République Algérienne Démocratique et Populaire
الجمهورية الجزائرية الديمقراطية الشعبية
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
وزارة التعليم العالي و البحث العلمي
École Nationale Polytechnique

Département électronique

Mémoire de projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état en électronique

Évaluation des techniques de génération de *deepfake*

Achref DJABER & Abdellah TAIBAOUI

Sous la direction de M. Sid-Ahmed BERRANI ENSIA, Sidi Abdellah

Présenté et soutenu publiquement le 01/07/2024 auprès des membres du jury :

Président	M. Mourad	ADNANE	ENP, Alger
Promoteur	M. Sid-Ahmed	BERRANI	ENSIA, Sidi Abdellah
Examinatrice	Mme. Nesrine	BOUADJENEK	ENP, Alger

ENP 2024

10, avenue des frères Oudek, Hassen Badi, BP. 182, 16200 El Harrach, Alger, Algérie.
www.enp.edu.dz

République Algérienne Démocratique et Populaire
الجمهورية الجزائرية الديمقراطية الشعبية
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
وزارة التعليم العالي و البحث العلمي
École Nationale Polytechnique

Département électronique

Mémoire de projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état en électronique

Évaluation des techniques de génération de *deepfake*

Achref DJABER & Abdellah TAIBAOUI

Sous la direction de M. Sid-Ahmed BERRANI ENSIA, Sidi Abdellah

Présenté et soutenu publiquement le 01/07/2024 auprès des membres du jury :

Président	M. Mourad	ADNANE	ENP, Alger
Promoteur	M. Sid-Ahmed	BERRANI	ENSIA, Sidi Abdellah
Examinatrice	Mme. Nesrine	BOUADJENEK	ENP, Alger

ENP 2024

10, avenue des frères Oudek, Hassen Badi, BP. 182, 16200 El Harrach, Alger, Algérie.
www.enp.edu.dz

ملخص

التقنيات المستخدمة في إنشاء التزييف العميق (ديب فيك) تُستخدم بشكل واسع في مجالات متعددة. ومع ذلك، بسبب تعقيدها والتحديات الأخلاقية والاجتماعية التي تطرحها، فإن تقييم تقنيات إنشاء التزييف العميق يبقى مهمة صعبة. تقترح بعض الدراسات هياكل وأدوات متقدمة لتجاوز هذه العقبات، ولكنها تتطلب موارد مادية كبيرة ووقت معالجة طويل. تهدف هذه الدراسة إلى اقتراح منهجية تُمكِّن من تقييم تقنيات إنشاء التزييف العميق بدقة، مع احترام قيود الوقت والموارد المحدودة.

كلمات مفتاحية : التزييف العميق، تقييم التقنيات، معالجة البيانات.

Abstract

Deepfake generation techniques are widely used in various fields. However, due to their complexity and the ethical and social challenges they pose, assessing deepfake generation techniques remains a demanding task. Some studies propose sophisticated architectures and tools to overcome these obstacles, but these require significant hardware resources and extensive processing time. This study aims to propose an approach to accurately evaluate deepfake generation techniques while respecting the constraints of limited processing time and resources.

Keywords : Deepfake, Technique assessment, Data processing.

Résumé

Les techniques de génération de *deepfakes* sont largement utilisées dans divers domaines. Cependant, en raison de leur complexité et des défis éthiques et sociaux qu'elles posent, l'évaluation des techniques de génération de *deepfakes* reste une tâche exigeante. Certaines études proposent des architectures et des outils sophistiqués pour surmonter ces obstacles, mais ceux-ci nécessitent des ressources matérielles importantes et un temps de traitement considérable. Cette étude vise à proposer une approche permettant d'évaluer avec précision les techniques de génération de *deepfakes*, tout en respectant les contraintes de temps de traitement et de ressources limitées.

Mots clés : *Deepfake*, Évaluation des techniques, Traitement des données.

Dédicace

"Je dédie ce travail à ma famille, qui m'a soutenu tout au long de ce parcours. Leur amour, leur encouragement, leurs sacrifices et leur soutien inconditionnel ont été les moteurs de ma réussite.

Je tiens à exprimer ma gratitude envers tous mes amis et toutes les personnes qui m'ont apporté leur aide, que ce soit par de petites attentions ou des mots qui m'ont rendu heureux. Je suis reconnaissant envers chaque personne qui m'a offert des expériences ou des leçons de vie, m'aidant ainsi à grandir en tant qu'individu. Merci à tous.

- Achref

Dédicace

C'est avec une profonde gratitude que je dédie ce projet de fin d'études à toutes les personnes qui ont contribué à sa réalisation. Le chemin jusqu'à ce moment n'a pas été sans embûches, mais grâce à votre soutien, votre amour et votre encouragement, j'ai pu franchir chaque étape avec détermination et persévérance.

Je dédie ce travail à ma famille, mes amis, mes camarades de l'école qui m'a soutenu tout au long de ce parcours. Leur amour, leurs encouragements, leurs sacrifices et leur soutien inconditionnel ont été les moteurs de ma réussite.

À tous ceux qui me sont chers, à vous tous."

- Abdellah

Remerciements

Nous souhaitons exprimer notre sincère gratitude à notre promoteur, le Dr Sid-Ahmed BERRANI, qui nous a soutenus tout au long de ce travail. Nous le remercions pour sa patience, sa motivation, sa disponibilité et son dévouement sans pareil. Nous n'aurions pas pu espérer meilleur superviseur et mentor.

Nous remercions tout particulièrement les membres du jury qui ont accepté de consacrer leur temps à examiner ce travail : Monsieur Mourad ADNANE, Professeur à l'Ecole Nationale Polytechnique, Madame Nesrine BOUADJENEK, Enseignante à l'Ecole Nationale Polytechnique.

Nous tenons à adresser nos remerciements et notre gratitude à tous nos professeurs de l'Ecole Nationale Polytechnique, qui nous ont guidé tout au long de nos cinq années d'études.

Nous remercions également tous nos collègues de l'Ecole Nationale Polytechnique, avec qui nous avons partagé de précieux moments.

Enfin, nous remercions tous ceux qui nous ont aidé de près ou de loin tout au long de notre cursus académique.

Achref et Abdellah.

Table des matières

Liste des tableaux

Table des figures

Liste des abréviations

GPU *Graphics Processing Unit.*

VAE *Variational autoencoder.*

CNN *Convolutional neural network.*

RNN *Recurrent neural network.*

AE *Auto-encodeur.*

MSE *Mean squared error.*

CE *Cross-entropy.*

RP *Rétro-propagation.*

SGD *Stochastic gradient descent.*

ACP *Analyse en Composantes Principales.*

K-L *Kullback-Leibler.*

GAN *Generative Adversarial Network.*

2DFAN *2D Face Alignment Network .*

MTCNN *Multi-task Cascaded Convolutional Neural Network.*

DFL *DeepFaceLab.*

SPSS *Synthèse vocale paramétrique statistique.*

HMM *Hidden Markov Model.*

DNN *Deep neural Network .*

ToBI *Tones and break indices.*

NeRF *Neural Radiance Fields.*

ReLU *Rectified Linear Unit .*

ASR *Automatic Speech Recognition.*

MLP *Multilayer perceptron.*

LLE *Locally Linear Embedding.*

GT *Ground Truth.*

TTS *Text To Speech.*

RVC *Retrieval Based Voice Conversion.*

GPT *Generative Pre-training Transformer.*

WF *Whole Face.*

PSNR *Peak Signal-to-Noise Ratio.*

SSIM *Structural Similarity Index Measure.*

VMAH *Video Multimethod Assessment Fusion.*

PQI *Picture Quality Index.*

VIF *Visual Information Fidelity.*

DLM *Detail Loss Metric.*

MCPD *Mean Co-Located Pixel Difference.*

Introduction générale

Introduction générale

L'évolution technologique a profondément transformé notre quotidien, rendant les dispositifs informatiques indispensables. Des ordinateurs portables aux smartphones en passant par les montres connectées, ces technologies facilitent nos tâches quotidiennes et enrichissent nos expériences. Cette omniprésence souligne l'importance de l'interaction homme-machine, où la reconnaissance et la génération de contenus jouent un rôle crucial.

Parmi les innovations technologiques, les techniques de génération de *deepfake* ont récemment attiré une attention particulière. Les vidéos, créés ou modifiés par des modèles de réseaux neuronaux tels que les *Generative Adversarial Networks* (GANs) et les autoencodeurs variationnels, permettent de produire des vidéos et des images très réalistes en modifiant des visages, des voix et d'autres éléments visuels ou sonores. Cependant, les *deepfakes* soulèvent des problèmes éthiques et de sécurité en raison de leur potentiel à manipuler l'information et à tromper les spectateurs. Ainsi, la détection et l'évaluation des *deepfakes* sont devenues des domaines de recherche essentiels pour prévenir les abus.

Ce mémoire explore en profondeur les techniques de génération des deepfakes pour créer des contenus audios et vidéos réalistes à l'aide d'outils existants. Il combine diverses méthodes et techniques afin d'atteindre cet objectif. Les contributions de ce travail portent principalement sur les points suivants :

1 Compréhension et maîtrise des différentes méthodes et outils de génération des deepfakes audio et vidéo.

2 Production de deepfakes audio réalistes dans différentes langues en combinant des techniques de synthèse vocale (TTS + RVC) associées à un entraînement de modèles de conversion vocale pour des personnes

parlant arabe (darija algérienne).

3 Évaluation approfondie des deepfakes générés. Cette évaluation a été réalisée à l'aide de métriques objectives (comme le PSNR ou le SSIM) mais aussi grâce à une étude impliquant des utilisateurs qui ont pu qualifier la qualité perçue des deepfakes. Il s'agit d'une évaluation subjective.

Le premier chapitre introduit les généralités sur les *deepfakes* et présente les travaux de recherche réalisés dans ce domaine. Nous y fournirons une vue d'ensemble des manipulations médiatiques, de l'historique du développement des *deepfakes* et des définitions essentielles pour comprendre cette technologie. Quelques statistiques pertinentes viendront illustrer l'ampleur et l'impact des *deepfakes* dans le paysage numérique actuel.

Le deuxième chapitre se concentre sur les méthodes de génération des *deepfakes*. Nous y définirons les concepts et présenterons une vue d'ensemble des modèles de réseaux neuronaux à la base des *deepfakes*, tels que les GANs et les autoencodeurs, y compris les autoencodeurs variationnels. Nous décrirons les techniques employées pour générer ces contenus, ainsi que les conséquences de leur utilisation.

Le troisième chapitre expose l'approche et le protocole d'expérimentation pour la génération des *deepfakes*. Nous y présenterons les outils les plus récents pour la génération des *deepfakes* et expliquerons leur principe de fonctionnement. En particulier, nous détaillerons le protocole de génération des vidéos ainsi que la technique de synthèse vocale (Text2Speech + RVC), en illustrant le processus de génération par des exemples concrets.

Dans le quatrième chapitre, nous présenterons les résultats expérimentaux et les techniques d'évaluation de la qualité des *deepfakes* générés par DINET, GENFACE++ et WAVE2LIP. Nous aborderons deux types d'évaluation : objective (PSNR, SSIM, VMAF) et subjective (protocole d'évaluation audio et visuel, incluant la précision de synchronisation des lèvres, la qualité vidéo, le réalisme vidéo et la qualité vocale). Nous inclurons également une évaluation utilisant des détecteurs de deepfakes. Enfin, nous discuterons des commentaires sur les performances des outils de généra-

tion.

Enfin, nous conclurons ce mémoire par une synthèse des résultats obtenus, les leçons apprises et les perspectives pour de futures recherches dans le domaine des *deepfakes*. ce travail vise une meilleure compréhension et à l'amélioration des techniques de génération des *deepfakes*, tout en offrant des méthodes d'évaluation rigoureuses pour garantir l'intégrité et la crédibilité des contenus multimédias dans notre société numérique.

Chapitre 1

Généralités sur les *deepfakes*

1.1 Introduction

Dans ce chapitre, nous aborderons l'aspect théorique du sujet en examinant en détail les principes fondamentaux des *deepfakes* et leur relation avec les techniques de manipulation vidéo. Nous plongerons dans les concepts théoriques essentiels pour comprendre le fonctionnement des *deepfakes*.

De plus, nous commencerons par définir les *deepfakes* et retracer leur évolution depuis leurs débuts jusqu'à leur état actuel. Cette exploration nous permettra de comprendre comment les *deepfakes* ont évolué au fil du temps et comment ils sont devenus une technologie significative dans le domaine de la manipulation de médias.

Nous présenterons ensuite quelques chiffres pertinents pour illustrer l'impact et la prévalence des *deepfakes* dans divers contextes. Ces données chiffrées nous permettront de mieux saisir l'ampleur de l'utilisation des *deepfakes* et leur impact potentiel dans la société.

Cette combinaison d'aspects théoriques, de définition, d'historique de développement et de statistiques nous permettra de fournir une vision globale et d'actualité du sujet, en soulignant les avancées qui ont contribué à une meilleure compréhension et à une utilisation plus efficace des *deepfakes* dans divers domaines de recherche et d'application.

1.2 Manipulation multimedia

1.2.1 Définition

La manipulation des médias consiste en des campagnes organisées par des acteurs pour tromper ou désinformer, en exploitant les caractéristiques des communications de masse ou des plateformes numériques afin de promouvoir leurs intérêts. Elle utilise des stratégies telles que les sophismes, la désinformation, et les techniques de propagande, souvent en supprimant ou détournant l'attention des informations. Jacques Ellul souligne que l'opinion publique est façonnée par les médias de masse, essentiels à la propagande [1]. Ces manipulations sont courantes en relations publiques, pro-

pagande et marketing, où les techniques employées sont souvent similaires malgré des objectifs différents. Les méthodes modernes de manipulation s'appuient souvent sur des distractions, en supposant que le public a une attention limitée.

Un exemple de manipulation télévisée est illustré dans le journal TV polonais Dziennik, qui diffamait le capitalisme dans la Pologne communiste en utilisant un langage émotionnel et chargé (voir figure. 1.1).

Fig. 1.1 : Exemple de manipulation télévisée, le journal TV polonais Dziennik diffamait le capitalisme dans la Pologne communiste en utilisant un langage émotionnel et chargé [2].

1.2.2 Techniques de manipulation des médias

1.2.2.1 Montage vidéo/audio

Ce processus implique la modification de vidéos ou d'audios pour altérer leur contenu. Cela peut inclure la suppression de segments, l'ajout de nouveaux éléments, ou même le changement de l'ordre des événements.

1.2.2.2 Slogans et narration émotionnelle

L'utilisation de slogans accrocheurs et d'une narration émotionnelle peut influencer les émotions du public et renforcer un message particulier. Ces techniques sont souvent utilisées dans la publicité politique et les campagnes de relations publiques.

1.2.2.3 Mauvaise interprétation des données

La présentation sélective de données statistiques ou l'utilisation de graphiques trompeurs peuvent être utilisées pour soutenir un argument spécifique. Cela peut inclure l'omission de données importantes ou la manipulation des échelles sur les graphiques.

1.2.2.4 Propagande

La propagande implique la diffusion de fausses informations ou de rumeurs dans le but d'influencer l'opinion publique. Cela peut être fait à travers les médias traditionnels, les réseaux sociaux, ou d'autres canaux de communication.

1.2.2.5 *Deepfakes* et synthèse de médias

Les *deepfakes* sont des médias synthétiques créés à l'aide de l'intelligence artificielle, tels que des vidéos où des personnes disent ou font des choses qu'elles n'ont jamais dites ou faites. Cette technologie peut être utilisée pour créer des faux discours, des interviews, ou même des scènes pornographiques.

1.3 Manipulation manuelle des images

1.3.1 Censure des images en Union Soviétique

La censure des images en Union soviétique est l'ensemble des mesures prises pour modifier les documents relatant l'histoire de l'Union soviétique. Il s'agit principalement de falsification d'images photographiques, d'où sont purement et simplement « éliminés » les personnages tombés en disgrâce. Il s'agissait de minimiser le rôle effectif de telle ou telle personnalité, mais également de montrer que les dirigeants n'avaient jamais été en contact avec certains leaders devenus infréquentables [3].

