Int. j. inf. tecnol.
https://doi.org/10.1007/s41870-018-0268-z

ORIGINAL RESEARCH

Tampering detection and localization in digital video using temporal difference between adjacent frames of actual and reconstructed video clip

Vaishali Joshi[1] · Sanjay Jain[1]

Received: 21 July 2018 / Accepted: 7 December 2018

Abstract The scientific, generalized and automatic methods for detecting forgery became the biggest challenge for scientists and researchers. This problem is true in case of all multimedia contents including audios, graphics and videos. It is harder when one doesn't know the source and background of video in hand and still expected to establish authenticity of it. However, there are algorithms suggested which can work for such tampering in videos captured with static GOP structure. The problem becomes even more difficult when video is captured using adaptive GOP structure (AGS) scheme in which variable sizes of GOP structures are used to improve coding efficiency and to provide strong temporal scalability. In this paper, an algorithm is proposed which is a passive tampering detection algorithm based on comparison of temporal difference between adjacent video frames of actual video clip and its reconstructed version using intrinsic temporal fingerprints, which can work on videos captured using variable size GOP structures. Firstly, all the video frames are extracted from given video sequence. Then, temporal difference is calculated for each pair of adjacent frames in video's actual and reconstructed from. Video is reconstructed using frame prediction error. Lastly, the calculated differences are used to find and localize tampering. Our proposed algorithm can effectively classify a video, irrespective of whether captured with fixed or AGSs, as genuine or forged using temporal difference between adjacent video frames in its actual and reconstructed form. Extensive experimental results show that the proposed method achieves promising accuracy in classifying genuine videos and forgeries. The results show that the proposed tampering detection algorithm can detect and precisely locate tampering with an average accuracy of 87.5%.

Keywords Video forgery · MPEG · GOP · Temporal fingerprints · Optical flow · Video tampering detection

1 Introduction

The fundamental process of digital video forensic system is to prove if the given video is intentionally modified or not but it might be more important if the system can tell where and when the alteration is made [1]. Temporal fingerprints based digital video forensic algorithms/methods make available information regarding digital video contents without depending on exterior descriptors such as metadata tags or extrinsically inserted information such as digital watermarks both fragile and semi-fragile. Alternatively, these techniques use intrinsic fingerprints left in digital video content itself by editing operations or the digital video capturing process [2]. When is video is captured by a video recording device, it records the scene in front of lens of capturing device, frame by frame. Thus a digital video sequence can be considered as a collection of successive frames with temporal dependency at previous frames, in a 3-D plane. When some malicious altering is carried out on a video sequence, it either attacks on contents of video or on the temporal dependency between the frames. So, basis regional property of video sequences, video tampering attacks can be broadly classified in three categories: spatial

✉ Vaishali Joshi
vaishali.joshi22@gmail.com

Sanjay Jain
sanjayjain@itmuniversity.ac.in

[1] Department of CSA, ITM University, Gwalior, Madhya Pradesh, India

Published online: 01 January 2019

tampering attacks, temporal tampering attacks and spatio-temporal, a combination of previous two, attacks [3].

1.1 Temporal fingerprints

Digital video is a collection of time varying images. The transformation of a four dimensional physical object 4D (x, y, z, t) into a three dimensional object 3D (x, y, t) is called a digital video. The dimension t stands for time which temporal difference between the two images.

Every digital video is captured with a lot of redundancy in each the frames of digital video sequence. MPEG digital video compression technique renders this redundancy by predicting some frames and encoding residual error between these predicted frames. The MPEG standard is developed to minimize both temporal redundancy across all video frames and spatial redundancy within each video frame [4]. In this video compression technique, to put a check on error propagation, the video sequence is divided into fragments, where each fragment is referred to as a group of pictures (GOP). Frame prediction is done within each fragment in case of fixed GOP videos, and across fragments in case of variable length videos. In a number of cases, a person intended to forge a video may wish to add some frames or delete few frames from a digital video sequence. To execute this forgery, the one must decompress the digital video before frames are added or deleted, then recompress it after it has been maliciously altered. In the previous work done by Wang and Farid, it has been shown that recompression of MPEG video using a fixed GOP structure results in two distinct, forensically detectable fingerprints; one spatial and the other temporal [5, 6]. The spatial fingerprint can be observed within a single I-frame and is similar in nature to the fingerprint left by double JPEG compression. This fingerprint occurs when either no frames are added or deleted, or when the number of frames added or deleted is an integer multiple of the fixed GOP length. The temporal fingerprint occurs in the sequence of P-frame or B-frame prediction errors and occurs only if frames have been added to or deleted from the video sequence prior to recompression.

