Analyse du bitstream pour la détection de Falsification Video

Paul CANCHON, Hugo JEAN, Hugo MERLY, Emmanuel GIGUET, Christophe CHARRIER

Normandie Univ., UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen
hugo.merly@ensicaen.fr, emmanuel.giguet@cnrs.fr, christophe.charrier@unicaen.fr

Résumé – Dans cet article, nous proposons une méthode de détection de falsification vidéo basée sur l'analyse du bitstream pour les vidéos au format H.264 ou MPEG-4 AVC. Cette méthode a pour objectif de détecter les altérations inter-frames : insertion, suppression, permutation, duplication. Les caractéristiques prises en compte pour la classification sont directement issues de la séquence de bits du fichier. La méthode a par conséquent l'avantage de ne pas requérir le décodage de la vidéo, ce qui permet d'obtenir une analyse rapide et économique des fichiers. La qualité de la détection reste par ailleurs très significative en terme de détection binaire, vidéo falsifiée / non falsifiée, avec une f-mesure de 94.89, et une f-mesure de 70.33 pour la classification multiclasse.

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 an f-measure of 94.89, and an f-measure of 70.33 for multiclass classification.

1 Introduction

De nos jours, les contenus vidéos sont transmis dans des volumes en croissance exponentielle. Elles ont, pour la plupart, vocation à être partagées sur les réseaux sociaux plébiscités par le grand public. Cet essor a été favorisé par la création d'outils d'édition puissants, faciles à utiliser, qui permettent de personnaliser les contenus vidéos. Les montages n'ont jamais été aussi simples à réaliser : les vidéos sont assemblées, certains passages sont effacés ou au contraire dupliqués, certaines portions d'images peuvent être altérées pour faire disparaître ou au contraire apparaître des éléments, et cela, au gré des envies ou des motivations. Les avancées technologiques en manière de manipulation d'images et de vidéos ont démocratisé les usages, qu'ils soient ludiques ou artistiques, mais également propagandistes ou complotistes. Dès lors, la légitimité, la fiabilité et l'authenticité des vidéos diffusées et relayées sur les réseaux sont devenues une préoccupation majeure, notamment pour détecter les tentatives de désinformation. En matière légale, les vidéos peuvent aujourd'hui servir de preuve devant la justice. La modification intentionnelle d'une vidéo à des fins de falsification, appelée contrefaçon vidéo, doit pouvoir être détectée. Le défi consiste à déterminer si la vidéo a été modifiée, et, dans la mesure du possible, à qualifier la nature des altérations.

De multiples méthodes de détection falsifications ont été proposées, mais elles sont généralement incapables de détecter simultanément les différents types de falsifications existantes et nécessitent de décoder préalablement la totalité de la vidéo afin d'effectuer ces détections.

Dans ce travail, nous proposons une technique de détection des falsifications inter-trames dans les vidéos au format H.264 (ou MPEG-4 AVC). Cette technique permet de détecter l'insertion, la suppression, la permutation et la duplication de trames. Notre approche est basée sur l'extraction de caractéristiques du domaine compressé, appelée approche *bitstream*. Plus précisément, nous exploitons la variation des statistiques des scènes naturelles pour détecter les anomalies dans une séquence vidéo. Les anomalies détectées font l'objet d'un examen plus approfondi, en prenant en compte la variation des vecteurs de mouvement avant et arrière dans les images B et P en optimisant le taux de faux positifs.

2 Etat de l'art

Dans la littérature, de nombreuses méthodes de détection de falsifications inter-images sont présentes. Que ces techniques s'appliquent au niveau local par LBP [1], par calcul de mesure de similarité [2] à l'instar de la mesure de qualité MS-SSIM [3], par calcul des moments de chromaticité opposés de Zernike [4], voire des histogrammes des gradients orientés et des images d'énergie de mouvement [5], les performances annoncées sont de haut niveau. Cependant elles décroissent rapidement lorsque les conditions d'apprentissage sont peu ou prou respectées (arrière plan des vidéos dynamique, vidéo statique, etc.).