1.3.2 Histoire

FIG. 1.2 : De gauche à droite : Nikolaï Antipov, Joseph Staline, Sergueï Kirov et Nikolaï Chvernik. Cette image a été modifiée à plusieurs reprises, supprimant chaque individu tombé en disgrâce. La version finale montre uniquement Staline [4].

Quand Joseph Staline prend les rênes du Parti communiste de l'Union soviétique et devient Président du Conseil des ministres d'URSS, il lance une série de purges visant à éliminer tout type d'ennemis du peuple, réels ou supposés. Au départ, une purge signifiait l'expulsion du Parti, mais à partir des Grandes Purges du milieu des années 1930, les ennemis du Parti – opposants, dissidents politiques, ou simplement ceux perçus comme une menace pour le pouvoir croissant de Staline, y compris d'anciens membres du Parti bolchevique – sont arrêtés, emprisonnés, envoyés au Goulag, exilés en Sibérie, ou exécutés. Le gouvernement soviétique s'efforce alors d'effacer de l'histoire l'existence de ces personnes par divers moyens : retouche de photographies, destruction de films, et dans les cas extrêmes, exécutions sommaires de toute la famille, comme illustré dans la figure 1.2. Après l'exécution de Lev Kamenev en 1936, son image est supprimée des photographies des célébrations de la révolution d'octobre de 1919. De nombreux membres de sa famille sont également éliminés.
Une autre falsification célèbre concerne une photographie de la signature du pacte germano-soviétique en 1939. Dans la version modifiée de l'image, seuls Ribbentrop et Molotov apparaissent. Sur l'originale, plusieurs dignitaires soviétiques, dont Staline, se tenaient également derrière eux dans un décor différent. Après l'attaque allemande contre la Russie, il s'agissait de minimiser l'implication de Staline dans ce traité. [3].

1.4 *Deepfake*

1.4.1 Définition

Les *deepfakes* sont une forme de média synthétique, généralement des vidéos, créés en utilisant des techniques d'intelligence artificielle avancées, notamment les réseaux antagonistes génératifs (GAN) [5]. Ces vidéos utilisent des algorithmes pour superposer le visage et les expressions faciales d'une personne sur le corps d'une autre personne dans une vidéo existante. Le résultat est une vidéo réaliste qui semble montrer la personne ciblée faisant ou disant quelque chose qu'elle n'a jamais fait ou dit en réalité. Les *deepfakes* peuvent être utilisés à diverses fins, notamment la satire, le divertissement et la recherche. Cependant, ils soulèvent également des préoccupations importantes en matière de désinformation, de confidentialité et de sécurité [6]. Leur capacité à créer des vidéos trompeuses et convaincantes peut être exploitée pour diffuser de fausses informations, discréditer des personnalités publiques, ou même créer du contenu pornographique non consenti.

Les *deepfakes* posent des défis pour la société et la technologie en termes de détection et de régulation. Des recherches sont en cours pour développer des méthodes de détection efficaces et des politiques sont élaborées pour reglementer leur utilisation et prévenir les abus [7].

1.4.2 Origine et évolution

Le terme « *deepfake* » est un mot-valise combinant « *deep learning* » (apprentissage profond) et « *fake* » (faux). La technologie a été popularisée à la fin des années 2010, notamment grâce à des forums en ligne et des applications accessibles au grand public. L'évolution rapide des techniques de *deep learning* a permis de rendre ces créations de plus en plus réalistes et difficiles à détecter .

1.4.3 Techniques utilisées pour la génération

Les *deepfakes* reposent principalement sur des techniques de *deep learning*, telles que les *Generative adversarial networks* (GANs) et les autoencodeurs. Les GANs mettent en concurrence deux réseaux neuronaux, l'un générant des images et l'autre tentant de les détecter comme fausses,

ce qui améliore progressivement la qualité des images générées. Les autoencodeurs sont utilisés pour compresser les images et extraire les caractéristiques essentielles du visage, permettant de recréer les expressions faciales sur un autre visage. Les détails et approfondissements de ces techniques seront abordés dans le chapitre 2.

1.4.4 Historique de dévelopement des *deepfakes*

La manipulation de contenu multimédia remonte à 1860, lorsqu'un portrait du politicien John Calhoun a été modifié pour y ajouter la tête du président américain de l'époque à des fins de propagande [8]. Ces manipulations incluent l'ajout (splicing), la suppression (inpainting) et la duplication (copy-move) d'objets entre images, suivies de traitements post-production comme le redimensionnement, la rotation et l'ajustement des couleurs pour améliorer l'apparence visuelle et la cohérence [9].

Avec les progrès des graphismes informatiques et de *deep learning* (DL), des approches automatisées pour la manipulation numérique ont émergé, offrant une meilleure cohérence sémantique. Les vidéos peuvent désormais être synthétisées à partir de zéro en utilisant des autoencodeurs ou des réseaux antagonistes génératifs (GANs) pour diverses applications [10], notamment la génération photoréaliste de visages humains [11]. Les «shallow fakes» ou «cheap fakes» utilisent des logiciels accessibles pour éditer de manière basique des vidéos en ralentissant, accélérant, coupant et assemblant des séquences existantes. En mai 2019, une vidéo de la présidente de la Chambre des États-Unis, Nancy Pelosi, a été modifiée pour donner l'impression qu'elle était confuse ou ivre, obtenant plus de 2,2 millions de vues en 48 heures [12].

L'industrie du divertissement utilise la manipulation vidéo depuis des décennies, notamment en production cinématographique. Un projet académique notable est le «Video Rewrite Program» publié en 1997, qui réanimait automatiquement les mouvements faciaux dans une vidéo existante pour correspondre à une piste audio différente [13]. En septembre 2017, un utilisateur de Reddit nommé «*deepfake*» a posté des vidéos générées par ordinateur où des visages de célèbres actrices étaient placés sur des contenus pornographiques [14]. Une autre application controversée, «deepNude», permettait de générer de fausses images de nues [15].

Aujourd'hui, des technologies comme FakeApp [16], FaceSwap [17], et ZAO [16] sont très accessibles, permettant aux utilisateurs de créer des vidéos truquées sans connaissances approfondies en ingénierie informatique. Des projets open-source comme DeepFaceLab [18] sont disponibles sur GitHub, avec des *tutoriels* sur YouTube. Parmi les projets académiques ayant conduit au développement des *deepfakes*, on trouve Face2Face [5] et Synthesizing Obama [19], publiés respectivement en 2016 et 2017. Face2Face capture les expressions faciales en temps réel et les applique à une autre personne dans une vidéo, tandis que Synthesizing Obama modifie les mouvements de bouche pour faire correspondre des paroles à une piste audio donnée.

Les *deepfakes* se sont étendus à la manipulation de tout le corps [20, 21, 22], à la génération à partir d'une seule image [23, 24] et à la synthèse vidéo temporellement fluide [25]. Bien que de nombreux *deepfakes* sur les réseaux sociaux soient inoffensifs, certains sont utilisés pour des fins malveillantes comme la vengeance pornographique, les canulars, et la fraude financière [26]. En 2018, une vidéo *deepfake* virale montrait l'ex-président Barack Obama insultant le président Donald Trump [27]. En juin 2019, une vidéo truquée du PDG de Facebook, Mark Zuckerberg, a été publiée sur Instagram [26]. Récemment, des vidéos très réalistes de Tom Cruise ont obtenu 1,4 million de vues sur TikTok [28].

Les *deepfakes* audio représentent une nouvelle forme de cyberattaque, causant des dommages significatifs grâce à des techniques sophistiquées de synthèse vocale comme WaveNet [29], Tacotron [30], et Deep Voice 1 [31]. En août 2019, un PDG européen a été dupé par une fausse voix, effectuant un transfert frauduleux de 243 000 $ [32]. Ces techniques peuvent imiter la voix de hauts responsables, posant des risques pour la sécurité nationale [33]. La figure 1.3 illustre une chronologie de l'évolution des techniques de *deepfakes*.

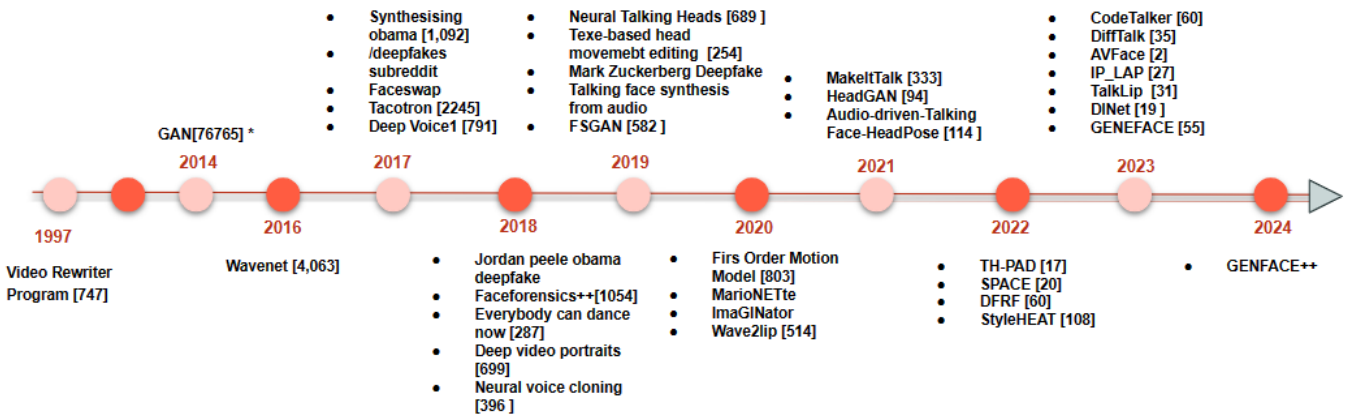


FIG. 1.3 : Chronologie de l'évolution des techniques de *deepfakes*.

1.4.5 Applications et Utilisations

1.4.5.1 Publicité et Marketing

Les marques utilisent des *deepfakes* pour créer des publicités innovantes et engageantes. Par exemple, certaines entreprises créent des avatars numériques de célébrités pour promouvoir leurs produits, permettant ainsi de réduire les coûts et les contraintes logistiques associées aux tournages avec des personnes réelles [34].

1.4.5.2 Divertissement et Cinéma

Les *deepfakes* sont utilisés dans l'industrie cinématographique pour recréer des acteurs décédés, rajeunir des acteurs, ou permettre à un acteur de jouer plusieurs rôles. Par exemple, dans le film «Rogue One : A Star Wars Story», la technologie *deepfake* a été utilisée pour recréer le personnage de la Princesse Leia tel qu'il apparaissait en 1977 (voir figure. 1.4).

FIG. 1.4 : Exemple de *deepfake* dans l'industrie cinématographique : recréation de la princesse Leia dans « Rogue One : A Star Wars Story » [35].

1.4.5.3 Médias et journalisme

Les *deepfakes* peuvent être utilisés à des fins éducatives ou pour reconstituer des événements historiques. Cependant, ils présentent également des risques significatifs pour la désinformation, car ils peuvent être utilisés pour créer de fausses déclarations de personnalités publiques, manipulant ainsi l'opinion publique [36].

1.4.5.4 Réseaux sociaux

Sur les plateformes de réseaux sociaux, les *deepfakes* peuvent être utilisés pour créer du contenu humoristique ou parodique. Toutefois, leur utilisation malveillante pour la cyberintimidation ou le chantage est une préoccupation croissante [36].

1.4.5.5 Éducation et formation

Les *deepfakes* peuvent être utilisés pour des simulations et des formations immersives. Par exemple, des programmes de formation peuvent utiliser des avatars réalistes pour enseigner des compétences interpersonnelles ou techniques dans des environnements contrôlés et réalistes [34].

1.4.5.6 Sécurité et authenticité

Dans le domaine de la cybersécurité, les *deepfakes* posent des défis majeurs. Ils peuvent être utilisés pour usurper l'identité de personnes dans des vidéos de phishing, rendant plus difficile la détection des fraudes. Des

technologies de détection avancées sont en développement pour contrer ces menaces [37].

1.4.6 Quelques chiffres sur les *deepfakes*

Les données de l'enquête d'iProov révèlent une progression de la sensibilisation aux *deepfakes* parmi les consommateurs : en 2019, seuls 13 % d'entre eux savaient ce qu'est un *deepfake*, contre 29 % en 2022. Cependant, étant donné que les *deepfakes* représentent une menace pour la confiance en ligne et potentiellement pour la sécurité nationale, il est crucial d'accroître cette sensibilisation. [38].

Les statistiques montrent également que 57 % des individus estiment pouvoir repérer un *deepfake*, mais en réalité, cela n'est probablement pas aussi facile, à moins que le *deepfake* soit de mauvaise qualité. L'analyse des propriétés spécifiques, comme la réflexion de la lumière sur la peau réelle par rapport à une peau synthétique, nécessite des technologies avancées telles que *deep learning* et la vision par ordinateur [38].

1.5 Conclusion

En conclusion, ce premier chapitre de la thèse entreprend une exploration approfondie des concepts préliminaires essentiels pour une compréhension complète de ce travail. Nous avons examiné en détail les principes fondamentaux des *deepfakes* et leur relation étroite avec les techniques de manipulation vidéo. En plongeant dans les concepts théoriques essentiels, nous avons établi une base solide pour comprendre le fonctionnement des *deepfakes*, de leur définition initiale à leur développement actuel.

Nous avons retracé l'évolution des *deepfakes* depuis leurs débuts jusqu'à leur état actuel, mettant en lumière comment cette technologie a progressivement gagné en sophistication et en importance dans le domaine de la manipulation des médias. En présentant des données chiffrées pertinentes, nous avons également illustré l'impact croissant et la prévalence des *deepfakes* dans divers contextes, permettant ainsi de mieux appréhender leur utilisation répandue et leur potentiel impact sociétal.

En combinant une approche théorique approfondie, une définition claire, un historique évolutif et des chiffres significatives, ce chapitre offre une vi-

sion complète et actuelle du sujet. Il met en lumière les avancées qui ont contribué à une meilleure compréhension et à une utilisation plus efficace des *deepfakes* dans les domaines de la recherche et des applications, soulignant leur importance croissante dans le paysage médiatique contemporain.

Chapitre 2

Méthode de génération des *deepfakes*

2.1 Introduction

Dans ce chapitre, nous abordons les différentes techniques à la base de la génération des *deepfakes* et nous exposons également les principaux types d'algorithmes qui sont essentiels pour générer ces *deepfakes*.

2.1.1 Définition

Les *deepfakes* sont générés à partir de la substitution ou superposition d'éléments dans une vidéo existante, comme le remplacement de visages, la modification de discours, ou la création de vidéos synthétiques de personnes fictives.

2.1.2 Type d'algorithmes pour la génération

L'élément central des *deepfakes* est l'apprentissage automatique. Grâce à l'apprentissage automatique, les ordinateurs peuvent désormais créer des vidéos et des audios de manière automatique, rapide et relativement simple.

2.1.3 Utilisation l'apprentissage automatique pour la génération

Les réseaux de neurones profonds sont entraînés avec des images d'une personne réelle pour apprendre à quoi elle ressemble et comment elle bouge dans certaines conditions. Ensuite, ce réseau est utilisé sur les images d'une autre personne, et des techniques d'infographie sont appliquées pour combiner la nouvelle personne avec les images originales.

2.2 Les modèles de réseau de neurones à la base des *deepfakes*

2.2.1 Les auto-encodeurs

Un auto-encodeur est un type d'architecture de réseau neuronal conçu pour compresser efficacement (encoder) les données d'entrée vers leurs caractéristiques essentielles, puis reconstruire (décoder) l'entrée d'origine à partir de cette représentation compressée (voir figure 2.1).

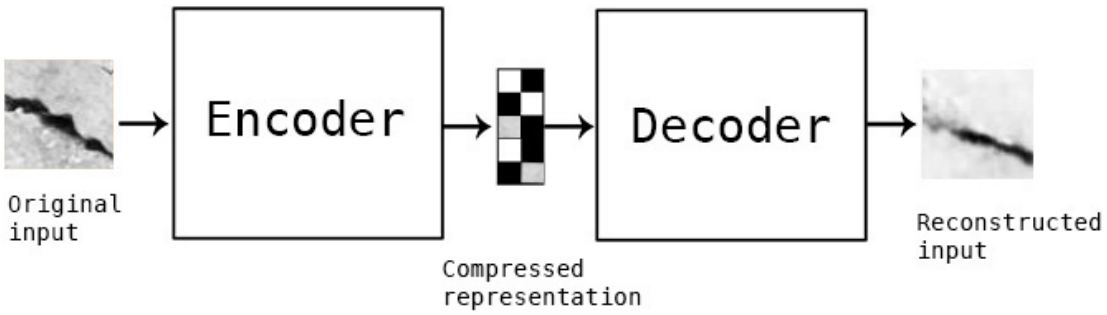


FIG. 2.1 : Structure schématique d'un auto-encodeur [39].

