Video Forgery Detection by Bitstream Analysis

Hugo Jean[1], Emmanuel Giguet[1] and Christophe Charrier[1]

[1]*Normandie Univ., UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen*

Abstract

In this paper, we propose a video tampering detection method based on bitstream analysis for videos in H.264 or MPEG-4 AVC format. This method aims at detecting inter-frame alterations: insertion, deletion, permutation, duplication. Features are extracted from the original bitstream. This method therefore does not require the decoding of the video, which improves the speed of analysis. The detection quality remains very significant in terms of binary detection, tampered / pristine video, with a F1 measure equal to 94.89. Concerning multiclass classification, F1 measure reaches 70.33 due to the difficulty to separate swap and duplication forgeries.

Keywords

Digital investigation, Video forensics, Video forgery, Forgery detection, Machine learning, Bitstream analysis

1. Introduction

Nowadays, video content is transmitted in exponentially increasing volumes. Most of them are intended to be shared on social networks which have become so popular. This growth has been facilitated by the creation of powerful, easy-to-use video editing tools. Video editing has never been so easy: videos can be combined, unwanted part can be deleted, amazing scenes can be duplicated, frames can be altered to make objects or people disappear or appear, and this, according to one's desires or motivations. Technological advances in image and video editing have unleashed creativity, for humorous or artistic purposes, but also for misinformation, propaganda, and conspiracy. Therefore, the legitimacy, the reliability and the authenticity of the videos which broadcasted and relayed on Internet have become a major concern, in particular to detect disinformation attempts. Legally speaking, videos can now be used as evidence in court. The intentional modification of a video for the purpose of falsification, called video forgery, must be detectable. The challenge is to determine whether the video has been altered and, if possible, to qualify the nature of the alterations.

Many forgery detection methods have been proposed, but they are generally unable to detect simultaneously the different types of existing forgeries. Moreover, they require the entire video to be decoded beforehand in order to perform these detections.

In this work, we propose an original method for detecting inter-frame forgery in H.264 (or MPEG-4 AVC) videos, using the bitstream approach. This method detects insertion, deletion,

The 11th Colour and Visual Computing Symposium, September 08–09, 2022, Gjovik, Norway

emmanuel.giguet@unicaen.fr (E. Giguet); christophe.charrier@unicaen.fr (C. Charrier)
https://giguete.users.greyc.fr/ (E. Giguet); https://charrierc.users.greyc.fr/ (C. Charrier)
0000-0002-3836-9829 (C. Charrier)

permutation and duplication of frames. It is based on feature extraction directly from the bistream, *i.e.*, from the compressed domain. Anomalies detection into a video are performed analyzing the variation of statistics computed on video fragments, taking into account the variation of the forward and backward motion vectors into the B and P frames by minimizing the false positives.

The paper is organized as follows. In Section 2, we introduce the current state-of-the-art for detecting inter-image falsification in video forgery. Our proposed methodology is then described in Section 3, including feature extraction and selected classification methods. In section 4, we provide a detailed description of the evaluation environment we set up for this study, including dataset construction, performance metrics, and two evaluation scenarii: a binary classification task and a multi-class classification task. In Section 5, we present the results we obtained for each scenario. Section 6 offers concluding remarks.

2. State-of-the-art

In the literature, many methods for detecting inter-image falsifications are present. Whether these techniques are applied at the local level by LBP [1], by similarity measure computation [2] like, for example, the MS-SSIM quality measure [3], by computing the Zernike opposite chromaticity moments [4], or even the histograms of oriented gradients and motion energy images [5], the announced performances are of high level. However, they decrease rapidly when the training conditions are more or less respected (dynamic video background, static video, and son on).