Ces dernières années, les techniques basées sur l'utilisation des CNN (3DCNN, 2DCNN, etc.) [6, 7, 8] ont été largement plébiscitées affichant des taux de performance importants.

Toutes les méthodes précédentes reposent sur l'accès aux pixels des trames de la vidéo pour ensuite éventuellement travailler dans un domaine transformé. Elles requièrent donc un décodage complet et réussi des fichiers vidéos codés, ce qui entraîne nécessairement un temps de calcul global conséquent, surtout lorsque des vidéos de plusieurs heures sont traitées.

3 La méthode proposée

Afin d'être diffusée sur internet, une vidéo est encodée sous forme d'une séquence de bits, communément appelée *bitstream*, à l'aide d'un algorithme de compression, ou codec. Parmi les plus utilisés figurent le codec H.264 et son successeur le codec H.265. Bien que ce dernier soit plus performant, le codec H.264 est encore aujourd'hui très largement utilisé sur internet grâce à sa meilleure compatibilité.

La méthode de détection de falsification vidéo proposée ici est illustrée dans la figure 1. À partir du *bitstream* de la vidéo, une extraction de caractéristiques est réalisée à l'aide d'un analyseur de flux. Cet ensemble de caractéristiques est ensuite utilisé pour entraîner des modèles d'apprentissage à classer les différents types de falsification recherchés.

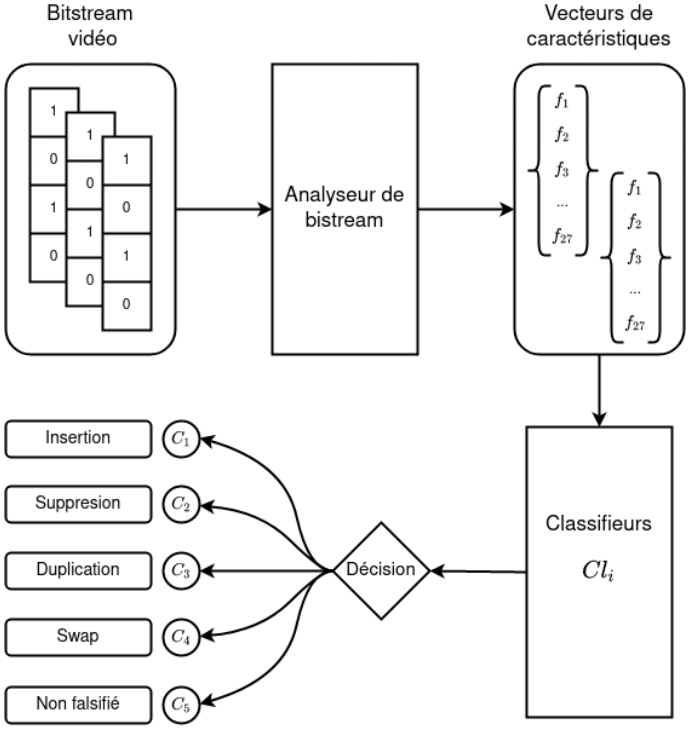

FIG. 1: *Illustration du modèle proposé*

3.1 Extraction des caractéristiques

Dans le domaine de la compression, une vidéo est représentée comme une suite d'images. Ces images, de type intra ou inter, sont organisés en groupe d'images (GOP). Chaque GOP est composé d'une image intra (I), dite image clé, encodée en JPEG. Une image intra est suivie de plusieurs images inter (B, P) représentées par un ensemble de vecteurs de mouvement. Les images P ne considèrent que les images précédentes comme images de références alors que les images B considèrent les images suivantes en tant que références supplémentaires. Le codec H.264 définit une image comme un ensemble de tranches, elles-mêmes composées de macroblocs. Un bitstream H.264 est, quant à lui, structuré selon trois couches. Le *Network Abstraction Layer* (NAL) contient les blocs de données vidéo, appelés *Video Coding Layers* (VCL). Chaque VCL décrit une tranche d'image, nommé le *Slice Layer*. Cette couche est représentée comme l'ensemble des macroblocs qui la compose. Chaque macrobloc est enfin décrit par ses caractéristiques propres au niveau du *Macroblock Layer*.