1.2 Role of temporal fingerprints

When frames are deleted from or added to a digital video, each GOP in the recompressed video will contain frames that belonged to different GOPs during the initial compression [2].

The effect can easily be seen in Fig. 1. When a P-frame or B-frame is predicted using an I-frame which belonged to different GOP before tampering, an increase in total frame prediction error is observed. This increase in the prediction error can be used to detect and localize tampering in digital video.

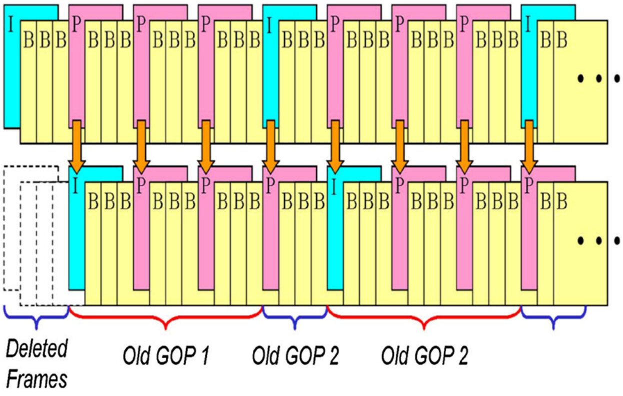


Fig. 1 The effect of tampering in terms of temporal fingerprints in a digital video

1.3 Optical flow

Optical flow refers to the visible motion of an object in an image, and the *apparent 'flow' of pixels* in an image. It is the result of 3D motion being projected on a 2-D image plane.

Real motion may or may not give rise to optical flow. For example consider a sphere which is uniformly illuminated is rotating about the axis parallel to image plane. It won't give any apparent information of motion of pixel flow in the image, and the sphere will appear still.

Similarly, a static object may give rise to optical flow in some cases. For example consider the same sphere, but this time it is stationary. Suppose the light source moves constantly, it will give rise to optical flow on the image. The uses of optical flow are mainly in the field of *Object Tracking*. The optical flow can be used as an estimation of object velocity and position of object in the next frame. It falls under the kernel tracking category of object tracking, and is referred to as KLT algorithm Also it can be used for stereo imaging systems and *registration* purpose. The calculation of optical flow is a bit of a tough task, since, in its original form there are more number of unknowns than the equations. It is also called *aperture problem*. But, some breakthrough research papers of 80s ad 90s gave a way to calculate the optical flow, by making some valid assumptions and formulating equations. The research papers by *Lucas–Kanade* and *Horn–Schunck* give a very detailed and easy-to-understand description of optical flow, its calculation and the uses. Please read those research papers for a clear understanding of this topic. Optical flow is used to find the relative change in between 2 frames and the feature points in them. It's very useful for pose-estimation, position holding and semi-dense visual odometery.

1.4 Frame prediction error

Any video is captured with a lot of redundancy between adjacent frames of video sequence. MPEG video compression technique uses this redundancy by predicting some frames and encoding residual error between these predicted frames. The MPEG video compression standard is developed basically to minimize both spatial redundancy within each video frame and temporal redundancy across all video frames. While applying MPEG video compression technique, to put a bay to error propagation, the video sequence is divided into fragments, where each fragment is referred to as a group of pictures (GOP). Frame prediction is done within each fragment, but never across fragments, thus preventing decoding errors in one frame from extending throughout the video sequence. Within each group of pictures (GOP), video frames are categorized into three types: intra-frames (I-frames), predicted-frames (P-frames), and bidirectional-frames (B-frames). Each group of picture starts with an I-frame followed by a number of P-frames and B-frames. No prediction is carried out when encoding I-frames therefore each I-frame is encoded and decoded independently using compression similar to JPEG compression. *A P-frame can reference only preceding I- or P-frames, while a B-frame can reference both preceding and succeeding I- or P-frames.* Video compression algorithms such as MPEG-4 and H.264 use inter frame prediction to reduce video data between a series of frames [2] (Fig. 2).