2.2.1.1 Principe des auto-encodeurs

Grâce à un mécanisme d'apprentissage automatique non supervisé, les auto-encodeurs sont formés pour identifier les variables latentes des données d'entrée. Ces variables cachées ou aléatoires, bien que non directement observables, influencent fortement la distribution des données. Ensemble, les variables latentes d'un ensemble de données spécifique sont appelées l'espace latent. Pendant l'entraînement, l'auto-encodeur apprend quelles variables latentes peuvent être utilisées pour reconstruire les données d'origine avec une grande précision. Cette représentation de l'espace latent ne contient donc que les informations les plus essentielles de l'entrée d'origine.

2.2.1.2 Utilisation des auto-encodeurs

La plupart des auto-encodeurs sont employés pour des tâches d'intelligence artificielle centrées sur l'extraction de caractéristiques, tels que la compression de données, le débruitage d'images, la détection d'anomalies et la reconnaissance faciale. Certains types, comme les auto-encodeurs variationnels (VAE) [40] et les auto-encodeurs adversariaux (AAE)[41], modifient leurs architectures pour être utilisés dans des tâches génératives, comme la création d'images ou la génération de données de séries temporelles.

2.2.1.3 Auto-encodeurs et encodeurs-décodeurs

Bien que tous les modèles d'auto-encodeurs incluent à la fois un encodeur et un décodeur, tous les modèles d'encodeurs-décodeurs ne sont pas

des auto-encodeurs.

Dans les schémas encodeurs-décodeurs, un réseau d'encodage extrait les principales caractéristiques des données d'entrée, et un réseau de décodage utilise ces caractéristiques comme entrée. Ces schémas sont utilisés dans divers modèles d'apprentissage profond, comme les architectures de réseaux neuronaux convolutifs (CNN) pour des tâches de vision par ordinateur telles que la segmentation d'images, ou les architectures de réseaux neuronaux récurrents (RNN) pour des tâches de séquence à séquence (seq2seq) [42].

Dans la plupart des applications des modèles encodeurs-décodeurs, la sortie du réseau est différente de l'entrée. Par exemple, dans les modèles de segmentation d'images comme U-Net [43], le réseau d'encodage extrait les caractéristiques de l'image d'entrée pour déterminer la classification sémantique des différents pixels. Ensuite, à partir de cette carte de caractéristiques et de la classification des pixels, le réseau de décodage crée des masques de segmentation pour chaque objet ou région de l'image. L'objectif de ces modèles encodeurs-décodeurs est de classer avec précision les pixels selon leurs catégories sémantiques. Ils sont entraînés via un apprentissage supervisé, en optimisant les prédictions du modèle par rapport à un ensemble d'images de référence étiquetées par des experts humains.

2.2.1.4 Différence entre les auto-encodeurs et encodeurs-décodeurs

Les auto-encodeurs font référence à un type particulier d'architectures encodeurs-décodeurs entraînées via un apprentissage non supervisé pour reconstruire leurs propres données d'entrée.

Étant donné qu'ils ne s'appuient pas sur des données d'apprentissage étiquetées, les auto-encodeurs ne sont pas considérés comme une méthode d'apprentissage supervisée. Comme pour toutes les méthodes d'apprentissage non supervisé, les auto-encodeurs sont entraînés à découvrir des motifs cachés dans les données non étiquetées, plutôt que de prédire des motifs connus présents dans des données étiquetées. Cependant, à la différence de la plupart des exemples d'apprentissage non supervisé, les auto-encodeurs comparent leur sortie à une vérité-terrain. L'entrée d'origine elle-même

(ou une version modifiée de celle-ci). C'est pourquoi ils sont souvent classés comme une forme d'apprentissage auto-supervisé, d'où le terme auto-encodeur.

2.2.1.5 Fonctionnement des auto-encodeurs

Auto-encodeurs :

Les auto-encodeurs découvrent les variables latentes en faisant passer les données d'entrée dans un « goulot d'étranglement » avant qu'elles atteignent le décodeur. Cela oblige l'encodeur à apprendre à extraire et à transmettre uniquement les informations essentielles et nécessaires à la reconstruction fidèle de l'entrée d'origine.

Structure et objectif :

Le modèle de base de l'auto-encodeur (AE) est composé d'une couche d'entrée, d'une couche cachée et d'une couche de sortie.

Un auto-encodeur (AE) prend un vecteur d'entrée et le transpose ensuite à la représentation cachée $y \in \mathbb{R}^{d'}$ en utilisant la transformation déterministe $y = f_\Theta(x) = s_f(Wx + b)$. La matrice W est une matrice de poids $d' \times d$, b est un vecteur de biais, et s_f est la fonction d'activation de l'encodeur (typiquement une non-linéarité élémentaire comme la sigmoïde ou la tangente hyperbolique, ou la fonction identité si l'on reste linéaire).
La représentation latente y, ou représentation cachée, est ensuite transposé (avec un décodeur) en un vecteur de reconstruction $z \in \mathbb{R}^d$ (z a la même forme que x).
La transformation est effectuée en utilisant une transformation similaire, par exemple $z = g_\Theta(y) = s_g(W'y + b')$, où $\theta = \{W, b, W', b'\}$ et s_g est la fonction d'activation du décodeur.
De plus, z peut être vu comme une prédiction de x donnée la représentation cachée y. Ce processus peut être résumé comme suit :
chaque entrée x_i est ainsi transposé à un y_j correspondant qui est ensuite transpose à une reconstruction z_i, de sorte que $z_i \approx x_i$. Il est judicieux de contraindre éventuellement la matrice de poids W' par $W' = W^T$. De cette façon, le nombre de paramètres libres est réduit, ce qui simplifie l'entraînement. Cela est appelé poids liés.

L'ensemble des paramètres θ de ce modèle est optimisé de sorte que la

fonction de perte soit minimisée, comme le montre l'équation suivante :

$$\theta^* = \arg\min_{\theta} \sum L(x, z) \tag{2.1}$$

où L est une fonction de perte. La méthode pour choisir s_g et L dépend largement de la plage et de la nature du domaine d'entrée. L peut être choisie comme l'erreur quadratique moyenne (MSE) traditionnelle, qui peut être exprimée comme l'équation (2.2). Ceci, couplé avec un décodeur linéaire (c'est-à-dire $s_g(a) = a$). À l'inverse, si les entrées sont bornées entre 0 et 1, utiliser s_g (sigmoïde) peut garantir une reconstruction de même borne. De plus, si l'entrée x est interprétée comme une séquence de bits ou une séquence de probabilités de bits (c'est-à-dire des vecteurs de probabilité de Bernoulli), alors l'entropie croisée (CE) peut être utilisée, comme défini dans l'équation (2.3).

$$L(x, z) = \frac{1}{2} \sum_i (x_i - z_i)^2 \tag{2.2}$$

$$L(x, z) = -\sum_i x_i \log z_i + (1 - x_i) \log(1 - z_i) \tag{2.3}$$

En particulier, il y a deux propriétés qui rendent raisonnable l'interprétation de la CE comme une fonction de coût. Premièrement, elle est non négative, c'est-à-dire $L(x, z) > 0$. Deuxièmement, la CE tend vers zéro à mesure que le neurone devient meilleur pour calculer la sortie désirée z, pour toutes les entrées d'entraînement x. À condition que les neurones de sortie soient des neurones sigmoïdes, la CE est presque toujours le meilleur choix. Cependant, si les neurones de sortie sont des neurones linéaires, alors la MSE ne posera aucun problème de ralentissement de l'apprentissage. Dans ce cas, la MSE est, en fait, une fonction de coût appropriée à utiliser [44].

Entrainement :

Le processus d'apprentissage dans les réseaux neuronaux artificiels ; il est généralement mis en œuvre à l'aide d'exemples et atteint en ajustant de manière itérative les poids de connexion. Les algorithmes d'entraînement pour les réseaux neuronaux artificiels se divisent en deux grandes catégories : basés sur le gradient et non basés sur le gradient.

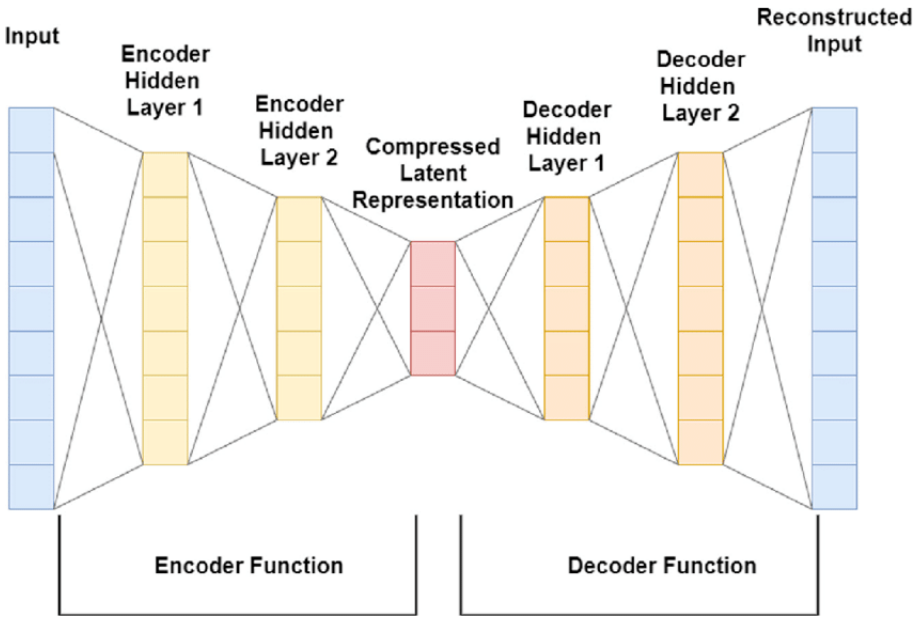


FIG. 2.2 : Une illustration simplifiée de l'architecture d'un autoencodeur profond [45].

Les auto-encodeurs (AE) peuvent être considérés comme des cas particuliers de réseaux de neurones et peuvent être entraînés avec toutes les mêmes techniques (voir figure 2.2). Dans cette section, nous nous concentrerons sur les méthodes basées sur le gradient car elles sont plus couramment utilisées de nos jours et convergent généralement beaucoup plus rapidement.

Comme mentionné dans la section 2.1.1, notre discussion s'est concentrée sur la mise en œuvre des fonctions qui calculent $L(\theta; x)$ avec l'ensemble de paramètres θ. Par conséquent, l'objectif du processus d'entraînement est de trouver un θ tel que $L(\theta; x)$ approxime la fonction que nous essayons de modéliser.
Soit $\nabla L(\theta; x)$ le gradient de $L(\theta; x)$ avec les paramètres θ. Le gradient n'a pas de solution en forme fermée. Au lieu de cela, il peut être implémenté efficacement en utilisant l'algorithme RP (Rétropropagation), qui est l'élément essentiel de l'apprentissage dans les réseaux neuronaux. Les paramètres θ d'un AE peuvent être le plus souvent entraînés avec les algorithmes d'optimisation suivant le gradient calculé en utilisant RP.

Une heuristique largement utilisée pour l'entraînement des réseaux neuronaux repose sur un cadre appelé SGD (la descente de gradient stochastique) [46]. Dans les réseaux neuronaux, la fonction de perte est fortement non convexe ; cependant, nous pouvons toujours implémenter les algorithmes SGD et trouver une solution raisonnable.

2.2.1.6 Type d'auto-encodeur

Comme discuté dans la section 2.2.1 , la structure générale d'un AE de base se compose de trois couches : une couche d'entrée, une couche cachée formant l'encodage, et une couche de sortie dont les unités correspondent à la couche d'entrée.
Étant donné que les sorties sont égales à l'entrée, cela revient à apprendre une approximation de la fonction d'identité.
Un AE dont la dimension cachée est inférieure à la dimension d'entrée est appelé sous-complet [47] (aussi appelé «étroit» ou «*bottleneck*»). Cette méthode permet de découvrir les caractéristiques les plus saillantes de l'ensemble de données qui reposent sur moins d'unités de la couche cachée.
Dans le cas d'un AE linéaire (encodeur et décodeur linéaires) avec une fonction MSE traditionnelle, la minimisation de l'équation (2.1) apprend un sous-espace similaire a celui de l'Analyse en Composantes Principales (ACP)

Auto-encodeurs convolutionnel : Les autoencodeurs traditionnels ne prennent pas en compte le fait qu'un signal peut être considéré comme une somme d'autres signaux. Les autoencodeurs convolutifs utilisent l'opérateur de convolution pour exploiter cette observation (figure 2.3). Ils apprennent à encoder l'entrée dans un ensemble de signaux simples, puis tentent de reconstruire l'entrée à partir de ces signaux, en modifiant la géométrie ou la réflectance de l'image. Il s'agit d'outils de pointe pour l'apprentissage non supervisé des filtres de convolution. Une fois ces filtres appris, ils peuvent être appliqués à n'importe quelle entrée pour extraire des caractéristiques. Ces caractéristiques peuvent ensuite être utilisées pour des tâches nécessitant une représentation compacte de l'entrée, comme la classification.

Auto-encodeurs variationnels : Les auto-encodeurs variationnels (VAE) sont des modèles génératifs qui apprennent à compresser leurs données d'entraînement sous forme de distributions de probabilités. Ces distributions sont ensuite utilisées pour générer de nouveaux échantillons de données en créant des variations à partir de ces représentations apprises.

La différence principale entre les VAE et les autres types d'auto-encodeurs est que, alors que la plupart des auto-encodeurs apprennent des modèles à

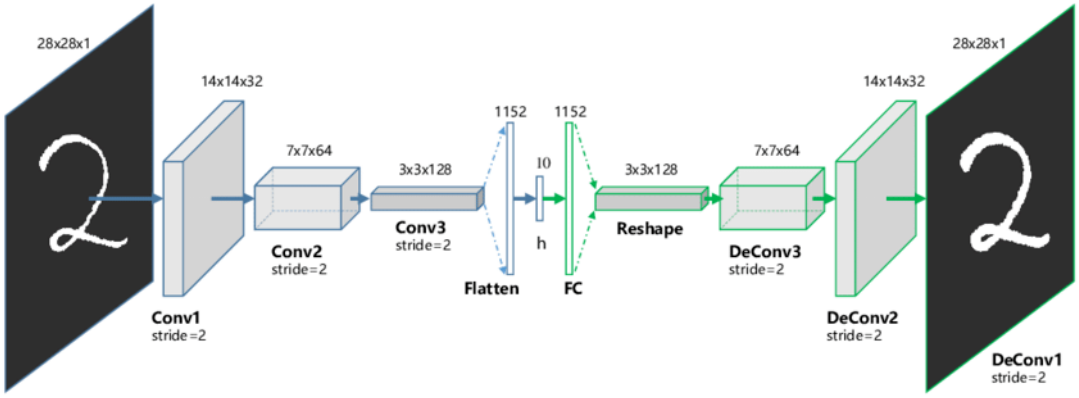


FIG. 2.3 : Une illustration de l'architecture d'un autoencodeur convolutionnel [48].

espace latent discret, les VAE apprennent des modèles à variables latentes continues. Plutôt qu'un seul vecteur d'encodage pour l'espace latent, les VAE utilisent deux vecteurs distincts :
un vecteur de moyenne (« μ ») et un vecteur d'écart type (« σ »)(figure 2.4). Ces vecteurs représentent les attributs latents sous forme de distributions de probabilités, ce qui signifie qu'ils apprennent un encodage stochastique plutôt qu'un encodage déterministe. Cela permet aux VAE de réaliser des interpolations et des échantillonnages aléatoires, augmentant ainsi leurs capacités et leurs applications potentielles. En d'autres termes, les VAE sont des modèles génératifs d'IA.