Afin d'extraire des caractéristiques sur les différentes couches du bitstream, chaque VLC est inspectée par l'analyseur de flux. Les paramètres extraits f_l sont les suivants :

- f_1 : le bitrate
- f_2 : le paramètre de quantification moyen (QP)
- f_3 : le delta QP (ΔQP)
- f_4,f_5 : la longueur moyenne et maximum des vecteurs de mouvement
- f_6,f_7 la longueur moyenne et maximum de l'erreur de prédiction sur les vecteurs de mouvement
- $f_8, \cdots, f_{10}$: le pourcentage de macroblocs de type intra (I), inter (B, P) et non codés (skip)
- $f_{11}, \cdots, f_{13}$: le pourcentage de macroblocs de type I ayant une taille de 16x16, 8x8 et 4x4.
- $f_{14}, \cdots, f_{17}$: le pourcentage de macroblocs de type P ayant une taille de 16x16, 16x8, 8x16 et 8x8.
- $f_{18}, \cdots, f_{20}$: le pourcentage de sous-macroblocs de type P ayant une taille de 8x4, 4x8 et 4x4.
- $f_{21}, \cdots, f_{24}$: le pourcentage de macroblocs de type B ayant une taille de 16x16, 16x8, 8x16 et 8x8.
- $f_{25}, \cdots, f_{27}$: le pourcentage de macroblocs de type B et P non codés et de macroblocs de type B codés en mode direct

Le paramètre f_1 est extrait directement au niveau du *Slice Layer* alors que le reste est extrait au niveau du *Macroblock Layer*. Les caractéristiques f_1,f_2 et f_3 représentent la distorsion de la vidéo tandis que le contenu en mouvement est symbolisé par les caractéristiques f_4 à f_7. Les choix de l'encodeur sont enfin retranscrits de f_8 à f_{27}. Un vecteur de caractéristique V_{GOP_k} pour chaque GOP k est finalement calculé :

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

où $f_{l,i,j}$ représente la l-*ième* caractéristique de la i-*ième* tranche d'image de la j-*ième* image du k-*ième* GOP de la vidéo.

3.2 Méthodes de classification

Il existe de nombreuses techniques d'apprentissage binaires et multiclasses dans la littérature. Leur performance varie selon le problème à résoudre.

Afin d'étudier l'adaptabilité des schémas de classification existants aux données du bistream, nous comparons, parmi les stratégies les plus performantes, les approches suivantes [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) et *K Nearest Neighbors* (KNN).

Nous avons également testé les méthodes suivantes : Ada Boost Classifier (ADA), Extra Trees Classifier (ETC), *Linear Discriminant*

Analysis (LDA), *Ridge Classifier* (RC), *Quadratic Discriminant Analysis* (QDA), *Dummy Classifier* (DC) et *Naive Bayes* (NB). Au final, quatorze méthodes sont comparées selon deux scenarii : classification binaire ou multiclasses.

4 Évaluation de la performance

4.1 Protocole expérimental

4.1.1 Création d'une base de vidéos altérées

Afin d'évaluer notre méthode de détection de falsification, une base dédiée a été créée, faute d'avoir pu identifier une base libre d'utilisation contenant les quatre types d'altération inter-images (insertion, suppression, duplication, et permutation de frames).

Pour construire notre base artificielle, nous avons procédé par dérivation de vidéos issues de la base LIVE Video Quality Challenge (VQC) [10, 11]. Cette base originale est constituée de 585 vidéos non altérées présentant des scènes de nature très diverse, captées à partir de 101 appareils représentant 43 modèles, filmés par 80 utilisateurs, et présentant des qualités d'enregistrement variées. Ces vidéos ont une durée moyenne de 10.03 secondes, des formats variables, portrait ou paysage, et des résolutions également variables.

À partir de 82 vidéos sélectionnées aléatoirement dans la base VQC, une base de 410 vidéos est créée en altérant chaque vidéo originale selon l'un des quatre types.