In 2014, Zhang *et al* [1] calculated the correlation between each adjacent frame encoded with the LBP approach, to decipher the frame insertion and frame deletion fakes in a video. If the number of frame deletions is small, the performance of this technique degrades. Li et al. [2] used the consistency of the quotient-of-mean structural similarity measure (QoMSSIM) to detect frame insertions and deletions. QoMSSIM is used as a feature and feeds an SVM classifier to detect the types of falsifications. However, the performance degrades when the videos are static, as is the case in video surveillance. Liu et al. [4] proposed an approach based on coarse-to-fine investigation to detect tampering types by inserting, deleting, duplicating, and replacing frames in videos. In coarse detection, abnormal frame locations are detected using Zernike Opponent Chromaticity Moments (ZOCM–Zernike Opponent Chromaticity Moments). All images are transformed into color opposition space, and the Zernike moment correlation is calculated over the color space to obtain the ZOCM value. The coarse Tamura feature is extracted from the detected anomalous images, and the fine detection algorithm is run to reduce false positives. However, this approach fails when the background of the videos is dynamic. Recently, Fadl et al. [5] used histograms of oriented gradients (HOG) and motion energy images (MEI) to design a passive detection technique to detect tampering by deleting, inserting, and reshuffling images. However, the performance of the proposed method quickly degrades when deleting images in a static scene video.

Concerning methods based on deep learning features, Long *et al.* [6] used a 3DCNN network to detect frame deletion in a single 16-frame video shot and checked the center of the shot (between frames 8 and 9). They refined the confidence scores using peak detection and temporal scaling to reduce false alarms. They also proposed another method [7] for image duplication using an I3D network (*Two-Stream Inflated 3D ConvNet*). The test video was divided into overlapping shots and the features of each shot were extracted using a pre-trained I3D network, and then the features of all the shots in the video were contacted to calculate the distance between them and detect similarity. Bakas et al. [8] used three pre-trained 3DCNN models to detect deletion, insertion, and duplication of frames in a single video shot. In the proposed model, a difference layer is added in the CNN, which is mainly aimed at extracting temporal information from videos. The authors claim significant performance rates.

In recent years, techniques based on the use of CNNs (3DCNN, 2DCNN, etc.) [6, 7, 8] have been widely used, showing significant performance rates.

All the previous methods rely on accessing the pixels of the video frames and then working in a transformed domain. They therefore require a complete and successful decoding of the encoded video files, which necessarily leads to a significant overall computation time, especially when processing several hours long videos.

3. The proposed approach

In order to be broadcast on the Internet, a video is encoded as a sequence of bits, commonly called a bitstream, using a compression algorithm, or codec. Among the most widely used are the H.264 codec and its successor the H.265 codec. Although the latter is more powerful, the H.264 codec is still widely used on the Internet today because of its better compatibility.

The video forgery detection method proposed here is illustrated in figure 1. From the *bitstream* of the video, a feature extraction is performed using a stream analyzer. This set of features is then used to train learning models to classify the different types of tampering sought.

In the field of compression, a video is represented as a sequence of images. These images, of the intra or inter type, are organized into groups of images (GOP). Each GOP is composed of an intra (I) image, known as the key image, encoded in JPEG. This algorithm takes into account spatial redundancies in order to reduce the amount of data to be encoded. An intra image is followed by several inter images (B, P) represented by a set of motion vectors. These vectors symbolize the displacement of a pixel of the current image with respect to the reference images. This representation abstracts from temporal redundancies while encoding the motion content.

P-frames consider only the previous frames as reference frames while B-frames consider the following frames as additional references. The H.264 codec defines a frame as a set of slices, which are composed of macroblocks. An H.264 bitstream is structured in three layers. The Network Ability Layer (NAL) contains the video data blocks, called Video Coding Layers (VCLs). Each VCL describes a slice of image, named the Slice Layer. This layer is reused as the set of macroblocks that compose it. Each macroblock is finally described by its own characteristics at the level of the Macroblock Layer.

