Détection de Deepfakes par Réseaux de Convolution : Performances et Limites Actuelles

Mohamed Mehdi Atamna, Iuliia Tkachenko, Serge Miguet

Détection de Deepfakes par Réseaux de Convolution : Performances et Limites Actuelles

Mohamed Mehdi ATAMNA, Iuliia TKACHENKO, Serge MIGUET

Univ Lyon, Univ Lyon 2, CNRS, INSA Lyon, UCBL, Centrale Lyon, LIRIS, UMR5205, F-69676 Bron, France
{mehdi.atamna, iuliia.tkachenko, serge.miguet}@liris.cnrs.fr

Résumé – Le perfectionnement des méthodes de trucage des visages (*deepfakes* en anglais) rend nécessaire le développement d'outils de détection efficaces. Même si les outils de détection les plus élaborés aujourd'hui—basés sur l'apprentissage profond—obtiennent de bons résultats, leur capacité à généraliser la détection à de nouveaux types de manipulations non rencontrés pendant l'entrainement reste très limitée. Dans cet article, nous réalisons une étude afin de montrer les performances ainsi que les limites des méthodes actuelles en évaluant XceptionNet [3], une architecture de l'état de l'art, sur un type de *deepfakes* non vu à l'entrainement et en utilisant t-SNE [8], un puissant outil de visualisation.

Abstract – The rapid development in face manipulation (i.e., deepfake) methods makes the development of effective detection tools necessary. Even if today's most effective deepfake detectors—based on deep learning—achieve good results, they still suffer from a limited ability to generalize their detection performance to novel types of deepfakes unseen during training. In this paper, we undertake a study to show the performance as well as the limits of the current methods by evaluating XceptionNet [3], a state-of-the-art architecture, on a type of deepfake that is unseen during training and by using t-SNE [8], a powerful visualization tool.

1 Introduction

Grâce aux progrès rapides de l'apprentissage profond, les *deepfakes*, images permettant de truquer un contenu en remplaçant le visage d'une personne par celui d'une autre ou en modifiant ses expressions faciales, deviennent de plus en plus sophistiqués et difficiles à détecter. Ceci soulève d'importantes questions vis-à-vis de la fiabilité du contenu numérique et de la propagation de fausses informations.

Par conséquent, la détection de ce type de trucages requiert le développement de méthodes robustes capables de reconnaître efficacement les trucages générés par différentes approches. Les approches de détection basées sur l'apprentissage profond atteignent actuellement les meilleures performances [9, 13, 12] sur les ensembles de données de *deepfakes* les plus utilisés dans l'état de l'art. Cependant, leur performance diminue fortement lorsqu'elles sont confrontées à des *deepfakes* générés par des méthodes non vues pendant l'entrainement.

Dans cet article, en plus de montrer les performances des méthodes de l'état de l'art, nous illustrons cette difficulté à généraliser la détection à de nouvelles manipulations en entrainant un réseau de convolution largement utilisé, XceptionNet [3], sur l'ensemble de données de vidéos FaceForensics++ [9] et en l'évaluant sur un autre ensemble de données, DeeperForensics-1.0 [5]. Nous présentons, en plus des résultats de classification, la projection en deux dimensions grâce à t-SNE [8] des caractéristiques apprises par le réseau afin de souligner les limites des méthodes de détection actuelles et de mettre en avant la nécessité de développer des méthodes plus robustes.

2 Procédure expérimentale

Dans un contexte d'apprentissage supervisé, nous posons le problème comme étant un problème de classification d'images multiclasse, où le réseau XceptionNet est entrainé à reconnaître les images non manipulées ainsi que chaque type de manipulation présent dans les ensembles de données utilisés.

Pour entrainer XceptionNet, nous reprenons la procédure et les hyperparamètres décrits dans [9]. Nous remplaçons la couche complètement connectée d'une version d'XceptionNet pré-entrainée sur ImageNet [4] par une couche adaptée au nombre de classes traitées ici et la pré-entrainons pendant trois époques. L'intégralité du réseau est ensuite entrainée pendant 15 époques. 75 % des données sont utilisées pour l'entrainement, 10 % pour la validation et 15 % pour le test. Les résultats fournis concernent les performances sur les données de test.

2.1 Ensembles de données

Dans cet article, nous utilisons deux ensembles de données : FaceForensics++ et DeeperForensics-1.0. Ces ensembles contiennent des vidéos sans trucage ainsi que des vidéos dont les visages des sujets sont manipulés.

