INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN NORMANDIE

cnrs Centre National de la Recherche Scientifique

Rapport de stage

Stage ingénieur

Paul Tessé

Tuteur de stage :

M. Christophe Charrier

Entreprise d'accueil :

GREYC

Tuteur INSA :

M. Maxime Gueriau

Dates du stage :

1er mars - 31 juillet 2023

Remerciements

De nombreuses personnes m'ont apporté leur aide ou leur soutient durant ce stage et je ne pourrai malheureusement pas toutes les remercier dans ce rapport. Je tiens néanmoins à dire combien je suis très reconnaissant envers chacun de mes collègues du laboratoire et plus particulièrement de l'équipe SAFE (Sécurité Architecture Forensique et biomEtrie) pour leur accueil chaleureux, leur bienveillance et leur confiance à mon égard qui m'ont permis de pleinement m'épanouir dans le cadre de ce stage au sein du GREYC.

Je tiens tout d'abord à remercier tout particulièrement Christophe Charrier, chef de l'équipe SAFE et mon tuteur de stage, pour m'avoir permis d'intégrer le GREYC au sein de son équipe. Je le remercie non seulement pour cette opportunité qu'il m'a offerte mais également pour son soutient inestimable tout au long de mon stage qu'il a su m'apporter en dépit du peu de temps dont il dispose.

Je tiens également à remercier Emmanuel Giguet, chercheur CNRS au sein de l'équipe SAFE et mon co-tuteur de stage, qui en dépit de son arrêt maladie m'a également accompagné tout au long de ce stage et a grandement contribué à sa réussite grâce à son suivi et son soutient.

Un grand merci également à Christophe Rosenberger, directeur du laboratoire et membre de l'équipe SAFE, qui m'a fait confiance et m'a permis de m'investir dans les nombreuses activités du laboratoire et de compléter ainsi mon expérience du métier d'enseignant chercheur. Je le remercie également pour ses conseils avisés vis-à-vis de mes choix de carrière futurs qui sont désormais plus éclairés.

Je remercie également chaleureusement mes collègues Tanguy, Brice et Abda pour leur soutient quotidien et leur bonne humeur qui m'ont donné envie de venir travailler chaque jour de ce stage. Et je remercie tout particulièrement Hugo Jean pour son aide précieuse dans le développement de ma solution logicielle et toutes les connaissances techniques qu'il m'a patiemment transmises.

Enfin, je tiens à remercier Mme Sophie Rastello, Mme Béatrice Frankinet, M. Benoît Gaüzere et M. Maxime Guériau sans qui ce stage n'aurait pas été possible.

Table des matières

1- Présentation du laboratoire

A) Le GREYC

Le Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen (GREYC) est une Unité Mixte de Recherche créée en 1995. Ce laboratoire de recherches en sciences du numérique est associé au Centre National de la Recherche Scientifique (CNRS), à l'Université de Caen Normandie (UNICAEN) ainsi qu'à l'Ecole Nationale Supérieure d'Ingénieurs de Caen (ENSICAEN).

Sa création est principalement motivée par la volonté de regrouper les chercheurs tout en renforçant les liens entre différentes équipes. Aujourd'hui, le GREYC est composé de six équipes aux thématiques différentes :

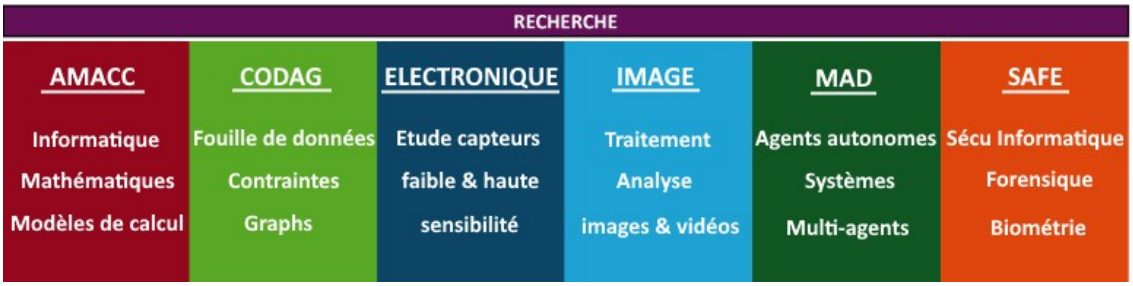


Figure 1: Equipes du GREYC

Aujourd'hui, le GREYC regroupe plus de 180 membres répartis sur sept sites différents localisés en Normandie : Cherbourg, Lisieux, Vire, Saint-Lô, Alençon et le site principal de Caen avec les Bâtiments F et Sciences 3. La direction du laboratoire est assurée par Christophe Rosenberger en tant que Directeur, et Gaël Dias et Olivier Lézoray en tant que Directeurs Adjoints.

Figure 4: G. Dias

Figure 3: C. Rosenberger

Figure 2: O. Lezoray

Les activités de recherche du laboratoire sont à la fois fondamentales, méthodologiques et appliquées et peuvent se résumer en trois axes que sont Algorithmes et Intelligence Artificielle, Capteurs et Instruments et Science des Données.

Si l'on se réfère aux derniers chiffres, le GREYC comptabilise un total de plus de 2100 publications, dont plus de 800 sur le dernier quinquennat, ce qui met en exergue la productivité du laboratoire. De plus, le laboratoire caennais cherche à toujours s'impliquer davantage dans les animations scientifiques, que ce soit par le biais des GDRs ,en tant que participants ou même animateurs, en organisant de nombreuses manifestations scientifiques, telles que CyberWorlds'2020, STACS'2018, PFIA'2017 mais également des colloques internationaux.

Figure 5: Conférence CyberWorlds2020

Ce sont les nombreuses thématiques abordées, le rayonnement du laboratoire et son dynamisme qui m'ont poussé à réaliser mon stage en son sein et plus précisément au sein de l'équipe SAFE.

B) L'équipe Sécurité, Architecture, Forensique et biomEtrie

L'équipe Sécurité Architecture Forensique et biomEtrie (SAFE) est une équipe composée de douze membres permanents dont trois Professeurs des Universités et six Maîtres de conférences dont quatre HDRs ainsi qu'un chargé de recherche CNRS. Entre les doctorants, les ingénieurs et les stagiaires, dont je fais partie, l'équipe comptabilise plus de vingt membres.

Tous ces membres viennent d'horizons culturels et scientifiques très variés. En effet, cela s'inscrit dans la volonté de l'équipe de cultiver cette diversité indispensable dans les activités de recherche. L'équipe est actuellement dirigée par Christophe Charrier qui est Maître de Conférences HDR et qui se trouve également être mon maître de stage.

Cette diversité permet à l'équipe de couvrir de nombreux sujets qui sont répartis en trois axes majeurs : la Biométrie, les Architectures et Modèles de Sécurité et les Sciences de l'Investigation (Forensique).

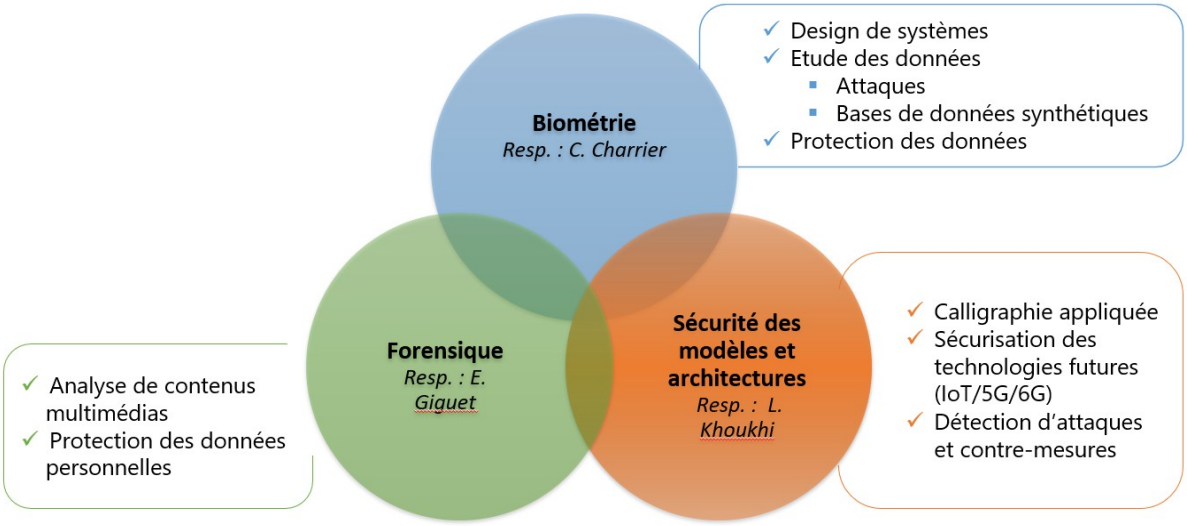


Figure 6: Thématiques équipe SAFE

Mon sujet quant à lui est en lien avec l'axe Forensique. Celui-ci correspond à l'ensemble des méthodes d'analyse et d'investigation basées sur des traces numériques. Ces traces peuvent être utilisées afin de profiler des personnes, détecter des fraudes, identifier des individus, etc. Cet axe est de plus en plus développé en collaboration avec l'axe biométrie, avec dernièrement l'objectif de développer une plateforme internet regroupant de nombreux outils d'investigation pour assister les forces de sécurité ainsi que les tribunaux face aux attaques numériques toujours plus nombreuses et efficaces.

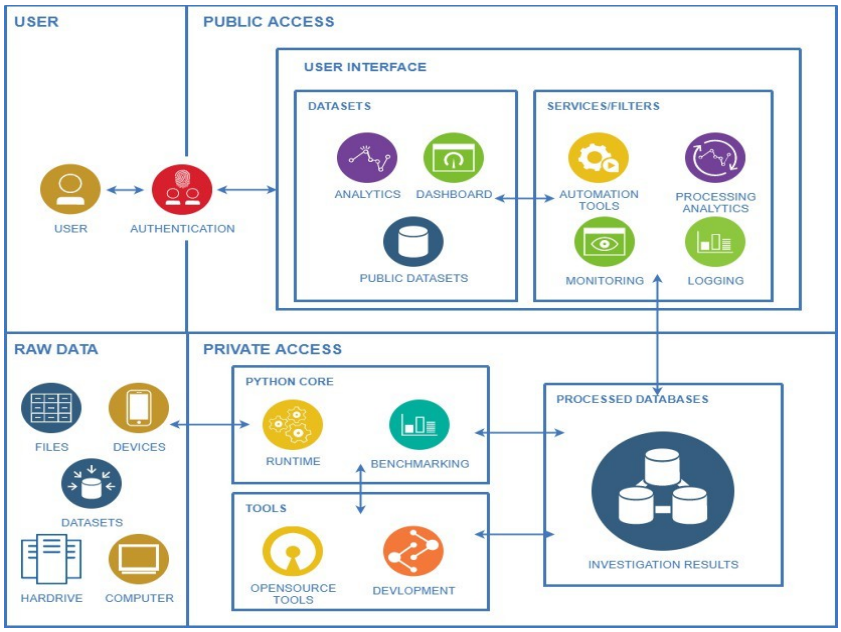


Figure 7: Plateforme d'investigation G'DIP

C) Site Campus II

Le bâtiment F où j'ai réalisé mon stage, de même que le bâtiment Sciences 3, est situé sur le site principal du GREYC : le Campus II de Caen. La présence de l'ENSICAEN, ainsi que de l'Université de Caen Normandie et de l'IUT Grand Ouest Normandie font du Campus un pôle très dynamique permettant au GREYC de développer de nombreuses collaborations entre de nombreux acteurs, et ce à différentes échelles.

Figure 8: Bâtiment Sciences 3

Figure 9: Bâtiment F

Ces très nombreux partenariats permettent également d'obtenir des financements qui sont essentiels au développement des activités du laboratoire et donc son rayonnement. De plus, la présence de ces nombreuses formations scientifiques attire les industriels qui cherchent plus que jamais à recruter de futurs talents et plus particulièrement en informatique. Le GREYC bénéficie ainsi d'une bonne visibilité et est très souvent sollicité pour de nombreux projets industriels, notamment dans le cadre de thèses CIFRE.

Grâce aux enseignements dispensés par les membres du laboratoire dans les différentes instances du campus, ainsi qu'aux nombreux évènements organisés pour promouvoir les formations du Campus II et autres manifestations scientifiques, le GREYC parvient à donner envie à de nombreux étudiants d'envisager une carrière dans la recherche. En effet, de nombreux membres de l'équipe SAFE encadrent des projets en lien avec les activités de l'équipe, ce qui permet aux étudiants de découvrir les thématiques abordées et d'avoir un aperçu du monde de la recherche. Le GREYC fait beaucoup d'efforts pour maintenir et accentuer cette proximité avec les étudiants et les industriels, indispensable à son développement et qui contribue grandement au développement de ce technopôle en constante évolution, faisant du Campus II un campus très dynamique où il fait bon étudier et travailler.

D) La vie au laboratoire

Comme je l'expliquais dans une section précédente, l'équipe SAFE est composée de nombreux membres à des postes bien différents. Les non permanents, dont je faisais partie, ont un nombre d'heures par jour avec une obligation de présence de 10h à 12h et de 14h à 16h. Ainsi, chacun est libre de gérer son temps de travail comme il le souhaite, ce que j'ai beaucoup apprécié étant donné qu'il est possible d'adapter ses horaires en fonction de son rythme qui varie évidemment souvent.

Les permanents quant à eux n'ont pas d'obligation de présence au laboratoire étant donné qu'ils ont un service d'enseignement à réaliser le plus souvent à l'Université ou à l'ENSICAEN et pour certains à l'IUT. Cette charge d'enseignement est plus ou moins importante et passe souvent avant les activités de recherche qui sont effectuées sur le temps restant. Beaucoup de permanents de l'équipe ont également des missions à responsabilité annexes qui les sollicitent également très souvent, telles que la recherche de financements, des responsabilités administratives ou encore un siège dans une des nombreuses instances du campus.

En plus de ces différentes missions, il y a les activités de recherche qui sont communes aux permanents et aux non permanents. Là encore, elles sont très nombreuses. Car en effet, il ne s'agit pas que d'avancer ses recherches mais également de les publier et ou de les présenter dans différentes manifestations scientifiques et journaux. Les membres du laboratoire passent beaucoup de leur temps à rédiger des papiers et à les soumettre et dans certains cas même à en examiner en tant que reviewers dans certaines revues telles que Cyberworlds. A cela s'ajoute également pour la majorité des permanents la charge d'encadrer des thésards et des stagiaires.

Avec toutes ces responsabilités et ces horaires flexibles, il est important de garder de la cohésion au sein de l'équipe et c'est pourquoi de nombreux dispositifs ont été mis en place dans l'équipe SAFE. Le plus efficace selon moi est le séminaire informel. Il s'agit d'un repas tous les premiers jeudi de chaque mois qui est payé sur le budget de l'équipe et pendant lequel des membres de l'équipe volontaires présentent leurs travaux. Ces présentations informelles permettent non seulement aux membres de l'équipe de se tenir au courant des travaux des autres membres mais également pour les présentateurs d'avoir un avis extérieur sur leurs travaux et de s'entraîner à les présenter en public.

2- Présentation du sujet

A) Le Contexte

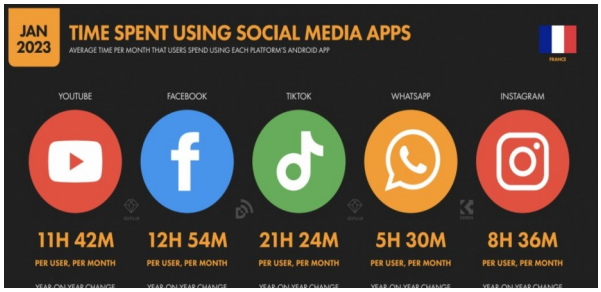


Figure 10: Statistiques Réseaux Sociaux

Avec l'évolution fulgurante de l'informatique, notre société est devenue hyperconnectée et voit transiter des quantités de données de plus en plus importantes. Que ce soit par la télévision, la vidéosurveillance, les réseaux sociaux ou plus généralement internet, c'est plus d'un milliard d'heures de vidéos visionnées dans le monde par jour. Comme le montre la Figure 10, rien que sur YouTube, on compte près de 12 heures de vidéos visionnées par mois par utilisateur. Il est de fait également devenu extrêmement facile de publier du contenu multimédia accessible à large échelle ce qui n'est pas sans risque.

En effet, avec les récentes avancées dans le domaine de l'apprentissage profond, et plus particulièrement avec la révolution des Réseaux Antagonistes Génératifs ou GANs en anglais, nous avons pu assister à l'arrivée de nombreux modèles permettant d'altérer les contenus multimédias. Un exemple très célèbre est celui des filtres sur Snapchat qui permettent de modifier les couleurs, les visages et même d'échanger les visages. Et c'est ainsi que depuis quelques années on entend parler du deepfake qui est actuellement une réelle menace pour notre société.

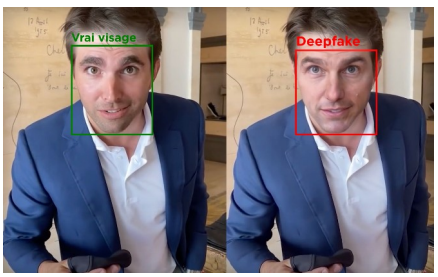


Figure 11: Exemple de Face Swapping [1]

Le deepfake est un média généré à l'aide d'une intelligence artificielle dont l'objectif est de générer du contenu multimédia d'apparence authentique dans le but de tromper le spectateur ou un système informatique. L'exemple le plus flagrant est celui de l'usurpation d'identité dont sont très souvent victimes les célébrités. Ici on peut voir un exemple de deepfake vidéo mais il existe malheureusement beaucoup d'autres types de deepfakes (textuel, audio, etc.). Dans ce rapport, je me focalise sur les deepfakes vidéos et plus particulièrement le face swapping.

B) Les Enjeux

Figure 12: Autre Exemple

Ces modèles deepfakes basés sur l'utilisation de réseaux de neurones sont de plus en plus accessibles. Actuellement, il est de plus en plus facile de générer des deepfakes en ligne sans même avoir à payer. Voici un exemple de deepfake que j'ai pu réaliser gratuitement en ligne avec faceswapper.ai à partir d'une image d'Emmanuel Macron en Figure 12.

Cette accessibilité n'est malheureusement pas contrebalancée par des conditions d'utilisation suffisamment restrictives ce qui pousse les utilisateurs à générer des faux. De plus, comme vous pouvez le voir, il n'y a aucun vrai signe visuel permettant d'identifier cette photo comme étant fausse.

Les images falsifiées sont de plus en plus crédibles si bien qu'il est fort probable que d'ici quelques années, il soit des plus difficile de déterminer si une image ou une vidéo est authentique.

Ce constat est non seulement alarmant du fait de la démocratisation du deepfake et de sa qualité mais également du fait du développement de ses utilisations. En effet, les utilisations de ces outils sont toujours plus nombreuses. Pour n'en citer que quelques-unes, nous avons la falsification d'identité, la propagande ou encore plus récemment, le revenge porn.

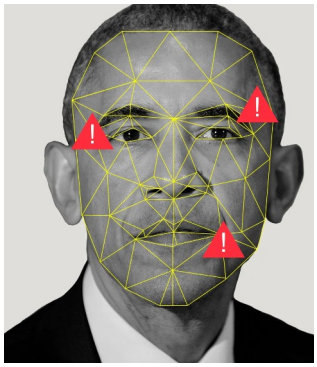


Figure 13: Analyse des caractéristiques d'un visage

Ces différentes utilisations représentent une réelle menace sociétale puisqu'elle met à mal les médias, le système judiciaire et même la réputation de personnes innocentes.

C'est pourquoi il est essentiel alors que les cas sont de plus en plus nombreux et difficiles à détecter, de trouver une solution face à cette menace.

De nombreux chercheurs se sont penchés sur cette question et un constat est rapidement tombé : les réseaux de neurones sont la cause de la menace mais également la solution [1]. En effet, les anciennes techniques d'investigation étant dépassées suite aux progrès phénoménaux dans le domaine du deepfake, de nombreux travaux sur la détection des deepfakes ont fait leur apparition dans la communauté scientifique.

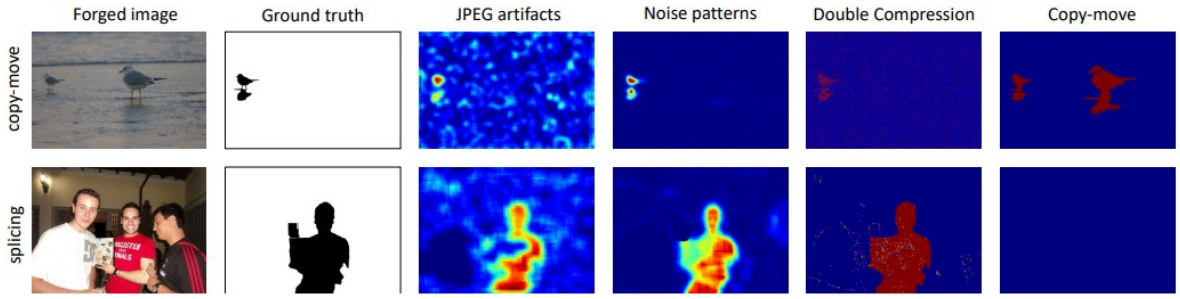


Figure 14: Exemple d'investigation forensique [1]

De plus, comme l'illustre le projet de règlement européen *Artificial Intelligence Act* publié le 21 avril 2021, les dirigeants de l'Union Européenne ont pris conscience du rôle à jouer de l'IA dans le futur de notre société et notamment pour la détection des deepfakes. La nécessité de poser un cadre à l'utilisation de tels outils s'est imposée et c'est ainsi que la pyramide des risques a été créée. Il y est clairement fait mention de la détection des deepfakes avec comme condition que l'IA utilisée soit transparente ce qui peut être associé à **l'explicabilité**. Malheureusement, même si les résultats obtenus par les chercheurs explorant la piste de l'apprentissage profond sont bons, ils ne sont pas suffisamment robustes et explicables au vu des exigences de l'UE pour pouvoir constituer une réelle solution au problème et c'est pourquoi mon stage traitait de ce sujet brûlant.

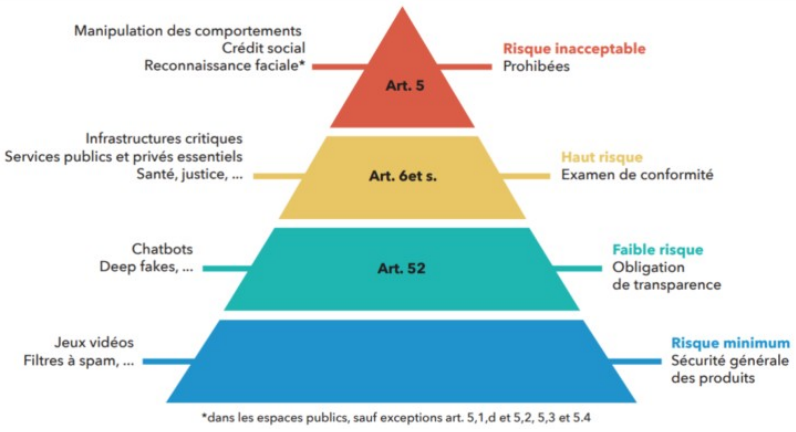


Figure 15: Pyramide des risques liés à l'IA

C) Mes Missions

Durant ce stage de 5 mois au sein de l'équipe SAFE (Sécurité Architecture Forensique et biomEtrie) du GREYC, j'ai donc effectué des travaux de recherche dans le domaine de la détection des deepfakes vidéos. L'objectif est de développer un modèle capable de déterminer si une vidéo est authentique ou falsifiée en analysant les signaux résiduels qu'elle contient. En basant notre analyse sur ces signaux très utilisés en forensique, j'espère obtenir le meilleur compromis possible entre explicabilité et précision du verdict, qui sera améliorée par l'utilisation d'un classifieur profond. Le terme **d'explicabilité** fait donc référence dans ce rapport à la possibilité de justifier le verdict rendu par notre modèle en analysant les données passées en entrée ainsi que son comportement.

Pour ce faire, ma première mission consiste en la réalisation d'un état de l'art sur la détection des vidéos deepfakes en général. L'objectif est d'avoir une vue d'ensemble sur le sujet afin de pouvoir cibler une catégorie de solutions.

La deuxième partie de mon état de l'art porte sur l'utilisation des signaux résiduels contenus dans les vidéos pour la détection des hypertrucages. J'expliciterai la notion de signaux résiduels plus loin dans ce rapport. L'objectif est de déterminer quels signaux permettent d'extraire des caractéristiques explicables et exploitables par un réseau de neurones.

Ma seconde mission durant ce stage consiste à développer des algorithmes extracteurs de caractéristiques. Ces algorithmes servent à extraire les caractéristiques basées sur les signaux résiduels retenus à l'état de l'art qui seront utilisées par le classifieur en guise d'entrées.

Ma troisième mission consiste à développer le classifieur basé sur l'apprentissage profond. Le rôle de ce classifieur est donc de déterminer, en analysant les caractéristiques extraites par les extracteurs, si la vidéo est authentique ou falsifiée.

En plus de ces missions, j'ai également été amené à effectuer des missions annexes liées aux activités de chercheurs, telles que la rédaction d'articles et la médiation scientifique.

3- Méthodologie de travail

Figure 16: Incertitudes

Ce stage étant orienté recherche, la question de l'organisation de mes tâches était essentielle. En effet, dans le cadre de mon Projet INSA Certifié avec le centre Henri Becquerel, j'ai eu la chance d'expérimenter la gestion d'un projet orienté recherche en apprentissage profond. J'ai découvert qu'il n'est pas simple de mener ce type de projet du fait de l'incertitude de la direction à prendre. De fait, ne pas savoir si un test va être concluant ou pas rend la planification complexe et les objectifs à atteindre sont amenés à changer régulièrement.