Pour produire une vidéo avec *insertion*, un fragment à insérer est extrait d'une vidéo sélectionnée de manière aléatoire. La durée de ce fragment est comprise entre 1 seconde et la durée totale. Le fragment est ensuite inséré dans la vidéo cible, à une position située entre le début de la vidéo cible et la fin de la vidéo cible diminuée de la durée de l'insertion.

Pour produire une vidéo avec *suppression*, nous sélectionnons au hasard un fragment à supprimer dont le début est situé entre le début et 75% de la vidéo, et une durée aléatoire comprise entre 20 et 100% de la durée restante.

Pour produire une vidéo avec *duplication*, nous sélectionnons de manière aléatoire un fragment à dupliquer de durée d'au maximum 33% de la vidéo, et commençant entre le début de la vidéo et la fin diminuée la durée de la copie.Le fragment est ensuite insérer a un point aléatoire de la vidéo.

Pour produire une vidéo avec *permutation*, nous choisissons de manière aléatoire deux fragments à permuter, sans plage de recouvrement. Pour garantir le non-recouvrement des extraits, nous choisissons de manière aléatoire une durée maximale de 33% de la vidéo, et deux points de départs distants.

4.1.2 Indices de performance

Les performances des différentes stratégies de classification retenues ont été comparées selon cinq critères :

1. *l'exactitude* qui est la fraction de prédictions correctes du modèle,
2. *la précision* qui correspond à la proportion d'identification positive réellement correcte,
3. *le rappel* qui est la proportion de positifs réels à avoir été correctement identifiés,

Modèle	Exact.	AUC	Rap.	Prec˙	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

TAB. 1: *Mesure de la performance des classifieurs binaires*

4. *le score F1* qui permet d'évaluer la capacité d'un modèle de classification à prédire efficacement les individus positifs, en faisant un compromis entre la précision et le rappel. Il est définit par la moyenne harmonique de la précision et du rappel,
5. *l'AUC* (*Area under the ROC Curve*) qui fournit une mesure agrégée des performances pour tous les seuils de classification possibles. Une façon d'interpréter l'AUC est la probabilité que le modèle classe un exemple positif aléatoire plus haut qu'un exemple négatif aléatoire.

4.1.3 Modèles de classification binaire

Dans ce scénario, l'objectif est de classer la vidéo en deux classes : vidéo falsifiée, vidéo non falsifiée. Les quatorze modèles présentés en 3.2 ont été testés afin de mesurer leur capacité à prédire la classe de la vidéo.

4.1.4 Modèles de classification multiclasse

Dans ce second scénario, nous avons testé la capacité de différents modèles de classification à prédire le type d'altération (insertion, suppression, permutation et duplication), ou l'absence d'altération, à l'aide d'approches multiclasses. Lors de cette approche, les 6 modèles considérés sont les suivants : GBC, L-GBM, LR, DTC, SVM et KNN.

4.1.5 Optimisation et entraînement des modèles

Que ce soit pour la classification binaire ou la classification multiclasse, la meilleure combinaison d'hyperparamètres a été effectuée à l'aide de la technique du *Grid Search*.

Lors de la phase d'apprentissage des divers schémas, 70% des exemples de la base tirés aléatoirement constituent la base d'apprentissage et les 30% restant alimentent la base de test. La validation croisée (*k-Fold cross-valisation*) à 10 sous-échantillons ($k = 10$) a été utilisée pour l'évaluation des modèles d'apprentissage automatique.

La technique de sélection des caractéristiques, ou *Features Selection*, n'a pas été retenue car non opportune. Cette technique est couramment utilisée pour sélectionner les caractéristiques contribuant à la performance du modèle et écarter les moins