Pour chaque vidéo, le visage du sujet est extrait des 120 premières images en utilisant le détecteur MTCNN [14], la boîte englobante élargie de 20 % en hauteur et en largeur et l'image résultante redimensionnée en 224×224.

Les images extraites sont nettoyées manuellement afin de ne garder que les 120 premières images du visage du sujet principal car les méthodes de détection des visages peuvent détecter

plusieurs sujets ou encore faire des détections erronées. [1] Ce nettoyage a permis d'améliorer les résultats lors de tests réalisés précédemment.

2.1.1 FaceForensics++

FaceForensics++ (FF++) est constitué de 1000 vidéos non manipulées (étiquetées comme *Real* dans cet article) et 5000 vidéos truquées en utilisant cinq différentes méthodes de trucage (DeepFakes [1], Face2Face [10], FaceShifter [7], FaceSwap [2], NeuralTextures [11]) de 1000 vidéos chacune. Ces méthodes reposent sur de l'apprentissage profond (DeepFakes, FaceShifter), sur des techniques d'infographie (Face2Face et FaceSwap), ou encore sur une combinaison des deux approches (NeuralTextures).

Nous utilisons la version de cet ensemble qui contient les vidéos faiblement compressées (H.264 avec le niveau de compression C23).

2.1.2 DeeperForensics-1.0

DeeperForensics-1.0 (DF-1.0) est constitué de 50,000 vidéos non manipulées et 10,000 vidéos manipulées en utilisant une méthode à base d'auto-encodeurs [6]. Nous utilisons le sous-ensemble standard et sans distortions (*std*) conformément aux recommandations des auteurs. Il est constitué de 1000 vidéos manipulées et des mêmes vidéos non manipulées (*Real*) que FaceForensics++ en compression C23.

3 Étude expérimentale

3.1 Performances en classification

La figure 1 représente la matrice de confusion d'XceptionNet entrainé sur FaceForensics++. La colonne la plus à droite indique les performances de ce modèle sur les images manipulées de DeeperForensics-1.0. Malgré les bons résultats sur FaceForensics++, on remarque une mauvaise capacité de généralisation sur la manipulation inconnue, le réseau confondant ces images, les images non manipulées (dans 75.64 % des cas) et les images d'une autre manipulation, NeuralTextures (dans 17.08 % des cas). Cette même difficulté est rencontrée en classification binaire, c'est-à-dire en regroupant toutes les images manipulées de FaceForensics++ en une seule classe ; en effet, lors d'un test réalisé précédemment, nous avons remarqué qu'environ 30 % seulement des images manipulées de DeeperForensics-1.0 étaient classées correctement.

En revanche, dès que les images truquées de DeeperForensics-1.0 sont ajoutées à l'ensemble d'entrainement et la classe correspondante créée, le réseau arrive à reconnaître ce type de trucage sans problème tout en maintenant d'excellentes performances sur les méthodes de FaceForensics++, comme illustré dans la figure 2.

1. La base contenant les images après nettoyage peut être fournie sur demande.

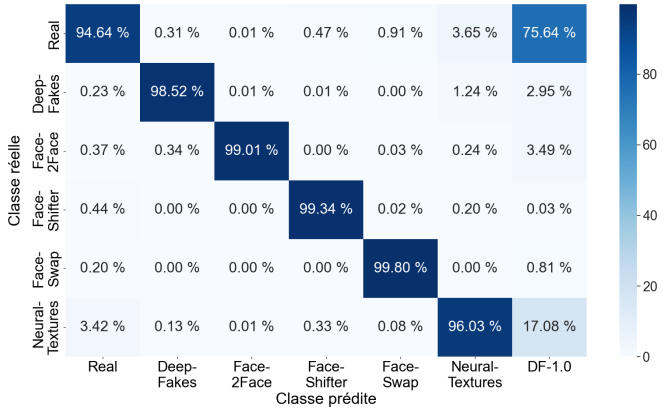

FIGURE 1 – Matrice de confusion d'XceptionNet entrainé sur FaceForensics++. La dernière colonne représente la performance de ce modèle en l'évaluant sur DeeperForensics-1.0.