3.1. Features extraction

In order to extract characteristics on the different layers of the bitstream, each VLC is inspected by the flow analyzer. The extracted parameters f_l are the following:

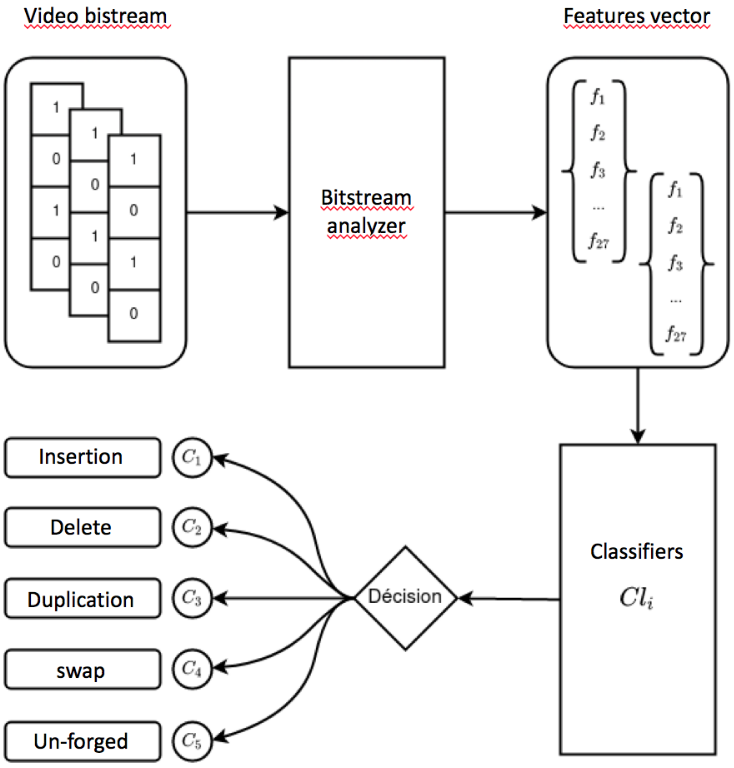

Figure 1: Synopsis of the proposed model.

- f_1 : the bitrate
- f_2 : the average Quantization Parameter (QP)
- f_3 : the QP delta (ΔQP)
- f_4, f_5 : the average and maximum length of the motion vectors
- f_6, f_7 the average and maximum length of the prediction error on the motion vectors
- $f_8, \cdots, f_{10}$: the percentage of intra (I), inter (B, P) and uncoded (skip) macroblocks
- $f_{11}, \cdots, f_{13}$: the percentage of macroblocks of type I having a size of 16x16, 8x8 and 4x4.
- $f_{14}, \cdots, f_{17}$: the percentage of macroblocks of type P having a size of 16x16, 16x8, 8x16 and 8x8.
- $f_{18}, \cdots, f_{20}$: the percentage of sub-macroblocks of type P having a size of 8x4, 4x8 and 4x4.
- $f_{21}, \cdots, f_{24}$: the percentage of type B macroblocks having a size of 16x16, 16x8, 8x16 and 8x8.
- $f_{25}, \cdots, f_{27}$: the percentage of uncoded type B and P macroblocks and direct-coded type B macroblocks

The f_1 parameter is extracted directly at the *Slice Layer* while the rest is extracted at the *Macroblock Layer*. The features f_1, f_2 and f_3 represent the distortion of the video while the moving content is symbolized by the features f_4 to f_7. The encoder choices are finally transcribed from f_8 to f_{27}. Each parameter is extracted for each image slice and then averaged over the current GOP size. The feature vector V_{GOP_k} for each GOP k is finally computed:

$$V_{GOP_k} = \left(\frac{1}{MN} \sum_{j=1}^{M} \sum_{i=1}^{N} f_{l,i,j} \right), \forall l \in [1, \ldots, 27] \tag{1}$$

where $f_{l,i,j}$ represents the l-*th* feature of the i-*th* frame slice of the j-*th* frame of the k-*th* GOP of the video.

3.2. Selected classification methods

There are many binary and multiclass learning techniques in the literature. Their performance varies according to the problem to be solved. However, they are all implemented in software libraries so that it is now possible and easy to test several of them to compare their performance on different data sets. One of the best known and most robust libraries for machine learning is Scikit-learn, also called sklearn.