Bien évidemment j'ai pu réitérer ce constat dans le cadre de mon stage au GREYC mais j'ai travaillé dès le début sur la définition du besoin, et ainsi organiser mes travaux afin d'être le plus efficace possible. En consacrant le début de mon stage à l'étude et la formalisation de la problématique, j'ai pu, en m'appuyant sur mon expérience de Product Owner, définir un cahier des charges avec mes deux encadrants. Une fois ce cahier des charges validé, j'ai pu commencer à organiser mon travail en sprints à l'image de la méthode SCRUM.

Figure 17: Idéation

Chaque sprint durait entre une et trois semaines et débutait par une réunion de présentation de mes résultats actuels. Ces réunions ont été cruciales dans le bon déroulé de ce stage et je remercie mes encadrants de m'avoir accordé de leur temps que je sais précieux. Je préparais des diapositives pour structurer et synthétiser la réunion au maximum afin de pouvoir garder le plus de temps possible pour les discussions scientifiques. Cette réunion permettait donc non seulement de faire le point sur les résultats du sprint précédent, mais également de discuter des résultats et pour finir de planifier le prochain sprint. Parfois, certains collègues de l'équipe venaient assister à nos réunions afin de suivre l'avancement et proposer certaines idées.

A l'issue de ces réunions, j'avais mes nouveaux objectifs et délais en tête, ce qui m'assurait de ne pas m'éparpiller et d'être productif. De plus, le suivi de mon projet par mes encadrants m'a permis d'être plus efficace et m'a surtout énormément motivé. Chaque réunion soulevait de nouveaux défis à relever ce qui me poussait à toujours m'investir davantage dans ce projet. J'ai beaucoup apprécié ce cadre offert par cette micro gestion de mon projet auquel s'est également ajoutée une planification plus macroscopique.

En effet, nous avons découpé le stage en trois grandes parties correspondant plus ou moins à mes trois missions. Durant la première partie je me suis vraiment focalisé sur l'étude de l'existant et la réalisation d'un état de l'art sur la détection des deepfakes vidéos et des signaux résiduels. La deuxième partie consistait en la réalisation de preuves de concepts simples afin de valider la viabilité de l'approche retenue. Enfin, la troisième partie consistait à développer un modèle plus performant et à améliorer le plus possible notre architecture.

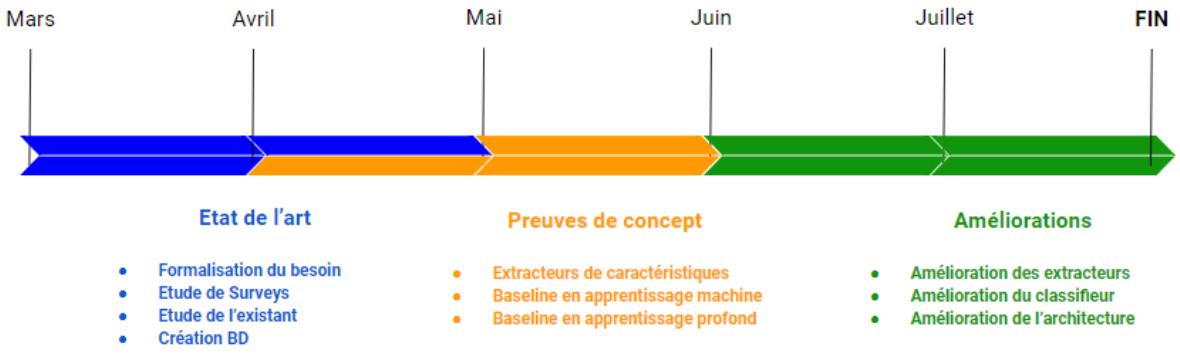


Figure 18: Macroplanning

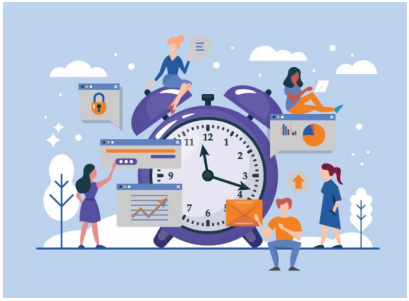


Figure 19: Planification

Ainsi, cette planification avait pour objectif d'assurer la cohérence entre les sprints et surtout de dimensionner le temps à investir dans mes différentes missions. Ce découpage du projet a permis une bien meilleure gestion du temps, ce qui était indispensable n'ayant que cinq mois devant moi pour traiter ce sujet complexe. De fait, cette notion de budget horaire m'a poussé à prioriser certaines pistes durant mes travaux de recherche, toujours dans cette optique d'être le plus efficace et productif possible. Cela m'a également conduit à estimer la faisabilité et la fiabilité des différentes pistes.

Enfin, j'ai également dû m'organiser pour allouer du temps à différentes missions annexes. J'ai effectivement été amené, dans le cadre des mes activités au laboratoire, à soumettre deux articles dont un accepté à CORESA et l'autre en attente d'être évalué à Electronic Imaging, participer en tant que présentateur ou simple spectateur à différents séminaires, évaluer deux soumissions pour la conférence internationale *Cyberworlds2023* ainsi que présenter mes travaux à l'INS2I. Toutes ces activités annexes à mes travaux de recherche m'ont également demandé un effort d'organisation et de prise de recul. Cette prise de recul m'a permis de tirer pleinement profit de toutes ces missions qui m'ont beaucoup apporté durant ce projet, sous la forme de veille scientifique ou d'audience pour présenter mes travaux et brasser toujours plus d'idées pour résoudre mon problème.

4- Cahier des charges

Comme je l'expliquais précédemment dans ce rapport, la spécification du problème a été ma première préoccupation. Que ce soit dans le cadre de ma démarche ingénieure ou tout simplement pour structurer mes travaux de recherche, la rédaction d'un cahier des charges le plus clair et le plus complet possible au vu du contexte était primordial. C'est pourquoi dès les premiers jours j'ai travaillé à m'approprier la problématique et les différentes contraintes mises en avant par mes encadrants afin de me les approprier. C'est ainsi qu'après plusieurs réunions avec M. Charrier et M. Giguet, nous avons pu affiner les différentes exigences liées au projet et ainsi concrétiser nos objectifs.

A) Principe de la solution logicielle

Le principe de la solution logicielle est simple. Il s'agit d'une architecture, basée sur des extracteurs de caractéristiques explicables ainsi qu'un classifieur profond, permettant de diagnostiquer si une vidéo passée en entrée est authentique ou falsifiée.

La solution logicielle a pour vocation d'être utilisable par toutes les instances pouvant être amenées à statuer de l'authenticité des vidéos, telles que les maisons d'éditions, les tribunaux, les réseaux sociaux, etc.

Tout l'enjeu réside dans l'équilibre entre explicabilité et performance. Au vu du contexte, les prédictions de l'architecture doivent pouvoir être justifiées tout en restant justes

et constantes. C'est pourquoi le produit final consistera en une librairie python permettant d'utiliser notre architecture prête à l'emploi avec au centre notre modèle entraîné ainsi que nos extracteurs de caractéristiques. De cette manière, cela simplifie au maximum l'utilisation de notre solution logicielle tout en garantissant sa stabilité et son évolutivité.

Afin de garantir la durée de vie de l'architecture, la librairie inclura également le pipeline d'entraînement qui pourra être amélioré et perfectionné afin de réentraîner le modèle et ainsi garantir sa durabilité dans le temps.

B) Exigences Fonctionnelles

- Déterminer si une vidéo est falsifiée ou authentique
- Détecter le ou les visages dans la vidéo et les extraire
- Entraîner un modèle de classification adapté à la détection des vidéos deepfake

Je reviendrai sur la nécessité de détecter et d'extraire les visages ultérieurement. A titre indicatif, cette contrainte vient du fait que je m'intéresse uniquement aux deepfakes basés sur le *face swapping* dans ces travaux.

C) Exigences Opérationnelles

- Explicable tout en restant performant → boîte grise
- Robuste à la durée de la vidéo et à la position de l'attaque
- Généralisable aux différents modèles de deepfakes
- Le diagnostic doit être effectué sans utiliser de vidéo de référence
- Fonctionnel pour des vidéos en haute qualité
- Pas de prise en compte de la voix, de l'éclairage et des images purement synthétiques
- Pas de contraintes en taille du modèle et temps d'apprentissage
- Temps d'inférence raisonnable (minutes)

D) Exigences de Qualité

- Application de la PEP8
- Application de la PEP257
- Documentation respectant la DocString

Ces exigences ne sont pas centrales dans ce contexte de recherche mais restent essentielles à la maintenabilité et la lisibilité du code produit durant le stage. C'est pourquoi une attention sera portée à l'application de ces exigences de qualité.

E) Exigences de Réalisation

- Code produit en Python3
- Pipelines distincts d'inférence et d'apprentissage
- Le modèle doit être réentraînable et il doit être possible de regénérer les données
- Si l'analyse à l'échelle de la vidéo est pertinente, entraîner un modèle spécifique ainsi qu'un autre pipeline adapté pour l'inférence et l'apprentissage
- Prise en compte de la dimension Développement Durable

5- Etat de l'art

A) Étude préliminaire

Afin de bien m'approprier le sujet et de mieux comprendre le choix de combiner signaux résiduels et apprentissage profond, j'ai réalisé une étude préliminaire sur la détection des deepfakes vidéos. Le premier constat qui s'est imposé était qu'il y avait beaucoup d'articles traitant du sujet de la falsification vidéo et plus récemment des deepfakes. Pour éviter toute confusion, lorsque l'on parle de deepfake, il s'agit d'outils utilisant des modèles d'apprentissage machine ou profond contrairement à certaines méthodes qui reposent sur des manipulations plus basique telles que le copy-move [2].

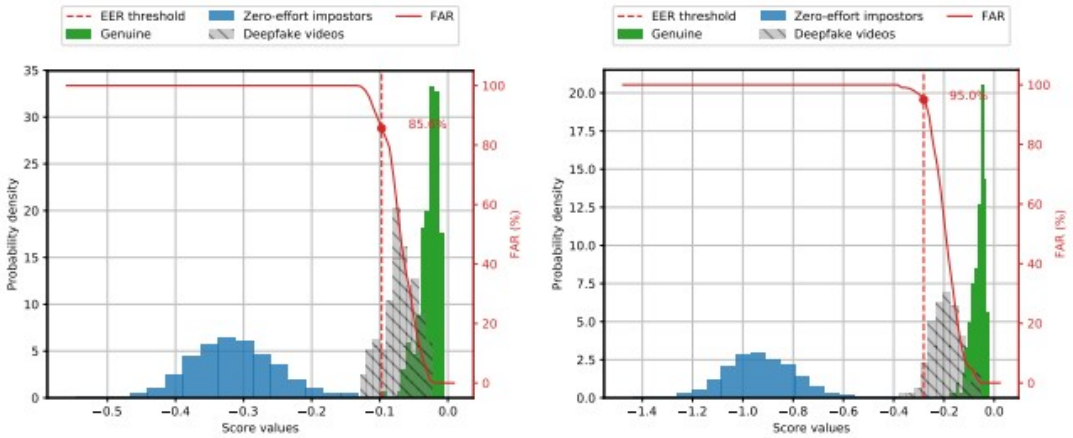
Dans leur article *DeepFakes: a New Threat to Face Recognition ? Assessment and Detection [3]* paru en 2018, Kurshonov et Marcel statuent sur la sensibilité des modèles classiques de détection face aux modèles générant des deepfakes vidéos par face swapping. Le problème a été formalisé comme un problème de classification binaire où le modèle doit déterminer si une vidéo est authentique ou falsifiée. Pour cela, les auteurs ont

extrait des paires similaires de visages de la base VidTIMIT[1], afin de générer grâce à des modèles de face swapping basés sur l'utlisation des RAGs [4] (Réseaux Antagonistes Génératifs), des deepfakes vidéos.

(g) Original 1 (h) Original 2 (i) LQ swap 1 → 2 (j) HQ swap 1 → 2 (k) LQ swap 2 → 1 (l) HQ swap 2 → 1

Figure 20: Exemple de génération des deep fakes [3]

Ici on se place donc dans un contexte où un individu essaie de se faire passer pour quelqu'un d'autre grâce au face swapping. L'évaluation de la vulnérabilité des modèles de détection a été menée par les auteurs en entraînant deux modèles de référence à partir des deepfakes et de leurs équivalents authentiques issus de VidTIMIT. Les modèles choisis sont VGG et FaceNet qui sont à l'état de l'art en matière de reconnaissance d'image. Ces derniers doivent déterminer si une vidéo est similaire aux autres vidéos authentiques qu'ils ont vues à l'entraînement.



(a) VGG-based face recognition (b) FaceNet-based face recognition

Figure 21: Vulnérabilité des modèles de reconnaissance d'images [3]

Les deux histogrammes de la Figure 21 représentent la densité de probabilité en fonction du score prédit par le modèle qui caractérise un score de confiance quant à l'appartenance de la vidéo à un ensemble de vidéos authentiques similaires. Il est clair au vu des distributions que les Deepfakes vidéos, contrairement aux attaques grossières, sont très

1 https://conradsanderson.id.au/vidtimit/

difficilement distinguables des vidéos authentiques ce qui pose problème étant donné que ces deux modèles sont à l'état de l'art et très utilisés.

Ce constat s'est répandu dans la communauté scientifique et de nombreux chercheurs ont tenté de créer des modèles de deep learning spécialisés dans la détection des deepfakes vidéos. Dans l'article *Deepfake Video Detection Using Recurrent Neural Networks [5]*, les auteurs présentent une nouvelle architecture de détection :

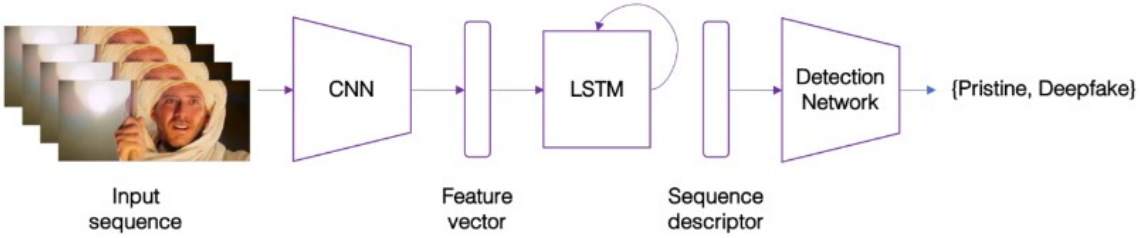


Figure 22: Architecture de détection C-LSTM [5]

Les couches de convolution (CNN) sont à l'état de l'art dans le domaine de l'imagerie. Elles permettent d'extraire un grand nombre de caractéristiques contenues dans les frames qui sont ensuite analysées par un réseau récurrent (LSTM). Ce LSTM permet de tenir compte de l'aspect temporel entre les frames et de l'incorporer à la première représentation de sorte à obtenir un vecteur de caractéristiques pour la séquences de frames. Le module de détection consiste en une couche complètement connectée suivie d'une softmax qui permet d'obtenir la probabilité d'appartenance aux deux classes.

Les résultats présentés dans l'article en Figure 23 ont été obtenus sur 300 vidéos authentiques issues du dataset HOHA et 300 vidéos deepfake scrappées sur internet. Le modèle proposé est très performant en termes de précision de classification (accuracy) mais sa capacité à généraliser n'est pas présentée.

Model	Training acc. (%)	Validation acc. (%)	Test acc. (%)
Conv-LSTM, 20 frames	99.5	96.9	96.7
Conv-LSTM, 40 frames	99.3	97.1	97.1
Conv-LSTM, 80 frames	99.7	97.2	97.1

Figure 23: Résultats présentés C-LSTM [5]

Comme illustré dans un article publié en 2022 [6], les capacités de généralisation sont limitées et ce même pour les modèles les plus récents et performants. Il est donc évident que les architectures deep learning ne sont pas la solution parfaite au problème.

Dans l'article *Media Forensics and Deepfakes: an overview [1]*, l'auteur présente de multiples méthodes permettant de détecter des traces de manipulations diverses, telles que des effets de distorsion, des artefacts colorimétriques ou encore des motifs de bruit. Ce sont ces informations invisibles à l'œil nu que l'on désigne dans nos travaux en tant que *signaux résiduels cf. Figure 24*. Il s'agit donc de caractéristiques intrinsèques des images étudiées qui sont altérées ou générées par les modèles de deepfakes. Ces différentes caractéristiques sont souvent utilisées en forensique car elles sont totalement explicables et variées ce qui, dans mon cas, est particulièrement intéressant.

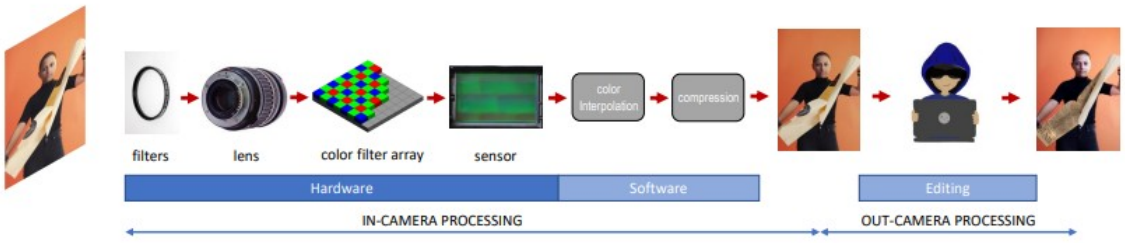


Figure 24: Sources de signaux résiduels [1]

Afin de comparer les avantages et inconvénients de ces deux approches classiques présentées succinctement, j'ai réalisé un tableau comparatif basé sur les exigences de notre cahier des charges :

	Architecture deep learning	Analyse forensique signaux
Précision	Très bonne sur ses sets de prédilection	Variable selon signaux & attaque
Fiabilité	Robuste mais difficile de généraliser	A améliorer
Explicabilité	Boîte noire	Boîte blanche
Solution légère	Modèles généralement lourds	Algorithmes sans apprentissage
Respect du DD	Fort impact environnemental	Impact environnemental acceptable

Tableau 1: Tableau comparatif des deux approches

Les architectures basées sur le deep learning offrent de bien meilleurs résultats en termes de fiabilité et de précision. Cette efficacité et cette stabilité dans les verdicts s'obtient malheureusement avec des entraînements très voraces en énergie de modèles très lourds.

A l'inverse, en plus d'offrir une excellente explicabilité, l'approche par analyse forensique des signaux résiduels demande moins de ressources et est beaucoup plus légère. En revanche, les performances en matière de verdict sont bien moins satisfaisantes. Il est évident que l'on ne peut se contenter d'une de ces deux approches en l'état. Et c'est pourquoi j'ai fait le choix d'expérimenter une architecture permettant de combiner les deux approches et ainsi compenser leurs faiblesses respectives.

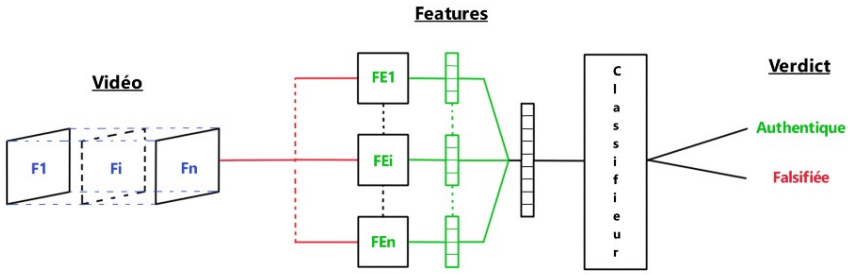


Figure 25: Ma proposition d'architecture

Notre architecture prend en entrée une vidéo qui est prétraitée, de sorte à extraire les frames. Je reviendrai par la suite sur le prétraitement mis en place. Celui-ci permet de fournir l'information utile aux extracteurs de caractéristiques (FE) qui sont des algorithmes explicables analysant les signaux résiduels afin de générer des vecteurs de caractéristiques.

Ensuite vient l'étape de concaténation des vecteurs de caractéristiques intermédiaires afin d'agréger l'information. Enfin, nous utilisons un classifieur basé sur de l'apprentissage profond afin de prédire si la vidéo est authentique ou non, en analysant les caractéristiques explicable extraites. De cette manière, nous combinons les performances en matière de verdict du deep learning avec les autres avantages de l'analyse des signaux résiduels présentés précédemment. Notre architecture constitue une boîte grise ce qui nous permet de satisfaire au mieux notre exigence principale d'explicabilité.

B) Etude et sélection des signaux résiduels

Il existe de très nombreux signaux résiduels et une étude plus approfondie est nécessaire afin de sélectionner certains d'entre eux pour nos expérimentations. En me basant sur l'état de l'art réalisé par Luisa Verdoliva [1] et d'autres articles présentés ultérieurement, j'ai réalisé une taxonomie répertoriant différents signaux que j'ai pu recenser afin d'avoir une vue globale de nos options.

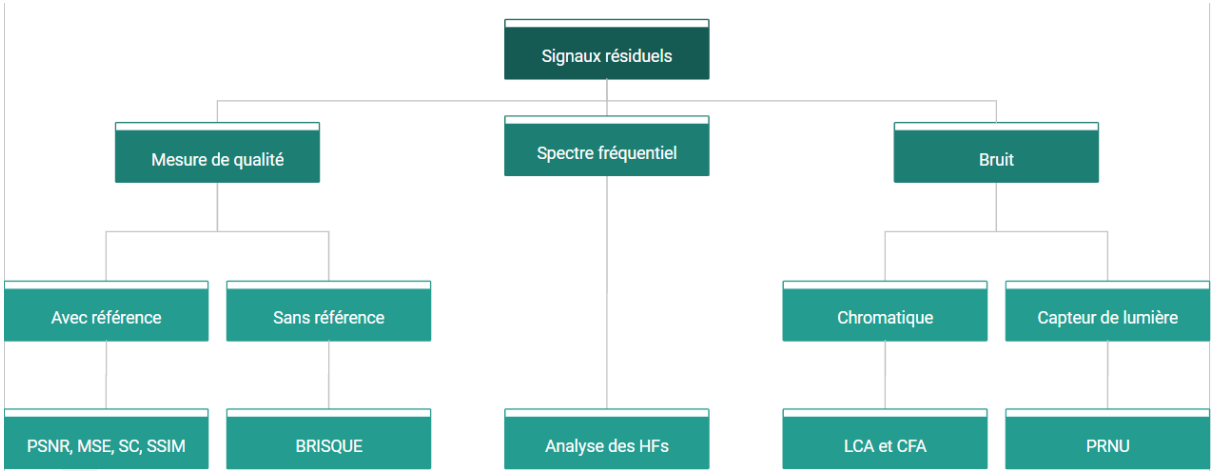


Figure 26: Taxonomie des signaux résiduels

I) Mesure de qualité

Au sein de l'équipe SAFE, j'ai découvert les fondamentaux de la biométrie. Une des problématiques les plus courantes est la Détection d'Attaque par Présentation (DAP). L'objectif est alors de détecter si l'élément présenté à un système biométrique équipé d'une caméra est authentique ou falsifié. Le parallèle avec notre problème est très intéressant bien qu'une différence de taille subsiste : ce genre de systèmes biométriques possèdent des références. On peut citer comme exemple les systèmes de reconnaissances d'empreintes digitales qui possèdent une référence de nos empreintes.

Dans l'article *Face Anti-Spoofing Based on General Image Quality Assessment* [7], les auteurs introduisent une architecture très simple basée sur l'utilisation de 14 mesures de qualité avec référence en guise de caractéristiques, ainsi qu'une Linear Discriminant Analysis (LDA) en guise de classifieur. Cette proposition est motivée par le fait que les échantillons falsifiés voient souvent leur qualité baisser pendant le processus de création et les résultats présentés sont encourageants.

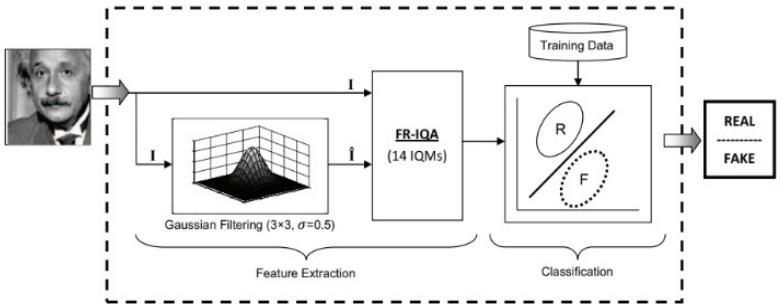


Figure 27: Architecture Anti-Spoofing [7]

L'article [3] cité précédemment utilise également des mesures de qualité avec référence en guise de caractéristiques mais qui sont cette fois-ci utilisées par un Support Vector Machine (SVM) en guise de classifieur. Les résultats présentés en Figure 28 sont probants au regard des méthodes à l'état de l'art en 2018. Bien que nous n'ayons pas de garanties quant aux performances en généralisation, la dégradation de la qualité semble être un facteur commun pouvant assurer de bonnes performances.