En termes plus simples, les VAE apprennent à encoder des caractéristiques importantes à partir des données d'entraînement de manière flexible et approximative, ce qui leur permet de générer de nouveaux échantillons similaires aux données d'origine. La fonction de perte utilisée pour minimiser l'erreur de reconstruction est régularisée par la divergence K-L entre la distribution de probabilités des données d'entraînement (distribution a priori) et la distribution des variables latentes apprises par le VAE (distribution a posteriori). Cette fonction de perte régularisée permet aux VAE de générer de nouveaux échantillons ressemblables aux données d'entraînement tout en évitant le sur-ajustement, qui produirait des échantillons trop similaires aux données originales.

Pour générer un nouvel échantillon, le VAE échantillonne un vecteur latent aléatoire (ϵ) à partir d'une distribution gaussienne standard. En d'autres termes, il choisit un point de départ aléatoire dans la distribution

normale, le déplace selon la moyenne de la distribution latente (μ) et le redimensionne en fonction de la variance de la distribution latente (σ). Ce processus, appelé «*reparameterization trick*», évite l'échantillonnage direct de la distribution variationnelle : étant donné que le processus est aléatoire, il n'a pas de dérivée, ce qui élimine le besoin de rétropropagation pendant l'échantillonnage.

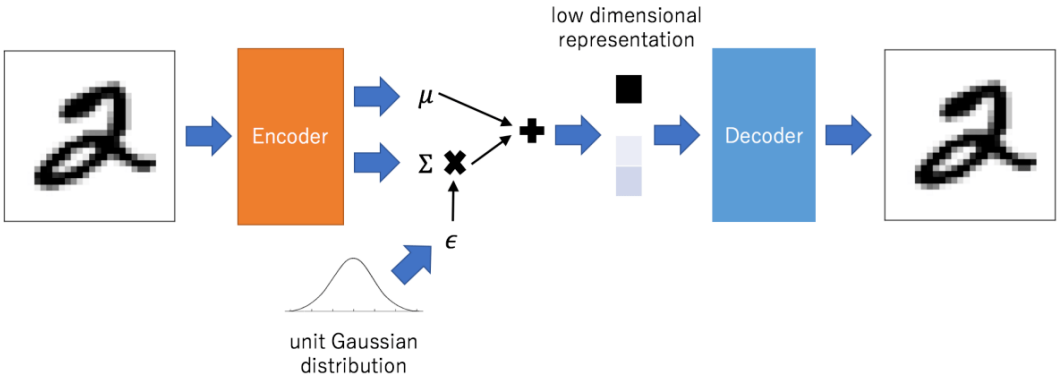


FIG. 2.4 : Une illustration de l'architecture d'un autoencodeur variationnels [49].

Lorsqu'un VAE est utilisé pour des tâches génératives, l'encodeur peut souvent être supprimé après l'entraînement. Les versions plus avancées des VAE, comme les VAE conditionnels [50], permettent aux utilisateurs de mieux contrôler les échantillons générés en fournissant des entrées conditionnelles qui modifient la sortie de l'encodeur.

2.2.2 Les réseaux génératifs adversariaux

Les réseaux génératifs adversariaux (ou GAN, pour «generative adversarial networks») ont été introduits en 2014 par Ian GoodFellow et son équipe dans la publication «Generative Adversarial Nets» de l'Université de Montréal [51].
Les GAN représentent une approche de modélisation générative utilisant des techniques de deep learning, telles que les réseaux de neurones convolutif. La modélisation générative est une tâche d'apprentissage non supervisé qui consiste à découvrir et apprendre automatiquement les régularités et les motifs dans les données d'entrée, permettant ainsi au modèle de générer de nouveaux échantillons ressemblant à ceux de l'ensemble de données d'origine (voir figure 2.5).

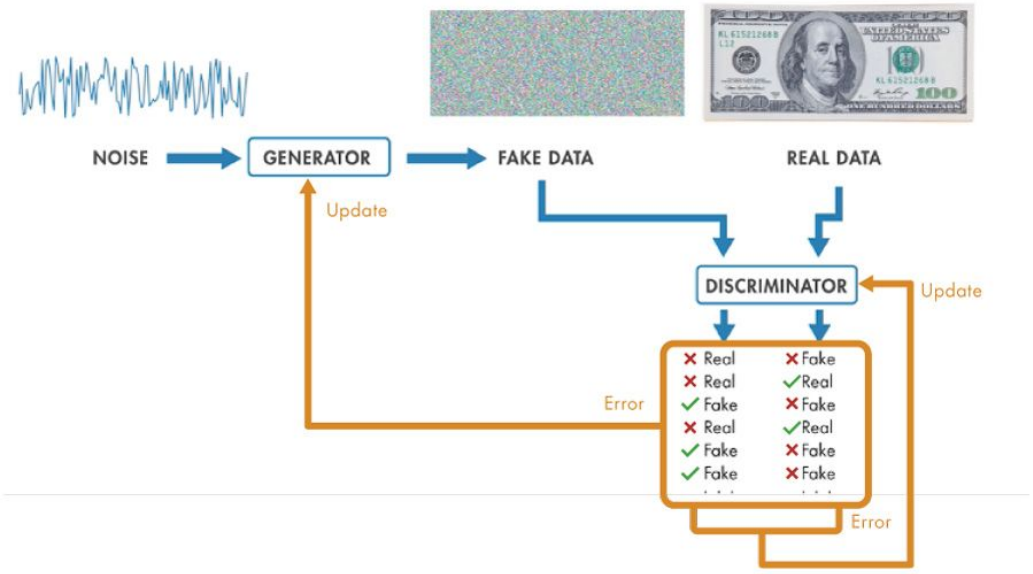


FIG. 2.5 : Une illustration simplifiée du fonctionnement des réseaux génératifs adversariaux (GAN) [52].

2.2.2.1 Architecture des GAN

Les GAN (réseaux génératifs adversariaux) sont un type de modèle génératif qui apprend à créer de nouvelles données. Leur architecture de base comprend deux réseaux de neurones : un réseau générateur (GeneratorNetwork) et un réseau discriminateur (DiscriminatorNetwork) (voir figure 2.6).

Le réseau générateur prend un vecteur de bruit aléatoire en entrée et produit un échantillon synthétique en sortie.

Le réseau discriminateur prend un échantillon de données en entrée et donne une valeur entre 0 et 1 en sortie, indiquant la probabilité que cet échantillon soit une donnée réelle (par opposition à une donnée synthétique générée par le réseau générateur).

Ces deux réseaux sont entraînés de manière concurrente : le réseau générateur essaie de créer des échantillons synthétiques qui ressemblent aux données réelles, tandis que le réseau discriminateur essaie de différencier les échantillons réels des échantillons synthétiques. Ce processus d'entraînement peut être vu comme un jeu à deux joueurs, où le réseau générateur tente de minimiser une fonction objectif et le réseau discriminateur tente de la maximiser.

La fonction objectif du réseau générateur est de minimiser l'expression

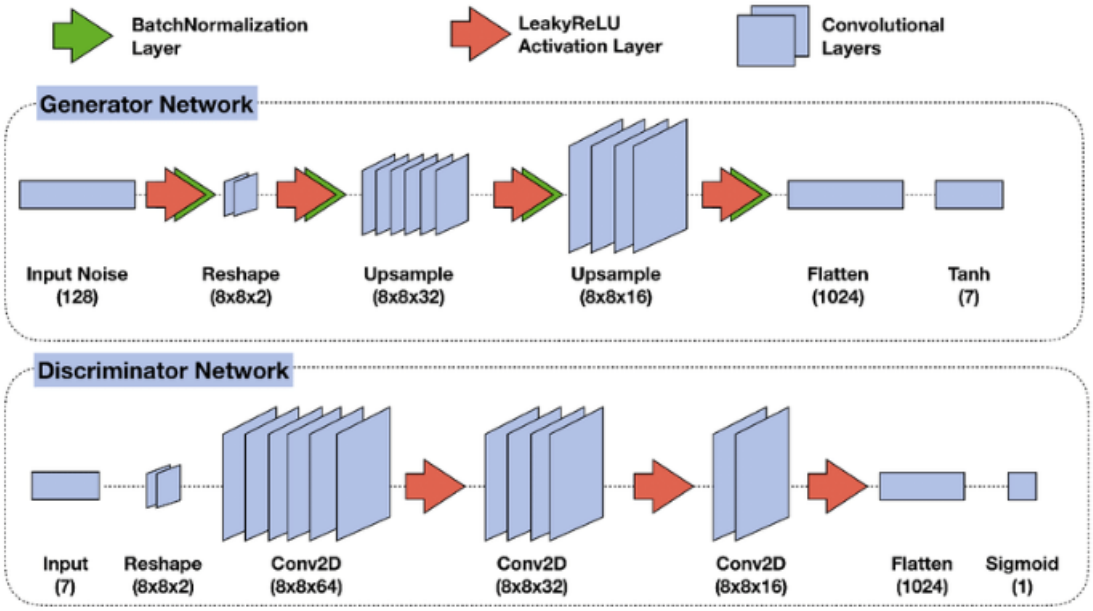


FIG. 2.6 : Architecture du réseau : générateur (en haut), discriminateur (en bas). Le GAN est composé en connectant la sortie du générateur à l'entrée du discriminateur [53].

suivante :

$$\min_G \max_D \mathbb{E}_{s\sim p_{\text{data}}(s)}[\log D(s)] + \mathbb{E}_{v\sim p_v(v)}[\log(1 - D(G(v)))] \tag{2.4}$$

où :

- $p_{\text{data}}(s)$ est la distribution réelle des données,
- v est le vecteur de bruit,
- $p_v(v)$ est la distribution a priori de v,
- $\mathbb{E}$ représente la valeur attendue.

Le premier terme de cette équation encourage le réseau discriminateur à classer correctement les échantillons réels, tandis que le second terme encourage le réseau générateur à créer des échantillons synthétiques que le réseau discriminateur classifie comme réels.

La fonction objectif du réseau discriminateur est de maximiser l'expression suivante :

$$\max_D \mathbb{E}_{s\sim p_{\text{data}}(s)}[\log D(s)] + \mathbb{E}_{v\sim p_v(v)}[\log(1 - D(G(v)))] \tag{2.5}$$

où le premier terme pousse le réseau discriminateur à bien classer les échantillons réels, et le second terme à bien classer les échantillons synthétiques comme étant faux.

2.3 Différentes techniques de génération des *deepfakes*

2.3.1 Reconstitution faciale (*face reenactment*)

La reconstitution faciale (*face reenactment*) est une tâche de synthèse faciale où les expressions et la pose d'un visage source sont transférées sur un visage cible, tout en préservant l'apparence et les détails du visage cible, voir figure 2.7.

FIG. 2.7 : Une représentation visuelle de *deepfake* basée sur la Réanimation faciale «*face reenactment*» [5].

2.3.2 Échange de visages (*face swapping*)

Le changement de visage (*faceswapping*) est une technique de synthèse vidéo où le visage d'une personne est remplacé par celui d'une autre dans une séquence vidéo. Utilisant les réseaux adverses génératifs (GANs) et les auto-encodeur, cette méthode permet de superposer de manière réaliste les caractéristiques faciales, expressions et mouvements d'un visage source sur un visage cible, tout en maintenant la continuité et la fluidité de la vidéo d'origine.
La figure 2.8 montre un pipeline de changement de visage qui se compose du prétraitement des données, d'un modèle de *deepfake* (contenant des

sous-modèles, tels que pour l'extraction d'identité), et du post-traitement pour améliorer les résultats.

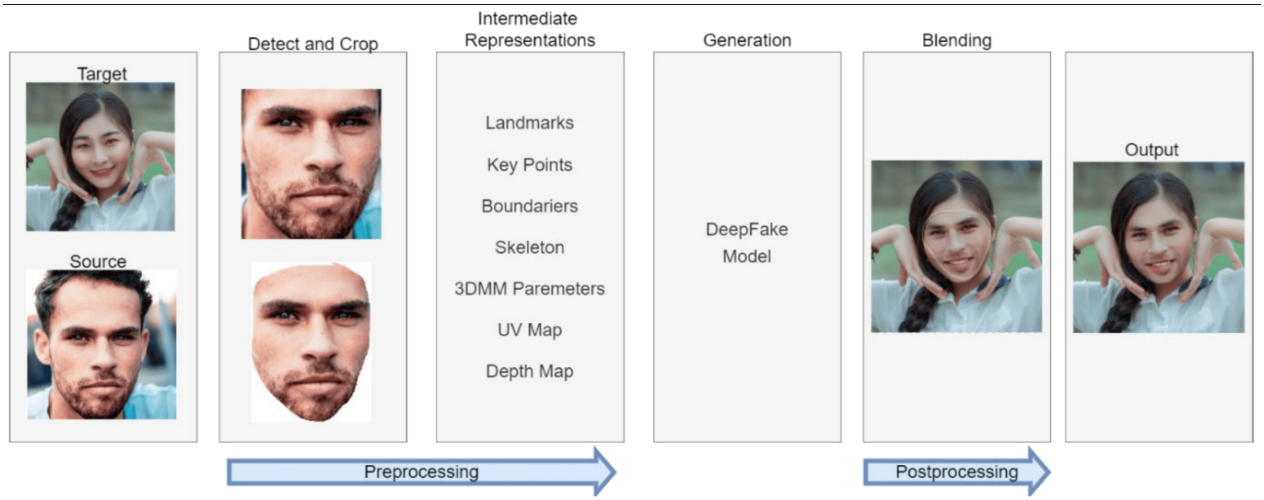


FIG. 2.8 : Un pipeline typique de changement de visage basé sur la source [54].

2.3.2.1 Processus d'échange de visages (*face swapping*)

- **Prétraitement** :
  L'un des principaux aspects, tant pour l'utilisation cible que pour l'entraînement, est le traitement approprié des données, par exemple, pour détecter un visage humain dans une image.
  Tout d'abord, il est essentiel d'acquérir une quantité considérable de données, telles que des photographies ou des films représentant la cible source. Ces données sont ensuite utilisées dans trois processus : détection, alignement et génération de masques, qui sont souvent similaires aux algorithmes discutés plus loin.

  – **La détection de visage** est le processus de localisation d'un visage à l'intérieur d'un cadre. Le détecteur examine l'image et identifie les régions ressemblant à un visage. Plusieurs modèles de détection peuvent être sélectionnés : le détecteur CV2 DNN (basé sur la bibliothèque OpenCV2, utilisant le modèle de détection à un seul coup pré-entraîné—ResNet [55]), les MTCNNs [56] (réseaux convolutifs multi-tâches en cascade), et le S3FD [57] (détecteur de visage invariant à l'échelle à un seul coup—le meilleur détecteur parmi ceux proposés).

  – **L'alignement** implique de trouver des «points de repère» (re-

présentations intermédiaires dans la figure 2.8) au sein du visage pour ainsi orienter ce dernier. Ce processus utilise le résultat du détecteur et détermine où se situent les principales caractéristiques du visage (yeux, bouche, nez, etc.).
Deux algorithmes peuvent être sélectionnés : 2DFAN (*2D Face Alignment Network* [58],(appliqué aux visages avec une pose standard) et PRNet [59] (utilisé dans les cas exceptionnels où un côté du visage est hors de vue).

De plus, DeepFacelab contient une fonctionnalité optionnelle avec un pas de temps configurable pour assurer la stabilité des points détectés. Ils utilisent l'approche traditionnelle de cartographie et de transformation des motifs de points d'Umeyama [60] pour produire la matrice de transformation de similarité nécessaire à l'alignement du visage (figure 2.9).