In order to study the adaptability of existing classification schemes to the bistream data, we compare, among the best performing strategies, the following approaches : [9]: *Gradient Boosting Classifier* (GBC), Light-Gradient Boosting Machine (L-GBM), *Logistic Regression* (RL), *Decision Tree Classifier* (DTC), *Random Forest Classifier* (RFC), *Support-Vector Machine* (SVM) and *K Nearest Neighbors* (KNN).

We also tested the following methods: Ada Boost Classifier (ADA), Extra Trees Classifier (ETC), *Linear Discriminant Analysis* (LDA), *Ridge Classifier* (RC), *Quadratic Discriminant Analysis* (QDA), *Dummy Classifier* (DC) and *Naive Bayes* (NB).

In the end, fourteen methods are compared according to two scenarios: binary or multiclass classification.

4. Experimental Setup

4.1. Video Dataset Design

In order to evaluate our tampering detection method, we had to create our own dedicated video database, a dedicated database has been created, as we could not identify a free database containing the four types of inter-frame alterations (insertion, deletion, duplication, and frame swapping).

To build our artificial database, we proceeded by deriving videos from the LIVE Video Quality Challenge (VQC) database [10, 11] created by the University of California, Berkeley. This database was created by the University of Texas at Austin as part of the LIVE Video Quality Challenge (VQC). This original database consists of 585 unaltered videos featuring a wide variety of scenes, captured from 101 cameras representing 43 models, shot by 80 users, and with varying recording qualities. These videos have an average duration of 10.03 seconds, with variable formats, portrait or landscape, and variable resolutions.

For our evaluation campaign, we automated the creation of the altered video database from the VQC database. We had to define a falsification process covering the 4 types of alterations targeted, with sufficiently varied positions and alteration durations. From 82 videos randomly selected in the VQC database, a database of 410 videos is created by altering each original video according to one of four types. We have created a database of 410 videos by altering each original video in one of four ways: insertion, deletion, duplication and permutation.

To produce a video with insertion, a fragment to be inserted is extracted from a randomly selected video. The duration of this fragment is between 1 second and the total duration. The fragment is then inserted into the target video, at a position between the beginning of the target video and the end of the target video minus the insertion time.

To produce a video with *delete*, we randomly select a fragment to be deleted with a start position between the beginning and 75% of the video, and a random duration between 20 and 100% of the remaining duration.

To produce a video with *duplication*, we randomly select a fragment to duplicate of maximum 33% of the video, and starting between the beginning of the video and the end decreased the duration of the copy. The fragment is then inserted at a random point in the video.

To produce a video with *permutation/swap*, we randomly choose two fragments to permute, without overlapping range. To guarantee the non-overlap of the excerpts, we randomly choose a maximum duration of 33% of the video, and two distant starting points: the beginning of extract1 starts between the beginning and 33% of the video, that of extract2 between 35% and 65% of the video. The two extracts are then swapped.

All such tampered videos are then re-encoded using the H.264 codec using the default Constant Rate Factor (CRF) value equal to 23 in order to get quite good quality videos. Actually, since CRF is a ”constant quality” encoding mode, as opposed to constant bitrate (CBR), it will compress different frames by different amounts, thus varying the Quantization Parameter (QP) as necessary to maintain a certain level of perceived quality.

4.2. Performance metrics

The performance of the 14 trail classification strategies selected was compared according to five criteria:

1. *the accuracy* which is the fraction of correct predictions of the model,
2. *the precision* which is the proportion of positive identification that is really correct,
3. *the recall* which is the proportion of real positives to have been correctly identified,
4. *The F1* score which allows to evaluate the capacity of a classification model to predict efficiently the positive individuals, by making a compromise between precision and recall. It is defined by the harmonic mean of precision and recall,
5. *The AUC* (Area under the ROC Curve) provides an aggregate measure of performance for all possible classification thresholds. One way to interpret the AUC is the probability that the model ranks a random positive example higher than a random negative example.