Database	Detection system	EER (%)	FRR@FAR10% (%)
LQ Deepfake	LSTM lip-sync [11]	41.8	81.67
	Pixels+PCA+LDA	39.48	78.10
	IQM+PCA+LDA	20.52	66.67
	IQM+SVM	3.33	0.95
HQ Deepfake	IQM+SVM	8.97	9.05

Figure 28: Performances caractéristiques IQM [3]

Le problème de cette approche réside cependant dans le fait que ces métriques nécessitent une référence que nous ne pouvons avoir compte tenu de notre cahier des charges. C'est pourquoi je me suis intéressé aux algorithmes de mesure de qualité sans référence afin de palier à ce problème. Dans le cadre de mes recherches, j'ai donc fait la découverte de l'indice de qualité sans référence BRISQUE. Je présenterai cet algorithme plus en détail dans une section ultérieure de ce rapport. Cet algorithme a également l'avantage d'être peu coûteux étant donné que le SVM utilisé est pré-entraîné et que l'essentiel des calculs correspond à des estimations de paramètres de gaussiennes.

II) Analyse des hautes fréquences

La grande majorité des méthodes de détection à l'état de l'art analysent les images dans leur représentation classique. C'est pourquoi, dans leur article *Leveraging Frequency Analysis for deep Fake Image Recognition* [8], les auteurs se penchent sur l'analyse des spectres fréquentiels des images deepfake. Le constat qu'ils présentent est que les images deepfakes ont des moyennes et hautes fréquences de plus forte intensité que les images naturelles. De plus, on peut même distinguer des motifs dans les spectres des deepfakes.

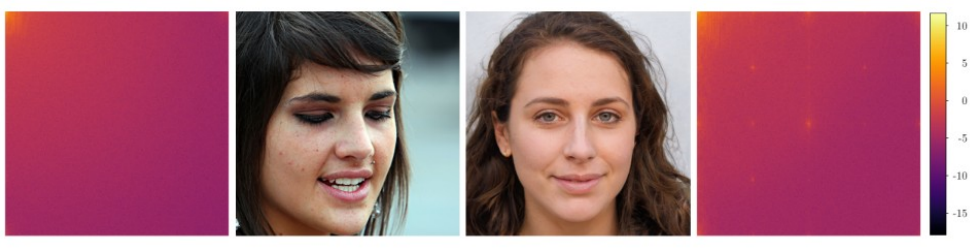


Figure 29: Spectres moyens fréquentiels des bases FFHQ et StyleGAN [8]

Afin de générer ces résultats, les auteurs ont calculé le spectre moyen de 10 000 images naturelles de haute qualité issues de la base de données FFHQ et d'autant d'images deepfakes générées avec un StyleGAN. Ces expérimentations semblent clairement indiquer qu'il est possible de distinguer les deux catégories d'image mais qu'en est-il de la capacité à généraliser ? Pour répondre à cette question, les auteurs ont répéter leur analyse présentée en Figure 29 avec des images deepfakes générées par quatre modèles (cf. Figure 30).

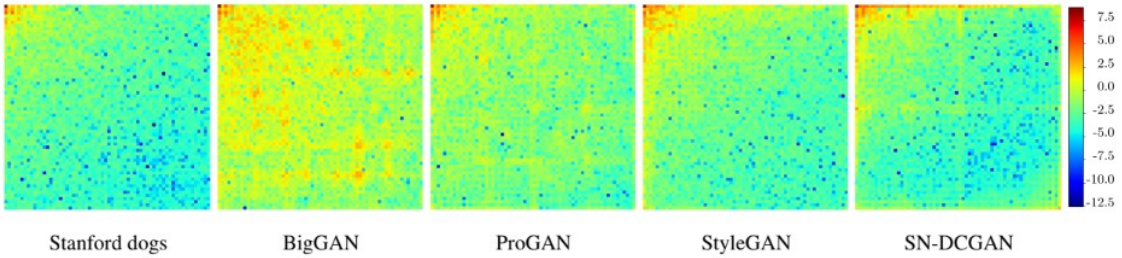


Figure 30: Spectres moyens fréquentiels pour différents modèles deepfake [8]

Bien que les résultats semblent variables, on constate que le phénomène reste visible pour ces quatre autres modèles. Le but final de cet article étant de quantifier le gain en précision en passant dans le domaine fréquentiel, ils ont testé différents modèles de classifieurs avec les deux représentations (pixels et fréquences) et ont comparé leurs performances.

Method	Accuracy	Gain
Ridge-Regression-Pixel	75.78 %	
Ridge-Regression-DCT	**100.00** %	+ 24.22 %

Figure 31: Gain en précision en fonction de la représentation [8]

Le gain en terme d'accuracy est significatif comme on peut le voir avec cet extrait de leurs résultats, présenté en Figure 31, générés avec une régression ridge. Je tiens à préciser que de nombreux modèles que je ne mentionne pas ont été testés pour effectuer cette analyse. Ces résultats complémentaires sont disponibles dans l'article. Ces résultats ont été obtenus sur les données de test qui sont au nombre de 20 000 images. Il est évident que l'étude de l'intensité des moyennes et hautes fréquences constitue une piste sérieuse en matière de détection de deepfakes. De plus, l'utilisation de la DCT assure un faible coût.

III) Analyse des aberrations chromatiques

Dans une autre partie du spectre des signaux résiduels existants, certains chercheurs se sont intéressés à l'étude des traces laissées par le système d'acquisition des images. En effet, chaque système d'acquisition est différent et ce sont ces traces laissées

par les différents composants et les différents traitements que l'on cherche à étudier. Parmi ces traces, on retrouve les aberrations chromatiques. Il s'agit d'incohérences liées à la colorimétrie des images qui sont imperceptibles à l'œil nu.

Dans l'article *Accurate and Efficient Image Forgery Detection Using Lateral Chromatic Aberration* [9], les auteurs se focalisent sur un phénomène optique, lié à l'acquisition des images, très connu : les Aberrations Chromatiques Latérales. En effet, toutes les lentilles d'objectifs ont un biais commun similarité. Ces dernières , comme l'illustre la Figure 32, vont créer un décalage entre les filtres de couleurs par rapport au centre optique qui va être plus ou moins important en fonction de la couleur et de la position de l'objet sur l'image.

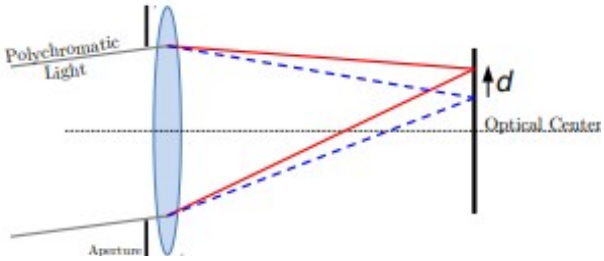


Figure 32: Phénomène d'aberration chromatique latérale [9]

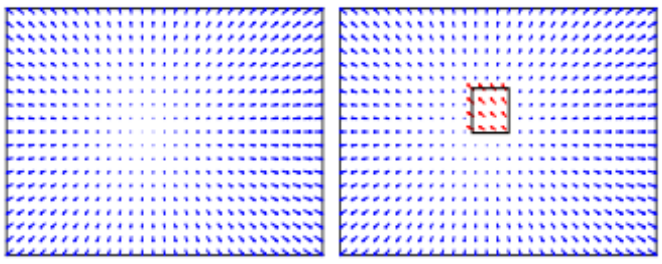


Figure 33: Visualisation de l'inconsistance de décalage [9]

C'est pourquoi, si l'on part du principe que dans le cadre du face swapping on utilise deux images, une cible et l'autre source, et que ces dernières n'utilisent pas une lentille identique, il est alors possible de caractériser une inconsistance dans ce décalage entre les filtres. Dans les images naturelles, les vecteurs caractérisant le décalage entre les filtres vont former un champ de vecteurs qui, dans le cas d'une attaque, va se retrouver altéré. C'est pourquoi, les auteurs développent un algorithme permettant de mesurer ce décalage en utilisant une estimation globale de ce dernier et une estimation plus locale grâce à des outils statistiques.

Les résultats présentés par les auteurs en Figure 34 semblent indiquer que la méthode était la plus efficace par rapport aux méthodes à l'état de l'art en 2018. En revanche, les résultats semblent varier selon le modèle de lentilles utilisé ce qui soulève des doutes quant à la robustesse et la capacité à généraliser les résultats de l'algorithme. Enfin la méthode nécessite de nombreux calculs de similarité coûteux.

DETECTION RATES OF RANDOM COPY PASTE FORGERIES COPIED FROM SONY DSC-H50 AND PASTED INTO AGFA SENSOR530S IMAGES

	$u=5$			$u=10$		
P_{FA}	0.01	0.05	0.10	0.01	0.05	0.10
Proposed	0.76	0.87	0.91	0.80	0.90	0.93
MS'16	0.44	0.77	0.87	0.49	0.82	0.90
Ang. Err.	0.10	0.38	0.57	0.12	0.41	0.59

DETECTION RATES OF RANDOM COPY PASTE FORGERIES ON ENTIRE IMAGE DATABASE

	$u=5$			$u=10$		
P_{FA}	0.01	0.05	0.10	0.01	0.05	0.10
Proposed	0.52	0.66	0.73	0.59	0.73	0.79
MS'16	0.23	0.54	0.66	0.28	0.61	0.72
Ang. Err.	0.07	0.29	0.46	0.08	0.32	0.50

Figure 34: Résultats de l'analyse de l'aberration chromatique latérale [9]

Une autre piste d'aberration liée à la chromatique des images est introduite dans l'article *Image Forgery Localization via Fine-Grained Analysis of CFA Artifacts* [10]. Dans cette étude, les auteurs se concentrent sur le processus d'acquisition des couleurs. En effet, contrairement à ce que l'on peut penser, les images ne sont pas acquises sur les trois canaux RGB directement. L'acquisition physique ne s'effectue qu'avec un unique filtre (Color Filter Array ou CFA) comportant l'information des trois canaux de couleurs mais avec une prédominance de vert. Cette prédominance du canal vert vient du fait que des études ont permis de constater que l'œil humain est tout simplement plus sensible à cette teinte.

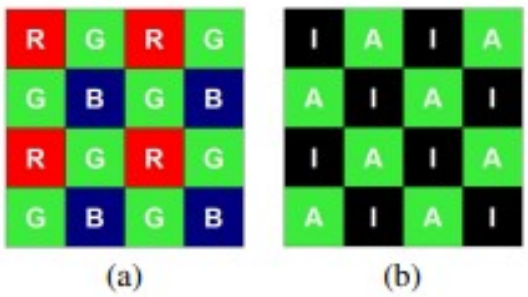


Figure 35: Acquisition avec un filtre de Bayer à gauche et interpolation pour le canal vert à droite [10]

Une fois cette étape de l'acquisition effectuée, l'appareil vient à l'aide d'un algorithme effectuer une interpolation à partir des informations du filtre pour reconstituer l'information manquante et ainsi générer nos trois canaux RGB. Il y a donc indubitablement un léger biais qui est engendré sous forme de bruit lors de ce processus d'interpolation qui dépend de l'objectif utilisé. Et c'est la présence de ces artefacts dans l'image que les auteurs cherchent à quantifier afin de déterminer si une image à été falsifiée ou non.

De fait, lors des attaques, les attaquants tentent de remplacer dans notre cas le visage d'une personne par celui de quelqu'un d'autre. Et pour ce faire, les attaquants vont venir homogénéiser lors de la fusion des deux images ce qui va avoir pour impact de gommer les imperfections et notamment celles présentes naturellement. Les auteurs mettent donc en place un détecteur basé sur des méthodes d'estimation statistiques afin de détecter la présence ou l'absence de ces artefacts et in fine la probabilité qu'une zone de l'image ait été modifiée.

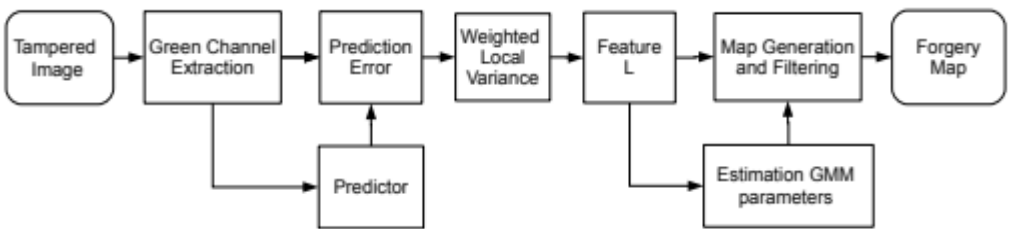


Figure 36: Algorithme d'analyse des artefacts CFA [10]

Les résultats présentés en Figure 37 sous la forme de courbe Receiver Operating Characteristic (ROC) par les auteurs correspondent à deux scenarii différents. Dans le premier cas, on connaît l'algorithme d'interpolation utilisé par l'objectif tandis que dans le second l'algorithme est inconnu. Ce deuxième cas d'utilisation est plus en adéquation avec notre cahier des charges, or il se trouve que l'on constate une baisse des performances qui semble indiquer des difficultés à généraliser.

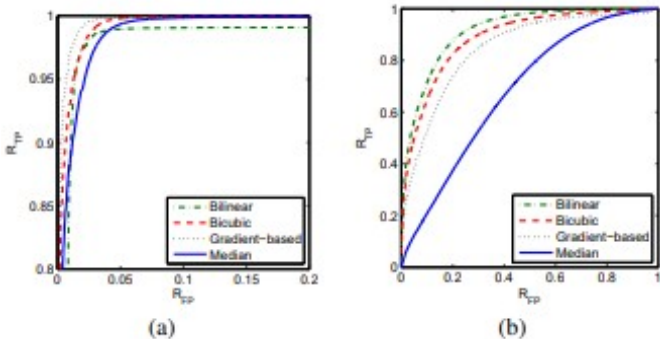


Figure 37: Courbes ROC détection par analyse CFA [10]

IV) Analyse de l'empreinte du capteur de lumière

Afin de compléter mon étude du spectre des signaux résiduels, je me suis intéressé à une autre source de bruit dans le processus d'acquisition : le capteur de lumière. Chaque appareil photo possède un capteur de lumière composé de silicium et qui lui est propre. Lors de l'acquisition, ce capteur laisse une empreinte fixe spécifique. Cette empreinte prend la forme d'un motif comme l'illustre la Figure 38. On peut en effet apercevoir sur ces images traitées des schémas de bruit différents que l'on appelle Photo-Response Non-Uniformity (PRNU).

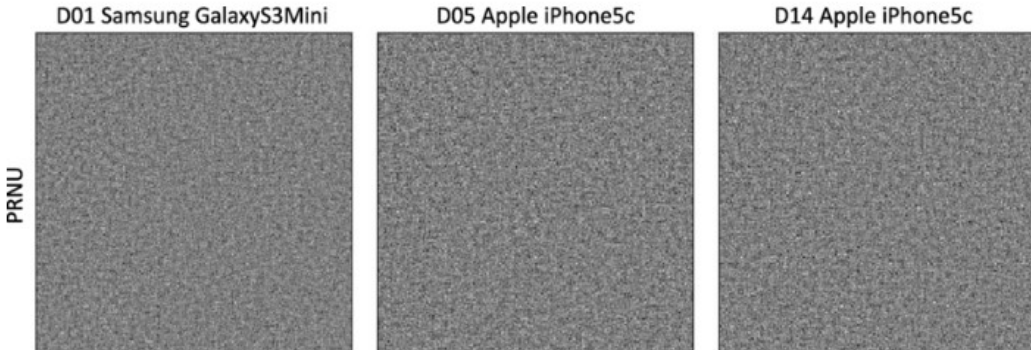


Figure 38: PRNU de différents capteurs

Ce PRNU est très utilisé, notamment en forensique, étant donné que cette information permet d'identifier l'appareil ayant pris l'image mais également de déterminer si deux images ont été prises par le même appareil. Le lien avec notre problématique est évident étant donné que dans le cadre du face swapping on remplace une partie d'une image par une autre.

Il existe de très nombreuses méthodes pour estimer ce PRNU et malheureusement, ces méthodes utilisent la plupart du temps une importante quantité de données pour lesquelles nous connaissons l'appareil utilisé pour réaliser l'acquisition. De fait, cette condition ne peut être satisfaite dans la majorité des cas d'applications, dont le mien. C'est pourquoi, dans leur article *PRNU-Based Forgery Localization in a Blind Scenario* [11], les auteurs présentent une méthode permettant d'estimer le PRNU sans connaissances ni données labellisées a priori.

L'architecture proposée s'articule autour de l'utilisation d'un algorithme de clustering pour regrouper par source les images contenues dans le dataset utilisé pour la comparaison avec l'image analysée. S'ensuit une estimation du PRNU par maximum de vraisemblance pour chaque cluster. Une fois les PRNUs calculés, ils déterminent à quel PRNU l'image testée est la plus proche par mesure de corrélation . Enfin, on vient mesurer la corrélation entre le PRNU de référence attribué mais localement grâce à une fenêtre glissante. Ce procédé classique permet ainsi de localiser les zones attaquées.

Les résultats présentés dans le papier sont satisfaisants mais nous n'avons pas de visibilité quant aux performances en généralisation. De plus, il est évident que les résultats, notamment en généralisation, sont dépendants des données collectées au préalable et de la précision du clustering. Enfin, cette étape de clustering s'avère également coûteuse bien qu'elle laisse la possibilité d'enrichir les références et donc améliorer la robustesse du modèle.

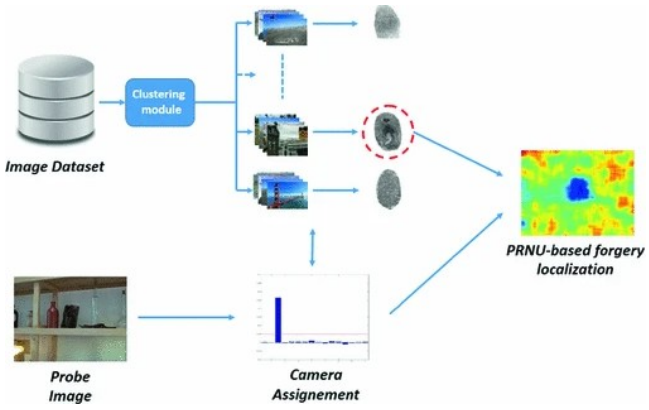


Figure 39: Architecture Blind PRNU [11]

V) Bilan

Voici dans cette section le tableau récapitulatif de cette étude comparative que j'ai menée entre les différents signaux résiduels et méthodes d'extractions associées étudiés. Quatre critères de comparaison qui me paraissaient pertinent au vu de nos contraintes et des particularités des différentes approches ont été mis en avant. Je tiens à préciser que le critère de compatibilité est un critère subjectif correspondant à la facilité de mise en place ainsi que l'adéquation des caractéristiques extraites à l'issue des différentes méthodes avec notre architecture proposée.

	BRISQUE	Hautes freqs.	LCA	CFA	Blind PRNU
Verdict	Très bon EER avec un SVM donc du potentiel	Très bonne précision avec régression	Précision variable mais bonne	Bonne AUC en blind	Très bonne AUC mais baisse pour compression JPEG
Robustesse/ Généralisation	Baisses de performances récurrentes	Très bons résultats présentés	Sensible aux changements de lentilles et peu de tests réalisés	Résultats variables mais restent bons	Sensible à la compression JPEG
Coût	SVM pré-entraîné et calculs de distributions peu coûteux	Normalisation + DCT donc très peu coûteux	Nombreux calculs de similarité coûteux	Estimations par maximum de vraisemblance donc peu coûteux	Clustering et beaucoup de données pour générer les références
Compatibilité	Score de qualité et paramètres de distributions	Spectre contenant toutes les fréquences	Possibilité de récupérer les valeurs de divergence locales	Nombreuses caractéristiques de distributions	Carte de corrélations

Tableau 2: Tableau comparatif signaux résiduels

En fonction des différents critères, j'ai décidé de me focaliser sur BRISQUE et l'analyse des hautes fréquences dans le cadre de mes travaux. Cette étude étant réalisée a priori et en me basant sur les informations que j'ai pu recueillir dans les différents article, aucune de ces pistes n'est définitivement exclue. De plus, les performances sont représentées par des métriques différentes et n'ont pas été actualisées face aux nouveaux modèles de deepfake. Comparer l'exactitude du verdict, critère principal, est donc difficile.

Selon le temps restant une fois mes expérimentations sur ces deux signaux réalisées, j'envisage de tester les autres signaux présentés dans cette étude comparative.

6- Extracteurs de caractéristiques

Comme expliqué précédemment, une de mes missions durant ce stage consistait au développement d'extracteurs de caractéristiques. Le rôle de ces extracteurs est de générer des caractéristiques explicables et exploitables par un réseau de neurones, basées sur l'analyse de signaux résiduels.

A l'image de l'architecture présentée en Figure 25, il y a autant d'algorithmes extracteurs de caractéristiques que de signaux résiduels pris en compte dans le processus de classification. A partir de l'étude comparative réalisée au préalable, je me focalise donc sur la qualité et les fréquences des images contenues dans les vidéos tout en gardant à l'esprit que l'architecture doit permettre l'ajout d'autres extracteurs sans trop de difficultés.

Dans le cadre de ce stage, j'ai réalisé un article publié en annexe n°1 qui détaille mon étude sur l'impact et la pertinence de ces signaux résiduels. Pour plus de détails relatifs aux méthodes présentées dans les sections suivantes, veuillez vous référer à cette annexe.

A) Blind/Referenceless Image Spatial Quality Evaluator

Blind/Referenceless Image Spatial Quality Evaluator, ou plus communément appelé BRISQUE, est un algorithme d'évaluation de la qualité des images introduit dans l'article *No-Reference Image Quality Assessment in the Spatial Domain* [12]. Comme son nom l'indique, cet algorithme a pour particularité de proposer une évaluation de la qualité sans référence, ce qui est indispensable dans notre cas, mais également quelque soit le type de distorsion appliquée. Ce dernier point est essentiel étant donné que les modèles deepfakes ne vont pas laisser les même traces. Ce sont ces deux atouts en matière d'algorithme d'évaluation de qualité qui m'ont poussés à sélectionner cet algorithme en tant qu'extracteur de caractéristiques de qualité.

BRISQUE est basé sur le constat selon lequel les images dîtes naturelles, par opposition aux images artificielles, possèdent des propriétés statistiques standardes qui se dégradent en présence de phénomènes de distorsion comme l'illustre la Figure 40.

Les courbes représentent la distribution des valeurs de luminance normalisées, respectivement pour une image naturelle (a) et une image artificielle (b) qui comporte de nombreux effets de distorsion. On peut facilement constater que si la distribution empirique des valeurs suit une gaussienne pour l'image naturelle, ce n'est pas le cas pour l'image artificielle.

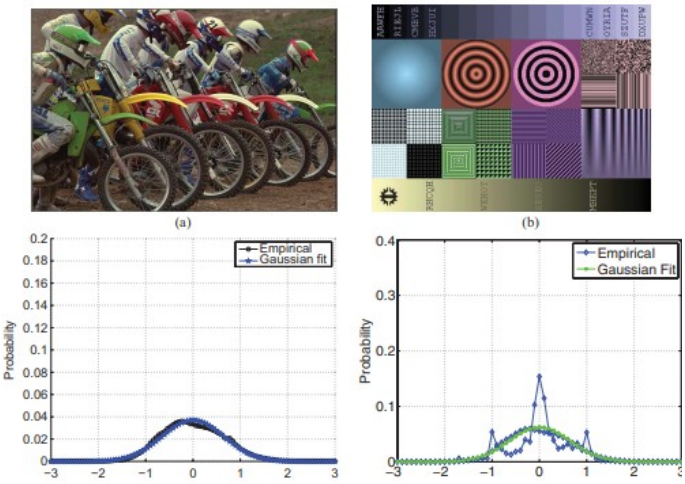


Figure 40: Distribution naturelle contre artificielle de la luminance normalisée [12]

Il est donc possible en utilisant des mesures statistiques, de quantifier cette différence dans les distributions et d'associer ces mesures à un score de qualité. Pour ce faire, comme expliqué par les auteurs, je commence par normaliser l'image de sorte à obtenir une nouvelle représentation. Cette normalisation est la *Mean Substracted Contrast Normalisation* (MSCN).

Cette normalisation de l'image d'origine se fait en soustrayant aux valeurs d'intensité le champs local moyen (μ) et en divisant par le champs local de variance (σ). De cette manière on obtient la luminance de l'image (Ï) :

$$\ddot{I}(i,j)=\frac{I(i,j)-\mu(i,j)}{\sigma(i,j)+C} \quad \text{avec} \quad \mu=W*I, \quad \sigma=\sqrt{W*(I-\mu)^2} \quad \text{et } W \text{ le flou gaussien de } I$$

Figure 41: Image avant normalisation

Figure 42: Image après normalisation

L'analyse de la distribution de cette nouvelle représentation permet de bien tenir compte de l'intensité des pixels mais manque de contexte par rapport au voisinage. C'est

pourquoi les auteurs s'intéressent à capturer l'information liée au voisinage des pixels. Par conséquent, quatre représentations dérivées de la forme normalisée sont utilisées. Il s'agit de quatre produits matriciels termes à termes entre l'image normalisée et une version décalée de celle-ci soit horizontalement (H), soit verticalement (V), soit sur la diagonale gauche($D1$) et enfin sur la diagonale de droite ($D2$).

$$H(i,j)=\hat{I}(i,j)*\hat{I}(i,j+1)$$
$$V(i,j)=\hat{I}(i,j)*\hat{I}(i+1,j)$$
$$D1(i,j)=\hat{I}(i,j)*\hat{I}(i+1,j+1)$$
$$D2(i,j)=\hat{I}(i,j)*\hat{I}(i+1,j-1)$$

Feature Range	Feature Description	Procedure
1 - 2	Shape and Variance.	GGD fit to MSCN coefficients.
3 - 6	Shape, Mean, Left Variance, Right Variance	AGGD fit to Horizontal Pairwise Products
7 - 10	Shape, Mean, Left Variance, Right Variance	AGGD fit to Vertical Pairwise Products
11 - 14	Shape, Mean, Left Variance, Right Variance	AGGD fit to Diagonal (left) Pairwise Products
15 - 18	Shape, Mean, Left Variance, Right Variance	AGGD fit to Diagonal (Right) Pairwise Products

Figure 43: Vecteur de caractéristiques

Nous nous retrouvons donc avec cinq représentations normalisées de notre image d'origine et la prochaine étape consiste à calculer les paramètres de distribution à partir de ces représentations. Pour cela, j'utilise des *Generalized Gaussian Distributions (GGD)* et des *Asymmetric Generalized Gaussian Distributions* (AGGD), comme préconisé par les auteurs [13], afin d'obtenir les différentes caractéristiques décrivant la distribution de la luminance Ï selon la représentation. J'applique ensuite ce même processus à l'image passée à l'échelle $1/2$ ce qui fait un total de 2×18 caractéristiques, soit un vecteur résultant de 36 caractéristiques.