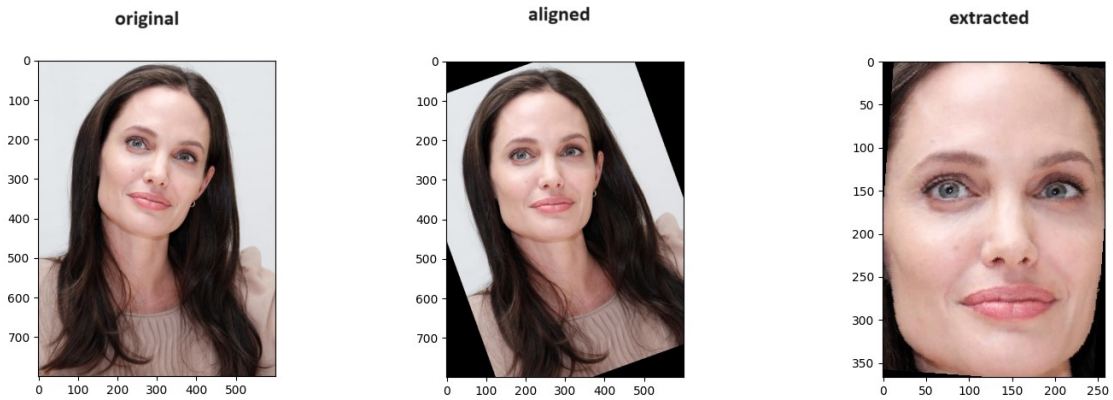


Fig. 2.9 : La détection l'alignement et l'extraction de visage [61]

– **La génération de masque** enlève l'arrière-plan et les obstacles de la zone de l'image, ne laissant que le visage.TernausNet [62] a été utilisé pour la segmentation et la génération de masques (figure 2.10).

FIG. 2.10 : La génération de masque

Les résultats obtenus lors du prétraitement sont utilisés à la fois dans la phase d'entraînement et dans l'utilisation réelle.

- **Extraction d'identité** :
  L'un des composants essentiels du pipeline responsable du remplacement de visage est l'extracteur d'identité. Sa tâche consiste à représenter le visage d'une personne dans l'espace caché approprié pour éliminer les attributs inutiles qui ne sont pas universels pour les photos de la même personne prises à différents moments et dans différentes situations.
  FaceSwap [17] et DeepFaceLab [18] sont des algorithmes un à un, donc l'entraînement de l'extracteur d'identité doit également se produire pendant l'entraînement de l'ensemble du pipeline. FaceSwap est basé sur l'idée d'utiliser des auto-encodeurs.
  Pour utiliser des auto-encodeurs pour échanger des visages, les auteurs entraînent deux auto-encodeurs : un pour la personne dont le visage doit être déplacé (la source) et un pour la personne dont le visage sera remplacé (la cible) puis lors de la génération, les décodeurs sont échangés, de sorte que le visage latent A est soumis au décodeur B pour générer le visage A avec les caractéristiques du visage B (voir figure 2.11).
  Les caractéristiques d'identité sont attribuées au modèle dans une telle solution en raison de la différence entre la dimension cachée à la dimension d'entrée utilisé. Les auto-encodeurs tentent finalement de reconstruire l'identité adaptée au modèle en se basant uniquement sur

des attributs non universels.
DeepFaceLab utilise un encodeur standard dans leur solution, et le transfert des caractéristiques d'identité se fait par le biais d'un décodeur qui est individuel pour chaque personne.

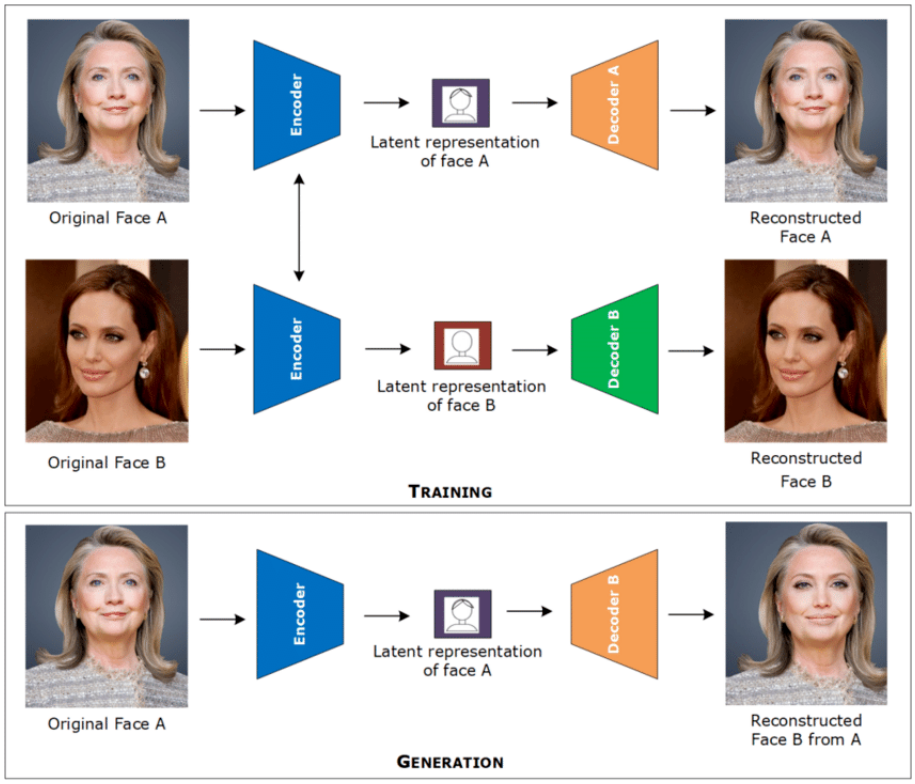


FIG. 2.11 : Création d'un *deepfake* en utilisant un auto-encodeur et un décodeur [63].

- **L'extracteur d'attributs** :
  Le deuxième composant du changement de visage est l'extraction des attributs qui ne sont pas les mêmes pour différentes photos de la même personne, c'est-à-dire la pose, l'expression faciale et l'éclairage. Avec FaceSwap et DeepFaceLab, les attributs sont extraits à l'aide d'encodeurs et encodés dans l'espace caché correspondant. En manipulant la taille de cet espace, il est possible d'augmenter la ressemblance à l'original au détriment des caractéristiques universelles et de l'identité. DeepFaceLab utilise également une extension du modèle avec des représentations intermédiaires supplémentaires qui ont été combinées et transférées au décodeur correspondant pour obtenir une meilleure représentation de l'éclairage et de la couleur.

- **Le générateur :**
  Un composant essentiel des sous-modèles de visage est le générateur,

qui est un encodeur combinant et traitant les représentations obtenues dans les étapes précédentes.
Dans le cas de FaceSwap, ainsi que de DeepFaceLab, le processus de génération, tout comme l'extraction, comme on a pu le voir dans les points précédents, est étroitement lié. Il n'y a pas de frontières claires quant à l'endroit où une division particulière se produit. Cependant, cela découle de la méthodologie adoptée et du swap final des décodeurs.
Aucune amélioration supplémentaire n'a été apportée à Faceswap, donc la fonction de coût prouve que le résultat est basé uniquement sur la reconstruction de l'image.
DeepFaceLab implémente des fonctions de coût telles que L1 (erreur absolue moyenne), L2 (erreur quadratique moyenne), Logcosh, DSSIM (différence de similarité structurelle) et GradientLoss. Des modèles GAN ont été ajoutés à DeepFaceLab pour produire des résultats plus réalistes ; néanmoins, leur utilisation est seulement suggérée vers la fin de l'entraînement. Par défaut, DFL utilise une perte mixte du DSSIM et du L2. Cette combinaison est motivée par le désir d'obtenir à la fois des avantages :
DSSIM généralise plus rapidement les visages humains, tandis que MSE offre une plus grande précision. Cette combinaison de pertes vise à trouver un équilibre entre la généralisation et l'expressivité. De plus, la fonction de coût décrite précédemment est appliquée lors de l'utilisation d'un GAN pour les réseaux de générateur et de discriminateur.

2.3.3 Synthèse vocale (*Speech synthesis*)

2.3.3.1 Technologie traditionnelle de synthèse vocale

La synthèse vocale ou TTS (*Text-to-Speech*) consiste à convertir toute information textuelle en un discours fluide et naturel. Elle implique de nombreuses disciplines telles que l'acoustique, la linguistique, le traitement numérique du signal, l'informatique, etc. Il s'agit d'une technologie de pointe dans le domaine du traitement de l'information, particulièrement pour les systèmes actuels d'interaction vocale intelligente.

2.3.3.2 Synthèse vocale paramétrique statistique (SPSS)

Un système complet de synthèse vocale paramétrique statistique (SPSS) est généralement composé de trois modules :

- un module d'analyse de texte.
- un module de prédiction des paramètres.
- un module de synthèse vocale (voir figure 2.12).

Le module d'analyse de texte prétraite principalement le texte d'entrée et le transforme en caractéristiques linguistiques utilisées par le système de synthèse vocale, incluant la normalisation du texte , la segmentation automatique des mots [64], et la conversion grapho-phonémique [65]. Ces caractéristiques linguistiques incluent généralement des caractéristiques au niveau du phonème, de la syllabe, du mot, de la phrase et de la phrase entière.
Le but du module de prédiction des paramètres est de prédire les paramètres des caractéristiques acoustiques de la parole cible selon la sortie du module d'analyse de texte.
Le module de synthèse vocale génère la forme d'onde de la parole cible selon la sortie du module de prédiction des paramètres en utilisant un algorithme de synthèse particulier.
Le SPSS est généralement divisé en deux phases, la phase d'entraînement et la phase de synthèse :

- Lors de **la phase d'entraînement**, les paramètres des caractéristiques acoustiques tels que F0 et les paramètres spectraux sont d'abord extraits du corpus, puis un modèle acoustique statistique est entraîné basé sur les caractéristiques linguistiques du module d'analyse de texte ainsi que sur les paramètres des caractéristiques acoustiques extraits.
- Lors de **la phase de synthèse**, les paramètres des caractéristiques acoustiques sont prédits en utilisant le modèle acoustique entraîné sous la guidance des caractéristiques linguistiques. Enfin, la parole est synthétisée basée sur les paramètres des caractéristiques acoustiques prédits en utilisant un vocodeur.

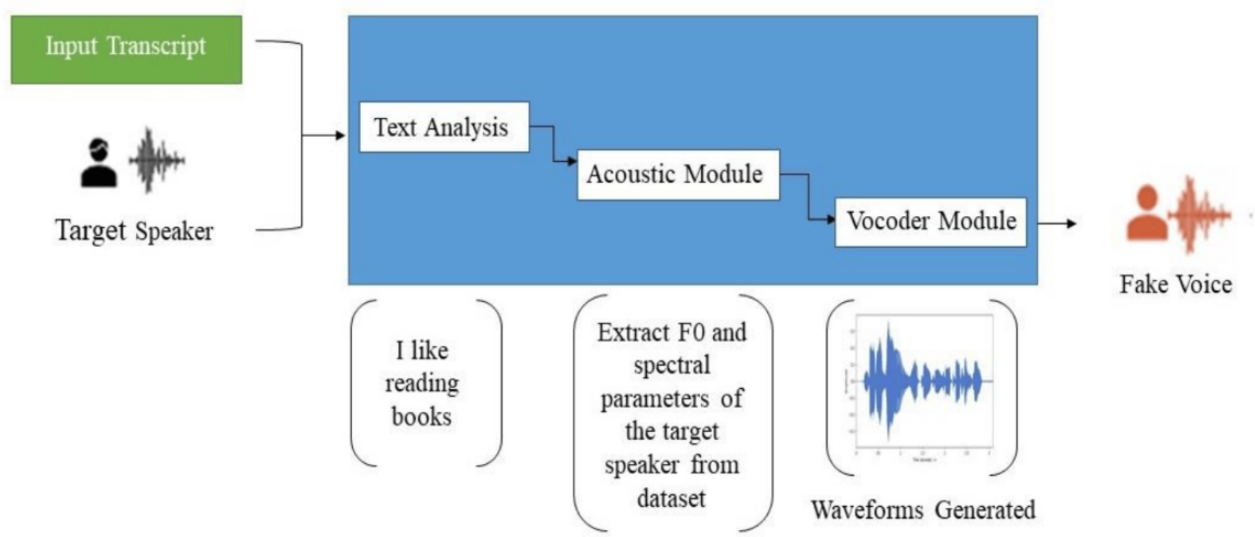


FIG. 2.12 : Les modules d'un système de synthèse vocale paramétrique statistique (SPSS) [66].

Analyse de texte : L'analyse de texte est un module important du modèle SPSS. Les méthodes traditionnelles d'analyse de texte sont principalement basées sur des règles, ce qui nécessite beaucoup de temps pour collecter et apprendre ces règles. Avec le développement rapide de la technologie de fouille de données, certaines méthodes basées sur les données ont progressivement été développées, telles que la méthode bigramme, la méthode trigramme, la méthode basée sur les HMM et la méthode basée sur les DNN. Lors de l'utilisation des deux dernières méthodes pour l'analyse de texte, le système Festival [67] est généralement utilisé pour effectuer la segmentation et l'annotation des phonèmes dans le corpus, ce qui inclut principalement les quatres niveaux suivants :

- Niveau des phonèmes ;
- Niveau des syllabes ;
- Niveau des mots ;
- Niveau des phrases.

Les caractéristiques linguistiques sont obtenues en extrayant les propriétés du texte afin d'obtenir davantage d'informations syntaxiques et sémantiques sur celui-ci, ce qui inclut généralement :

- Normalisation du texte : Il est essentiel de normaliser le texte brut fourni en entrée au format parlé pour la synthèse vocale (TTS). Par exemple, les nombres, comme *58 dinars*, doivent être convertis en *cinquante-huit dinars*. Il en va de même pour les symboles, \$ → *dollar*, & → *et*.

- La segmentation des mots sera obligatoire pour certaines langues en raison de la façon dont la langue est écrite, comme le chinois, afin de créer une séparation entre les mots. Les étapes suivantes, telles que l'étiquetage des parties du discours (POS) et la conversion des phonèmes, nécessitent une segmentation appropriée des mots.

- Étiquetage des parties du discours (POS) : Cette étape fournit la majeure partie des informations sémantiques de la phrase. Durant cette phase, nous étiquetons les parties du discours des mots respectifs dans les phrases, telles que les noms, les verbes, les adjectifs, et ainsi de suite. Les informations extraites de cette phase seront utiles pour la conversion des graphèmes en phonèmes.

- Prédiction de la prosodie : La prosodie désigne le schéma de l'accentuation et de l'intonation (montée et descente de la voix lors de la parole) dans une langue. À cette phase de l'analyse textuelle, nous étiquetons les tons (par exemple, les accents de ton, les accents de phrase et les tons de limite) ainsi que les pauses (l'intensité de la pause entre les mots). Il existe différents modèles pour la prédiction de la prosodie ; pour l'anglais, nous avons le système ToBI (*tones and break indices*).

- Conversion graphème-phonème : C'est la partie principale de la phase d'analyse où nous convertissons les graphèmes (qui sont les caractères d'une langue) en phonèmes (unités sonores de la parole). Les mots sont effectivement convertis en prononciation durant cette phase.

La prédiction des paramètres (*Acoustic model*) Utilisée pour prédire les paramètres des caractéristiques acoustiques tels que la fréquence fondamentale (F0), les paramètres spectraux et la durée en se basant sur le résultat du module d'analyse de texte et le modèle acoustique entraîné.

Module de synthèse vocale (*Vocodeur model*) Le synthétiseur vocal, ou vocodeur, est un composant important de la synthèse vocale paramétrique statistique, dont le but est de synthétiser une forme d'onde vocale basée sur les paramètres acoustiques estimés et les vocodeurs peuvent être divisés en différentes catégories :

- Vocoders basés sur VAE (encodeurs variationnels)
- Vocoders basés sur GAN (réseaux génératifs antagonistes)
- Vocoders autorégressifs

Synthèse vocale de bout en bout(*End to End*) Un système TTS se compose généralement d'un front-end d'analyse de texte, d'un modèle acoustique et d'un synthétiseur vocal. Comme ces composants sont entraînés indépendamment et reposent sur une expertise de domaine approfondie, ce qui est laborieux, les erreurs de chaque composant peuvent s'accumuler. Pour résoudre ces problèmes, les méthodes de synthèse vocale de bout en bout, qui combinent ces composants en un cadre unifié, sont devenues courantes dans le domaine de la synthèse vocale.
Les systèmes TTS de bout en bout présentent de nombreux avantages :

- ils peuvent être entraînés sur un large ensemble de paires <texte, parole> avec un minimum d'annotations humaines
- ils ne nécessitent pas d'alignement au niveau des phonèmes
- les erreurs ne peuvent pas s'accumuler puisqu'il s'agit d'un modèle unique.