Model	Accuracy	AUC	Recall	Precision	F1
L-GBM	**91.63**	**93.55**	97.39	92.6	**94.89**
ADA	91.60	93.4	95.61	**94.16**	94.81
GBC	90.21	93.18	96.96	91.56	94.13
ETC	89.51	89.99	**99.57**	88.87	93.87
RFC	89.16	90.43	98.26	89.44	93.58
LR	87.77	87.85	94.33	91.23	92.51
LDA	87.44	88.49	93.87	91.19	92.3
RC	87.06	0.0	96.5	88.76	92.31
DTC	82.88	71.22	90.43	88.49	89.36
QDA	80.09	75.19	87.39	87.72	87.34
DC	80.09	0.5	1.0	80.09	88.94
KNN	78.04	75.31	89.51	84.14	86.64
SVM	76.56	0.0	89.35	82.81	85.
NB	72.04	84.85	68.99	94.68	79.16

Table 1
Binary classifiers performance evaluation.

4.3. Evaluation scenarii

4.3.1. Binary Classification Models

In this scenario, the goal is to classify the video into two classes: forged video, un-forged video. The fourteen patterns presented in were tested to measure their ability to predict the class of the video.

4.3.2. Multi-class classification models

In this second scenario, we tested the ability of different classification models to predict the type of forgery (insertion, deletion, permutation and duplication), or the absence of forgery, using multiclass approaches. In this approach, the 6 models considered are: GBC, L-GBM, LR, DTC, SVM and KNN.

5. Performance Evaluation

5.1. Optimization and model training

Whether for binary or multiclass classification, the best combination of hyperparameters was performed using the *Grid Search* technique.

During the learning phase of the various schemes, 70% of the randomly drawn examples of the database constitute the learning database and the remaining 30% feed the test database. The 10-sub-sample cross-validation ($k = 10$) was used to evaluate the machine learning models.

The feature selection technique, or *Features Selection*, was not chosen as it was not appropriate. This technique is commonly used to select the features contributing to the performance of the model and to discard the less relevant ones. However, this process is not compatible with the

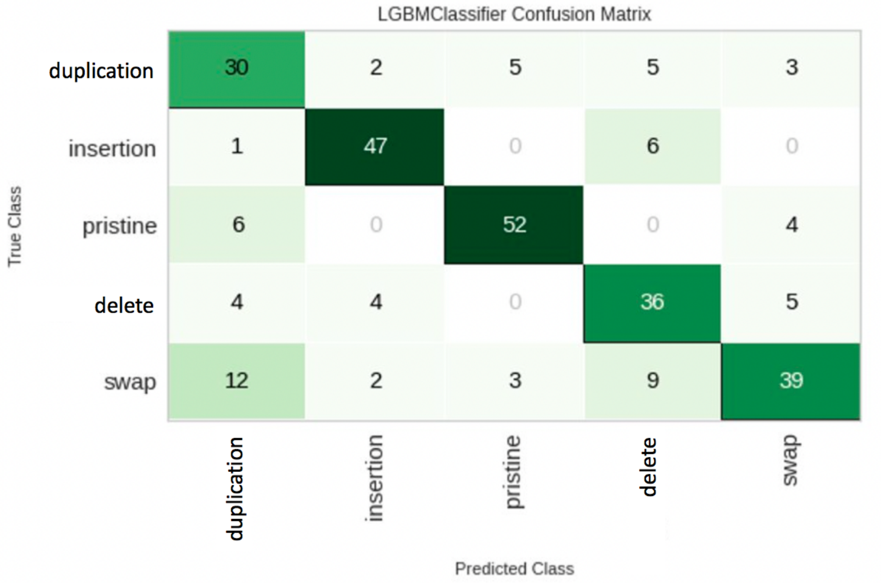

Figure 2: Confusion matrix for results obtained from the LGBM Classifier

fact that the detection of different types of alterations requires considering different subsets of features.

5.2. Results

Table 1 shows the results obtained for the binary classification. The Light Gradient Boosting Machine (L-GBM) obtains the best accuracy (91.63) and the best F1 measure (94.89).