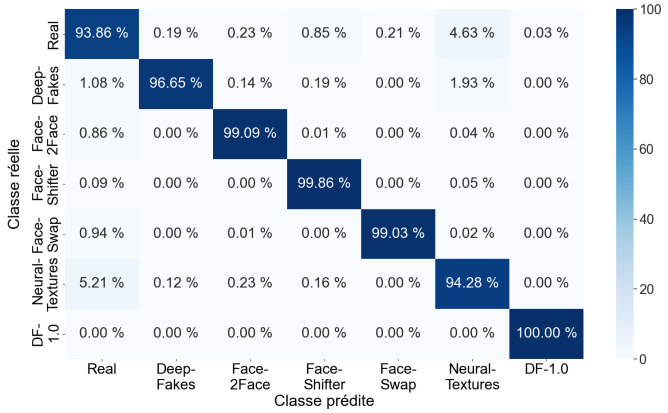

FIGURE 2 – Matrice de confusion d'XceptionNet entrainé sur FaceForensics++ et DeeperForensics-1.0.

3.2 Visualisation par t-SNE

Afin d'illustrer visuellement ces résultats, nous avons utilisé la méthode de réduction de dimension t-SNE [2] [8] afin d'obtenir une projection en deux dimensions du vecteur de caractéristiques [3] qui précède la couche complètement connectée. Ce vecteur est calculé pour chaque image de l'ensemble global de test, qui regroupe les données de test de FaceForensics++ et DeeperForensics-1.0. Chaque point obtenu est ensuite coloré selon la vraie classe (la vérité terrain) de l'image qui a servi à l'obtenir. Les nuages de points obtenus sont représentés dans la figure 3 (entrainement sur FaceForensics++ uniquement) et la figure 4 (entrainement sur FaceForensics++ et DeeperForensics-1.0).

Une bonne séparation entre les différentes couleurs signifie que le réseau arrive à bien distinguer les différentes classes. Dans la figure 3, les points correspondant à des images truquées

2. Nous avons choisi des valeurs d'hyperparamètres pour t-SNE adaptées à la quantité de données exploitées : une perplexité d'une valeur de 50 ainsi que 50000 itérations.

3. Ce vecteur est de dimension $d = 2048$.

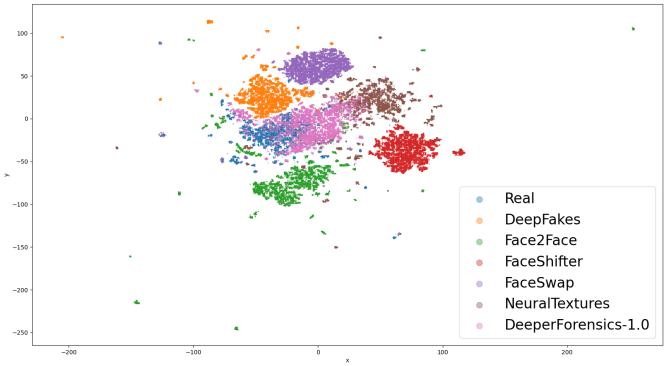

FIGURE 3 – Projections 2D des vecteurs de caractéristiques d'XceptionNet obtenus en utilisant t-SNE (XceptionNet entrainé sur FaceForensics++ uniquement).

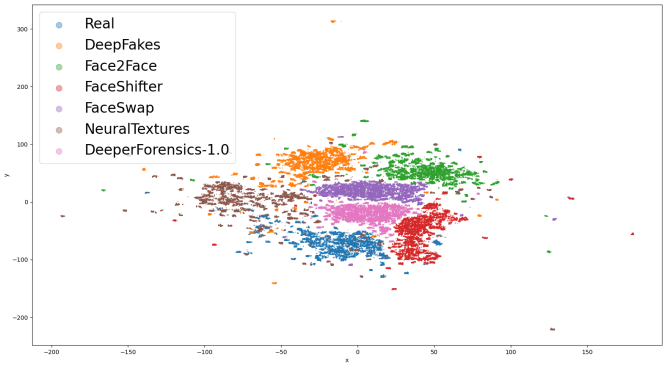

FIGURE 4 – Projections 2D des vecteurs de caractéristiques d'XceptionNet obtenus en utilisant t-SNE (XceptionNet entrainé sur FaceForensics++ et DeeperForensics-1.0).

de DeeperForensics-1.0 (en rose) recouvrent largement les points Real (en bleu) et, dans une moindre mesure, NeuralTextures (en marron), ce qui confirme les résultats numériques de la sous-section 3.1. Le réseau, n'ayant pas été exposé à cette méthode de trucage pendant l'entrainement, a tendance à la confondre avec du contenu non manipulé. L'entrainement sur une nouvelle méthode permet de bien la détecter comme indiqué par la bonne séparation sur la figure 4, mais ce cas de figure n'est pas représentatif d'un cas d'usage réel. En effet, avec le développement continu et l'émergence régulière de nouvelles méthodes de trucage, il est peu probable de croiser un type de *deepfake* déjà observé pendant l'entrainement par un système de détection.