Synthèse vocale basée sur WaveNet WaveNet [29] est un puissant modèle génératif d'ondes audio brutes, dérivé des modèles PixelRNN [68] utilisés dans le domaine de la génération d'images. Proposé par DeepMind en 2016, il ouvre la voie à la synthèse vocale de bout en bout. WaveNet est capable de générer des voix humaines relativement réalistes en modélisant directement les formes d'ondes à l'aide d'un modèle de réseau de neurones profonds (DNN) entraîné avec des enregistrements de discours réel. Il s'agit d'un modèle autoregressif probabiliste complet qui prédit la distribution de probabilité de l'échantillon audio actuel en se basant sur

tous les échantillons générés auparavant. Un composant essentiel de WaveNet est l'utilisation de convolutions causales dilatées, qui garantissent que WaveNet ne peut utiliser que les points d'échantillonnage de 0 à $t-1$ lors de la génération du point d'échantillonnage t.

2.3.4 Synchronisation labiale (*Lip synchronization])*

Un *deepfake* de synchronisation labiale est une vidéo manipulée numériquement dans laquelle les mouvements des lèvres d'une personne sont créés de manière convaincante à l'aide de modèles d'intelligence artificielle pour correspondre à un audio modifié ou entièrement nouveau (voir figure 2.13). Les *deepfakes* de synchronisation labiale sont particulièrement dangereux car les artefacts sont limités à la région des lèvres, ce qui les rend plus difficiles à détecter.
En général, les systèmes modernes de génération de synchronisation labiale visent à atteindre les objectifs suivants :

- **Synchronisation généralisée audio-lèvres** : Étant donné que les gens sont sensibles au moindre décalage entre les mouvements du visage et le son de la parole, il est essentiel de maintenir la précision des mouvements des lèvres et la fluidité temporelle des mouvements faciaux prédits.
- **Bonne qualité vidéo** : Une bonne qualité vidéo globale se caractérise généralement par une haute fidélité d'image, une transition fluide entre les images adjacentes, et un réalisme convaincant.

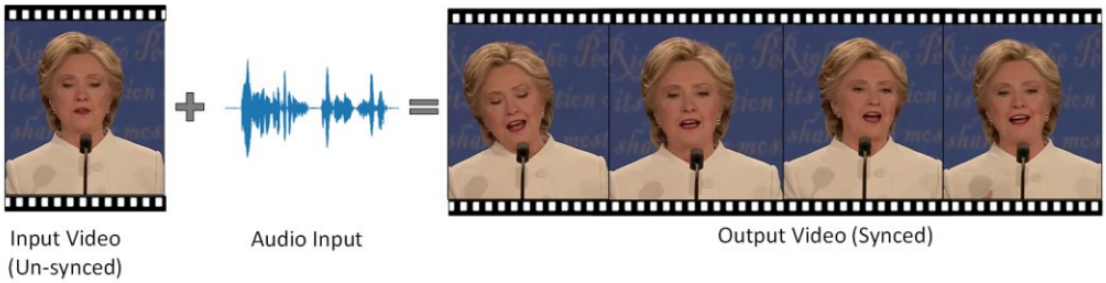


Fig. 2.13 : Une représentation visuelle de la synchronisation labiale d'une vidéo existante avec un extrait audio arbitraire [63].

2.3.4.1 Synchronisation labiale basé sur les GAN

Les manipulations basées sur les GAN, telles que wav2lip[69], utilisent un modèle indépendant du locuteur qui peut synchroniser avec précision les mouvements des lèvres dans un enregistrement vidéo avec un clip audio donné. Cette approche emploie un discriminateur de synchronisation labiale pré-entraîné qui est ensuite affiné sur des vidéos générées bruyantes en l'absence d'un générateur. Ce modèle utilise plusieurs images consécutives au lieu d'une seule dans le discriminateur et emploie une perte de qualité visuelle ainsi qu'une perte contrastive, augmentant ainsi la qualité visuelle en prenant en compte la corrélation temporelle.

Les approches récentes peuvent synthétiser des vidéos à partir de la parole (audio-vidéo) ou du texte (texte-vidéo) avec des résultats vidéo convaincants. Ces approches se concentrent principalement sur la synchronisation des mouvements des lèvres en synthétisant la région autour de la bouche.

2.3.4.2 Synchronisation labiale basé sur Neural Radiance Field (NeRF)

Avec le développement des techniques de Neural Radiance telles que NeRF [70] , il est possible de construire un système de génération de visage parlant en 3D de haute fidélité à partir d'une vidéo de quelques minutes de la personne cible.

Nerf : est une technique utilisée pour reconstruire une représentation tridimensionnelle d'une scène à partir d'images bidimensionnelles dispersées. La synthèse de nouvelles vues, la géométrie de la scène et les propriétés de réflectance de la scène peuvent être appréhendées grâce au modèle NeRF. Il est également possible d'acquérir des caractéristiques supplémentaires de la scène, comme les poses de la caméra, en collaboration. NeRF offre la possibilité de créer des images photoréalistes depuis de nouveaux points de vue. Initialement présenté en 2020, il a suscité une grande attention pour ses potentielles applications dans les domaines des graphiques informatiques et de la création de contenu. Une scène est représentée par l'algorithme NeRF comme un champ de radiance configuré par un réseau de neurones profonds (DNN). Le réseau prédit une densité volumique et une radiance émise dépendante de la vue, étant donné la position spatiale (x, y, z) et

la direction de vue en angles d'Euler (θ, Φ) de la caméra. En échantillonnant de nombreux points le long des rayons de la caméra, les techniques traditionnelles de rendu volumique peuvent produire une image [70].

2.4 Conclusion

Dans ce chapitre, nous avons expliqué les techniques de base pour créer des *deepfakes* et présenté les principaux algorithmes nécessaires pour les générer.
Ce chapitre fournit une vue d'ensemble sur les principaux algorithmes de génération de *deepfakes*, en soulignant l'importance de développer des méthodes plus avancées pour produire des *deepfakes* plus réalistes.

Chapitre 3

Approche et protocole d'expérimentation

3.1 Introduction

Durant la phase d'expérimentation, nous avons pu tester plusieurs méthodes de génération de *deepfakes* audio et visuels, incluant le remplacement de visage et la synchronisation labiale. L'objectif était de trouver les meilleurs outils pour créer les *deepfakes* les plus réalistes possibles.
Ces tests nous ont permis d'identifier les technologies les plus performantes pour produire des *deepfakes* de haute qualité, en tenant compte de la fluidité des mouvements, de la précision des expressions faciales et de la synchronisation audio-visuelle.

3.2 Les outils les plus récents pour la génération des *deepfakes*

3.2.1 DeepFaceLab (2021)

Actuellement DeepFaceLab (DFL) [18] est une outil de *deepfake* pour l'échange de visages. Il fournit les outils nécessaires dans un environnement facile à utiliser permettant de réaliser des échanges de visages de haute qualité. Il offre également une structure flexible.

3.2.1.1 DeepFaceLab pipeline

DeepFaceLab fournit un ensemble de flux de travail qui forment un pipeline flexible. Dans DFL, nous pouvons abstraire le pipeline en trois phases : extraction, entraînement et conversion.
Ces trois parties sont présentées de manière séquentielle. Il est également à noter que DFL s'inscrit dans un paradigme typique d'échange de visages un-à-un, ce qui signifie qu'il n'y a que deux types de données : *src* et *dst*, abréviations pour source et destination.

Extraction : La phase d'extraction est la première phase de DFL, visant à extraire un visage à partir des données source et destination. Cette phase se compose de nombreux algorithmes et parties de traitement, à savoir la détection de visage, l'alignement de visage et la segmentation de visage (voir figure 3.1). DFL offre de nombreux modes d'extraction (c'est-à-dire, demi-visage, visage complet, visage entier), qui représentent la zone de couverture du visage de la phase d'extraction. En général, nous prenons

par défaut le mode visage complet.

- **Détection de visage** : la première étape de la phase d'extraction consiste à trouver le visage cible dans les données fournies : *src* et *dst*. DFL utilise S3FD [57] comme détecteur de visage par défaut.
- **l'alignement du visage** : DFL propose deux types canoniques d'algorithmes d'extraction de points de repère faciaux :
  - l'algorithme de points de repère faciaux 2DFAN (pour les visages avec une pose standard).
  - PRNet avec des informations préalables de visage 3D (pour les visages avec un grand angle d'Euler).
- **Segmentation de visage** : après l'alignement du visage, un dossier de données avec un visage en poses frontal et de profil (*src* alignée ou *dst* alignée) est obtenu.
  DFL utilise un réseau de segmentation de visage à grain fin (TernausNet) [62] sur le visage aligné (*src* ou *dst* aligné), grâce auquel un visage avec des cheveux, des doigts ou des lunettes peut être segmenté avec précision.

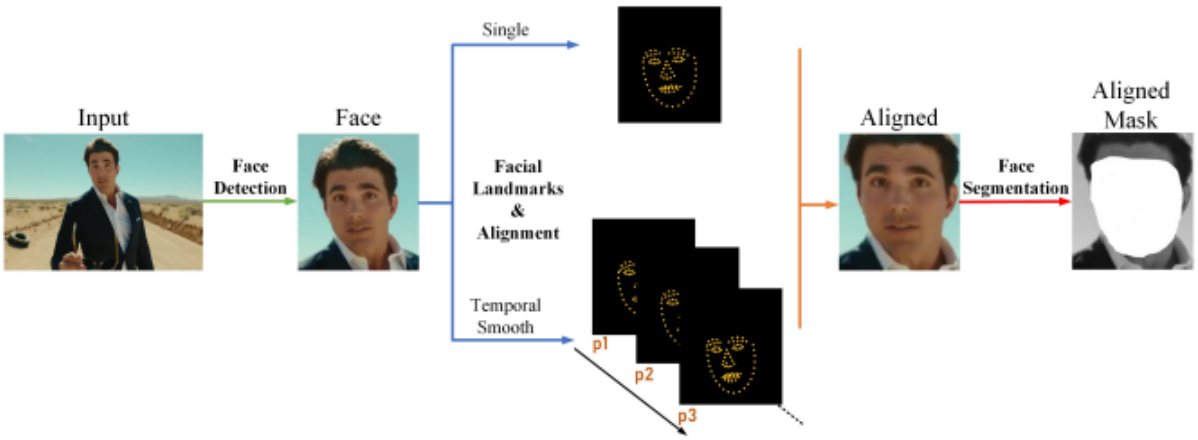


FIG. 3.1 : Aperçu de la phase d'extraction dans DeepFaceLab [18].

Entraînement : La phase d'entraînement joue un rôle crucial dans l'obtention de résultats photoréalistes pour l'échange de visages avec DFL. DFL propose deux structures, la structure DF et la structure LIAE :
Comme montré dans la figure 3.2, la structure DF se compose d'un Encodeur ainsi que d'un *Inter* avec des poids partagés entre *src* et *dst*, et de

deux décodeurs appartenant respectivement à *src* et *dst*. La généralisation de *src* et *dst* est réalisée par l'encodeur et l'*Inter* partagés. La structure DF peut accomplir la tâche d'échange de visages mais ne peut pas hériter suffisamment d'informations de *dst*, telles que l'éclairage.

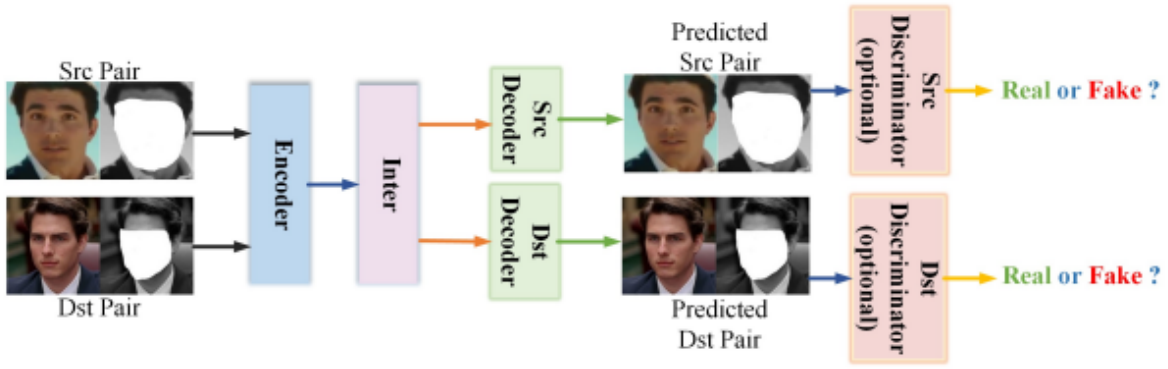


Fig. 3.2 : Structure de DF [18] .

Pour améliorer davantage la de cohérence de l'éclairage, DFL propose LIAE. Comme illustré dans la figure 3.3, la structure LIAE est une structure plus complexe avec un encodeur, un décodeur et deux *Inters* indépendants avec des poids partagés. La principale différence par rapport à la DF est qu'*InterAB* est utilisé pour générer les codes latents de *src* et de *dst*, tandis qu'*InterB* ne produit que le code latent de *dst*.

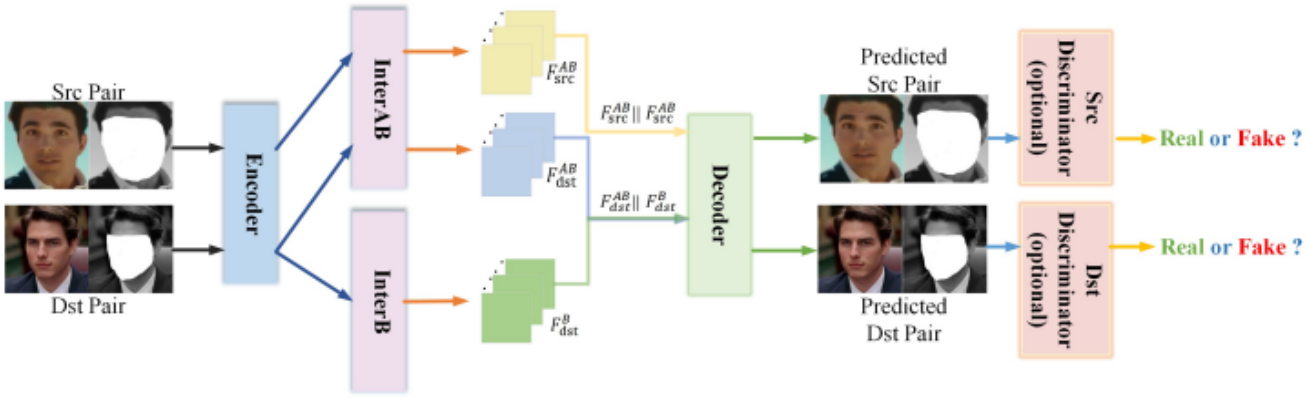


Fig. 3.3 : Structure de LIAE [18].

Conversion : La phase de conversion est la dernière et comme illustré dans la Figure 3.4, les utilisateurs peuvent échanger les visages de *src* à *dst* et vice versa.
Dans le cas de *src to dst*, la première étape du schéma proposé d'échange de visages dans la phase de conversion est de transformer le visage généré ainsi que son masque du décodeur *dst* à la position d'origine de l'image cible dans *src* grâce à la réversibilité de Umeyama [60].

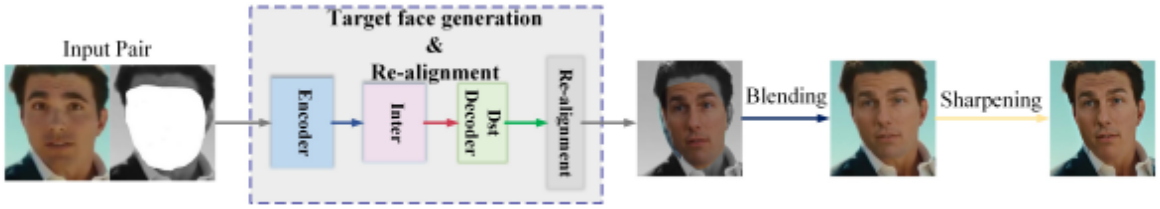


FIG. 3.4 : Aperçu de la phase de conversion dans DeepFaceLab [18].