Model	Accuracy	Precision	Recall	F1	AUC
GBC	**70.63**	**71.50**	68.70	**70.33**	90.53
LGB	68.19	69.16	66.38	67.83	90.55
LR	69.93	70.84	**68.74**	69.02	**91.34**
DTC	66.44	68.14	64.68	66.48	79.27
RFC	64.41	63.50	62.06	63.46	85.81
SVM	39.51	42.50	38.27	32.54	0.00
KNN	36.32	39.58	34.92	36.16	67.26

Table 2
Multiclass classifiers performance evaluation

For multiclass classification, the table 2 presents the obtained results. The *Gradient Boosting Classifier* (GBC) obtains the best accuracy (70.63) and the best F1 measure (70.33).

Figure 2 presents the confusion matrix for the best classifier: LGBM. As we can observe, the classifier makes confusion for two kind of forgery: 1) swap and duplication. This is not really surprising since both swap and duplication are performed uusing an extract of the same video and thus, it, in general, is difficult to detect the difference between the a swap of two extracts of a duplication of an extract if a long term memory strategy is not used. One solution would to add such a strategy to be able to distinguish those two kinds of forgery. Except for this case, the obtained results clearly show that the LGBM classifier performs well to identify the type of forgery, and un-forged video.

6. Conclusion

In this paper, we have proposed a video tampering detection method based on bitstream analysis for videos in H.264 or MPEG-4 AVC format. This forgery detection method aims at identifying inter-frame alterations: insertion, deletion, permutation, duplication. In our approach, the features taken into account during classification are directly derived from the file's bit sequence.

This video forgery detection method has the advantage to prevent decoding the video. Thus, it permits very fast and memory efficient analysis of the files. The binary classification, forged / un-forged video, remains very qualitative with an F1 measure equal to 94.89. It is obtained with the *Light-Gradient Boosting Machine* classification model. The multi-class classification task leads to promising results, with an F1 measure value equal to 70.33. It is obtained with the *Gradient Boosting Classifier* classification method.

References

[1] Z. Zhang, J. Hou, Q. Ma, Z. Li, Efficient video frame insertion and deletion detection based on inconsistency of correlations between local binary pattern coded frames, Security and Communication Networks 8 (2015) 311–320.

[2] Z. Li, Z. Zhang, S. Guo, J. Wang, Video inter-frame forgery identification based on the consistency of quotient of mssim, Security and Communication Networks 9 (2016) 4548–4556.

[3] Z. Wang, E. P. Simoncelli, A. C. Bovik, Multi-scale structural similarity for image quality assessment, in: IEEE Asilomar Conference on Signals, Systems, and Computers, 2003, pp. 1398–1402.

[4] Exposing video inter-frame forgery by zernike opponent chromaticity moments and coarseness analysis, Multimedia Systems 23 (2017) 223–238.

[5] S. Fadl, Q. Han, Q. Li, Surveillance video authentication using universal image quality index of temporal average, in: C. D. Yoo, Y.-Q. Shi, H. J. Kim, A. Piva, G. Kim (Eds.), Digital Forensics and Watermarking, Springer International Publishing, 2019, pp. 337–350.

[6] C. Long, E. Smith, A. Basharat, A. Hoogs, A c3d-based convolutional neural network for frame dropping detection in a single video shot, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 1898–1906.

[7] C. Long, A. Basharat, A. Hoogs, A coarse-to-fine deep convolutional neural network framework for frame duplication detection and localization in video forgery, ArXiv abs/1811.10762 (2018).

[8] J. Bakas, R. Naskar, A digital forensic technique for inter-frame video forgery detection based on 3d cnn, in: Information Systems Security, Springer International Publishing, 2018, pp. 304–317.

[9] C. M. Bishop, N. M. Nasrabadi, Pattern recognition and machine learning, volume 4, Springer, 2006.

[10] Z. Sinno, A. C. Bovik, Large-scale study of perceptual video quality, IEEE Transactions on Image Processing 28 (2019) 612–627. doi:10.1109/TIP.2018.2869673.

[11] Z. Sinno, A. C. Bovik, Large scale subjective video quality study, in: 2018 25th IEEE International Conference on Image Processing (ICIP), 2018, pp. 276–280.