Enfin, j'utilise le Support Vector Machine (SVM) entraîné par les auteurs, modèle très classique de machine learning, afin de prédire le score de qualité compris entre zéro et cents à partir du vecteur de caractéristiques. C'est ce score, avec zéro la qualité maximale, qui correspond à la sortie de l'algorithme BRISQUE. Le SVM quant à lui a été entraîné sur la base de données LIVE IQA pour lesquelles le score de qualité a été évalué par l'Homme.

Dans le cadre de cette étude, je me suis donc basé sur l'implémentation python de BRISQUE proposée par OpenCV que j'ai adaptée et ai réutilisé le modèle SVM entraîné par les auteurs du papier. Cela m'a permis de gagner un temps précieux et d'avoir moins de sources d'erreur. Une fois mon algorithme BRISQUE développé, mes efforts ont surtout été consacrés à l'intégration dans mon pipeline, que je détaillerai par la suite, de ce module python.

La différence principale avec l'implémentation d'OpenCV réside dans le fait que j'ai décidé de conserver le vecteur des trente-six valeurs en plus du score de qualité. En effet, je ne pouvais me contenter de récupérer un score prédit dans un soucis de transparence. De plus, cette modification permet de collecter un maximum d'informations, ce qui est primordial dans le cadre du machine learning et plus particulièrement du deep learning.

B) Mesure de l'intensité des hautes fréquences

Tout d'abord, afin de compléter ma description précédente vis-à-vis des résultats présentés dans l'article [8], il est important de préciser à quoi est due cette augmentation des hautes fréquences que j'ai mentionnée dans mon étude comparative. Si l'on répertorie les modèles de deepfake de l'état de l'art, le constat selon lequel ces modèles reposent sur l'utilisation de GANs est évident. Je ne vais pas m'attarder sur le fonctionnement de ces derniers mais il est important de comprendre que le générateur doit à partir d'un vecteur 1D obtenir une image 2D.

Pour ce faire, les GANs utilisent des blocs de sur-échantillonnage, dont le rôle est d'augmenter la dimension des données générées dans l'espace latent entre les différentes couches. Ces blocs utilisent donc des algorithmes d'interpolation pour générer les données manquantes, ce qui est la cause de cette augmentation de l'intensité des hautes fréquences. Vous trouverez un exemple de GAN en annexe n°2.

L'algorithme permettant l'extraction des fréquences présenté par les auteurs est assez simple, aussi l'ai-je implémenté pour l'intégrer à mon pipeline. La première étape consiste à normaliser les valeurs d'intensité des pixels entre $[-1,1]$. Ensuite, je passe l'image dans le domaine fréquentiel grâce à l'utilisation de la *Discrete Cosine Transform*

effectuée en deux dimensions. Afin d'obtenir une meilleure représentation, j'utilise la fonction log et je normalise par rapport à la moyenne et à l'écart-type les fréquences ainsi obtenues.

Dans l'article, les données sont intégralement transmises au modèle classifieur, ce qui n'était pas forcément la meilleure approche selon moi. En effet, en faisant de la sorte, les informations sensibles liées à l'intensité des hautes fréquences sont bruitées par celles relatives aux basses fréquences. C'est pourquoi j'ai décidé de synthétiser l'information pertinente liée aux hautes fréquences. Je détaillerai par la suite les différentes approches étudiées ainsi que les résultats obtenus en fonction des caractéristiques résumées utilisées en sortie de cet extracteur.

7- Pipeline de traitement des données

Mes précédents projets et stages m'ont permis de prendre conscience d'un enjeu à ne pas sous-estimer lorsque l'on projette de faire du machine/deep learning. La question des moyens à disposition est essentielle. En effet, sans puissance de calcul adaptée au modèle à entraîner et, encore plus important , sans données en quantités suffisantes, il est très difficile d'aboutir à des résultats. C'est pourquoi j'ai consacré une importante partie de mon temps, essentiellement au début du stage, afin de m'assurer d'avoir les ressources nécessaires.

Pour ce qui est de la question de la puissance de calcul, l'objectif étant de n'entraîner qu'un modèle classifieur sans la partie extracteurs de caractéristiques, la puissance de calcul n'était pas une contrainte forte. Néanmoins, j'ai demandé un accès à un des serveurs de calcul du GREYC, possédant deux GPUs Tesla K40M 2880 Cores avec 12G de RAM et 512G de stockage en RAM. Ainsi, j'ai pu stocker de gros volumes de données et lancer mes apprentissages en tâche de fond.

En ce qui concerne les données, j'ai dû réaliser un travail de recherche et de sélection parmi les nombreux datasets accessibles sur Internet avant de m'attaquer à la question essentielle des pipelines de prétraitement et d'extraction. En effet, ces pipelines sont indispensables pour automatiser et standardiser les traitements de nos données avant utilisation. Les sections suivantes présentent ainsi ces travaux relatifs aux données.

A) Collecte de données

Afin de pouvoir entraîner correctement mes modèles, il y avait plusieurs points relatifs au cahier des charges à prendre en compte dans cette collecte. Bien que la quantité de données soit la priorité, il était essentiel que les vidéos soient de bonne qualité et de durées variables. De plus, afin de maximiser mes chances d'obtenir un modèle robuste et généralisable aux attaques par face swapping, j'ai cherché à rassembler des vidéos pouvant contenir plusieurs visages, générées à partir de modèles deepfakes variés et issues de datasets différents. J'ai donc utilisé cinq jeux de données : VidTIMIT [14], DeepfakeTIMIT [3], Celeb-DF [15], FaceForensics (FF++) [16] et DFDC [17]. Des exemples de deepfakes sont présentés en annexe n°3, notamment un par modèle de deepfake utilisé pour générer FF++. Les données collectées comptent donc six modèles différents de deepfake utilisés pour générer les vidéos truquées.

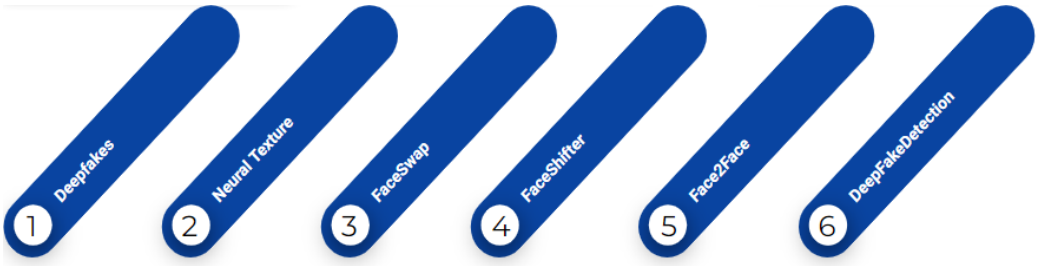


Figure 44: Différents modèles deepfake utilisés

J'ai ainsi pu rassembler 3148 vidéos parfaitement adaptées à mon problème de classification binaire, dont 1413 authentiques pour 1735 falsifiées. Il est important de souligner le léger déséquilibre entre les deux classes que l'on peut constater à l'échelle des vidéos. Ceci n'aura pas réellement d'impact étant donné que lors de l'extraction des frames, je fais en sorte d'équilibrer les classes en utilisant de la décimation.

Une fois toutes ces vidéos récupérées, j'ai travaillé à la réalisation d'une base de données uniforme et structurée. De fait, les vidéos n'étaient pas de même résolution et certaines bases de données fournissent les vidéos découpées en images contrairement à d'autres. Pour ce faire, j'ai développé un pipeline de prétraitement automatisé.

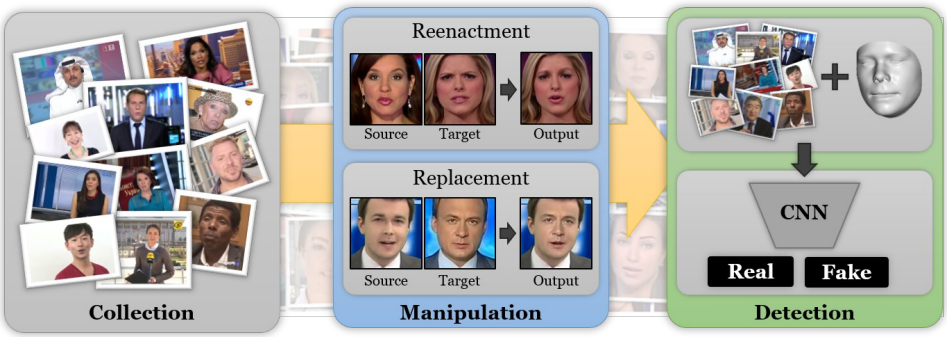


Figure 45: Cas d'usage bases de données deepfake

B) Pipeline de prétraitement

Mon pipeline se décompose en trois parties principales. La première partie sert à extraire les images des vidéos sources lorsque celles-ci n'ont pas déjà été prédécoupées par les créateurs des bases de données. Pour ce faire, j'ai utilisé la fonction VideoCapture fournie par la librairie OpenCV qui permet notamment de définir le ratio auquel on va extraire les frames du flux vidéo. De cette manière je peux définir la proportion d'images que j'extrais. A la suite de cette première étape, les images extraites sont passées à un module d'extraction de visages.

Lorsque l'on développe des modèles, il est plus simple d'avoir des entrées de même taille. C'est pourquoi je me suis intéressé à la possibilité de redimensionner mes images de sorte à ce qu'elles soient de même taille. En étudiant la question, la possibilité de rogner ou de compléter l'image (zero padding) m'est venue à l'esprit et je me suis alors demandé s'il ne serait pas avantageux de rogner autour du visage. Etant donné que dans mon étude je me concentre sur les attaques par face swapping, j'ai décidé de privilégier cette option qui constitue la deuxième partie du pipeline. Les extracteurs de caractéristiques se focaliseront sur la zone sensible ce qui peut permettre de réduire le bruit. De plus, les images ainsi générées sont moins lourdes que ce soit en termes de stockage ou de temps de calcul.

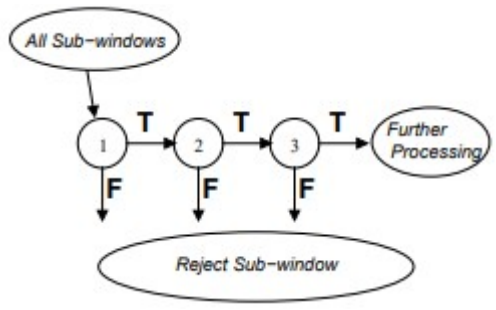


Figure 46: Processus de classification Haar [18]

J'ai donc étudié les outils les plus récurrents et j'ai donc comparé deux solutions : les filtres de Haar en cascade et le Multi-task Cascaded Convolutional Network. La première solution est présentée dans l'article *Rapid Object Detection using a Boosted Cascade of Simple Features [18]*. Il s'agit d'une méthode d'extraction de visages basées sur l'utilisation de plus de 6000 caractéristiques analysées par Adaboost en utilisant 38 classifieurs se focalisant sur un sous-ensemble de ces caractéristiques apprises. Ce modèle créé en 2001 présente, très bonnes performances ce qui en fait un incontournable.

En 2016, les auteurs de l'article *Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks* [19] présentent une nouvelle méthode d'extraction basée

sur les travaux relatifs au filtre de Haar. Il s'agit d'une version plus sophistiquée qui se base sur l'utilisation de trois réseaux d'extraction de caractéristiques basés sur les réseaux convolutionnels. Ces trois réseaux en cascade permettent d'extraire des caractéristiques profondes qui sont utilisées par un classifieur pour déterminer si le patch étudié contient un visage ou non. Depuis, le MTCNN est la méthode à l'état de l'art en matière de détection de visages. Voici le tableau comparatif que j'ai réalisé sur les deux méthodes que j'ai testées. J'ai finalement choisi d'utiliser le MTCNN.

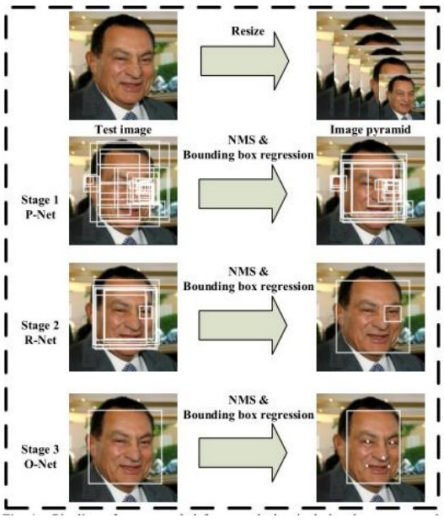


Figure 47: CNNs en cascades

	Haar	MTCNN
Précision	Bons résultats	Très bons résultats
Multi-visages	Permet d'extraire deux visages	Permet d'extraire deux visages
Explicabilité	Machine learning basé sur Adaboost donc verdict lisible	Caractéristiques relativement explicables par l'étude des filtres des CNNs donc verdict relativement lisible
Ergonomie	Modèle pré-entraîné mais plus difficile à prendre en main et à ajuster (marges autour du visage)	Modèle clés en mains fourni par FaceNet très simple d'utilisation et adaptable facilement
Coût	Relativement coûteux en temps de calcul mais peu coûteux en poids	Peu coûteux en temps de calcul mais modèle lourd à cause des nombreux poids des CNNs

Tableau 3: Tableau comparatif Haar & MTCNN

Une fois les visages extraits, la troisième étape est celle du tri et du rangement de la base de données ainsi générée. J'ai mis au point de nombreux scripts permettant d'organiser les différentes bases de données de manière uniforme tout en supprimant les éléments superflus afin de ne conserver que les visages extraits. De cette manière, j'ai pu stocker sur le serveur du GREYC les images de visage de dimension 256x256 pixels et ainsi diminuer l'espace de stockage consommé, et par extension, l'impact environnemental. Les différentes étapes de ce pipeline sont illustrées par la Figure 48.

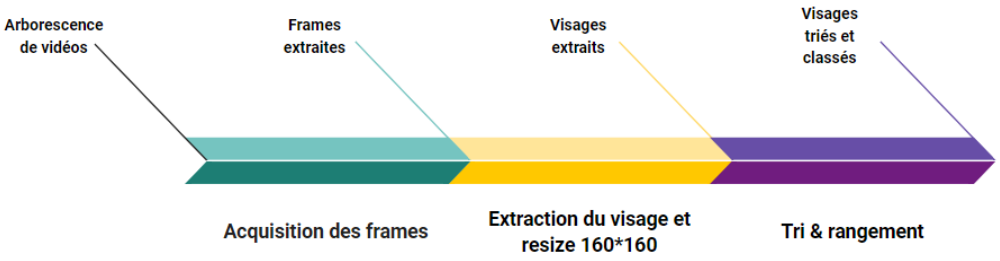


Figure 48: Pipeline de prétraitement

C) Pipeline d'extraction

Une fois les visages extraits, ces derniers doivent encore être traités afin d'obtenir les vecteurs de caractéristiques. Pour ce faire, j'ai réalisé le pipeline schématisé en Figure 49.

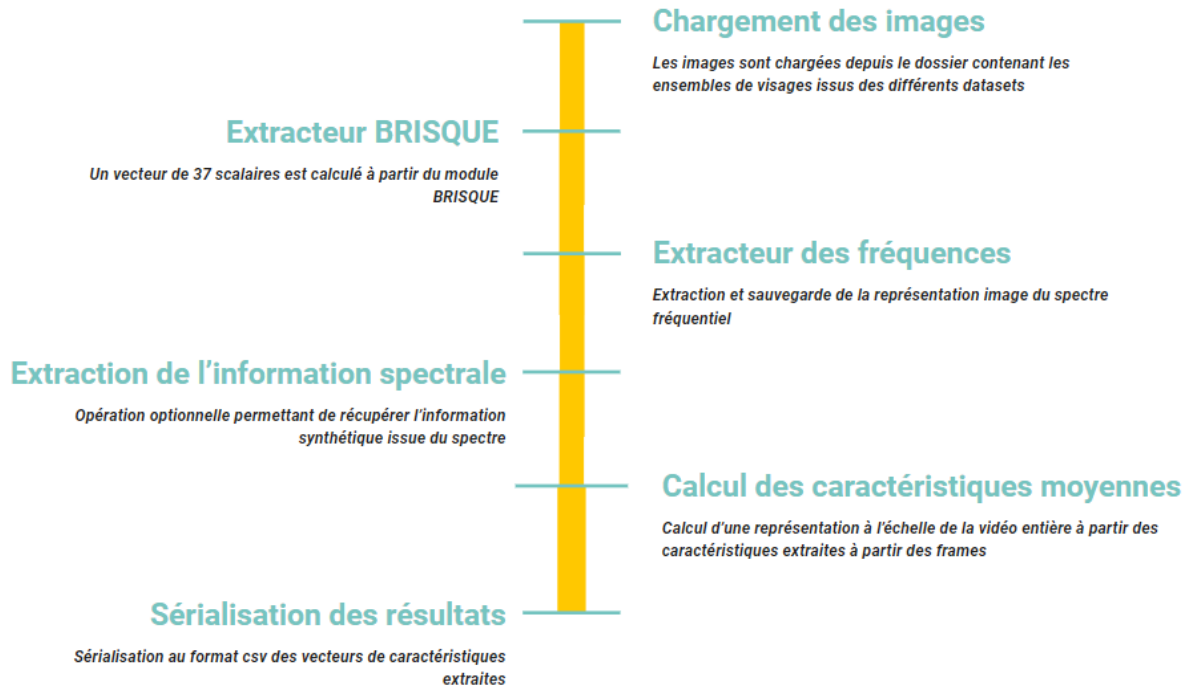


Figure 49: Pipeline d'extraction

Ce pipeline permet non seulement d'extraire les caractéristiques pour les frames prétraitées mais également de calculer des caractéristiques à l'échelle de la vidéo entière. En effet, une fois les vecteurs calculés pour chaque frame, il suffit par exemple de calculer le vecteur moyen associé afin de résumer les informations. Plusieurs aspects m'ont semblé dignes d'intérêt avec cette approche, et voici un tableau comparatif entre ces deux méthodes.

	Échelle vidéo	Échelle frames
Précision	Compression de l'information donc probablement moins bonne à cause du bruit et difficultés à généraliser	Précis car toutes les informations sont analysées sans bruit
Explicabilité	Moins de détails vis à vis du verdict mais verdict général donc moins discutable	Possibilité de cibler les frames attaquées mais quelle tolérance par rapport aux faux positifs ? Si une frame trafiquée il s'agit d'une erreur ou d'une attaque ?
Temps d'entraînement	Moins de données en entrée (plus de 100 000 à 1500) donc bien plus rapide	Beaucoup plus long
Temps d'inférence	Pas besoin de prédire le label pour chaque frame mais seulement pour la vidéo donc beaucoup plus rapide	Autant de prédictions que de frames

Tableau 4: Tableau comparatif échelles

Après avoir comparé dans le cadre de ma démarche ingénieur ces deux approches, le constat selon lequel il y a un compromis entre précision des résultats et coûts s'est levé. J'ai alors décidé de conserver les deux approches pour mes expérimentations. De fait, si l'approche à l'échelle des vidéos peut être moins précise, celle-ci est suffisamment rapide pour réaliser une analyse préliminaire et a l'avantage d'être plus respectueuse du Développement Durable. Ainsi, il serait possible de n'analyser à l'échelle des frames que si la première analyse révèle une probabilité non négligeable que la vidéo soit un deepfake.

Enfin, afin de simplifier mes apprentissages, j'ai décidé de sérialiser ces vecteurs de caractéristiques sous la forme de quatre fichiers csv. Chaque fichier csv correspond à un ensemble différent parmi les ensembles de train, de validation, de test et de généralisation. L'ensemble de généralisation correspond à un jeu de données issues d'une base de données à part servant à quantifier la capacité à généraliser du modèle.

	VidTIMIT	DeepfakeTIMIT	FF++	Celeb -DF	DFDC	Nb frames
Train	210	160	100/299	X	371/384	53811/56502
Validation	105	80	50/149	X	185/190	24447/26546
Test	105	80	50/149	X	186/192	24765/25422
Généralisation	X	X	X	51/52	X	20694/20905

Tableau 5: Composition de la base de données

8- Développement d'un modèle de référence

A) Mon premier modèle

Dans le cadre de mes enseignements, j'ai appris qu'il est primordial d'initialiser ses recherches avec des modèles simples lorsque l'on expérimente en machine/deep learning. De cette manière, on obtient un premier résultat facilement et tôt dans notre phase exploratoire que l'on va pouvoir dépasser sans trop de difficulté. C'est pourquoi j'ai fait le choix de commencer avec l'approche **« échelle vidéo »** présentée dans la section précédente à des fins de simplification.

Mon modèle de référence consiste donc en un Support Vector Machines (SVM), plus précisemment le Nu-Support Vector Classification (Nu-SVC), implémenté par la librairie scikit-learn. Ce choix était principalement motivé par le fait que le Nu-SVC est très simple d'utilisation et très utilisé dans l'état de l'art pour ses bonnes performances en classification notamment. De plus, ces modèles sont beaucoup moins coûteux que les modèles de deep learning.

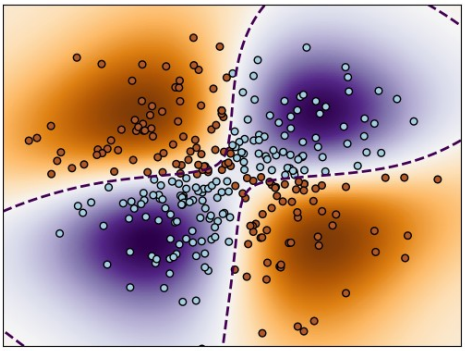


Figure 50: Illustration NuSVC avec un noyau non linéaire

Pour ce modèle, chaque vidéo était représentée par un vecteur de caractéristiques moyen calculé à partir de la moyenne des vecteurs générés, pour chaque frame de la vidéo, par nos extracteurs. **Le jeu de données DFDC n'était pas encore intégré à ce stade de mes travaux.** Les hyperparamètres n'ont pas été modifiés et sont donc ceux par défaut.

Les résultats présentés sont obtenus en appliquant un bootstrapping à 999 réplications et en prenant l'**accuracy** moyenne afin de quantifier le plus finement possible les performances en classification du modèle. Il est important de préciser qu'à ce stade de mes travaux, je me contentais de calculer le ratio du nombre de fréquences à haute intensité par rapport au nombre total de fréquences afin de quantifier la hausse d'intensité des hautes fréquences.

Caractéristiques	Train	Validation	Test	Gen
BRISQUE	77,00 %	77,00 %	76,00 %	54,00 %
Ratio fréquentiel	60,00 %	59,00 %	59,00 %	50,00 %
Concaténation	77,00 %	77,00 %	77,00 %	55,00 %

Tableau 6: Accurary modèle baseline en fonction des caractéristiques

Sans grandes surprises, nos résultats obtenus pour les caractéristiques prises individuellement sont inférieurs à ceux présentés dans les papiers cités précédemment, ce qui peut s'expliquer par le fait que nous n'avons que peu de données et que notre classifieur n'est pas optimisé. On peut néanmoins affirmer que nos caractéristiques explicables sont exploitables. Il semblerait également que conformément à nos hypothèses, le fait de concaténer les caractéristiques puisse améliorer les performances, notamment en généralisation.

B) Recherche du meilleur modèle de machine learning

J'ai fait le choix de pousser à son paroxysme la piste du machine learning avant de m'attaquer au deep learning pour avoir une référence plus compétitive. Pour cela, je me suis mis à la recherche du meilleur modèle en réalisant un benchmark avec la librairie Pycaret que j'ai pu découvrir dans le cadre de ce stage. Parmi les nombreux outils que cette librairie gratuite propose, on retrouve un outil de sélection de modèles.

Vous trouverez en annexe n°4 les résultats du benchmark réalisé pour plus de détails quant aux différents modèles testés et les performances obtenues.