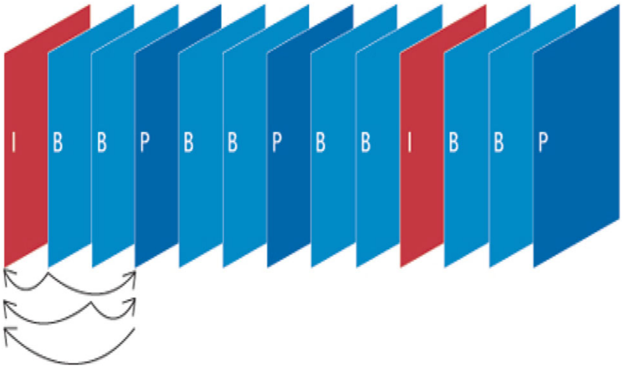


Fig. 2 A typical video sequence with I-, B- and P-frames

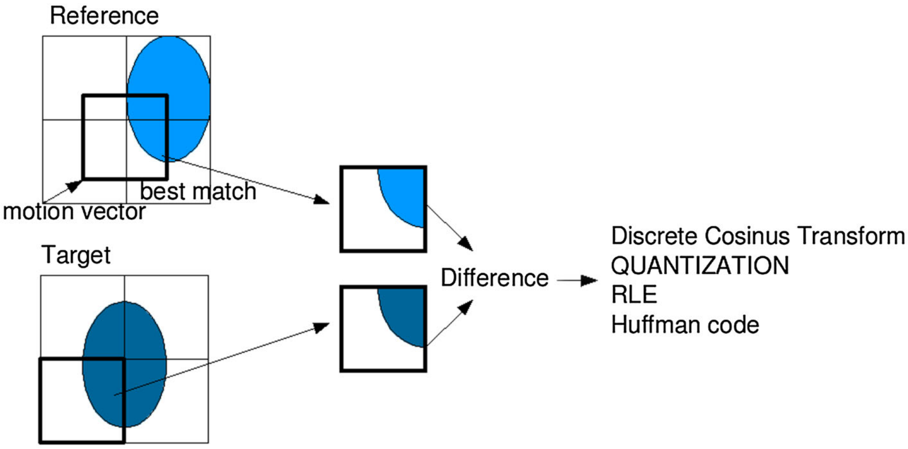
An inter coded frame is divided into blocks known as macroblocks. After that, instead of directly encoding the raw pixel values for each block, the encoder will try to find a block similar to the one it is encoding on a previously encoded frame, referred to as a reference frame. This process is done by a block matching algorithm. If the encoder succeeds on its search, the block could be encoded by a vector, known as motion vector, which points to the position of the matching block at the reference frame. The process of motion vector determination is called motion estimation.

In most cases the encoder will succeed, but the block found is likely not an exact match to the block it is encoding. This is why the encoder will compute the differences between them. Those residual values are known as the prediction error and need to be transformed and sent to the decoder [7].

To sum up, if the encoder succeeds in finding a matching block on a reference frame, it will obtain a motion vector pointing to the matched block and a prediction error. Using both elements, the decoder will be able to recover the raw pixels of the block. The following image shows the whole process graphically (Fig. 3).

In this case, there has been an illumination change between the block at the reference frame and the block which is being encoded: this difference will be the prediction error to this block.

Fig. 3 Inter-frame prediction process



2 Proposed method

2.1 Method overview

A number of videos have been taken with the help of capturing devices like digital camcorder and mobile phone. The formats of the captured videos are MP4 and AVI. Each video clip is manipulated/forged with the help of Windows default video editing software. Frame prediction error is used to detect forgery. The frame prediction errors of all the frames of video in hand and its reconstructed form are calculated and stored in excel sheets. These excel sheets are used to draw plots for frame prediction error of the video in hand and its reconstructed form and compare these plots. Already proven optical flow feature is used to verify the classification of video in hand as forged or genuine. The proposed method calculates optical flow for the given video clip and its motion estimated version. By manual comparison of the two, tampering can be detected and localized.

2.2 Reason for choosing frame prediction error as feature to detect tampering

One of the challenges was to find a robust feature which can be used to identify and localize tampering in digital video [8]. Feature extraction in highly compressed video was very challenging. Another issue was presence of a single video file in hand without the knowledge of its source. That available single file is to be classified as forged or genuine. The proposed method can classify the video file as forged of genuine without knowing its source. The method used frame prediction error as a strong feature to accurately classify the video as forged or genuine. The frame prediction error can be calculated for each with the help of its reference frame. This information is inbuilt into the video. No forger can remove these fingerprints and prevent researcher from finding these. These fingerprints are automatically inserted into the video when one tries to alter the video sequence [9]. Alteration may include frame insertion, frame deletion, frame duplication etc.

