing the hash value, the tampered frames can be detected.

C. *Evaluate tampered frame and highlight forged region*

When the tampered frames are detected the images and again decomposed into small tiles. These tiles are generated for both source and target frame.

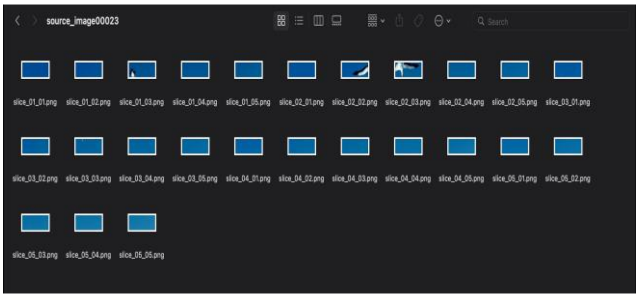


Figure 4 Frame tiles

As shown in figure 3, the frame is decomposed again into 25 tiles with ".png" format. These frames are again processed to generate hash values with MD5 algorithm. The hash values generated are compared to detect the forged region of the frame.

When different hash values are detected, that particular tiles is then taken for adding border for user are a result of the analysis.

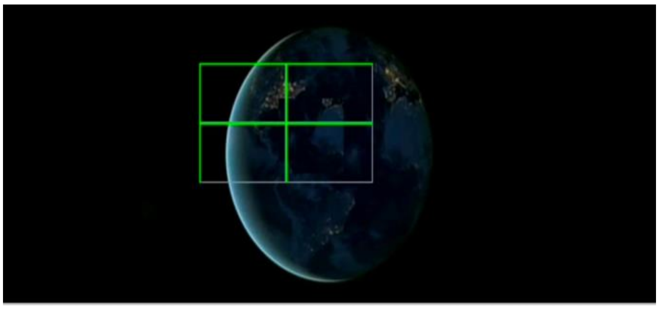


Figure 5 Updated frame after analysis

The tiles are stitched back to form the original frame back with addition information embedded into the image. The figure 4 demonstrates the updated frame with highlighted tempered region.

D. *Compress frames back to vedio*

After analyzing the tampered frames and the detecting the tampered part of each image these frames are again compressed back into video.

The target frames are taken and the frames rates per second mentioned by the user's configurations are taken and a ".mkv" video format file is generated.

IV. Experiment Results

This algorithm was given 4 different video inputs with have a different scenario of with tampering like color correction, adding new object, removing an object and duplication of object among the video. Among this dataset some tampered regions are human interpretable, but many regions were not interpreted so this makes a base of comparison and use case for the proposed method. The algorithm managed to detect all the tampered regions during experiments thus have a recall rate of 100%.

V. Conclusion

The hashing method can some time proves to be a costlier algorithm as it can take more compute cost when compared to other methods proposed. This scenario would likely to happen when the frames generated are of higher resolution thus resulting in longer processing time for generating results. But on the other side the algorithm is robust to any tampering method usually used by an attacker.

References

[1] S. Lee and C. D. Yoo, "Robust video fingerprinting for content-based video identification," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 7, pp. 983-988, July 2008, doi: 10.1109/TCSVT.2008.920739.

[2] C. De Roover, C. De Vleeschouwer, F. Lefebvre and B. Macq, "Robust video hashing based on radial projections of key frames," in IEEE Transactions on Signal Processing, vol. 53, no. 10, pp. 4020-4037, Oct. 2005, doi: 10.1109/TSP.2005.855414.

[3] Chen, H., Wo, Y. and Han, G. (2017) “Multi-granularity geometrically robust video hashing for tampering detection”, Multimedia Tools and Applications, 77(5), pp. 5303-5321. doi: 10.1007/s11042-017-4434-2.