Au vu des résultats de ce benchmark, il semblerait que le meilleur modèle, compte tenu de nos données, soit la Linear Discriminant Analysis (LDA). Il s'agit d'un modèle de régression très utilisé et également disponible avec scikit-learn. Afin de pousser ce modèle au maximum de son potentiel, j'ai réalisé une hyperparamétrisation par gridsearch. Voici le comparatif des résultats obtenus en prenant l'ensemble des caractéristiques (concaténation) :

Modèle	Train	Validation	Test	Gen
Baseline NuSVC	77,00 %	77,00 %	77,00 %	55,00 %
LDA	88,00 %	88,00 %	85,00 %	57,00 %

Tableau 7: Accurary du modèle LDA avec gridsearch

Il est évident que ce modèle est bien plus performant que le modèle précédent. J'ai également procédé à l'hyperparamétrisation du NuSVC et suis arrivé au même constat. Ce modèle constitue donc mon point de repère pour la suite mais la mesure de l'accuracy ne suffit pas à analyser les performances. C'est pourquoi j'ai fait le choix d'utiliser quatre métriques supplémentaires : le F1-score, l'AUC, la Précision et le Recall. Je vais donc brièvement détailler l'utilité de chacune de ces métriques étant donné que je les ai utilisées tout le long de mes travaux afin de pouvoir comparer et analyser plus finement les performances de mes modèles. Tout d'abord, il est important de rappeler que dans notre cas de classification binaire, notre modèle possède la matrice de confusion suivante :

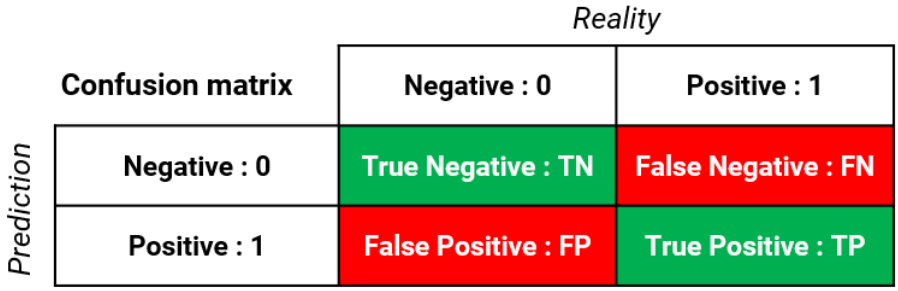


Figure 51: Matrice de confusion

- Le **Recall** permet de déterminer le pourcentage de positifs prédits. Dans notre cas cela permet de déterminer à quel point notre modèle prend le risque de rater un deepfake.

$$recall = \frac{TP}{TP + FN}$$

- La **Précision** est très liée au Recall. Elle permet de déterminer le pourcentage de positifs correctement prédits. Plus elle est élevée et moins le modèle prend des vidéos authentiques pour des deepfakes.

$$precision = \frac{TP}{TP + FP}$$

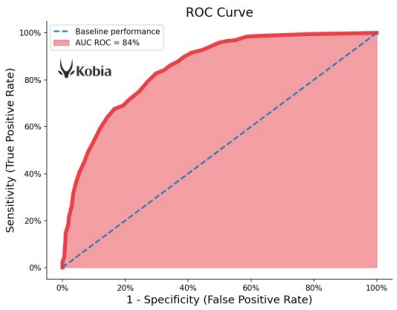
- Le **F1 Score** est une métrique permettant de tenir compte à la fois de la Précision et du Recall. Elle permet ainsi d'évaluer complètement et avec précision les performances d'un modèle tout en étant robuste au déséquilibre des classes contrairement à l 'Accuracy. Plus cette valeur est élevée et plus le modèle est performant.

$$f1\,score = 2\,x\,\frac{recall\,x\,precision}{recall + precision}$$

- L'**Accuracy** est une métrique très utilisée en classification afin d'évaluer la performance des modèles. Elle mesure le taux de prédictions correctes en accordant autant d'importance aux éléments négatifs que positifs et est donc un incontournable. Plus cette valeur est élevée et plus le modèle est performant.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- L'**AUC (Area Under the Curve)** est très utilisée pour quantifier la robustesse d'un modèle. Cette métrique mesure la capacité à maximiser à la fois la Sensibilité (capacité à correctement reconnaître les éléments de la classe positive) et la Spécificité (capacité à correctement reconnaître les éléments de la classe négative) du modèle. Il s'agit de l'aire sous la courbe ROC (Receiver Operating Characteristic). Plus cette valeur est élevée et plus le modèle est performant.



Set/Métriques	F1	Accuracy	AUC	Precision	Recall
Train	87,00 %	88,00 %	89,00 %	86,00 %	95,00 %
Validation	87,00 %	88,00 %	88,00 %	86,00 %	95,00 %
Test	84,00 %	85,00 %	84,00 %	85,00 %	91,00 %
Généralisation	55,00 %	57,00 %	57,00 %	54,00 %	94,00 %

Tableau 8: Performances du modèle LDA avec gridsearch

C) Analyse des performances

Comme on peut le constater au vu des résultats présentés dans le Tableau 8, cette version préliminaire offre de bonnes performances. En effet, les valeurs des différentes métriques en apprentissage, validation et test sont autour des 85 % en moyenne. De plus, ce modèle peu coûteux a été entraîné sur peu de données (1500 vidéos) ce qui a l'avantage de rendre le stockage et les entraînements moins coûteux. Néanmoins, le modèle est loin d'être parfait étant donné que les modèles concurrents à l'état de l'art basés sur du deep learning restent meilleurs. Enfin, une baisse significative des performances est visible sur l'ensemble de généralisation ce qui signifie que le modèle n'est pas assez robuste et indépendant de la base étudiée. Pour cela, il va être nécessaire d'améliorer notre modèle et/ou nos caractéristiques.

Afin d'avoir une meilleure compréhension des résultats, j'ai développé un module permettant de visualiser les données tout en étudiant leur corrélation et importance. J'ai mené cette étude à la fois à l'échelle des vidéos et des frames pour quantifier la difficulté à discriminer les vraies vidéos des trafiquées à partir de nos caractéristiques extraites. J'ai réalisé une Analyse en Composantes Principales ainsi qu'une Analyse discriminante linéaire. Vous trouverez en annexe n°5 les résultats obtenus lors de cette première étude.

Il semblerait que les vidéos soient plus simples à discriminer, mais avec les deux approches, nos caractéristiques ne permettent pas de discerner d'éventuels clusters. En effet, si l'on visualise nos données suivant les deux axes ayant la plus grande variance expliquée, on observe beaucoup de superposition. De plus, la variance expliquée décroît très fortement ce qui indique que certaines variables sont peu utiles pour le modèle. En effet, lorsque l'on observe le cercle des corrélations on constate dans les deux cas une potentielle redondance d'information.

Pour résoudre ces problèmes, j'ai fait le choix d'améliorer la partie classifieur en utilisant de l'apprentissage profond. En effet, le deep learning est moins dépendant de la représentation des données et cette piste doit être explorée afin de pouvoir comparer nos résultats avec l'état de l'art. Enfin, étant donné que nos caractéristiques prennent la forme d'un vecteur de scalaires, il n'est pas difficile d'adapter le modèle à la nouvelle représentation. De plus, la redondance et la présence de variables corrélées est moins problématique puisqu'il est nécessaire d'avoir le plus de données possibles.

9- Développement d'un classifieur profond

En utilisant les résultats obtenus avec mon modèle de référence, j'ai travaillé au développement d'un classifieur profond, conformément à ma troisième mission dans le cadre de ce stage. L'objectif principal est donc d'obtenir les meilleurs résultats possibles.

A) Développement du pipeline d'entraînement

Afin de développer un pipeline fiable et durable, je me suis focalisé sur quatre aspects : la supervision de mon modèle (logger), le chargement des données (dataloader), le processus d'entraînement (trainer) et enfin le modèle en lui-même. De part ma formation et mes expériences passées dans le domaine, je préfère travailler sous PyTorch qui offre selon moi beaucoup de contrôle sur ces différents aspects contrairement à TensorFlow. Cependant, développer des modèles en PyTorch peut vite s'avérer complexe et chronophage. De plus, il est facile d'aboutir à des modèles sous-optimisés, ce qui risque de ralentir grandement le processus d'apprentissage. C'est pourquoi les chercheurs travaillent souvent avec PyTorch Lightning qui est un framework reprenant l'ensemble des fonctionnalités de PyTorch tout en améliorant de nombreux points :

Figure 52: Logo Pytorch Lightning

- classes Dataloader et Trainer optimisées pour accélérer les apprentissages
- logger intégré pour la supervision
- classes principales de PyTorch et métriques conservées
- facilités de développement et de déploiement
- indépendance des appareils
- parallélisme et optimisation des ressources GPU facilités

Grâce à ce framework, j'ai été en mesure d'implémenter rapidement un pipeline léger, optimisé et complet gérant les quatre aspects cités précédemment. Vous trouverez le diagramme de classes associé en annexe n°6. Pour ce qui est de la supervision, j'ai configuré WandB en utilisant l'option logger de sorte à archiver chacun de mes entraînements avec les résultats obtenus pour chaque métrique citée précédemment.

B) Premiers résultats

Le premier modèle mis en place était également très simple, basé sur l'utilisation de couches denses. L'objectif principal était de déterminer s'il y avait du potentiel dans cette approche plus coûteuse que notre modèle de référence. Deux couches denses suivies de couches d'activations RELU se succèdent avec 38 neurones (nombre de caractéristiques en entrée), suivies d'une dernière couche dense avec une sigmoïde en sortie. La fonction de coût utilisée est la CrossEntropy binaire de PyTorch étant donné que nous sommes dans un contexte de classification binaire avec des classes plutôt équilibrées (autant de vidéos truquées que de vidéos authentiques). Le nombre d'epochs a été fixé arbitrairement à 100, le ratio d'apprentissage à 10e-4 et la taille des batchs à 10. Les vecteurs de caractéristiques utilisés sont identiques à ceux utilisés par le précédent modèle **(dataset DFDC pas inclus)**.

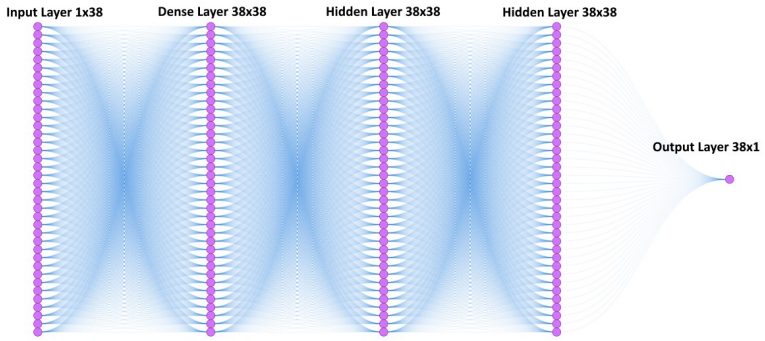


Figure 53: Modèle Deep Learning basique

Les résultats obtenus à l'échelle des vidéos sont présentés dans le Tableau 9. On peut constater une légère baisse des performances par rapport à celles obtenues avec notre modèle de référence mais un gain en généralisation. Étant donné que le modèle reste très basique, la marge de progression est très importante et il est nécessaire de creuser.

Set/Métriques	F1	Accuracy	AUC	Precision	Recall
Train	81,00 %	78,00 %	78,00 %	81,00 %	84,00 %
Validation	83,00 %	80,00 %	78,00 %	82,00 %	85,00 %
Test	84,00 %	80,00 %	77,00 %	80,00 %	89,00 %
Généralisation	68,00 %	53,00 %	52,00 %	52,00 %	100,00 %

Tableau 9: Performances du modèle DL basique échelle vidéos

J'ai également testé ce modèle à l'échelle des frames où un vecteur correspond aux caractéristiques d'une frame et non plus d'une vidéo entière. L'avantage de cette approche réside principalement dans le fait que l'on obtient un verdict beaucoup plus précis et donc explicable. De plus, les frames sont beaucoup plus nombreuses et mieux équilibrées en terme de classes que les vidéos (1500 vidéos pour plus de 150 000 frames) ce qui peut avoir un gros impact lors de l'apprentissage. Les résultats obtenus sont présentés dans le Tableau 10.

Set/Métriques	F1	Accuracy	AUC	Precision	Recall
Train	83,00 %	83,00 %	86,00 %	89,00 %	77,00 %
Validation	83,00 %	81,00 %	86,00 %	83,00 %	82,00 %
Test	81,00 %	81,00 %	88,00 %	84,00 %	77,00 %
Généralisation	67,00 %	52,00 %	62,00 %	51,00 %	96,00 %

Tableau 10: Performances du modèle DL basique échelle frames

Comme on pouvait s'y attendre, on observe que les performances à l'échelle des frames ont tendance à être supérieures à celles obtenues à l'échelle des vidéos. L'AUC est la métrique qui augmente le plus, ce qui signifie que le modèle discrimine mieux les vidéos authentiques des vidéos falsifiées. C'est pourquoi j'ai fait le choix de me concentrer sur l'approche à l'échelle des frames qui selon moi a plus de potentiel et est plus adaptée à notre contrainte d'explicabilité du verdict.

C) Amélioration du modèle

Afin d'améliorer les performances, j'ai réalisé de nombreuses expérimentations dans le but de trouver un modèle plus performant. Je me suis intéressé à de nombreux aspects que sont :

- le nombre de couches denses/de neurones
- les hyperparamètres (taille des batchs, nombre d'epochs, etc.)
- la normalisation
- le dropout
- la fonction de perte/la couche d'activation de sortie utilisée

Pour des besoins de synthétisme, je ne présenterai pas toutes ces expérimentations mais vous pourrez trouver une partie des résultats obtenus en annexe n°7. J'ai ainsi pu observer que quatre couches denses étaient suffisantes pour obtenir de bons résultats et ai trouvé un nombre de neurones adapté pour chaque couche.

Ensuite, le modèle étant extrêmement léger et optimisé, j'ai pu utiliser un scheduler pour l'optimiseur de sorte à adapter le ratio d'apprentissage et éviter le sur-apprentissage. L'ajout d'une couche de batchnormalisation en entrée et d'une couche de droupout a permis d'améliorer/stabiliser légèrement les performances du modèle en test et en généralisation.

Enfin, pour ce qui est de la fonction de perte utilisée et de l'activation de sortie, j'ai conservé une sigmoïde avec une BCEWithLogitsLoss qui est une version alternative de la BCELoss (CrossEntropie Binaire) classique proposée par PyTorch. La couche d'activation sigmoïde permet d'obtenir une sortie à 0 ou 1 et la BCEWithLogitsLoss quant à elle est plus stable numériquement que la BCELoss classique. Cependant, toujours dans l'optique d'avoir le verdict le plus explicable, j'ai également testé l'utilisation d'une softmax en sortie pour obtenir une probabilité et non une classe directement, ce qui est plus interprétable et permet d'avoir plus de visibilité quant à la confiance du modèle dans son verdict. Pour cela, j'ai du utiliser une CrossEntropy classique et modifier la sortie du modèle de sorte à avoir deux neurones et non plus un.

Les résultats présentés dans le Tableau 11 et le Tableau 12 sont obtenus avec le même modèle, une taille de batch de 1024 pour 100 epochs et un ratio évolutif d'apprentissage de départ de 5e-3. La seule différence se situe donc dans la fonction de perte utilisée et la couche d'activation finale comme expliqué précédemment.

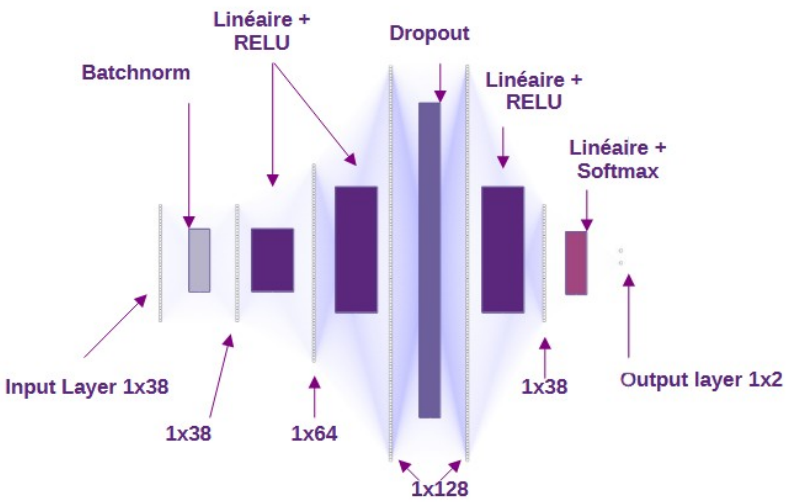


Figure 54: Modèle avancé avec Softmax

Set/Métriques	F1	Accuracy	AUC	Precision	Recall
Train	96,00 %	96,00 %	99,00 %	97,00 %	95,00 %
Validation	84,00 %	82,00 %	88,00 %	83,00 %	86,00 %
Test	85,00 %	84,00 %	89,00 %	83,00 %	87,00 %
Généralisation	60,00 %	49,00 %	52,00 %	50,00 %	75,00 %

Tableau 11: Modèle amélioré BCE et Sigmoïde

Set/Métriques	F1	Accuracy	AUC	Precision	Recall
Train	97,00 %	97,00 %	97,00 %	97,00 %	97,00 %
Validation	85,00 %	84,00 %	84,00 %	83,00 %	88,00 %
Test	86,00 %	87,00 %	86,00 %	83,00 %	91,00 %
Généralisation	60,00 %	49,00 %	49,00 %	50,00 %	77,00 %

Tableau 12: Modèle amélioré CrossEntropy et Softmax

Il est évident que les résultats obtenus avec ces deux nouveaux modèles sont bien supérieurs à ceux présentés précédemment au regard des résultats en entraînement (83 % contre 97 % pour le F1). Cela prouve que notre modèle apprend efficacement. Le principal inconvénient réside dans le fait que les performances sont moins bonnes en généralisation pour un gain assez faible en validation et en test.

On peut également constater que le changement de fonction de perte et de couche d'activation en sortie a un léger impact sur les résultats. En effet, les performances générales sont légèrement meilleures et surtout en test pour le modèle Softmax au détriment de l'AUC qui diminue légèrement contrairement au modèle Sigmoïde qui a une très bonne AUC. Cela peut s'expliquer du fait que la Softmax renvoit des probabilités et non pas une classe, ce qui rend les verdicts plus nuancés. J'ai donc fait le choix de conserver ce modèle Softmax qui offre davantage de lisibilité dans le verdict et de très bonnes performances.

Afin de m'assurer que l'écart entre les résultats en entraînement et le reste ne soit pas dû à du sur-apprentissage, j'ai étudié les courbes des fonctions de perte présentées en Figure 56 et Figure 55 . On peut constater au vu de la forme de la courbe en validation qu'il n'y a pas de sur-apprentissage, ce qui indique que le dropout et la batchnormalisation font bien effet. Le reste des courbes pour ces deux modèles sont disponibles en annexe n°8.

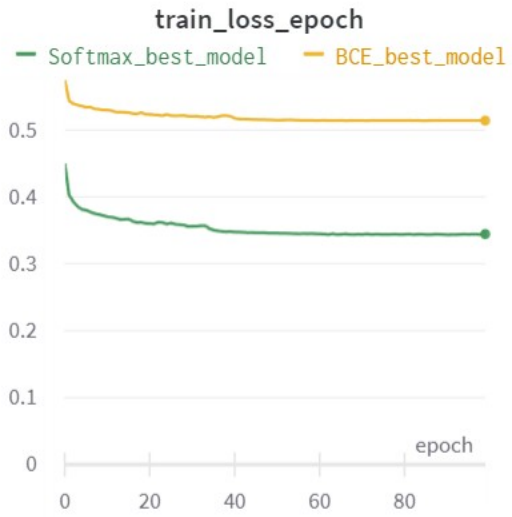


Figure 56: Fonction de perte CrossEntropy train

Figure 55: Fonction de perte CrossEntropy validation

D) Amélioration des données

L'écart entre les résultats en test de l'ordre de 80 % et ceux en généralisation de l'ordre de 50 % est significatif et c'est pourquoi j'ai cherché un moyen de corriger cet écart. Les données utilisées en généralisation proviennent du jeu de données Celeb-DF or ce dernier utilise des modèles de deepfake plus élaborés que ceux rencontrés à l'entraînement. Il est toutefois important de préciser que je ne m'étais pas rendu compte de ce biais et c'est pourquoi, plutôt que de changer de jeu de données pour la généralisation, j'ai cherché à intégrer des vidéos falsifiées générées par des modèles tout aussi performants.

C'est dans ce but que j'ai décidé d'intégrer des échantillons de la base de données DFDC développée par Facebook et publiée publiquement dans le cadre de la compétition DeepFake Detection Challenge. Le jeu de données entier regroupe plus de 500Go de vidéos

et c'est pourquoi, pour respecter la dimension développement durable, j'ai fait le choix de n'utiliser qu'un sous-ensemble contenant 1500 vidéos sur les 100 000 disponibles. Une fois les données prétraitées en utilisant mon pipeline, j'ai été en mesure d'extraire les caractéristiques et de les intégrer à mes données d'apprentissage qui sont donc passées de 165 000 frames à plus de 250 000.

Si l'un des facteurs pouvant impacter mes résultats correspond bien à la quantité de mes données et à leur nature, l'extraction de caractéristiques pertinentes reste décisive. Plusieurs choix s'offraient à moi, à savoir créer de nouveaux extracteurs de caractéristiques ou bien améliorer les existants. Pour des raisons de manque de temps, j'ai privilégié l'amélioration des extracteurs existant.

J'ai donc cherché une approche plus appropriée afin de quantifier l'intensité des hautes fréquences. En effet, je m'étais contenté de calculer le ratio des pixels à haute intensité par rapport au nombre de pixels total dans les spectres fréquentiels convertis en images. Mon hypothèse était que l'augmentation de l'intensité des hautes fréquences ferait augmenter ce ratio en partant du principe que l'intensité des basses fréquences ne diminuerait pas ou peu suite au trucage. Pour réaliser cette mesure, j'ai donc utilisé un seuillage à 150 étant donné que les valeurs d'intensités d'une image sont comprises entre 0 et 255. Ce seuil relativement bas avait pour objectif de ne pas trop lisser le signal et ne pas relever que les pics d'intensité uniquement et ainsi passer à côté d'une augmentation plus subtile. C'est également le seuil qui expérimentalement donnait les meilleurs résultats.

$$ratio = \frac{len(DctImage > 150)}{len(DctImage)}$$

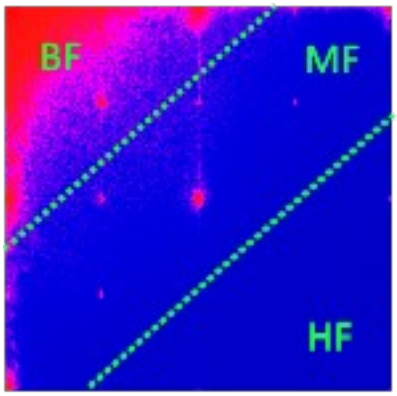


Figure 57: Analyse de l'intensité des fréquences

Comme illustré dans le Tableau 3 de l'annexe n°1, les résultats expérimentaux semblent bien indiquer que le ratio augmente pour les images trafiquées. Cependant, cette variation est trop faible pour être satisfaisante. De plus, la mesure contient beaucoup trop de bruit puisque l'on prend en compte toute l'image du spectre et donc les basses fréquences dans la mesure. C'est pourquoi j'ai cherché un moyen d'isoler les hautes fréquences.

Comme vous pouvez le voir en Figure 57, le spectre fréquentiel d'une image peut être représenté sous la forme d'une image avec pour valeurs de pixels, l'intensité de la fréquence associée. Les spectres sous cette forme peuvent être découpés en trois parties qui correspondent aux valeurs des basses, moyennes et hautes fréquences (BF, MF et HF).

En rehaussant le contraste de l'image ainsi générée, on voit clairement les pics d'intensité dans les moyennes et hautes fréquences contrairement aux basses fréquences qui ont une intensité moyenne très élevée. C'est pourquoi il est nécessaire de ne pas prendre en considération les valeurs dans la partie supérieure gauche des spectres. Pour ce faire, j'ai étudié un algorithme de parcours qui s'appelle le parcours zigzag. Comme son nom l'indique, le principe est de lire les valeurs contenues dans les pixels dans un ordre dessinant un zigzag.

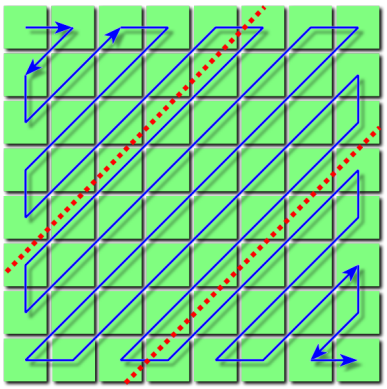


Figure 58: Parcours zigzag

Cette lecture particulière me permet ainsi comme illustré par la Figure 58 de stocker les valeurs d'intensité dans un vecteur à une dimension et de les ordonner par valeur de fréquences. Ainsi, j'obtiens un vecteur à une dimension que je peux découper en trois parties égales. Chaque tiers correspond aux valeurs d'intensités associées à une catégorie de fréquences parmi les basses, les moyennes et les hautes. J'ai finalement fait le choix de conserver les hautes mais également les moyennes fréquences étant donné qu'elles sont également impactées et de manière moins subtile comme l'illustre la Figure 57.

Grâce à cet algorithme de parcours en zigzag que j'ai pu mettre au point en python, j'ai pu exclure une importante quantité de bruit. Néanmoins, le fait de n'avoir qu'une seule valeur mesurant la surreprésentation des hautes fréquences n'était pas satisfaisant. En effet, dans l'article d'origine présenté précédemment, les auteurs utilisaient le spectre entier. C'est pourquoi le fait de compresser autant d'information à un seul scalaire me paraît problématique.

Afin d'obtenir un maximum d'information explicable sur ces valeurs d'intensité des fréquences, je me suis orienté sur des mesures statistiques. J'ai mis au point une fonction permettant grâce au module scikit-learn de calculer efficacement la moyenne, l'écart-type et

les quartiles afin d'estimer la distribution gaussienne de l'intensité. Pour affiner ma mesure, je calcule également la kurtosis et la skewness. La kurtosis permet de mesurer le degré d'aplatissement de la courbe tandis que la skewness mesure son degré d'asymétrie.