3.2.2 Wav2Lip (2020)

Wav2Lip [69] est une méthode de synchronisation labiale qui est :

- **Personne générique** : En utilisant un cadre de référence et en s'entraînant sur un large ensemble de données, Wav2Lip peut fonctionner sur n'importe quelle vidéo et n'importe quel audio.
- **Basé sur la 2D** : Wav2Lip fonctionne directement au niveau de l'image, ce qui entraîne généralement des résultats de qualité inférieure. Cependant, cela facilite la généralisation du modèle pour qu'il soit applicable à différentes personnes.
- **Seulement les lèvres** : Wav2Lip ne remplace que la moitié inférieure du visage, en prenant la moitié supérieure de la vidéo fournie.

3.2.2.1 Architecture

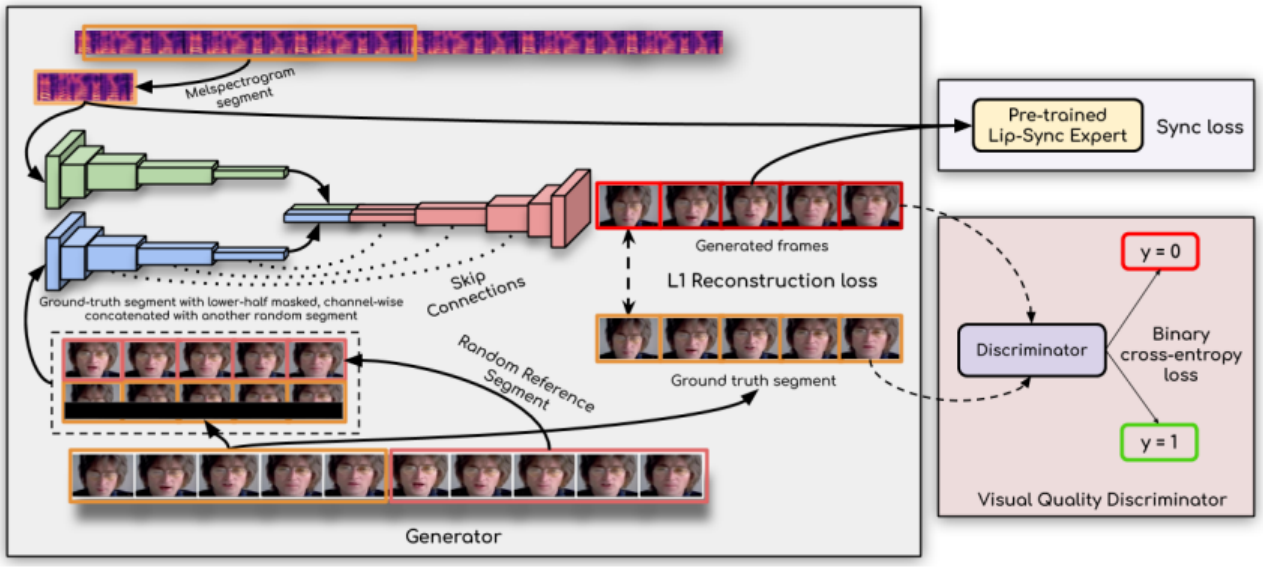


FIG. 3.5 : Le diagramme d'architecture de Wav2Lip. Des encodeurs audio (vert) et vidéo (bleu) distincts convertissent leurs entrées respectives en un espace latent, tandis qu'un décodeur (rouge) est utilisé pour générer les vidéos [69].

Un modèle spécifique à une personne pourrait simplement masquer la moitié inférieure du visage et apprendre une correspondance pour prédire cette moitié inférieure à partir du signal audio, car il aurait vu de nombreux exemples des lèvres de cette personne. Cependant, un modèle générique comme Wav2Lip n'a aucune idée de l'apparence des lèvres, des dents, de la langue ou de toute autre partie de la moitié inférieure du visage de cette personne. Ainsi, en plus de la moitié supérieure du visage, Wav2Lip prend également une autre image de référence aléatoire en entrée. Cela permet d'intégrer au réseau un certain niveau d'information sur la moitié inférieure du visage de la personne.

Pour utiliser efficacement toutes les informations disponibles, Wav2lip [69] utilise une architecture d'autoencodeur. Un encodeur vidéo (en bleu dans le diagramme du réseau de la figure 3.5) prend la image actuelle, avec la moitié inférieure masquée, et une image de référence aléatoire du même vidéo et l'encode dans un espace latent. Il en va de même pour l'encodeur audio (en vert figure 3.5), qui fonctionne avec des MEL-spectrogrammes. Un décodeur (en rouge dans la figure 3.5) prend ensuite ces deux codes latents et produit la image finale.

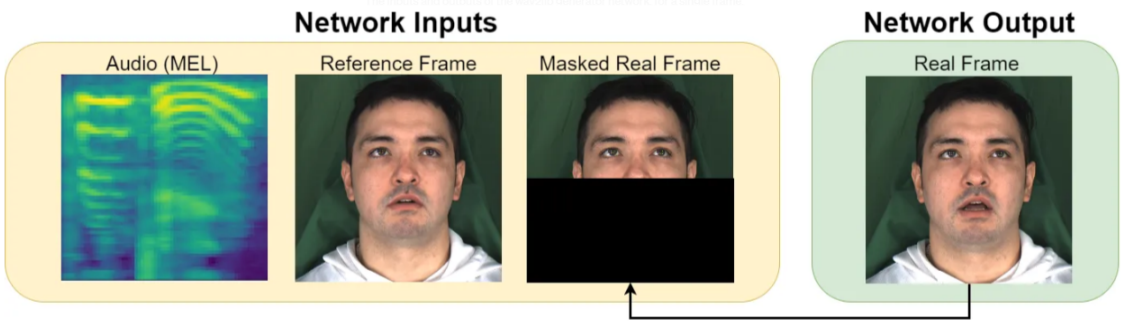


FIG. 3.6 : Les entrées et sorties du réseau générateur wav2lip, pour une seule image [71].

Wav2lip [69] tente de reconstruire complètement les images de vérité terrain à partir de leurs copies masquées (figure 3.6). Il calcule la perte de reconstruction L1 entre les images reconstruites et les images de vérité terrain. Ensuite, les images reconstruites sont passées à travers un détecteur de synchronisation labiale « expert » pré-entraîné, tandis que les images reconstruites et les images de vérité terrain sont passées à travers le Discriminateur de Qualité Visuelle. Le Discriminateur de Qualité Visuelle tente de distinguer les images reconstruites des images de vérité terrain afin de promouvoir la qualité visuelle du générateur de images.

3.2.2.2 Fonction de perte

Le générateur :

Le générateur vise à minimiser la perte L1 entre les images reconstruites L_g et les images de vérité terrain L_G,

$$L_{\text{recon}} = \frac{1}{N} \sum_{i=1}^{N} \|L_g - L_G\|_1 \tag{3.1}$$

où $\boldsymbol{N}$ est la notation généralement acceptée pour désigner la taille du lot.

Le Discriminateur de Synchronisation Labiale :

Pour la synchronisation labiale, ils utilisent la similarité cosinus avec une perte de cross-entropy binaire, ce qui permet de calculer la probabilité qu'une paire d'images données soit synchronisée. Plus précisément, la perte est calculée entre les embeddings vidéo et audio activés par $ReLU$,v et s. Cela produit une liste de probabilités, une pour chaque échantillon, indiquant la probabilité que l'échantillon correspondant soit synchronisé.

$$P_{sync} = \frac{vs}{\max\left(\|v\|_2 \|s\|_2, \epsilon\right)} \tag{3.2}$$

où l'activation $ReLU$ appliquée peut être décrite comme suit :

$$ReLU(x) = \begin{cases} 0 & \text{if } x \leq \epsilon \\ x & \text{otherwise} \end{cases} \tag{3.3}$$

La perte complète du discriminateur expert est calculée en prenant la cross-entropy de la distribution P_{sync} comme suit :

$$E_{\text{sync}} = \frac{1}{N} \sum_{i=1}^{N} -\log\left(P_{\text{sync}}^{i}\right) \tag{3.4}$$

Le Discriminateur de Qualité Visuelle :
est entraîné à maximiser la perte suivante :

$$L_{\text{disc}} = \mathbb{E}_{x \sim L_C}[\log(D(x))] + L_{gcn} \tag{3.5}$$

où la perte du générateur L_{gen} est formulée comme suit :

$$L_{gcn} = \mathbb{E}_{x \sim L_g} \mid \log(1 - D(x))\big] \tag{3.6}$$

En conséquence, le générateur tente de minimiser la somme pondérée de la perte de reconstruction, de la perte de synchronisation et de la perte adversariale (rappelons que nous avons affaire à deux discriminateurs) :

$$L_{\text{total}} = (1 - s_w - s_g) \cdot L_{\text{recon}} + s_w * E_{\text{symc}} + s_g * L_{gen} \tag{3.7}$$

où s_w est une valeur de pondération indiquant la pénalité attribuée à la synchronisation, et s_g est la perte adversariale.

Ces deux discriminateurs distincts permettent au réseau d'atteindre une précision de synchronisation et une qualité de génération visuelle supérieures.

3.2.3 DINet (2023)

DINet [72] est un outil permettant de synchroniser les mouvements des lèvres avec l'audio.
Les détails structurels de DINet sont présentés dans la Figure 3.7.

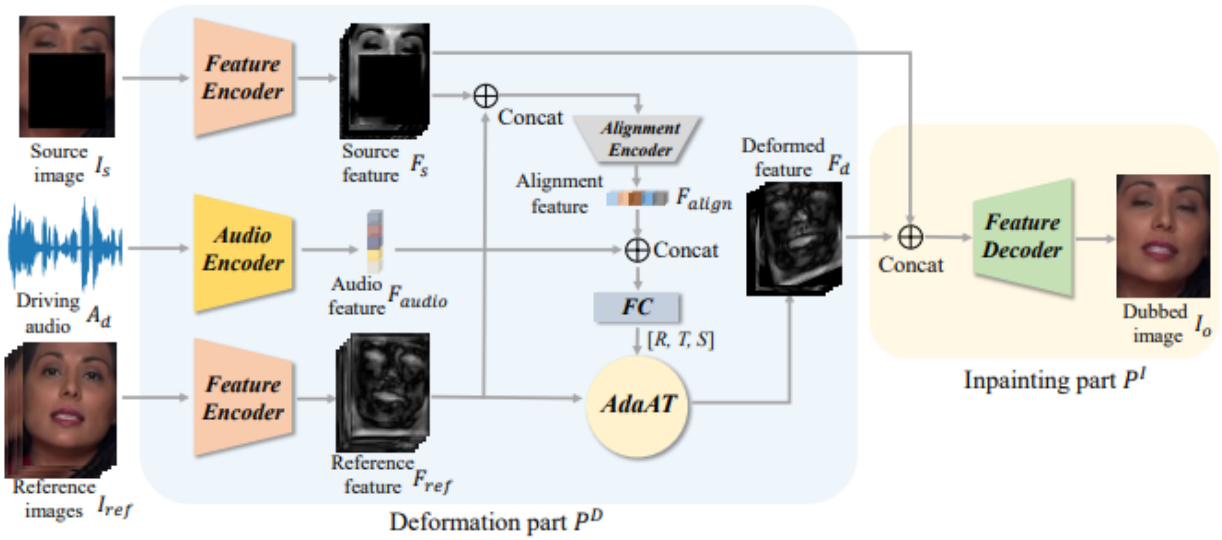


FIG. 3.7 : Illustration de framework DINet. DINet se compose d'une partie de déformation P_D et d'une partie d'inpainting P_I [72].

DINet se compose de deux parties une partie de déformation P_D et une partie de retouche P_I :

- **La partie de déformation** P_D se concentre sur la déformation spatiale des cartes de caractéristiques des images de référence afin de synchroniser la forme de la bouche avec l'audio pilote et d'aligner la position de la tête avec l'image source.
- **La partie de retouche** P_I vise à utiliser les résultats déformés pour réparer la région de la bouche sur le visage source.

3.2.3.1 La partie de déformation

Le rectangle bleu dans la Figure 3.7 illustre la structure de P_D.
Étant donné une image source $I_s \in \mathbb{R}^{3\times H\times W}$, un audio de conduite $A_d \in \mathbb{R}^{T\times 29}$ (29 est une dimension de la caractéristique audio, utilisons la caractéristique de deepspeech [73] et cinq images de référence $I_{ref} \in \mathbb{R}^{15\times H\times W}$, P_D vise à produire des caractéristiques déformées $F_d \in \mathbb{R}^{256\times \frac{H}{4}\times \frac{W}{4}}$ qui ont une forme de bouche synchrone avec A_d et une pose de tête alignée avec

I_s.
Pour réaliser cet objectif, A_d est d'abord entré dans un encodeur audio pour extraire la caractéristique audio $F_{audio} \in \mathbb{R}^{128}$. F_{audio} encode le contenu vocal de A_d. Ensuite, I_s et I_{ref} sont entrés dans deux encodeurs de caractéristiques différents pour extraire la caractéristique source $F_s \in \mathbb{R}^{256\times\frac{H}{4}\times\frac{W}{4}}$ et la caractéristique de référence $F_{ref} \in \mathbb{R}^{256\times\frac{H}{4}\times\frac{W}{4}}$. Ensuite, F_s et F_{ref} sont concaténés et entrés dans un encodeur d'alignement pour calculer la caractéristique d'alignement $F_{align} \in \mathbb{R}^{128}$. F_{align} encode les informations d'alignement de la pose de tête entre I_s et I_{ref}.
Enfin, F_{audio} et F_{align} sont utilisés pour déformer spatialement F_{ref} en F_d.

3.2.3.2 Partie d'inpainting

Le rectangle jaune dans la Figure 3.7 illustre la structure de P_I.
P_I vise à produire une image doublée $I_o \in \mathbb{R}^{3\times H\times W}$ à partir de F_s et F_d.
Pour réaliser cet objectif, F_s et F_d sont d'abord concaténés dans le canal de caractéristiques. Ensuite, un décodeur de caractéristiques avec des couches convolutionnelles est utilisé pour restaurer la bouche masquée et générer I_o.

3.2.4 GeneFace (2023)

GeneFace [74] est un système de génération de visages parlants.
Le champ de radiance neuronale (NeRF) a été exploré dans la génération de visages parlants [70]. Comparé aux techniques de rendu basées sur GAN, les générateurs NeRF peuvent préserver plus de détails et offrir une meilleure naturalité en 3D car ils modélisent une scène 3D continue dans l'espace caché.
GeneFace ce décompose en trois partie :

- ***Audio-to-motion*** : GeneFace propose un générateur de mouvement variationnel pour générer des points de repère faciaux précis et expressifs à partir de l'audio en entrée.

- **Adaptation de domaine pour le mouvement** : Pour surmonter le décalage de domaine, GeneFace propose un pipeline d'entraînement semi-supervisé adversarial pour former un post-réseau adaptatif au domaine, qui affine les points de repère 3D prédits du domaine multi-locuteur vers le domaine de la personne cible.

- **Mouvement vers image** : GeneFace propose un render basé sur NeRF pour générer des images haute-fidélité conditionnées aux points de repère 3D prédits.

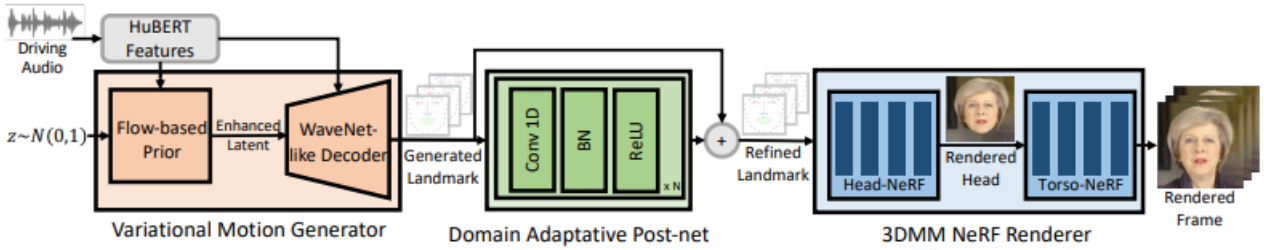


FIG. 3.8 : Le processus d'inférence de GeneFace. BN désigne la normalisation par batch [74].

GeneFace se compose de trois parties (figure 3.8) :
1) Un générateur de mouvement variationnel qui transforme les caractéristiques HuBERT [75] en points de repère faciaux 3D ;
2) Un post-réseau pour affiner le mouvement généré dans le domaine de la personne cible ;
3) Un générateur basé sur NeRF pour synthétiser des images haute-fidélité.