Le défi majeur est donc de développer un outil de détection qui puisse extraire et exploiter des caractéristiques riches d'un point de vue sémantique et communes au plus large éventail de trucages possibles.

3.3 Prise en compte de la dimension temporelle

Les méthodes de trucage actuelles travaillent indépendamment image par image, ce qui peut laisser des incohérences temporelles sur la vidéo. La figure 5 montre qu'un simple vote majoritaire pour la classe la plus fréquente associée aux images d'une vidéo de 120 images permet d'augmenter le taux de bonne classification pour toutes les manipulations (+ 2.39 % sur NeuralTextures notamment) ainsi que pour les images non manipulées (+ 4.14 %).

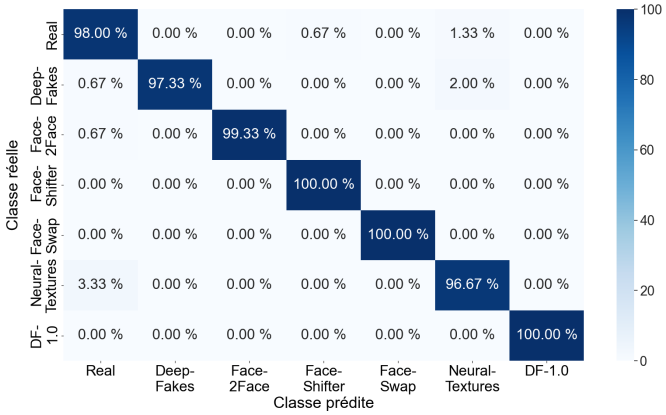

FIGURE 5 – Matrice de confusion d'XceptionNet entrainé sur FaceForensics++ et DeeperForensics-1.0 en utilisant le vote majoritaire.

3.4 Bilan de l'étude effectuée

Le tableau 1 résume les résultats de notre étude expérimentale. Dans ce tableau, toutes les méthodes de trucage sont regroupées en une seule classe, *Fake*. C'est-à-dire que l'on considère ici qu'une image est truquée si elle est classée comme résultant d'une méthode de trucage. Ce tableau présente donc les résultats sur deux classes, *Real* et *Fake*.

Nous remarquons d'abord que les performances chutent lorsque le modèle est entrainé sur FF++ et évalué sur FF++ et DF-1.0 car la majorité des images de DF-1.0 sont détectées comme étant non manipulées (*Real*). Elles chutent davantage en utilisant le vote majoritaire car ce biais est renforcé. Cependant, l'ajout de DeeperForensics-1.0 à l'ensemble d'apprentissage permet de maintenir de bons résultats et le vote majoritaire permet de les améliorer dans ce cas. Plus de tests sont cependants requis afin de mieux chiffrer l'impact du vote majoritaire sur les différents ensembles de données et méthodes de trucage.

Entrainement	Test	Taux de classification binaire correcte
FF++	FF++	98.36 %
FF++	FF++ et DF-1.0	87.76 %
FF++	FF++ et DF-1.0 (vote majoritaire)	86.95 %
FF++ et DF-1.0	FF++ et DF-1.0	97.95 %
FF++ et DF-1.0	FF++ et DF-1.0 (vote majoritaire)	99.05 %

TABLEAU 1 – Tableau récapitulatif de l'étude. Les images manipulées sont regroupées ici en une seule classe (*Fake*).