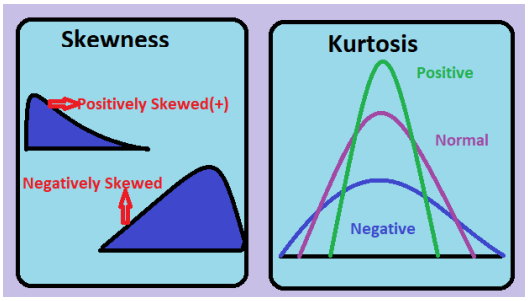


Figure 59: Skewness et Kurtosis

Ainsi, l'information relative à l'intensité des fréquences n'est plus compressée à une unique valeur bruitée mais à 14 indicateurs statistiques décrivant la distribution de l'intensité pour les moyennes et les hautes fréquences. C'est pourquoi j'espérais que cette nouvelle représentation plus détaillée, combinée à l'augmentation du nombre et de la qualité de mes données, améliorerait les performances du modèle. Afin de quantifier cette amélioration, j'ai donc repris le modèle Softmax sans modifier les hyperparamètres présenté précédemment et ai juste adapté la dimension de la couche en entrée pour correspondre au nombre de caractéristiques qui passe donc de 38 à 51. Les résultats obtenus sont présentés dans le Tableau 13 et les courbes sont présentées en annexe n°9.

Set/Métriques	F1	Accuracy	AUC	Precision	Recall
Train	96,00 %	96,00 %	96,00 %	96,00 %	96,00 %
Validation	87,00 %	86,00 %	86,00 %	85,00 %	90,00 %
Test	87,00 %	86,00 %	86,00 %	82,00 %	93,00 %
Généralisation	61,00 %	53,00 %	53,00 %	53,00 %	74,00 %

Tableau 13: Performances du Softmax final

Les résultats obtenus sont prometteurs puisque l'on observe une légère amélioration en généralisation comme escompté et ce sans diminuer les performances en test. Il est néanmoins évident que les performances sont loin d'être parfaites et qu'une importante

marge de progression subsiste (surtout en généralisation). En utilisant mon module de visualisation, j'ai été en mesure de constater que nos caractéristiques sont encore difficiles à discriminer, ce qui se confirme au vu de nos valeurs d'AUC qui bien qu'élevées stagnent à 86 % en test. Les résultats des analyses par PCA et LDA sont présentés en annexe n°10.

J'ai également mené quelques expérimentations similaires concernant l'échelle des vidéos avec un modèle de deep learning mais n'ai pas eu le temps de creuser suffisamment. C'est pourquoi j'ai fait le choix de ne pas développer cette partie dans cette section de mon rapport. Des résultats sont néanmoins disponibles en annexe n°11.

10- Livrable final

Dans le cadre du Projet INSA Certifié (PIC) et de mes précédents stages, j'ai appris à quel point la livraison d'un projet est une étape décisive dans son cycle de vie. Néanmoins, cette étape qui constitue la fin de mes travaux sur ce projet a été anticipée tout au long de mon stage. L'un des principaux outils que j'ai utilisés à cet effet est GitLab. GitLab permet non seulement d'archiver et de partager du code simplement mais également de gérer efficacement les versions par le mécanisme des branches et des tags. Ayant acquis de l'expérience dans l'utilisation de ce logiciel j'ai donc fait le choix de l'utiliser pour ce projet. J'ai ainsi pu archiver de très nombreuses versions du projet tout au long de ce stage qui sont conservées et étiquetées afin de pouvoir revenir facilement à une version antérieure.

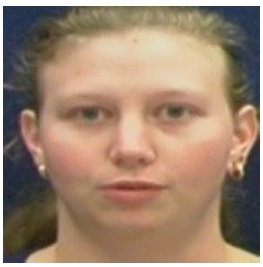
La version finale qui correspond à la version 6.0 a donc été livrée sur cet espace GitLab avec les éléments principaux suivants :

- un ReadMe.txt contenant la documentation relative au livrable
- l'ensemble des données au format json et csv utilisées en entraînement
- un échantillon de données de test brutes (images/vidéos) et un module de tests unitaires
- l'ensemble des modèles entraînés
- l'ensemble des modules développés et utilisés dans la version finale du projet documentés (DocString)
- les pipelines distincts d'entraînement et d'inférence ainsi que leurs diagrammes de classe associés (annexe n°12 pour l'inférence)

Le pipeline d'inférence est donc un module contenant plusieurs classes permettant de charger et d'utiliser le modèle afin de prononcer un verdict. Les poids du modèles sont donc figés et ces derniers sont chargés à partir d'un fichier pytorch contenant ces derniers. Les poids en question sont générés à l'apprentissage du modèle et enregistrés par le biais du pipeline d'entraînement. J'ai fait le choix de ne pas sauvegarder le modèle en entier (la structure également) mais uniquement les poids afin de diminuer l'espace de stockage requis et d'avoir plus de souplesse quant à l'instanciation du modèle à utiliser.

De la même manière que celui d'apprentissage, le pipeline d'inférence permet de charger des données brutes (vidéo, gif ou ensemble de frames) et d'en extraire les caractéristiques appropriées. Une fois les caractéristiques générées, celles-ci sont passées en entrée du modèle chargé qui retourne une prédiction. Ce processus est presque identique à celui de l'apprentissage, à la différence près qu'il n'y a pas d'apprentissage des poids et donc de rétro-propagation du gradient.

Ce module python d'inférence est donc utilisable en l'état et permet, comme décrit dans le cahier des charges, de générer un verdict pour chaque frame qui, grâce au modèle Softmax, peut être associé à une probabilité interprétable pour plus de transparence. Voici un exemple de ce que ce module produit :



Exemple authentique

Exemple deepfake

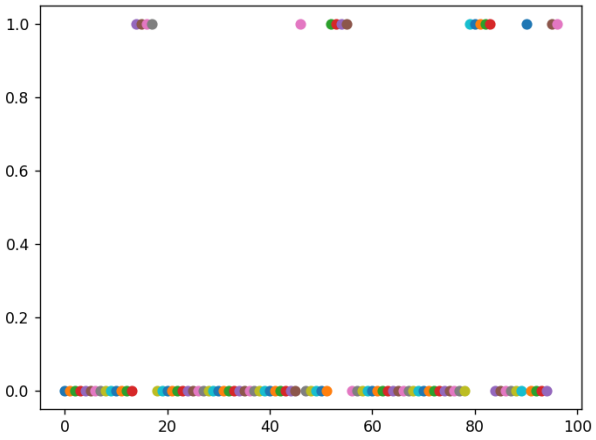


Figure 60: Prédiction du modèle Softmax

La Figure 60 correspond aux prédictions pour chaque frames extraites d'un GIF trafiqué en intercalant des images deepfake. Une prédiction à 1 correspond à une frame truquée et 0 correspond à une authentique. On constate bien à travers cet exemple que le modèle est capable de discriminer le vrai du faux avec un verdict lisible et explicable grâce aux caractéristiques extraites exploitées.

11- Développement Durable et Responsabilité Sociétale (DDRS)

Dans le cadre de mes travaux, j'ai été amené à faire de nombreux choix de conception et de réalisation. Le respect du développement durable, ainsi que la prise en compte des impacts sociétaux, ont été des critères essentiels dans mes études comparatives. En ce qui concerne la dimension environnementale, j'ai principalement concentré mes efforts sur les aspects de dimensionnement du projet. C'est pourquoi, en limitant ma quantité de données et en créant un modèle le plus léger possible, je pense avoir été en mesure de limiter l'impact environnemental de ce projet. De plus, ceci a contribué à ce que la solution soit viable économiquement de part son faible coût tout en conservant la notion d'équité.

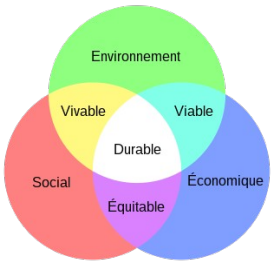


Figure 61: Diagramme de Venn du DD

Un autre point essentiel est donc celui de l'impact sociétal qui était au cœur de mes travaux avec la question de l'explicabilité et de la justesse des verdicts rendus. En effet, ces contraintes ont pour objectif majeur de pouvoir interpréter et justifier les verdicts, afin d'éviter toute injustice. Cette démarche s'inscrit également dans la dynamique Européenne de réglementation de l'Intelligence Artificielle qui se développe de plus en plus comme je m'indiquais précédemment dans mon rapport.

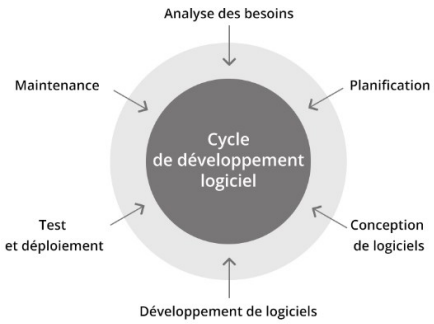


Figure 62: Cycle de vie du projet

Toujours dans cette optique de respect du développement durable, j'ai fait tout mon possible pour garantir la durée de vie de ce projet. L'architecture développée a de fait été conçue de manière modulaire, afin de permettre d'ajouter ou de supprimer des extracteurs de caractéristiques facilement. De cette manière, on peut espérer que le modèle pourra évoluer continuellement. De plus, les deux pipelines développés sont optimisés et peuvent tourner sur toutes les machines grâce à PyTorch Lightning, ce qui assure un meilleur portage. Enfin, la documentation développée en respectant les normes de la DocString, ainsi que les tests unitaires développés permettront une meilleure maintenance et reprise du projet et donc augmenteront drastiquement sa durée de vie.

Conclusion

Dans le cadre de ce stage, j'ai pu conformément aux objectifs définis dans notre cahier des charges développer un modèle capable de détecter les deepfakes vidéos en ce concentrant plus particulièrement sur le face swapping grâce à un mécanisme d'extraction de caractéristiques basé sur l'analyse de signaux résiduels. Les performances obtenues sont très satisfaisantes pour une première étude si l'on se compare à certains résultats présentés à l'état de l'art :

- 97 % d'accuracy en train/validation/test pour le réseau profond convolutionnel récurrent [5] contre 96 %/84 %/86 % pour mon modèle
- 91 % d'AUC en moyenne en test avec un ensemble de réseaux convolutionnels [20] contre 86 % pour mon modèle
- 80 % d'accuracy en moyenne en train/validation/test pour un réseau convolutionnel utilisant le flux optique en tant que signal résiduel [21] contre 89 % pour mon modèle

Si les performances atteintes par le modèle présenté dans ces travaux sont inférieures à celles obtenues par des réseaux convolutionnels profonds plus classiques, les résultats restent néanmoins très encourageants. De plus, conformément à notre cahier des charges, notre architecture possède d'autres avantages tels que son explicabilité grâce à l'utilisation de caractéristiques explicables (signaux résiduels), ainsi que sa durabilité (modularité des extracteurs de caractéristiques) et dans une moindre mesure sa capacité à généraliser. Ensuite, les contraintes liées au dimensionnement du modèle en terme de temps d'entraînement, stockage et temps d'inférence étaient assez souples l'objectif premier étant l'obtention de résultats significatifs. J'ai néanmoins réussi, afin de diminuer l'impact environnemental, à produire un modèle très léger, rapide et peu coûteux avec 15 000 paramètres seulement, un temps d'apprentissage en minutes et d'inférence en secondes.

Parvenir à ce résultat n'a pas été chose facile et ce stage a été l'occasion de non seulement approfondir mes connaissances du monde de la recherche mais également d'apprendre à mener des activités de recherche. Ce stage orienté recherche m'a permis de développer mes compétences en matière d'analyse de l'existant par le biais de la réalisation d'états de l'art tout en me poussant à développer un regard scientifique critique nécessaire pour expérimenter de nouvelles choses. Mes travaux m'ont également permis de plus clairement percevoir la différence entre les métiers d'ingénieur et de chercheur dont j'ai eu les deux casquettes dans le cadre de ce stage. De fait, j'ai utilisé de nombreuses connaissances acquises dans le cadre de ma formation aussi bien à l'INSA qu'à l'Université avec le Master SID.

Le plus difficile selon moi a été de ne pas m'éparpiller. En effet, le champs des possibles étant très vaste, j'ai été à de multiples reprises submergé par mes idées. J'ai donc du apprendre à prendre du recul afin de définir mes priorités correctement et je perdrai moins de temps à l'avenir dans mes phases d'expérimentation. Selon les sujets, il peut être difficile de comparer les résultats que l'on obtient, ce qui a été un des problèmes que j'ai rencontrés. Dans le cadre de mes soumissions d'articles, on m'a encouragé à comparer davantage mes résultats à l'état de l'art et j'ai compris qu'il aurait été préférable que je m'inquiète plus tôt de cette nécessité de trouver des résultats comparables afin de pouvoir analyser les performances de mon modèle plus précisément. Enfin, ce stage a été pour moi l'occasion de m'approprier la librairie PyTorch et de découvrir PyTorch Lightninig ce qui me sera très utile pour la suite de ma carrière.

Même si j'ai commis quelques erreurs, j'ai également réussi à éviter certains biais. Je pense avoir pu, grâce à mes expériences passées, concrétiser le besoin en un cahier des charges détaillé, ce qui m'a souvent fait défaut dans le cadre d'activités de recherche. Mes efforts en matière de méthodologie de travail et plus précisément de planification m'ont également permis de maintenir un rythme soutenu et constant. Mes formations à l'INSA et à l'Université m'ont permis d'expérimenter de nombreuses pistes grâce à mon champs d'expertise élargi et d'être force de proposition lors des discussions scientifiques, non seulement relatives à mon sujet, mais également aux travaux de l'équipe SAFE. Ensuite, je pense avoir su m'intégrer rapidement à l'équipe, de sorte à faciliter les échanges qui sont la clef dans la recherche scientifique, notamment en participant à de nombreuses activités du laboratoire (séminaires, soumissions/corrections d'articles, etc.). Enfin, je pense avoir été en mesure de produire un code propre et documenté, respectant les principaux enjeux du développement durable ainsi que les principaux objectifs de ce stage orienté recherche.

Ces travaux constituent selon moi une preuve de concept, une base qu'il faut approfondir. Pour cela, il est nécessaire d'utiliser davantage de signaux résiduels parmi ceux étudiés et de réaliser un réseau profond non explicable entraîné sur les mêmes données afin de pousser le comparatif entre l'approche classique en boîte noire et l'approche explicable. De plus, cela permettra de trouver une meilleure représentation étant donné qu'actuellement, au vu des résultats de notre PCA et LDA, les classes sont difficilement distinguable. Le modèle est également perfectible et il serait intéressant d'ajouter de l'attention afin d'améliorer l'utilisation de nos caractéristiques explicables par le modèle et d'étudier la possibilité d'utiliser des couches convolutionnelles ainsi qu'un LSTM. De cette manière, on pourrait travailler avec des caractéristiques plus efficaces sans perdre en explicabilité. Ce faisant, on peut espérer augmenter les performances notamment en généralisation du modèle qui constituent selon moi le principal point faible actuellement.

Index des tableaux

Bibliographie

1: Luisa Verdoliva, Media Forensics and Deepfakes: An Overview, IEEE Journal of Selected Topics in Signal Processing, 2020

2: Ewerton Silva, Tiago Carvalho, Anselmo Ferreira , Going deeper into copy-move forgery detection: Exploring image telltales via multi-scale analysis and voting processes, 2015

3: Pavel Korshunov, Sébastien Marcel, DeepFakes: a New Threat to Face Recognition? Assessment and Detection., arXiv, 2018

4: Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative adversarial networks, ACM, 63, 139-144, 2020

5: David Güera, Edward J. Delp, Deepfake Video Detection Using Recurrent Neural Networks, 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 1-6, 2018

6: Ruben Tolosana, Sergio Romero-Tapiador, Ruben Vera-Rodriguez, Ester Gonzalez-Sosa, Julian Fierrez, DeepFakes detection across generations: Analysis of facial regions, fusion, and performance evaluation, Engineering Applications of Artificial Intelligence, 2022

7: Javier Galbally, Sebastien Marcel, Face Anti-spoofing Based on General Image Quality Assessment, 22nd International Conference on Pattern Recognition, 1173-1178, 2014

8: Joel Frank, Thorsten Eisenhofer, Lea Schonherr, Asja Fischer, Dorothea Kolossa, Thorsten Holz, Leveraging Frequency Analysis for Deep Fake Image Recognition, 2020

9: Owen Mayer, Matthew C. Stamm, Accurate and Efficient Image Forgery DetectionUsing Lateral Chromatic Aberration, IEEE Transactions on Information Forensics and Security, 13, 1762-1777, 2018

10: Pasquale Ferrara, Tiziano Bianchi, Alessia De Rosa, Alessandro Piva, Image Forgery Localization

via Fine-Grained Analysis of CFA Artifacts, IEEE Transactions on Information Forensics and Security, 7, 1566-1577, 2012

11: Davide Cozzolino, Francesco Marra, Giovani Poggi, Carlo Sansone et Luisa Verdoliva, PRNU-Based Forgery Localization in a Blind Scenario, Image Analysis and Processing, 569-579, 2017

12: Anish Mittal, Anush Krishna Moorthy, Alan Conrad Bovik, No-Reference Image Quality Assessment in the Spatial Domain, IEEE Transacations on Image Processing, 21, 2012

13: Nour-Eddine Lasmar, Youssef Stitou, Yannick Berthoumieu, Multiscale skewed heavy tailed model for texture analysis, International Conference on Image Processing, 2281-2284, 2009

14: C. Sanderson and B.C. Lovell, Multi-Region Probabilistic Histograms for Robust and Scalable Identity Inference, Lecture Notes in Computer Science (LNCS), 5558, 2009

15: Y. Li, X. Yang, P. Sun, H. Qi and S. Lyu, Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 3204-3213, 2020

16: A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies and M. Niessner, FaceForensics++: Learning to Detect Manipulated Facial Images, IEEE/CVF International Conference on Computer Vision (ICCV), 2019

17: Dolhansky, Brian and Bitton, Joanna and Pflaum, Ben and Lu, Jikuo and Howes, Russ and Wang, Menglin and Ferrer, Cristian, The DeepFake Detection Challenge (DFDC) Dataset, arXiv:2006.07397, 2020

18: P. Viola, M. Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001

19: K. Zhang, Z. Zhang, Z. Li and Y. Qiao, Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks, IEEE Signal Processing Letters, 23, 1499-1503, 2016

20: Nicolò Bonettini, Edoardo Daniele Cannas, Sara Mandelli, Luca Bondi, Paolo Bestagini, Stefano Tubaro, Video Face Manipulation Detection ThroughEnsemble of CNNs, 2021

21: Amerini, Irene & Galteri, Leonardo & Caldelli, Roberto & Bimbo, Alberto, Deepfake Video Detection through Optical Flow Based CNN, 2019

Annexes

Annexe N°1

Contribution des signaux résiduels pour la détection de la permutation de visages dans les vidéos hypertruquées

P. Tessé C. Charrier E. Giguet

Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France

{paul.tesse, christophe.charrier, emmanuel.giguet}@unicaen.fr

Résumé

L'évolution fulgurante de l'apprentissage profond et plus particulièrement la découverte des réseaux antagonistes génératifs (RAG) a révolutionné le monde du Deepfake. Les falsifications sont de plus en plus réalistes et par conséquent de plus en plus difficiles à détecter. Attester si un contenu vidéo est authentique est de plus en plus sensible et le libre accès aux technologies de falsification rend la menace d'autant plus inquiétante. De nombreuses méthodes ont été proposées pour détecter ces faux et il est difficile de savoir quelles méthodes de détection sont encore d'actualité face aux progrès. Dans cet article, nous présentons notre approche pour la détection de permutation de visages dans les vidéos hypertruquées basée sur l'analyse des signaux résiduels.

Mots clefs

Vidéos hypertruquées, permutation de visages signaux résiduels, investigation numérique, apprentissage profond.

1 Introduction

Notre société hyperconnectée voit transiter des quantités de contenus multimédia de plus en plus importantes, que ce soit via la télévision, la vidéo surveillance, les réseaux sociaux et plus généralement internet. Ceci est dû aux progrès réalisés ces dernières années en matière de création et de partage de contenus vidéos. En couplant ces progrès avec les avancées réalisées dans le domaine de l'apprentissage machine, et plus particulièrement de l'apprentissage profond, nous assistons à une hausse très significative du nombre de faux contenus multimédia, en particulier les vidéos hypertruquées, aussi appelées *deepfakes*. De nouveaux outils de falsification très performants sont librement accessibles et de plus en plus simples d'utilisation. Certains de ces modèles sont d'ores et déjà intégrés à des réseaux sociaux tels que Snapchat et accessibles à tout utilisateur sous le nom de "filtres". Cette démocratisation des outils de falsification vidéo est à l'origine de la hausse significative du nombre de fake news, vidéos de propagande, tentative d'usurpation vidéo, etc. La détection de ces vidéos falsifiées représente par conséquent un enjeux sociétal majeur. En effet, il est de plus en plus difficile d'attester l'authenticité d'une vidéo, ce qui est très préoccupant dans notre société où chaque jour, les heures de visionnage uniquement sur Youtube se comptent en milliards.

La détection des vidéos hypertruquées est un sujet particulièrement ardent ces derniers temps bien que de nombreux chercheurs travaillent sur le sujet depuis des années. De nombreux articles traitent de ce sujet sous des angles variés. Parmi les approches les plus performantes, celles basées sur l'apprentissage profond sont majoritairement plébiscités, ce qui n'est pas sans poser de problèmes en terme d'explicabilité et du biais récurrent induit durant la phase d'apprentissage, voire du transfert d'apprentissage. Afin de pallier ces deux inconvénients, l'approche que nous avons retenue est fondée sur l'utilisation conjointe d'informations extraites des signaux résiduels et de réseaux de neurones.

La structure de l'article est la suivante. Une formalisation du problème est proposée dans la section 2. La section 3 dresse un panorama des méthodes de détection de permetutation de visages, basées notamment sur l'utilisation des modèles génératifs adverses et sur les techniques issues de la criminalistique des images. La section 4 décrit la méthode d'analyse que nous proposons. Les résultats sont présentés en section 5. La conclusion met en avant les perspectives de ce travail.

2 Formalisation du problème

La problématique étudiée étant la détection des deepfakes vidéos basés sur la permutation de visages, les éléments essentiels pris en considération dans la formalisation du problème sont les suivants :

- les vidéos sont de durée variable avec un trucage pouvant survenir à n'importe quel endroit ou moment ;
- un mécanisme de détection des visages est nécessaire puisqu'il permet de cibler la zone à étudier ;
- le modèle doit être le plus robuste et généralisable possible ;
- le modèle devant pouvoir être utilisé pour éclairer la Justice, une attention toute particulière doit être accordée à l'explicabilité des résultats ;
- le modèle doit fonctionner sans référence pour prononcer son diagnostic ;
- l'analyse d'images synthétiques n'est pas prise en compte dans ces travaux.

Ces aspects pris en compte, notre objectif est de développer un module prenant en entrée une vidéo et retournant en sortie un verdict concernant l'authenticité de cette dernière. Le problème est donc envisagé comme un problème de classification binaire où les classes sont "authentique" et "falsifiée".

3 Etat de l'art

De très nombreuses méthodes de détection de vidéos hypertruquées ont été proposées au cours des dernières années. Parmi les méthodes existantes, nous nous sommes tout d'abord intéressés aux méthodes d'apprentissage profond qui ont montré un niveau de performance élevé dans les tâches de classification au détriment de l'explicabilité du verdict [1]. C'est pourquoi nous avons laissé de côté ces modèles et avons concentré nos efforts sur les méthodes d'analyse des signaux résiduels, celles-ci étant totalement explicables. Voici les deux signaux résiduels que nous avons sélectionnés jusqu'à présent.

3.1 Evaluation de la qualité des images

Parmi les méthodes les plus répandues, on retrouve la mesure de la qualité des images (IQA–*Image Quality Assessment*). En effet, de nombreuses études ont montré que la qualité des images est altérée suite à la falsification [2]. Cette information est *de facto* pertinente et sera exploitée comme telle dans la tâche de classifation. Etant donné que nous ne disposons pas de l'image de référence, on s'attachera à utiliser une mesure de qualité des images *sans référence*. Parmi toutes les méthodes existantes, nous avons sélectionné l'indice de qualité BRISQUE (*Blind/Referenceless Image Spatial Quality Evaluator*) [3]. Ce dernier ne calcule pas les caractéristiques spécifiques aux distorsions, telles que l'effet le flou, de ringing ou de bloc, mais utilise les statistiques de scènes naturelles des coefficients de luminance normalisés localement pour quantifier les éventuelles pertes de « naturel » dans l'image dues à la présence de distorsions, ce qui aboutit à une mesure holistique de la qualité.

3.2 Analyse du spectre fréquentiel

Une autre approche consiste à étudier le spectre fréquentiel des images. Ce changement de représentation est motivé par un constat très intéressant présenté dans [4]. En effet, les auteurs ont mis en exergue un phénomène lié à l'utilisation des GANs dans les modèles générateurs de deepfake tel que le StyleGAN [5]. L'utilisation des opérations d'upsampling est nécessaire dans le processus de génération afin d'augmenter la dimensionnalité tout au long du processus. Cette opération utilise une opération d'interpolation qui est à l'origine d'une augmentation de l'utilisation des hautes fréquences dans la représentation de l'image. Cette introduction de hautes fréquences est alors un indice qui peut être exploité afin de déterminer si une vidéo est authentique ou non.

3.3 Autres signaux résiduels

D'autres signaux résiduels sont également exploitables. Nous avons pour l'instant concentré nos efforts sur les deux premiers mais l'on peut en citer de nombreux autres tels que l'analyse de la Lateral Chromatic Aberration [6] ou encore des Color Filter Array [7] Artefacts, qui se concentrent sur l'analyse des d'artéfacts induits par les différences dans les systèmes d'acquisition des sources des images mélangées pour générer le deepfake.