2.3 Framework description

a. A video clip in MP4 or AVI format is taken as input and passed to proposed system for computation of prediction error vector and optical flow.
b. The full length video sequence given as input is divided into frames and saved in a folder in form of JPEG images.
c. The structural similarity is extended to measure similarity between two frames of video clip.
d. Then the similarities between frames in the temporal domain are measured and used to calculate prediction error between two frames. Bi –directional motion estimation can be calculated using:

$$\mathrm{F(x,y,t)} = \mathrm{F_b(x - d_b x, n - d_b y, t - 1) + F_f(x - d_f x, n - d_f y, t - 1).}$$

where Ff and Fb stands for forward and backward frames while (d_bx, d_by) and (d_fx, d_fy) stands for pixel difference in forwards and backward frames.
e. Frame prediction error for each frame is calculated and frames are reconstructed based on this error. The calculated errors are stored in Excel sheets for further use.
f. Frame prediction errors of these reconstructed frames are also calculated. And again stored in excel sheets.
g. These stored frame prediction errors are used to draw plots of actual video and its reconstructed form.
h. These plots easily show the variation between frame prediction errors of forged video and genuine video. So, by comparing these plots one can easily classify between forged and genuine video.
i. Finally, optical motion is calculated using method proposed by Lucas-Kanade for input video clip as well as predicted video clip. A plot is drawn to show optical flow of each video clip. With the help of plot tampering can be accurately detected and localized. Already proven feature of optical flow is used to verify correct classification of video.

3 Experiment

3.1 Tools used

MATLAB tool is used to implement and test the proposed algorithm. MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-*use* environment where problems and solutions are expressed in familiar mathematical notation. Typical *uses* include: Data analysis, exploration, and visualization. Mainly image processing and computer vision toolboxes are used for implementation. Computer Vision System Toolbox$^{\mathrm{TM}}$ provides algorithms and tools for video processing workflows. You can read and write from common video formats, perform common video processing algorithms such as de-interlacing and chroma-resampling, and display results with text and graphics burnt into the video. Video processing in MATLAB uses system objects, which avoids excessive memory use by streaming data to and from video files [10].

Another tool called ffmpeg is also used to generate detailed information about each and every frame of a video including pict_type which can be I,B or P, stream_index, whether it is an index frame or not, pkt_duration, pkt_position and many more useful features. FFmpeg is the leading multimedia framework, able to *decode*, *encode*, *transcode*, *mux*, *demux*, *stream*, *filter* and *play* pretty much anything that humans and machines have created. It supports the most obscure ancient formats up to the cutting edge. No matter if they were designed by some standards committee, the community or a corporation. It is also highly portable: FFmpeg compiles, runs, and passes our testing infrastructure across Linux, Mac OS X, Microsoft Windows, the BSDs, Solaris, etc. under a wide variety of build environments, machine architectures, and configurations.

The PLOT function in MATLAB is used to show the frame-wise temporal differences in actual as well as reconstructed video. The plot function in MATLAB is used to create a graphical representation of some data. It is often very easy to "see" a trend in data when plotted, and very difficult when just looking at the raw numbers.

3.2 Data set

Experiment is performed on around 200 video clips of length between 8 and 23 s. The whole dataset is randomly divided into four small datasets (DS1, DS2, DS3 and DS4) containing 50 video clips each. The full length videos are divided into frames and stored in separate folders. The number of extracted frames in these videos ranges from 197 to 568 frames. Format of input videos are MP4 and AVI. MATLAB code is used to divide videos into pictures and to save these pictures in JPEG format in separate folders. In this paper, an experiment performed on a video with 329 frames and 13 s in length is explained. First of all, all the 329 frames are extracted a JPEG pictures in a folder. Motion vectors are calculated by subtracting old image from current image. Based on these calculated motion vectors, Frame prediction error is calculated for each frame from second frame onwards. Following is the piece of code which shows the code for calculation of prediction error [10]:

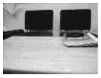
$$\begin{aligned}&\%\,\text{frame prediction error (current image} - \text{old image moved}\\&\text{by motion vectors pf (rows:rows} + \text{N} - 1,\ \text{cols:cols} + \text{N} - 1)\\&= \text{im\_new1 (rows} + \text{N:rows} + \text{N2} - 1,\ \text{cols} + \text{N:cols} + \text{N2} - 1)\\&- \text{im\_old1 (rows} + \text{N} + \text{y1:rows} + \text{y1} + \text{N2} - 1,\\&\text{cols} + \text{N} + \text{x 1 : cols} + \text{x1} + \text{N2} - 1).\end{aligned}$$

The calculated temporal differences are stored in an excel sheet using table() and writetable() commands of MATLAB. Table 1 shows sample of frame wise temporal differences of actual video in hand. Table 2 shows sample of frame wise temporal differences of reconstructed video frames.

Table 1 Sample of frame-wise temporal differences of actual video in hand

Frame1	Frame2	Temporal_Difference
		0.005138
		0.003847
		0.004006
		0.004182
		0.00206
		0.010293
		0.00316
		0.002813

Table 2 Sample of frame-wise temporal differences of reconstructed form of actual video

Frame1	Frame2	Temporal_difference
		0.003929
		0.004102
		0.004304
		0.002118
		0.010277
		0.003218
		0.002961
		0.003573

As the next step, frames are reconstructed based on calculated motion vector and temporal differences. Now, frame wise temporal differences of actual video frames and reconstructed video frames are compared. For comparison, MATLAB plots are used which clearly show it difference lies between the two. If there are phenomenal or remarkable differences in the plot, the video is classified as tampered one. If the two plots closely resemble each other, the video is genuine.

For verifying correctness of proposed algorithm, the same experiment has been performed on forged video as well. The video is deliberately forged by overwriting 40 frames starting from frame number 121. Total number of frames in actual and forged video are kept same i.e. 329. The temporal difference of forged video and its reconstructed video are shown in Tables 3 and 4, respectively.

4 Result and discussion

Based on calculated temporal difference for frames of both the actual and reconstructed video, plots has been drawn in MAT LAB. While plotting, X-axis is used to show number of frames with a class difference of 50 frames. On Y-axis, prediction error is put with a class difference of 0.002 units. Figure 4 shows first plot which is for actual video in hand (which is genuine in this case).

Figure 5 shows second plat with double arguments with same scale, one for actual video and other for its reconstructed form. Blue colour is used for actual video while red colour is used for its reconstructed form. It can be clearly seen that there are subtle differences between the two plots.

Figures 6 and 7 shows the above two plots for forged video and its reconstructed form. It can be clearly seen from these plots that the natural temporal difference between the frames are disturbed around frame number 121 where forgery has been deliberately introduced.

It can be clearly seen from Fig. 7 that the natural temporal difference from frame number 121 is disturbed. The blue and red plots are quite different in region from frame number 121–40 frames beyond it. Hence, if the two plot differs, the video can be classified as forged video.

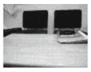
Table 5 shows prediction error in actual video at frame number 121 onwards. Table 6 shows prediction error in forged video at frame number 121 onwards due to disturbed temporal difference.

It is clearly evident from the above two tables that the temporal differences are different when a malicious altering has been performed on the video. Few performance measures are also calculated for the proposed algorithm. These are test efficiency (TE), sensitivity, precision and F1 score (Table 7):

Table 3 Sample of frame-wise temporal differences of forged video

Frame1	Frame2	Temporal_Difference
		0.005138
		0.003847
		0.004006
		0.004182
		0.00206
		0.010293
		0.00316
		0.002813

Table 4 Sample of frame-wise temporal differences of reconstructed form of forged video

Frame1	Frame2	Temporal_Difference
		0.003929
		0.004102
		0.004304
		0.002111
		0.010275
		0.003218
		0.002961
		0.003573