4 Discussion

Parmi les pistes de recherche qui nous semblent prometteuses en détection de *deepfakes*, on peut envisager :

— la détection des traces de manipulations sur les vidéos, par exemple en identifiant les empreintes laissées par les caméras ou par les algorithmes de compression à l'emplacement de l'incrustation d'un nouveau visage,

— l'étude de l'incidence des algorithmes de compression, et notamment le type de *frames* (I, P, B), sur la capacité à bien classer une image. Une première étude que nous avons effectuée montre qu'en entrainant XceptionNet sur FaceForensics++ uniquement, 97.22 %, 97.98 % et 97.86 % des images I, P et B, respectivement, sont classées correctement, ce qui ne semble pas représenter une différence significative en fonction du type de *frame*. Néanmoins, des tests sur des *deepfakes* utilisant différents types d'encodage sont nécessaires car sur nos séquences de 120 images par vidéo, seule la première image est de type I pour chaque vidéo, ce qui est insuffisant pour tirer des conclusions.

5 Conclusion

Les méthodes de détection de *deepfakes* basées sur l'apprentissage profond donnent de bons résultats, et ce même sur des trucages visuellement aboutis. Cependant, l'inconvénient majeur reste leur capacité limitée à généraliser la détection à de nouvelles méthodes de trucage non rencontrées pendant l'entraînement. L'apparition continue de nouveaux types de *deepfakes* nécessite le développement de méthodes de détection capables de s'adapter avec peu ou pas de données.

Dans cet article, nous illustrons par le biais d'une étude expérimentale les performances ainsi que les limites des méthodes actuelles en évaluant une méthode de l'état de l'art sur un type de manipulation inconnu et en utilisant t-SNE, un puissant outil de visualisation pour mieux comprendre le comportement de la méthode de détection utilisée.

Nous montrons que l'évaluation sur DeeperForensics-1.0, un nouveau type de trucage plus élaboré que ceux de FaceForensics++ déjà appris, induit le réseau à confondre ces images et les images non manipulées (avec néanmoins un rapprochement dans 17.08 % des cas avec NeuralTextures, l'une des méthodes les plus élaborées de FaceForensics++). Cependant, l'étude montre que ce nouveau type peut être intégré à l'entrainement, garantissant un taux de succès en détection très élevé tout en maintenant des performances très satisfaisantes sur les manipulations déjà connues. Les nouvelles méthodes pourront ainsi, au fur et à mesure de leur apparition, être intégrées à un outil généraliste de détection de *deepfakes*, notamment grâce à l'apprentissage par transfert.

Nous espérons que ce travail puisse pousser à la réflexion vis-à-vis de la nécessité d'une détection efficace de contenu manipulé au vu des implications sociétales et sécuritaires de ce sujet.

Références

[1] *DeepFakes*. `https://github.com/deepfakes/faceswap`. Consulté le 17 mars 2022.

[2] *FaceSwap*. `https://github.com/MarekKowalski/FaceSwap`. Consulté le 17 mars 2022.

[3] F. Chollet. *Xception : Deep Learning with Depthwise Separable Convolutions*. CVPR (2017).

[4] J. Deng, W. Dong, R. Socher, L. Li, K. Li et L. Fei-Fei. *ImageNet : A large-scale hierarchical image database*. CVPR (2009).

[5] L. Jiang, R. Li, W. Wu, C. Qian, et C. Loy. *DeeperForensics-1.0 : A Large-Scale Dataset for Real-World Face Forgery Detection*. CVPR (2020).

[6] D. Kingma et M. Welling. *Auto-Encoding Variational Bayes*. ICLR (2014).

[7] L. Li, J. Bao, H. Yang, D. Chen et F. Wen. *FaceShifter : Towards High Fidelity And Occlusion Aware Face Swapping*. CVPR (2020, Oral).

[8] L. Maaten et G. Hinton. *Visualizing Data using t-SNE*. JMLR (2008).

[9] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies et M. Niessner. *FaceForensics++ : Learning to Detect Manipulated Facial Images*. ICCV (2019).

[10] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt et M. Niessner. *Face2Face : Real-time Face Capture and Reenactment of RGB Videos*. CVPR (2016).

[11] J. Thies, M. Zollhöfer et M. Niessner. *Deferred Neural Rendering : Image Synthesis using Neural Textures*. ACM TOG (2019).

[12] L. Verdoliva. *Media Forensics and DeepFakes : An Overview*. IEEE JSTSP (2020).

[13] X. Wu, Z. Xie, Y. Gao et Y. Xiao. *SSTNet : Detecting Manipulated Faces Through Spatial, Steganalysis and Temporal Features*. ICASSP (2020).

[14] K. Zhang, Z. Zhang, Z. Li et Y. Qiao. *Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks*. IEEE SPL (2016).