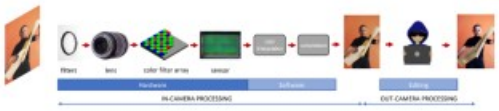


FIGURE 1 – *Système d'acquisition image numérique [8]*

4 Architecture proposée

Afin de combiner la puissance des modèles d'apprentissage profond avec l'explicabilité des méthodes basées sur l'analyse des signaux résiduels que nous avons présentés précédemment, nous proposons l'architecture suivante. L'architecture proposée, telle qu'illustrée dans la figure 3, se décompose en quatre étapes :

1. La vidéo est prétraitée pour obtenir les frames (F) et ne conserver que le visage qui est la zone de l'attaque pour plus de précision et une optimisation en terme de coûts.
2. Les images sont ensuite passées à différents extracteurs de caractéristiques (FE) qui vont extraire des caractéristiques pertinentes telles que le score de qualité via la mesure BRISQUE, la représentation fréquentielle de l'image ou le ratio des hautes fréquences.
3. Ces différentes caractéristiques sont ensuite concaténées en une seule représentation pour le Classifieur afin d'augmenter la robustesse de ce dernier.
4. Le Classifieur qui sera à terme un modèle d'apprentissage profond à définir quant à lui procède à la classification binaire entre les classes *Authentique* et *Falsifiée*.

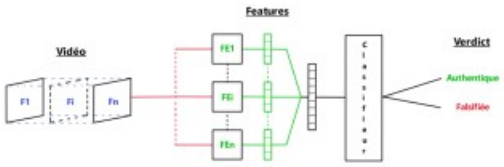


FIGURE 2 – *Architecture proposée*

L'intérêt de cette architecture est selon nous de proposer une alternative en boîte grise. En effet, les extracteurs de caractéristiques sont des boîtes blanches puisqu'ils n'utilisent pas d'apprentissage profond et seul le classifieur sera une boîte noire. De cette manière nous pensons pouvoir conserver un bon équilibre entre performance et explicabilité. Enfin, cette architecture est évolutive puisque la modularité permet d'ajouter simplement de nouveaux extracteurs de caractéristiques et seul le classifieur sera à entraîner, ce qui assure une meilleure durabilité du modèle dans le temps.

5 Expérimentations et résultats

Afin d'étudier ces signaux résiduels et leur pertinence plus en détail, nous avons réalisé plusieurs expérimentations. A l'heure actuelle nous n'avons pu nous intéresser qu'à BRISQUE ainsi qu'à la représentation fréquentielle. Ces expérimentations ont été réalisées sur les vidéos issues des base de données VidTIMIT [9] et DeepfakeTIMIT [10] qui contiennent respectivement les échantillons authentiques et falsifiés. Ces vidéos de haute qualité ont été traitées de sorte à ne conserver que les visages dans les images d'origines. Il est important de préciser que nous utilisons un SVM en guise de classifieur dans cette étude préliminaire au vu du peu de données que nous avons.

5.1 Indice de qualité

En reprenant l'implémentation fournie par les auteurs sur Github [3], nous avons été en mesure de calculer le score de qualité pour une image. Notre modèle recevant une vidéo en input, nous nous sommes intéressés au calcul de ce score à l'échelle de la vidéo. C'est pourquoi nous avons calculé la moyenne et l'écart-type de ce score à partir du score de chaque frame. Voici un échantillon des résultats obtenus en appliquant notre extracteur de caractéristiques sur les deux bases de données pour les vidéos authentiques (Tableau 1) et les vidéos truquées (Tableau 2).

VideoId	faks0-sa1	fcft0-sa2	mdab0-sx49	BDD
mean±std	21.4 ± 1.81	31.9 ± 1.68	26.6 ± 1.80	24.28 ± 2.88

TABLEAU 1 – *BRISQUE Scores vidéos Authentiques*

VideoId	faks0-sa1	fcft0-sa2	mdab0-sx49	BDD
mean±std	31.5 ± 2.15	42.5 ± 1.40	38 ± 1.45	32.9 ± 1.93

TABLEAU 2 – *BRISQUE Scores vidéos Hypertruquées*

On peut constater, et ce à l'échelle de l'ensemble des paires de vidéos authentiques/falsifiées, que la qualité moyenne semble se dégrader systématiquement et ce de manière significative. Nous rappelons que le score varie entre 0 et 100 avec 0 qui correspond à la qualité optimale. Pour ce qui est de l'écart-type, la variation est moins significative mais celle-ci a tendance à diminuer contrairement à la moyenne. Cette tendance dans les résultats semble conservée à l'échelle des bases de données au vu de la colonne BDD. Cela tend à confirmer que ces scores pourraient bien servir de caractéristiques pour notre classifieur.

5.2 Hautes fréquences

De la même manière que pour les tests sur BRISQUE, nous avons repris l'implémentation des auteurs [4] et l'avons reprise afin de permettre de générer la représentation fréquentielle d'une vidéo. Encore une fois, nous avons généré les résultats sous la forme de paires dont un échantillon est présenté sur la figure 3.

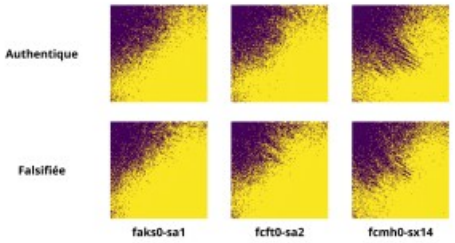


FIGURE 3 – *Visualisation spectre fréquentiel vidéos*

On peut observer une hausse des hautes fréquences que nous avons essayé de quantifier plus finement avec la différence entre les ratios des vidéos authentiques et falsifiées, correspondant au rapport entre le nombre de valeurs de pixels supérieures à 150 et le nombre de pixels total.

VideoId	faks0-sa1	fcft0-sa2	mdab0-sx49	BDD
ΔHF	+2.1%	+2.6%	+1.8%	+0.5%

TABLEAU 3 – *Variation des hautes fréquences*

Les résultats du tableau 3 confirment bien notre analyse qualitative des spectres. Il y a une augmentation légère mais qui peut rester perceptible pour notre classifieur qui est persistante d'après les résultats à l'échelle de la base de données (BDD) bien que l'augmentation soit plus faible. Ceci peut s'expliquer par le fait que nous calculons une première moyenne entre les frames, puis entre toutes les vidéos ce qui produit un effet de lissage. Néanmoins, on constate qu'il reste une variation qui pourrait être exploitable par notre classifieur. Dans notre cas, nous avons fait le choix d'utiliser ce ratio en guise de caractéristique étant donné qu'il s'agit d'un score normalisé représenté par un simple scalaire.

5.3 Classification par SVM

Afin de statuer sur la pertinence des caractéristiques présentées, nous avons testé la détection des deepfakes en utilisant le classifieur SVC de Scikit Learn [3] avec les réglages par défaut. Pour cela nous avons extrait les différentes caractéristiques des vidéos issues des bases de données Vid-

TIMIT et DeepfakeTIMIT. Les caractéristiques ainsi obtenues ont été divisées en un jeu d'entraînement (Train) et un jeu de validation. Ce même processus a été appliqué à un échantillon de la base de données Celeb-DeepFake [11] afin de générer des données de test (Test) pour avoir un aperçu des performances en généralisation. Les résultats obtenus avec les différents ensembles sont présentés dans le tableau 4. Les résultats présentés ont été obtenus en appliquant un Bootstrap à 999 réplications. La composition des ensembles utilisés est présentée dans le tableau 5.

SVM	BRISQUE	Somme HFs	Concaténés
Train	88% ± 0.008	60% ± 0.006	86% ± 0.006
Val	87% ± 0.02	59% ± 0.02	85% ± 0.03
Test	48% ± 0.01	45% ± 0.03	46% ± 0.01

TABLEAU 4 – *Précision Classification SVM*

Ensembles	Train	Validation	Test
Source(s)	VidTIMIT DeepfakeTIMIT	VidTIMIT DeepfakeTIMIT	CelebDeepFake
Taille	580	286	103
Repartition	A=436/F=256	A=110/F=64	A=51/F=52

TABLEAU 5 – *Composition des ensembles où A correspond au nombre de vidéos non falsifées et F au nombre de vidéos hypertruquées*

Nos résultats sont au dessus des 50% ce qui signifie que nos prédictions sont plus fiables que le hasard bien que l'on observe une baisse systématique et significative des performances en généralisation. Cette baisse de performance peut être due à plusieurs facteurs tels que la quantité de données qui reste assez faible, le fait que les données d'entraînement ne soient issues que d'un seul jeu de données, ou encore tout simplement le modèle en lui-même qui reste trop simple. Les résultats relatifs à l'utilisation du ratio des hautes fréquences nous laisse penser qu'il est nécessaire d'utiliser un CNN afin d'exploiter au maximum les informations contenues dans le spectre et non pas un simple ratio. De plus, la combinaison des deux semble bien indiquer que l'utilisation du ratio des hautes fréquences n'améliore pas les performances obtenues avec BRISQUE.

6 Conclusion

Nous avons présenté dans cet article nos travaux préliminaires relatifs à la détection de vidéos hypertruquées, aussi appelées *Deepfakes*. Nous avons pu démontrer que les signaux résiduels constituent bel et bien une piste sérieuse de caractéristiques pertinentes et explicables. Il est en effet possible pour un classifieur, comme le montre les résultats obtenus, d'exploiter ces signaux afin de résoudre notre problème de détection. Nous n'avons pour le moment pu tester que deux extracteurs de caractéristiques avec un simple SVM en guise de classifieur. C'est pourquoi il nous faut procéder à davantage de tests sur ces derniers afin de confirmer ces premiers résultats expérimentaux. Dans un second temps nous incorporons d'autres signaux résiduels tout en améliorant le classifieur afin d'améliorer les performances de notre architecture. Enfin, un travail de passage à l'échelle reste à effectuer afin d'obtenir le plus de précision et de recul possible quant à l'évaluation de ces performances.

Références

[1] David Güera et Edward J Delp. Deepfake video detection using recurrent neural networks. Dans *2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pages 1–6. IEEE, 2018.

[2] Javier Galbally et Sébastien Marcel. Face anti-spoofing based on general image quality assessment. *Proceedings - International Conference on Pattern Recognition*, pages 1173–1178, 08 2014.

[3] Anish Mittal, Anush Krishna Moorthy, et Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12) :4695–4708, 2012.

[4] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, et Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. Dans *International conference on machine learning*, pages 3247–3258. PMLR, 2020.

[5] Tero Karras, Samuli Laine, et Timo Aila. A style-based generator architecture for generative adversarial networks. Dans *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 4396–4405, 2019.

[6] Owen Mayer et Matthew C. Stamm. Accurate and efficient image forgery detection using lateral chromatic aberration. *IEEE Transactions on Information Forensics and Security*, 13(7) :1762–1777, 2018.

[7] Pasquale Ferrara, Tiziano Bianchi, Alessia De Rosa, et Alessandro Piva. Image forgery localization via fine-grained analysis of cfa artifacts. *IEEE Transactions on Information Forensics and Security*, 7(5) :1566–1577, 2012.

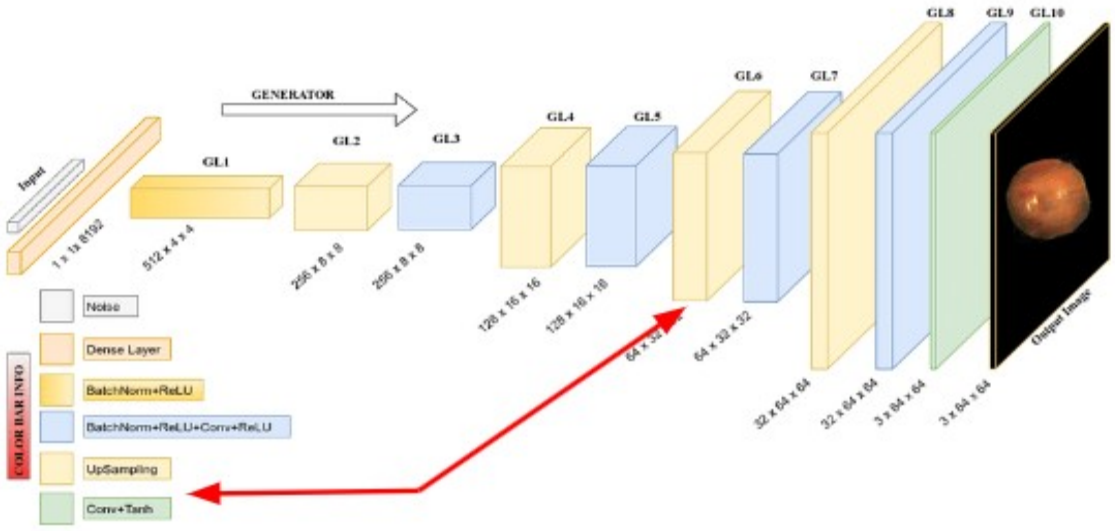
[8] Luisa Verdoliva. Media forensics and deepfakes : An overview. *IEEE Journal of Selected Topics in Signal Processing*, 14 :910–932, 2020.

[9] C. Sanderson et B.C. Lovell. Multi-region probabilistic histograms for robust and scalable identity inference. *Lecture Notes in Computer Science (LNCS)*, 5558 :199–208, 2009.

[10] Pavel Korshunov et Sébastien Marcel. Deepfakes : a new threat to face recognition ? assessment and detection. *ArXiv*, abs/1812.08685, 2018.

[11] Pu Sun Honggang Qi Yuezun Li, Xin Yang et Siwei Lyu. Celeb-df : A large-scale challenging dataset for deepfake forensics. Dans *IEEE Conference on Computer Vision and Patten Recognition (CVPR)*, 2020.

Annexe N°2



Architecture GAN

Annexe N°3

Exemples Celeb-DF

FaceForensics
FaceShifter

FaceForensics
NeuralTextures

FaceForensics
Face2Face

FaceForensics
Faceswap

FaceForensics
Deepfakes

FaceForensics
DeepfakeDetection

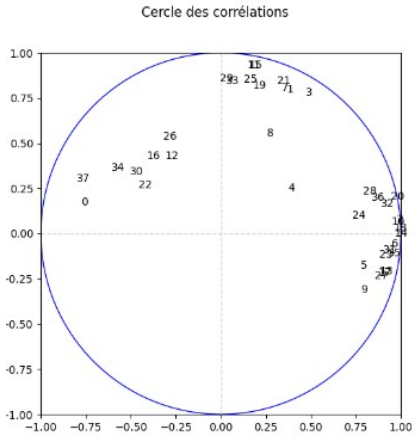
Annexe N°4

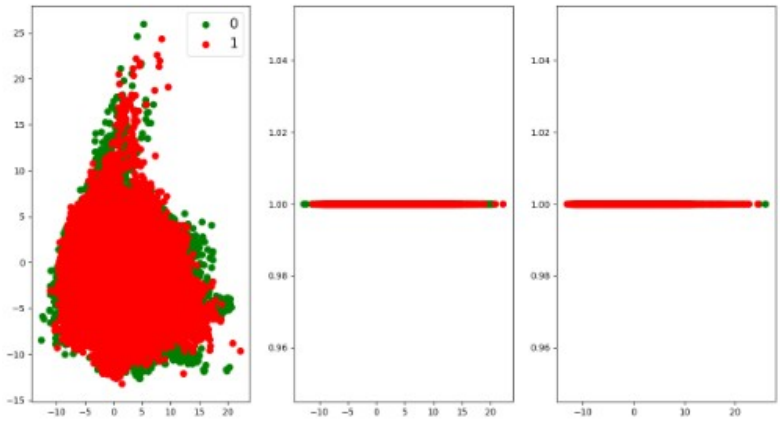
	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lda	Linear Discriminant Analysis	0.8606	0.8851	0.9396	0.8440	0.8887	0.7036	0.7127	0.033
ridge	Ridge Classifier	0.8597	0.0000	0.9396	0.8428	0.8881	0.7017	0.7109	0.032
lr	Logistic Regression	0.8571	0.8782	0.9279	0.8464	0.8847	0.6978	0.7048	1.083
qda	Quadratic Discriminant Analysis	0.8371	0.8454	0.9087	0.8327	0.8685	0.6556	0.6617	0.035
et	Extra Trees Classifier	0.8301	0.8619	0.9175	0.8187	0.8647	0.6383	0.6477	0.153
rf	Random Forest Classifier	0.8283	0.8724	0.9131	0.8193	0.8628	0.6352	0.6443	0.112
knn	K Neighbors Classifier	0.8205	0.8492	0.9072	0.8125	0.8567	0.6185	0.6270	0.078
gbc	Gradient Boosting Classifier	0.8161	0.8821	0.8822	0.8213	0.8500	0.6132	0.6175	0.044
lightgbm	Light Gradient Boosting Machine	0.8144	0.8796	0.8748	0.8244	0.8477	0.6105	0.6152	0.095
svm	SVM - Linear Kernel	0.8083	0.0000	0.8484	0.8330	0.8382	0.6024	0.6074	0.034
ada	Ada Boost Classifier	0.8031	0.8581	0.8645	0.8150	0.8385	0.5868	0.5897	0.045
dt	Decision Tree Classifier	0.7604	0.7521	0.7983	0.7978	0.7972	0.5044	0.5061	0.033
nb	Naive Bayes	0.7300	0.7802	0.7482	0.7868	0.7662	0.4470	0.4491	0.043
dummy	Dummy Classifier	0.5915	0.5000	1.0000	0.5915	0.7433	0.0000	0.0000	0.028

Résultats benchmark Pycaret

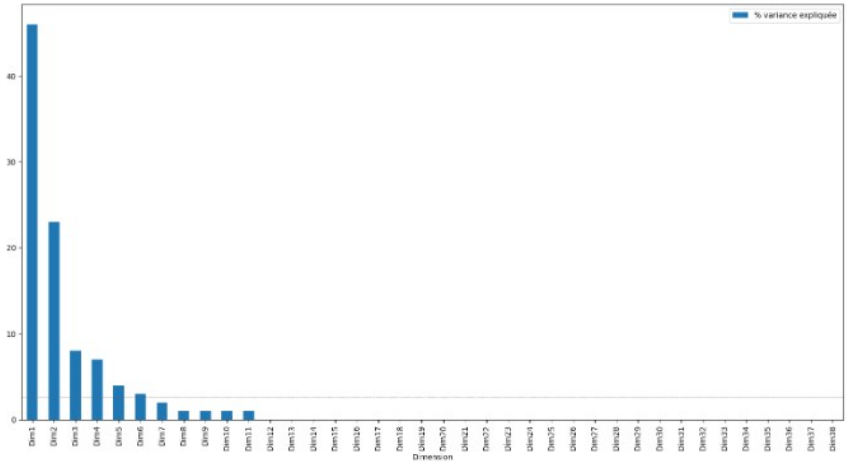
Annexe N°5

Échelle frames

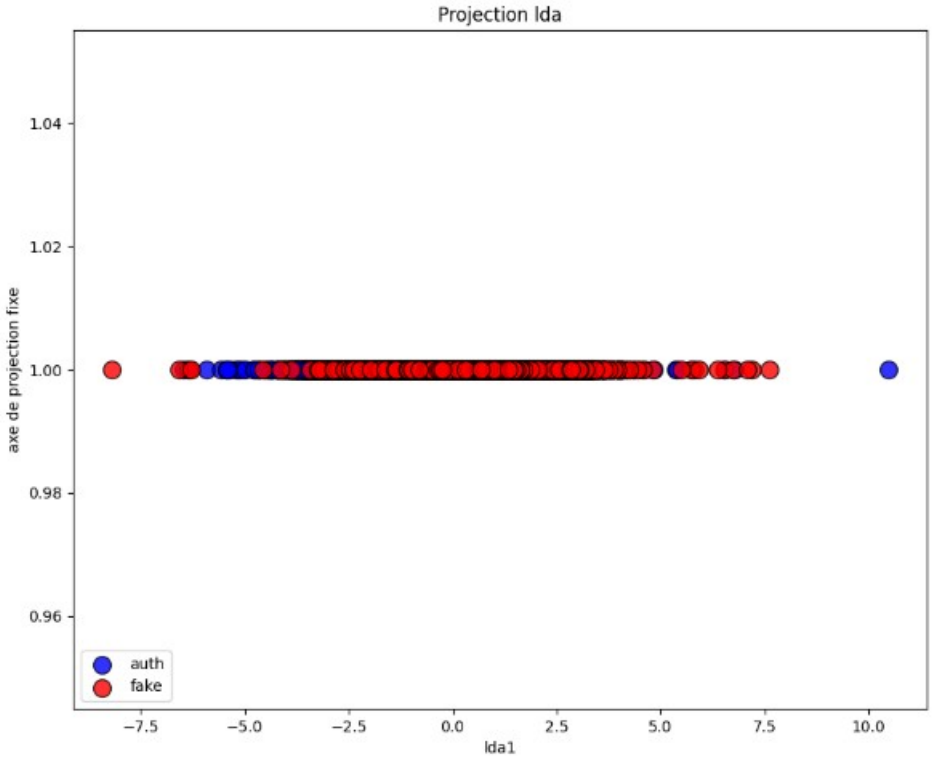




Analyse PCA échelle frames - Projections 1D

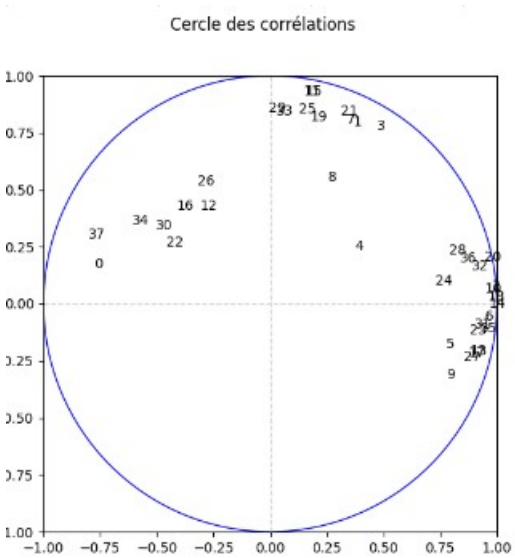


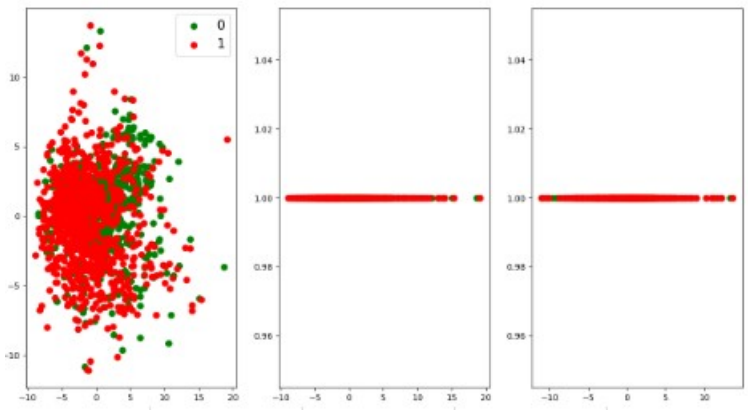
Mesure de la variance expliquée selon les axes