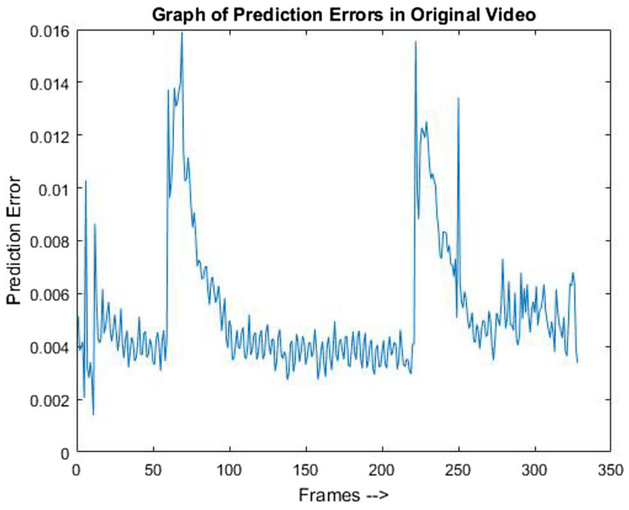


Fig. 4 Frame-wise temporal differences in actual video in hand

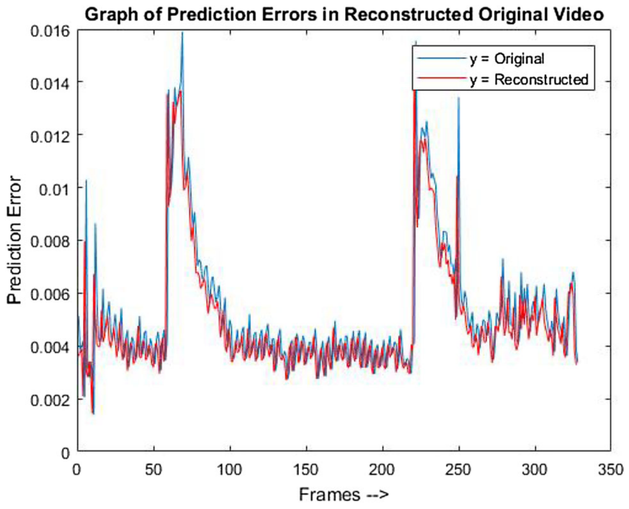


Fig. 5 Frame-wise temporal differences in actual video and its reconstructed form

Test efficiency = (TP + TN)/total number of clips analysed,

Sensitivity or TP rate = TP/total number of actual positive results,

Precision = TP/total number of positive results given by the test,

F1 score = (2 × sensitivity × precision)/(sensitivity + precision).

5 Conclusion

The proposed algorithm, extracts robust and compact features from digital videos in an efficient manner. These features are extracted from digital video clips containing both spatial and temporal information about a video segment. The proposed algorithm detects and localizes the

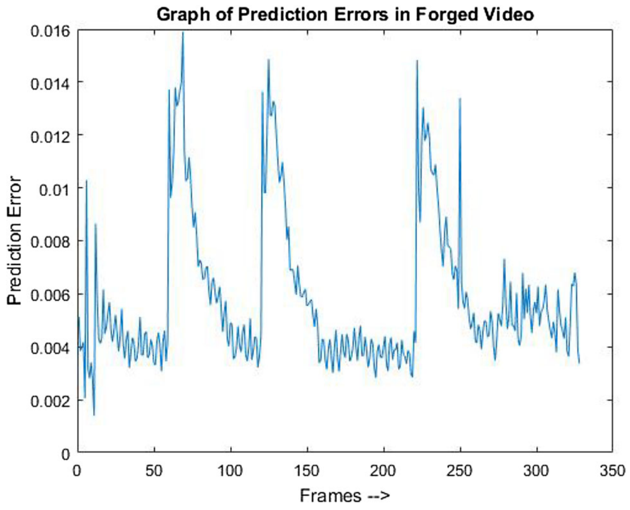


Fig. 6 Frame-wise temporal differences in forged video

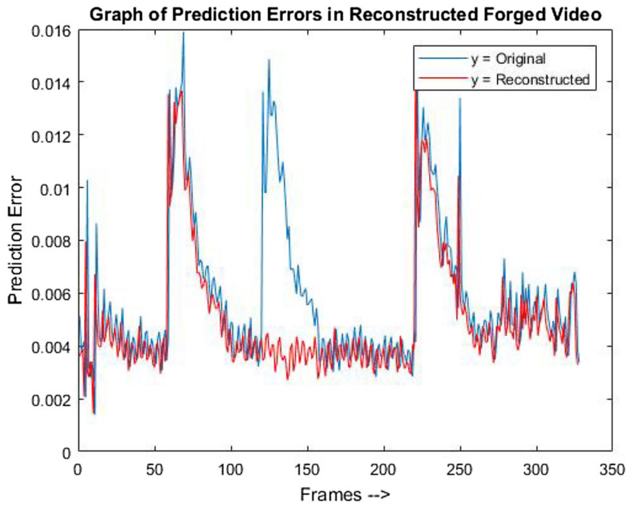


Fig. 7 Frame-wise temporal differences in forged video and its reconstructed form