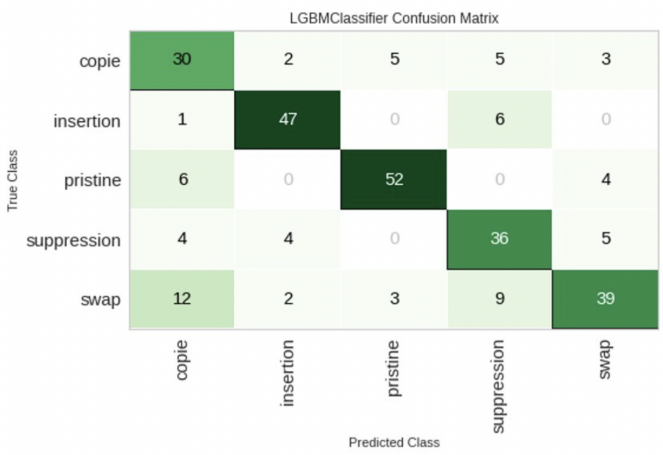

FIG. 2: *Matrice de confusion pour le classifieur LGBM.*

pertinentes. Ce procédé n'est cependant pas compatible avec le fait que la détection des différents types d'altération nécessite de considérer des sous-ensembles de caractéristiques différents.

4.2 Résultats

Le tableau 1 présente les résultats obtenus pour la classification binaire. Le Light Gradient Boosting Machine (L-GBM) obtient la meilleure exactitude (91.63) et la meilleure f-mesure (94.89).

Pour la classification multiclasse, le tableau 2 présente les résultats obtenus. Le *Gradient Boosting Classifier* (GBC) obtient la meilleure exactitude (70.63) et la meilleure f-mesure (70.33).

Modèle	Exact.	Prec.	Rap.	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

TAB. 2: *Mesure de la performance des classifieurs multiclasses*

La figure 2 présente la matrice de confusion pour le classifieur LGBM. A l'exception de l'échange et la duplication, les résultats obtenus montrent clairement que le classifieur LGBM est performant pour identifier le type de contrefaçon, ainsi que les vidéos non falsifiées.

5 Conclusion

Dans cet article, nous avons proposé une méthode de détection de falsification vidéo basée sur l'analyse du bitstream pour les vidéos au format H.264 ou MPEG-4 AVC. Cette méthode a pour objectif de détecter les altérations inter-frames : insertion, suppression, permutation, duplication. Dans cette approche, les caractéristiques sont directement issues de la séquence de bits du fichier. Cette méthode a ainsi l'avantage de ne pas requérir le décodage de la vidéo, ce qui permet d'obtenir une analyse rapide et économique des fichiers. La classication binaire, vidéo falsifiée / non falsifiée, reste par ailleurs très qualitative avec une f-mesure de 94.89 obtenue avec le modèle de classification *Light-Gradient Boosting Machine*. Les essais de classification multiclasse nous ont par ailleurs permis d'obtenir des résultats prometteurs, avec une f-mesure de 70.33 obtenue avec la méthode classification *Gradient Boosting Classifier*.

Références

[1] Zhenzhen Zhang, Jianjun Hou, Qinglong Ma, and Zhaohong Li. Efficient video frame insertion and deletion detection based on inconsistency of correlations between local binary pattern coded frames. *Security and Communication Networks*, 8(2):311–320, 2015.

[2] Zhaohong Li, Zhenzhen Zhang, Sheng Guo, and Jinwei Wang. Video inter-frame forgery identification based on the consistency of quotient of mssim. *Security and Communication Networks*, 9(17):4548–4556, 2016.

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

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

[5] Sondos Fadl, Qi Han, and Qiong Li. Surveillance video authentication using universal image quality index of temporal average. In Chang D. Yoo, Yun-Qing Shi, Hyoung Joong Kim, Alessandro Piva, and Gwangsu Kim, editors, *Digital Forensics and Watermarking*, pages 337–350. Springer International Publishing, 2019.

[6] Chengjiang Long, Eric Smith, Arslan Basharat, and Anthony 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)*, pages 1898–1906, 2017.

[7] Chengjiang Long, Arslan Basharat, and 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] Jamimamul Bakas and Ruchira Naskar. A digital forensic technique for inter-frame video forgery detection based on 3d cnn. In *Information Systems Security*, pages 304–317. Springer International Publishing, 2018.

[9] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[10] Zeina Sinno and Alan Conrad Bovik. Large-scale study of perceptual video quality. *IEEE Transactions on Image Processing*, 28(2):612–627, 2019.

[11] Zeina Sinno and Alan C. Bovik. Large scale subjective video quality study. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 276–280, 2018.