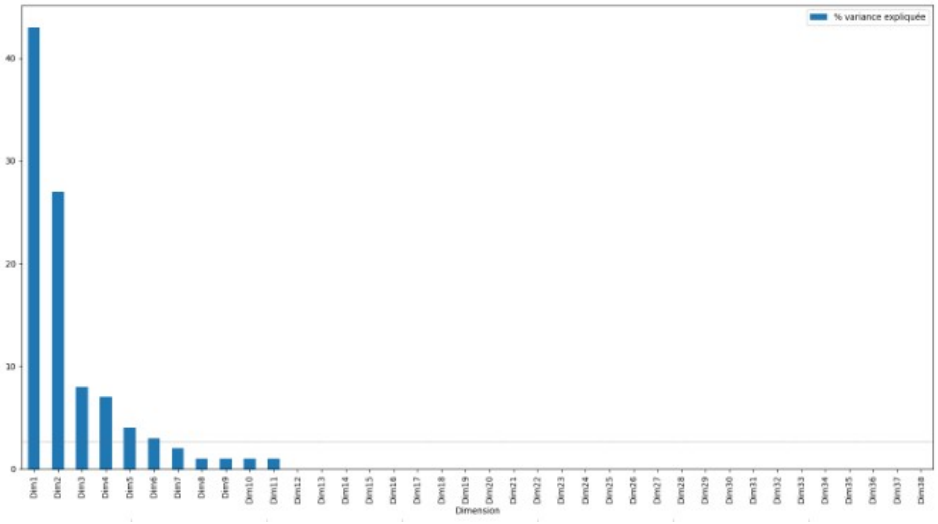
Projection 1D LDA

Échelle vidéos

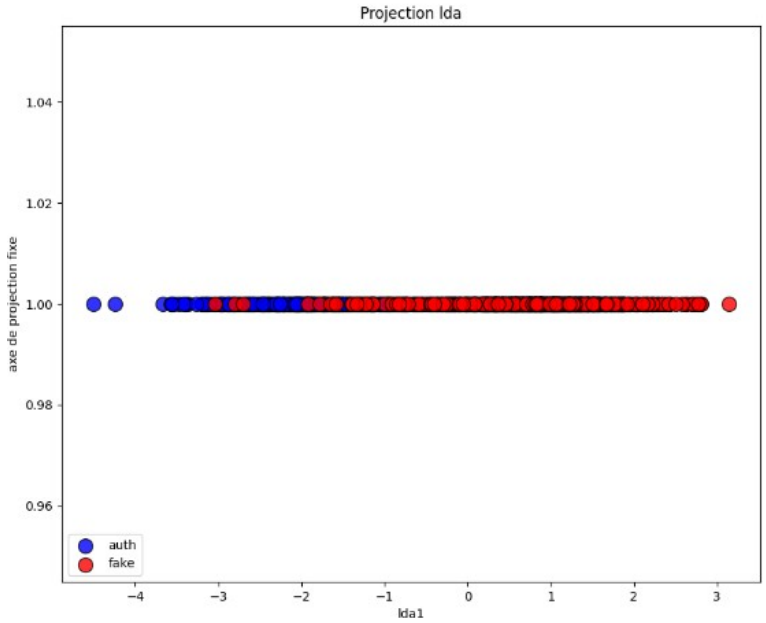




Analyse PCA échelle vidéo - Projections 1D



Mesure de la variance expliquée selon les axes



Projection 1D LDA

Annexe N°6

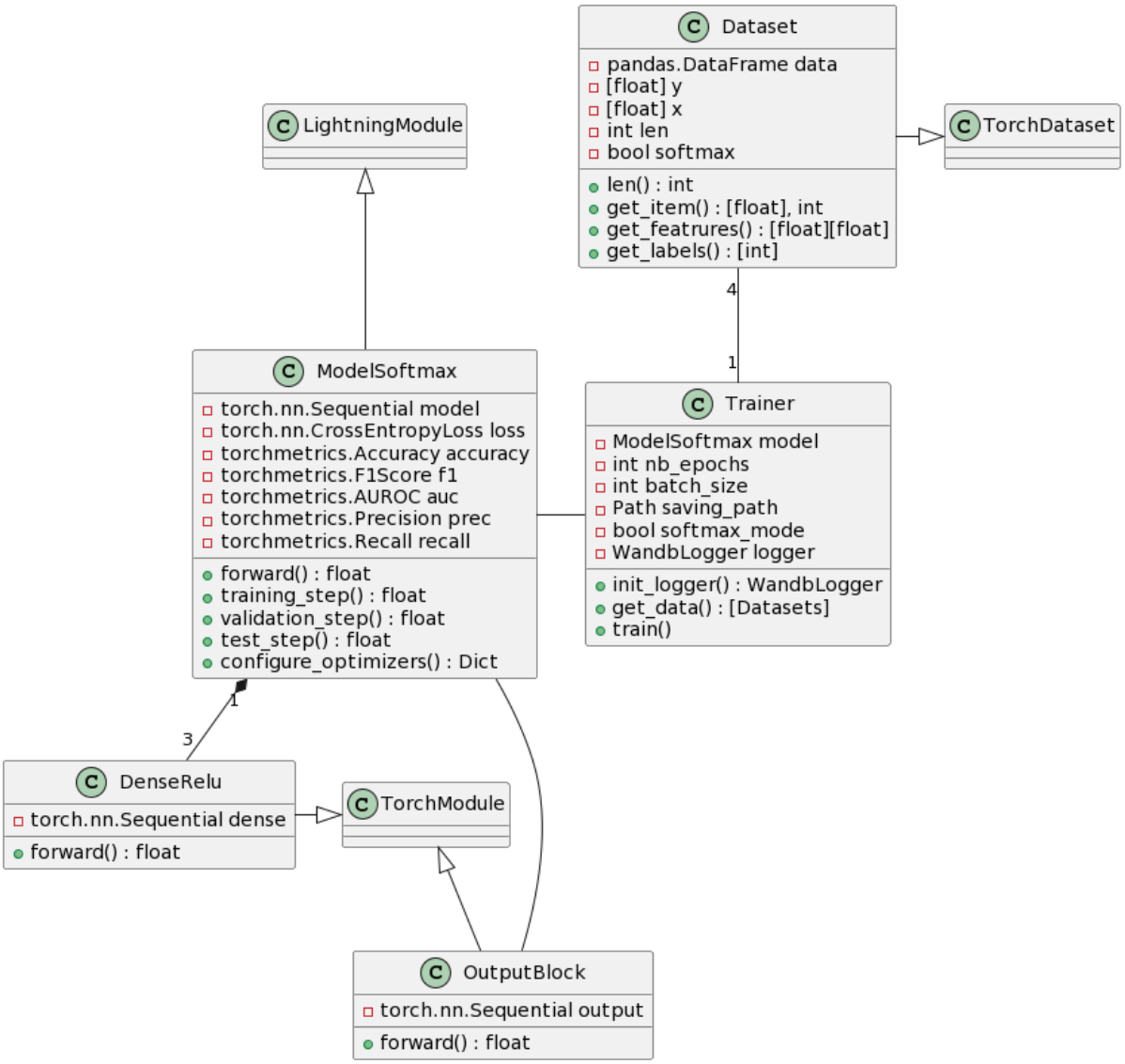


Diagramme de classes pipeline d'entraînement

Annexe N°7

- Performances échelle **frames** avec ajout de la **batch normalisation** au modèle deep basique :

Set/Métriques	F1	Accuracy	AUC	Precision	Recall
Train	96,00 %	96,00 %	99,00 %	97,00 %	95,00 %
Validation	84,00 %	82,00 %	87,00 %	83,00 %	85,00 %
Test	84,00 %	83,00 %	88,00 %	83,00 %	84,00 %
Généralisation	58,00 %	47,00 %	51,00 %	48,00 %	73,00 %

- Performances échelle **frames** avec le modèle deep basique avec **batch normalisation** et **dropout** à 33 % :

Set/Métriques	F1	Accuracy	AUC	Precision	Recall
Train	89,00 %	88,00 %	95,00 %	92,00 %	86,00 %
Validation	82,00 %	80,00 %	86,00 %	81,00 %	83,00 %
Test	81,00 %	81,00 %	89,00 %	83,00 %	78,00 %
Généralisation	67,00 %	53,00 %	59,00 %	52,00 %	92,00 %

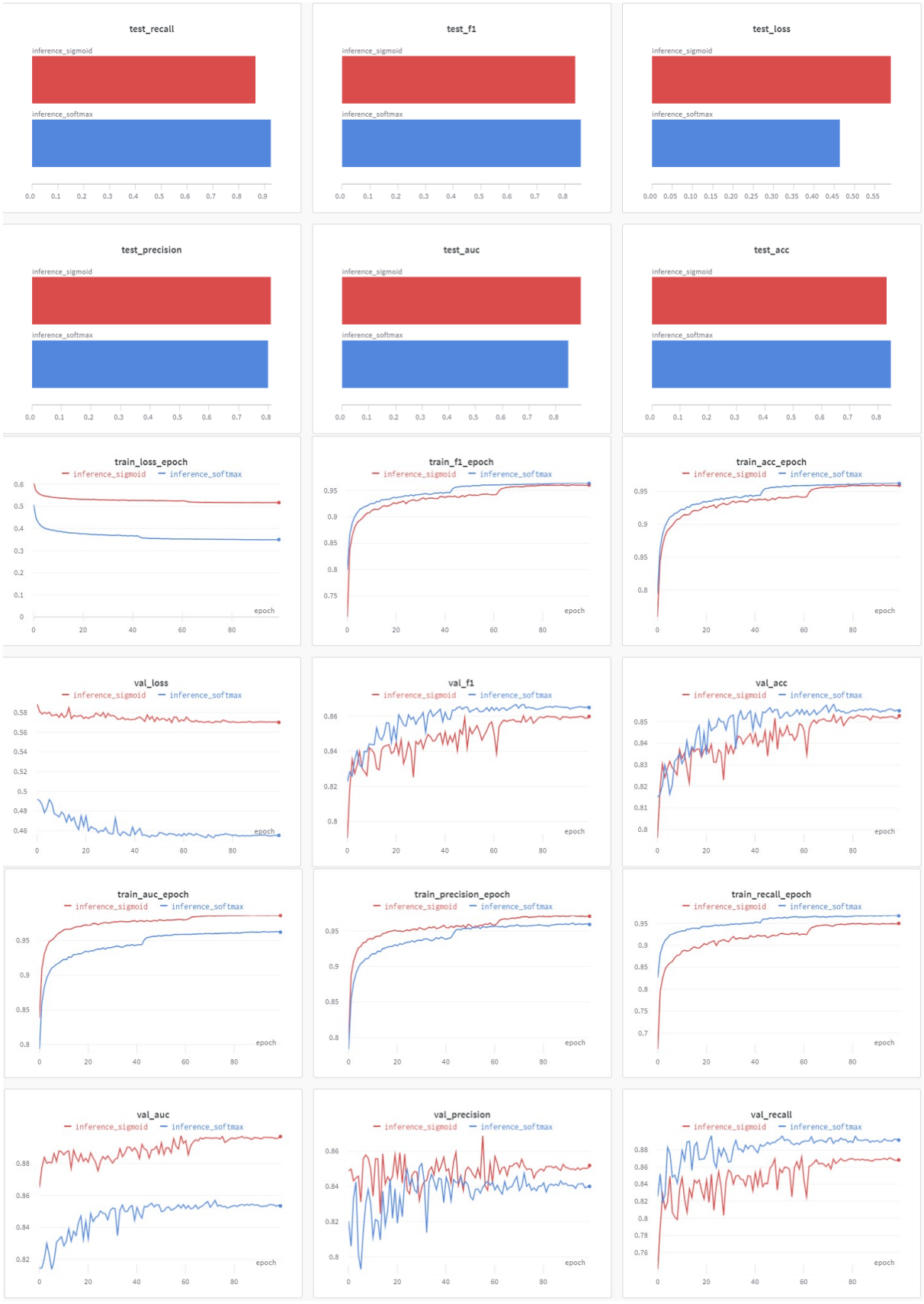
- Performances échelle **frames** avec le modèle deep basique avec **batch normalisation**, **dropout** à 33 % et **augmentation** du **nombre** de **neurones** :

Set/Métriques	F1	Accuracy	AUC	Precision	Recall
Train	96,00 %	96,00 %	99,00 %	97,00 %	95,00 %
Validation	84,00 %	82,00 %	88,00 %	83,00 %	86,00 %
Test	85,00 %	84,00 %	89,00 %	83,00 %	87,00 %
Généralisation	60,00 %	49,00 %	52,00 %	50,00 %	75,00 %

Annexe N°8

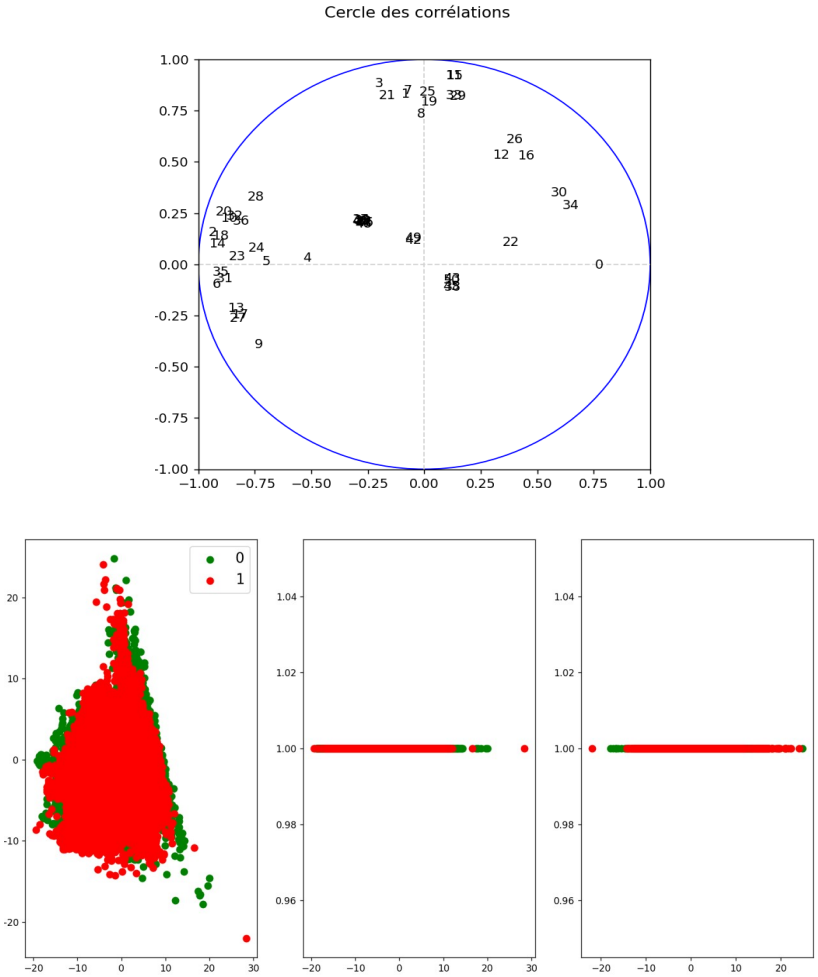

test_recall
Softmax_best_model
BCE_best_model
test_f1
test_loss
test_precision
test_auc
test_acc
train_loss_epoch
Softmax_best_model
BCE_best_model
epoch
train_f1_epoch
train_acc_epoch
val_loss
val_f1
val_acc
train_auc_epoch
train_precision_epoch
train_recall_epoch
val_auc
val_precision
val_recall

Annexe N°9


test_recall
inference_sigmoid
inference_softmax
test_f1
test_loss
test_precision
test_auc
test_acc
train_loss_epoch
train_f1_epoch
train_acc_epoch
val_loss
val_f1
val_acc
train_auc_epoch
train_precision_epoch
train_recall_epoch
val_auc
val_precision
val_recall
epoch

Annexe N°10

Échelle frames



Analyse PCA - Projection 1D dims 1-2/1-38

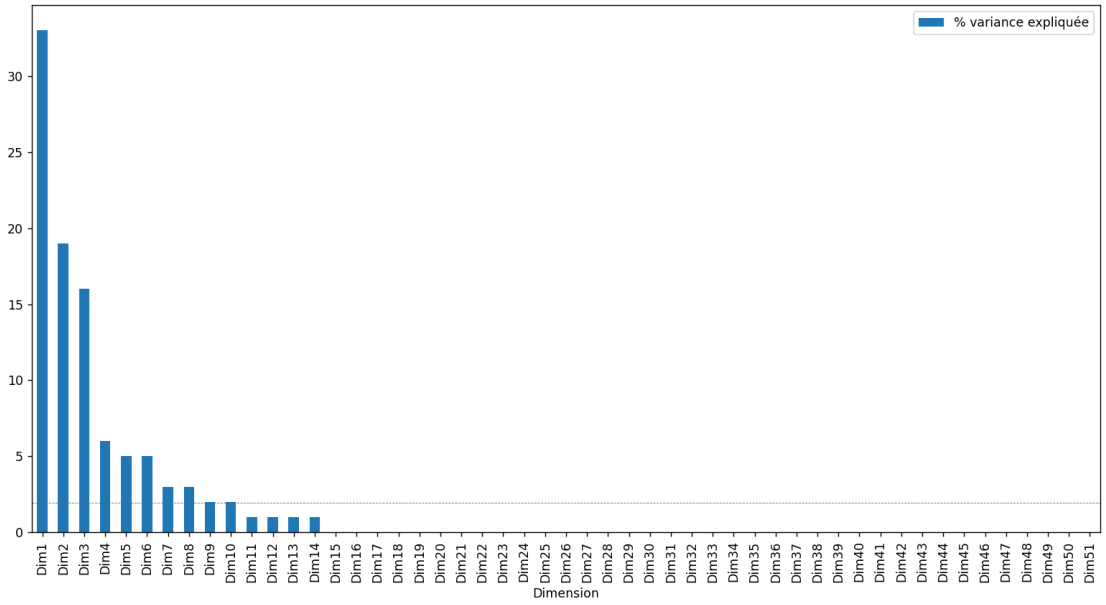
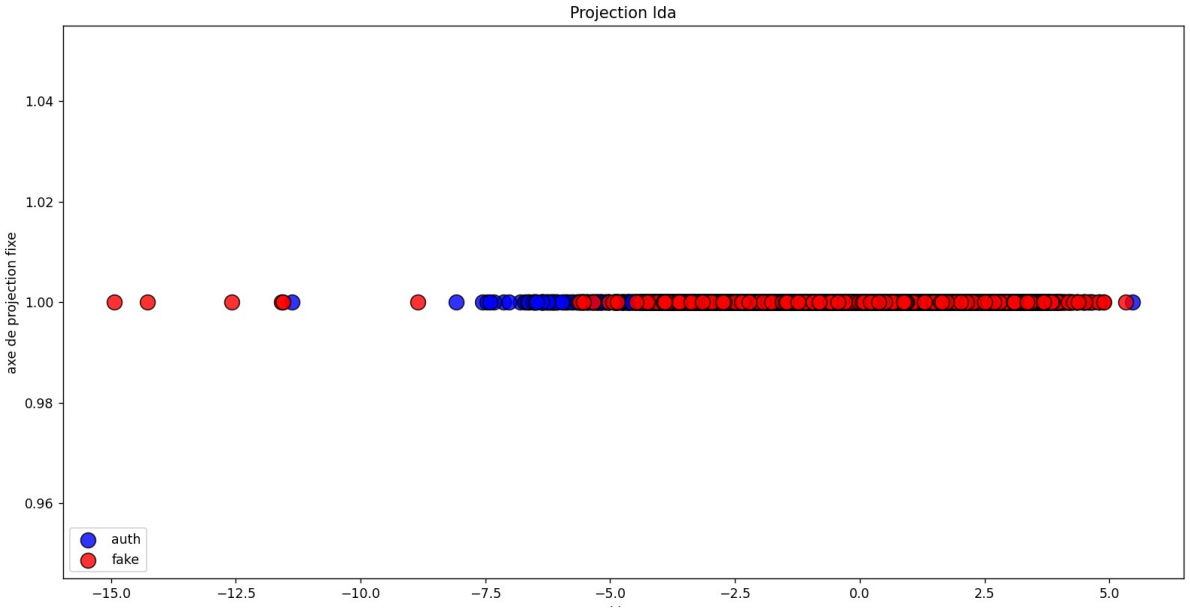


Figure 63: Mesure de la variance expliquée selon les axes



Projection 1D LDA

Annexe N°11

- Performances échelle **vidéos** avec le modèle basique deep en utilisant la **moyenne** comme opération pour synthétiser l'information contenues dans chaque frames de la vidéo :

Set/Métriques	F1	Accuracy	AUC	Precision	Recall
Train	81,00 %	78,00 %	78,00 %	81,00 %	84,00 %
Validation	83,00 %	80,00 %	78,00 %	82,00 %	85,00 %
Test	84,00 %	80,00 %	77,00 %	80,00 %	89,00 %
Généralisation	68,00 %	53,00 %	52,00 %	52,00 %	100,00 %

- Performances échelle **vidéos** avec le modèle basique deep en utilisant les **quartiles** comme opération pour synthétiser l'information contenues dans chaque frames de la vidéo :

Set/Métriques	F1	Accuracy	AUC	Precision	Recall
Train	85,00 %	83,00 %	82,00 %	83,00 %	91,00 %
Validation	86,00 %	82,00 %	81,00 %	82,00 %	90,00 %
Test	86,00 %	82,00 %	80,00 %	81,00 %	91,00 %
Généralisation	67,00 %	52,00 %	51,00 %	51,00 %	98,00 %

- Performances échelle **vidéos** avec le modèle basique deep en utilisant les **quartiles** comme opération pour synthétiser l'information contenues dans chaque frames de la vidéo et **1 frame toutes les 10 frames** dans le processus d'analyse :

Set/Métriques	F1	Accuracy	AUC	Precision	Recall
Train	84,00 %	83,00 %	83,00 %	83,00 %	88,00 %
Validation	86,00 %	84,00 %	83,00 %	83,00 %	89,00 %
Test	87,00 %	86,00 %	85,00 %	86,00 %	89,00 %
Généralisation	68,00 %	53,00 %	52,00 %	52,00 %	98,00 %

Annexe N°12

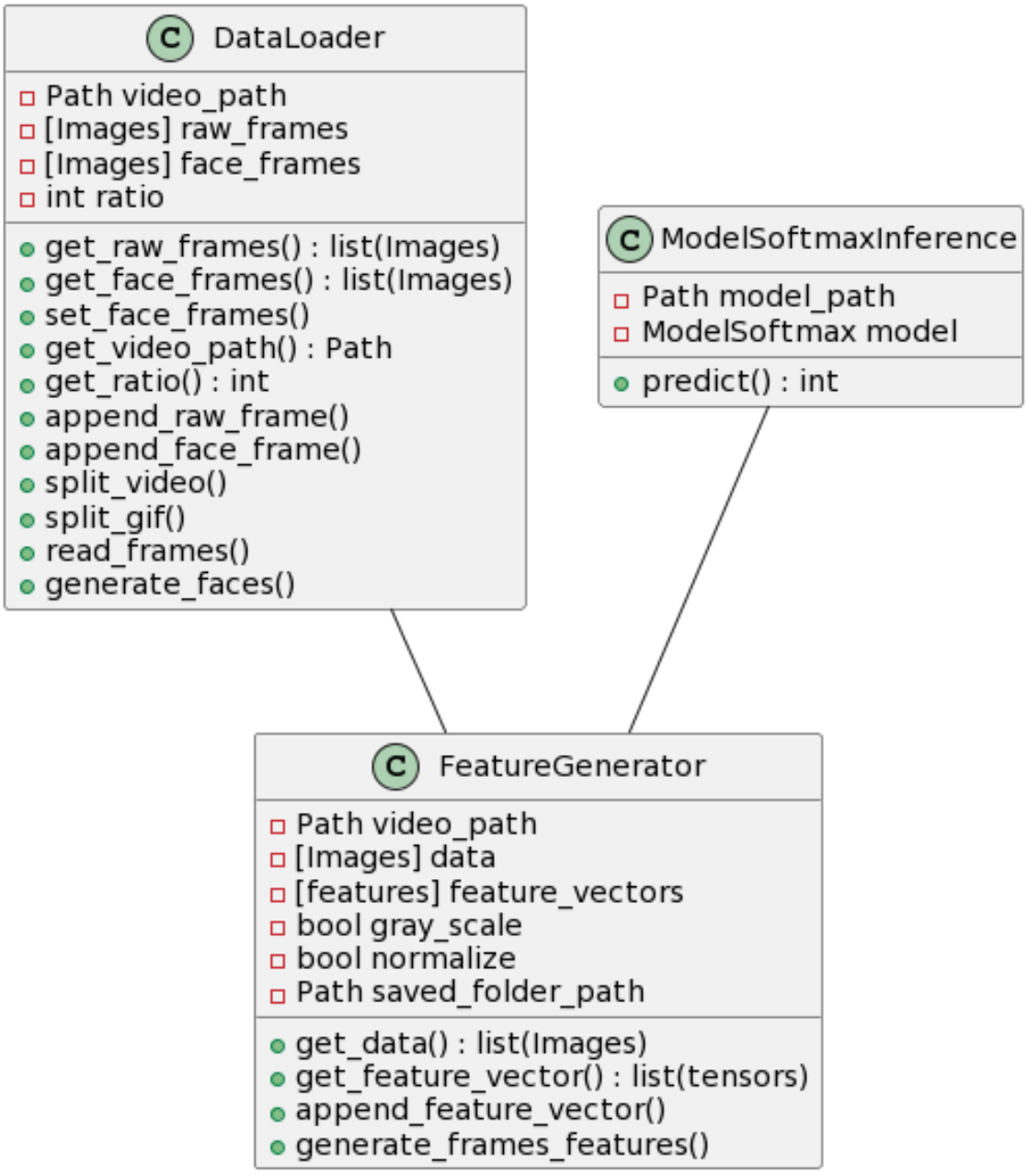


Diagramme de classes pipeline d'inférence

Résumé

Ce rapport de stage propose une synthèse du travail effectué dans le cadre de mon stage ingénieur orienté recherche au sein du GREYC du 1er mars au 31 juillet 2023 et rend compte de l'essentiel de mes travaux de recherche au sein de l'équipe Sécurité Architecture Forensique et biomEtrie dans le domaine de la détection des deepfakes vidéos. Ces travaux portent plus précisément sur l'utilisation des signaux résiduels contenus dans les vidéos pour la détection des vidéos hypertruquées par face swapping. De fait, il existe de nombreux travaux traitant de la détection de ces deepfakes mais les approches à l'état de l'art, bien qu'efficaces, sont loin d'être explicables ce qui pause problème dans un cadre de réglementation. C'est dans ce contexte qu'a vu le jour ce projet orienté forensique ayant pour objectif final de créer une architecture basée sur de l'apprentissage profond, ayant le juste équilibre entre performances et explicabilité du verdict, capable de détecter les vidéos hypertruquées en se basant sur l'analyse de caractéristiques issues des signaux résiduels. Vous trouverez dans ce rapport une synthèse des mes principales réflexions et expérimentations, ainsi que les principaux choix et résultats obtenus pour mener à bien ce projet qui m'aura beaucoup appris, tant sur le plan de l'ingénierie pour la réalisation et les choix techniques, que de la recherche par l'étude de l'existant, la conception et l'innovation.

This internship report provides a summary of the work carried out as part of my research-oriented engineering internship at GREYC from 1 March to 31 July 2023 and describes the main aspects of my research work in the Forensic Architecture and Biometrics Security team in the field of video deepfake detection. More specifically, this work focuses on the use of residual signals contained in videos to detect videos that have been hypertrupled by face swapping. In fact, there is a great deal of work dealing with the detection of these deepfakes, but the state-of-the-art approaches, although effective, are far from being explainable, which poses a problem in a regulatory context. It was against this backdrop that this forensics-oriented project was born, with the ultimate aim of creating a deep learning-based architecture that strikes the right balance between performance and the explicability of the verdict, and that can detect hypertrusted videos based on the analysis of characteristics derived from residual signals. In this report, you will find a summary of my main reflexions and experiments, as well as the main choices and results obtained in the course of this project, which has taught me a great deal, both in terms of engineering for the implementation and technical choices, and in terms of research for the study of the existing system, the design and the innovation part.