tampered region in a digital video. Experimental results show that the proposed method achieves promising accuracy in classifying genuine videos and forgeries. The performance of the proposed system gives high true positive rate and low false positive rate. The limitation of the all fingerprint based techniques is that, they work well with long video sequences but do not perform that effectively with short video clips. To be more specific, the proposed algorithm work very accurately for video clips more than 7 s in length or containing more than 200 frames. Extensive experimental results show that the proposed method achieves promising accuracy in classifying genuine videos and forgeries. The Results show that the proposed tampering detection algorithm can detect and precisely locate tampering with an average accuracy of 87.5%.

Table 5 Sample prediction error in actual video at frame number 121 onwards

Frame1	Frame2	Temporal_Difference
		0.004619
		0.003519
		0.003664
		0.004521
		0.004844
		0.004023
		0.003684
		0.004279

Table 6 Sample prediction error in forged video at frame number 121 onwards

Frame1	Frame2	Temporal_Difference
		0.013635
		0.009832
		0.009803
		0.012128
		0.01487
		0.012722
		0.012743
		0.013275

Table 7 Performance measures of proposed algorithm

Dataset	No. of clips < 10 s	No. of clips > 10 s	TP	TN	FP	FN	Test efficiency	Sensitivity	Precision	F1 score
DS1	3	47	27	20	3	0	94.0	100	90.00	94.73
DS2	6	44	10	34	1	5	88.0	66.67	90.90	76.92
DS3	12	38	33	5	9	3	76.0	91.67	78.57	84.62
DS4	4	46	8	38	3	1	92.0	88.89	72.72	79.97
Combined	25	175	78	97	16	9	87.5	89.88	83.33	86.48

6 Future extension

Future scope for the proposed method is to implement this algorithm using machine learning algorithm for classification. Binary classifier with supervised learning would be helpful for implementing proposed algorithm. The linear regression would be the specific choice. Also, the algorithm has a scope to be more generic which can be applied to a wide range of video formats.

References

1. Bestagini P, Fontani M, Milani S, Barni M (2012) An overview on video forensics. In: 20th European signal processing conference (EUSIPCO 2012). IEEE, Bucharest, Romania
2. Stamm MC, Lin WS, Liu KJR (2012) Temporal forensics and anti-forensics for motion compensated video. IEEE Trans Inf Forensics Secur 7(4):1315–1329. https://doi.org/10.1109/TIFS.2012.2205568
3. Yin P, Yu HH (2001) Classification of video tampering methods and countermeasures using digital watermarking. In: Proceedings of the SPIE, multimedia systems and applications IV, vol 4518, pp 239-246
4. Joshi V, Jain S (2015) Tampering detection in digital video—a review of temporal fingerprints based techniques. In: 2015 2nd international conference on computing for sustainable global development (INDIACom), New Delhi, 2015, pp 1121–1124
5. Wang W, Farid H (2006) Exposing digital forgeries in video by detecting double MPEG compression. In: Proceedings of the ACM multimedia and security workshop, Geneva, Switzerland, 2006, pp 37–47
6. Jia S, Xu Z, Wang H, Feng C, Wang T (2018) Coarse-to-fine copy-move forgery detection for video forensics. IEEE Access 6:25323–25335. https://doi.org/10.1109/ACCESS.2018.2819624
7. Joshi V, Jain S, Bansal C (2018) B-frames: efficiency analysis for digital video tampering detection in videos with variable GOP structure. Int J Comput Sci Eng 6(5):808–815
8. Bestagini P, Milani S, Tagliasacchi M, Tubaro S (2013) Local tampering detection in video sequences. In: MMSP'13, Sept. 30–Oct. 2, 2013, Pula (Sardinia), Italy. 978-1-4799-0125-8/13/$31.00 ©2013 IEEE
9. Kingra S, Aggarwal N, Singh RD (2017) Inter-frame forgery detection in H. 264 videos using motion and brightness gradients. Multimed Tools Appl 76:25767. https://doi.org/10.1007/s11042-017-4762-2
10. A library for MATLAB code support. https://sites.google.com/site/santhanarajarunachalam/. Accessed July 2018