3.2.4.1 Générateur de mouvement variationnel (*Audio-to-motion*)

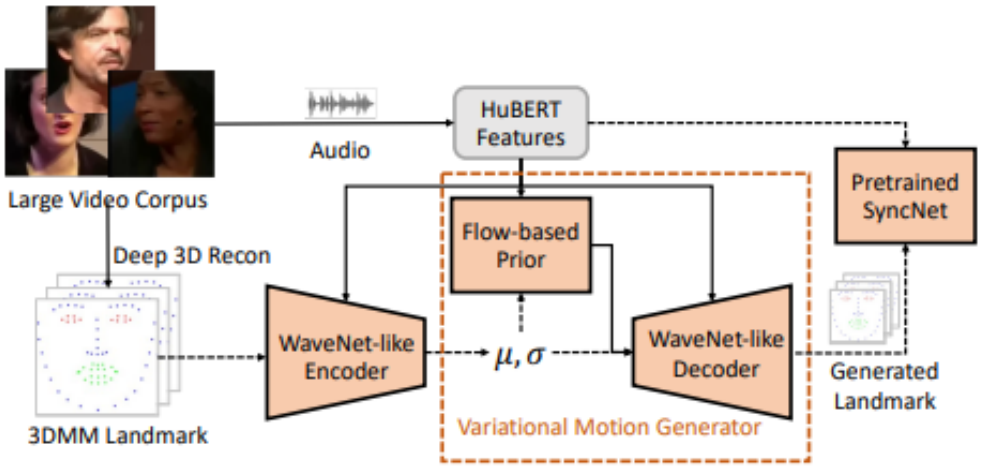


FIG. 3.9 : La structure du générateur de mouvement variationnel. Les flèches en pointillés signifient que le processus est uniquement effectué pendant l'entraînement ; et seule la partie du rectangle en pointillés est utilisée pendant l'inférence [74].

Pour réaliser une génération de mouvement de tête 3D expressive et diversifiée, GeneFace présente un autoencodeur variationnel (VAE) pour

effectuer une transformation générative de l'audio vers le mouvement, nommément le générateur de mouvement variationnel, comme illustré à la Figure 3.9.

Pour mieux extraire l'information sémantique, GeneFace utilise HuBERT [75], un modèle ASR (*Automatic Speech Recognition*) de l'état de l'art , pour obtenir des caractéristiques audio à partir du signal d'entrée et les utilisons comme condition pour le générateur de mouvement variationnel. En ce qui concerne la représentation du mouvement, pour représenter les mouvements faciaux détaillés dans l'espace euclidien, GeneFace sélectionne 68 points clés à partir du maillage 3D reconstruit de la tête et utilisons leur position comme représentations des actions (3D *Landmarks*).

3.2.4.2 Adaptation de domaine pour le mouvement

Comme nous entraînons le générateur de mouvement variationnel sur un large ensemble de données multi-locuteurs, le modèle peut bien se généraliser avec divers entrées audio. Cependant, comme la taille de la vidéo de la personne cible est relativement petite (environ 4-5 minutes) par rapport à l'ensemble de données de lecture labiale multi-locuteurs (environ des centaines d'heures), il existe un décalage de domaine entre les points de repère 3D prédits et le domaine de la personne cible. En conséquence, le générateur basé sur NeRF ne peut pas bien se généraliser avec les points de repère prédits, ce qui entraîne des images rendues floues ou non réalistes. À cette fin, GeneFace propose une solution qui consiste à affiner le générateur variationnel dans l'ensemble de données de la personne cible.

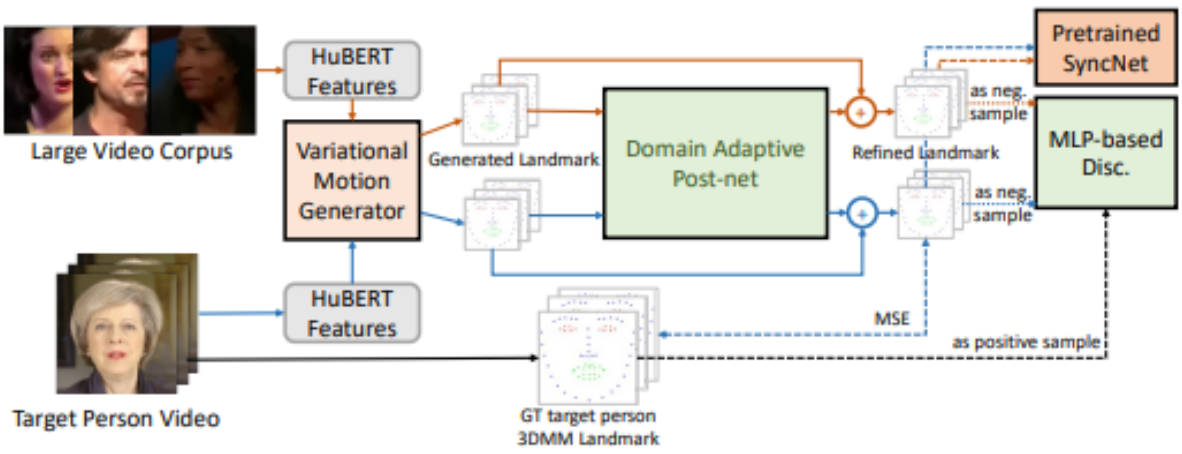


Fig. 3.10 : Le processus d'entraînement de Domain Adaptative Post-net [74].

Dans de telles circonstances, GeneFace propose un pipeline d'entraînement semi-supervisé adversarial pour effectuer une adaptation de domaine (figure 3.10). Plus précisément, un post-net pour affiner les points de repère 3D prédits par le VAE dans le domaine personnalisé. avec ces deux exigences pour ce mapping :

- Il doit préserver la cohérence temporelle et la synchronisation labiale de la séquence d'entrée en utilisons un CNN 1D comme structure du post-net et adoptons l'expert en synchronisation pour superviser la synchronisation labiale.
- Il doit correctement mapper chaque image dans le domaine de la personne cible en entraînons conjointement un discriminateur de niveau image structuré en MLP (Perceptron multicouche) qui mesure la similarité d'identité de chaque image de points de repère avec la personne cible.

3.2.4.3 Moteur de rendu basé sur NeRF

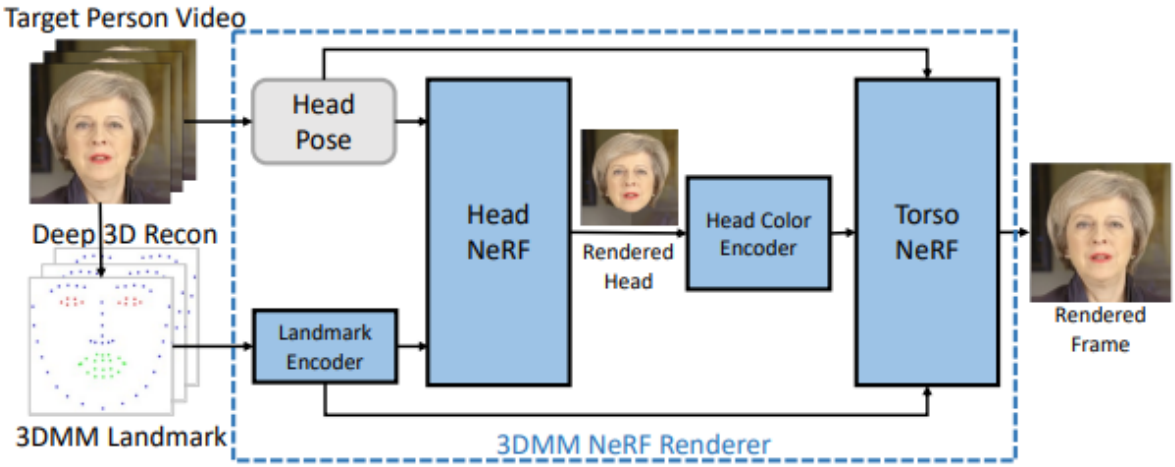


Fig. 3.11 : Le processus d'entraînement du rendu basé sur NeRF [74].

Conditionné par les *Landmarks* 3D NeRF Inspiré, GeneFace présente un NeRF conditionnel pour représenter la tête parlante dynamique. En plus de la direction de vue d et de la position 3D x, les *Landmarks* 3D l agiront comme condition pour manipuler la couleur et la géométrie de la tête représentée implicitement. Plus précisément, la fonction implicite F

peut être formulée comme suit :

$$F_\theta : (x, d, l) \rightarrow (c, \sigma) \tag{3.8}$$

où c et σ représentent la couleur et la densité dans le champ de radiance. Pour améliorer la continuité entre les images adjacentes, nous utilisons les *Landmarks* 3D des trois images voisines pour représenter la forme faciale. Pour mieux modéliser le mouvement de la tête et du torse, nous entraînons deux NeRFs pour rendre respectivement les parties de la tête et du torse. Comme illustré à la figure 3.11, nous commençons par entraîner un NeRF pour la tête afin de rendre la partie tête, puis nous entraînons un NeRF pour le torse afin de rendre la partie torse avec l'image de rendu du NeRF pour la tête comme arrière-plan. GeneFace suppose que la partie torse est dans l'espace canonique et fournissons la pose de la tête h au NeRF pour le torse en tant que signal pour inférer le mouvement du torse.

Nous extrayons des repères 3D à partir des images vidéo et utilisons ces paires image-repère pour entraîner notre moteur de rendu basé sur NeRF. L'objectif d'optimisation de head-NeRF et torso-NeRF est de réduire l'erreur de reconstruction photométrique entre les images rendues et les images de référence. Plus précisément, la fonction de perte peut être formulée comme suit :

$$L_{\text{NeRF}}(\theta) = \sum_{r \in R} \| C_\theta(r, l) - C_g \|_2^2 \tag{3.9}$$

où R est l'ensemble des rayons de la caméra, C_g est la couleur de l'image de référence.

3.2.5 Geneface++ (2024)

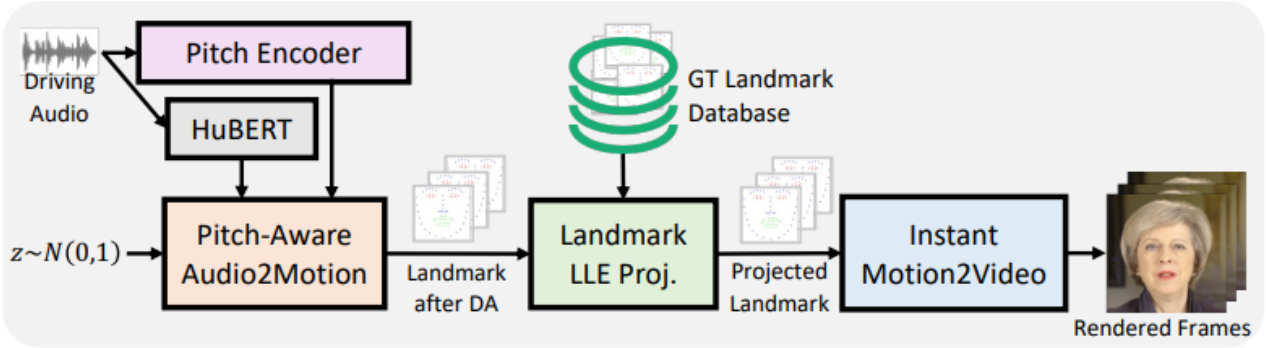
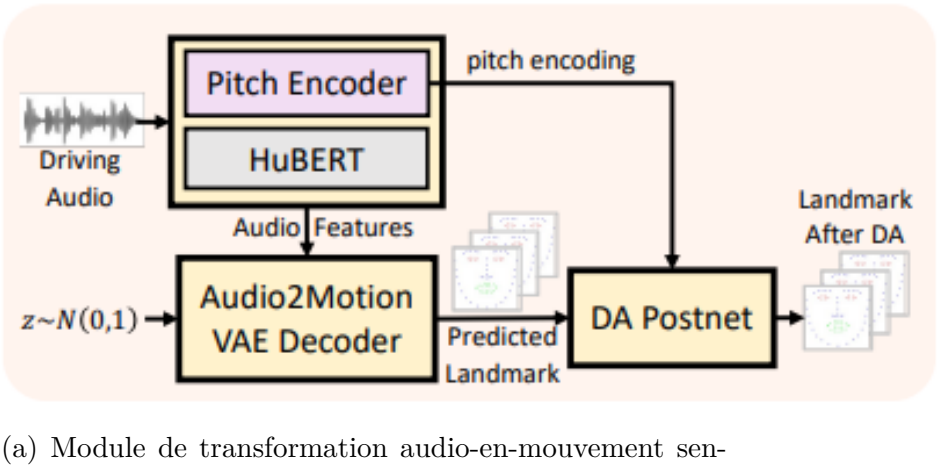


FIG. 3.12 : Pipeline global de GeneFace++ [76].

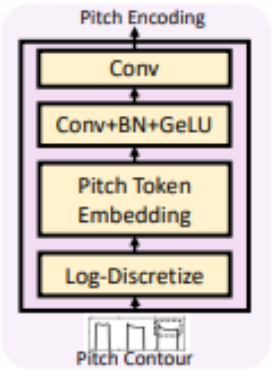
GeneFace++ [76], qui vise à améliorer GeneFace pour obtenir une synchronisation audio-labiale plus naturelle, une qualité vidéo plus robuste et une plus grande efficacité du système. Comme le montre la figure 3.12), GeneFace++ est composé de trois parties :
1) un module audio-à-mouvement sensible à la hauteur (pitch) qui transforme les caractéristiques audio en mouvements faciaux ;
2) une méthode d'incorporation linéaire localisée des points de repère pour post-traiter le mouvement prédit ;
3) un module de mouvement-à-vidéo instantané qui peut rendre efficacement la vidéo finale du visage parlant.

3.2.5.1 Transformation Audio-à-Mouvement Sensible à la Hauteur(pitch)

La motivation pour prendre en compte l'information de la hauteur dans la cartographie audio-à-mouvement est que la hauteur est connue pour être fortement corrélée aux expressions faciales. Par exemple, un contour de hauteur élevé et stable peut correspondre à un mouvement des lèvres large et stable (figure 3.13).



(a) Module de transformation audio-en-mouvement sensible à la hauteur [76].



(b) Encodeur de hauteur [76].

FIG. 3.13

3.2.5.2 Localisation linéaire des points de repère(*Landmark* LLE)

GeneFace++ propose l'incorporation linéaire localisée des points de repère (*Landmark* LLE) (figure 3.14), une méthode de post-traitement basée sur la projection de variétés qui garantit que chaque point de repère prédit est correctement mappé dans (ou à proximité de) l'espace d'entrée du moteur de rendu conditionné par les points de repère. En d'autres termes,

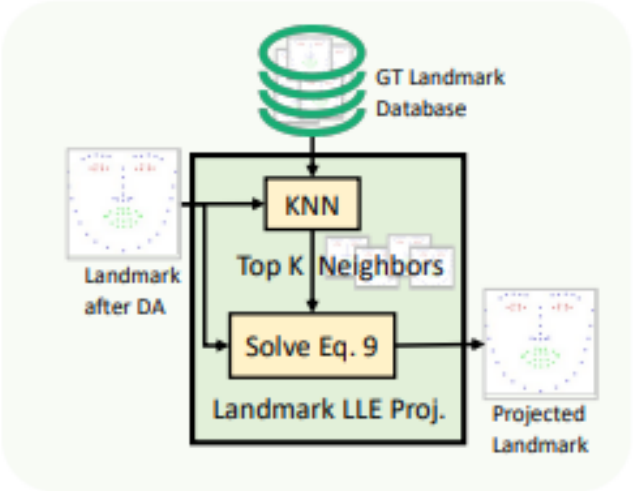


FIG. 3.14 : Une représentation visuelle de la synchronisation labiale d'une vidéo existante avec un extrait audio arbitraire [76].

avec l'aide de *Landmark* LLE, chaque point de repère prédit est rapproché de l'ensemble des points de repère GT (*Ground Truth*) utilisés pour entraîner le moteur de rendu.

3.2.5.3 Module de mouvement-à-vidéo

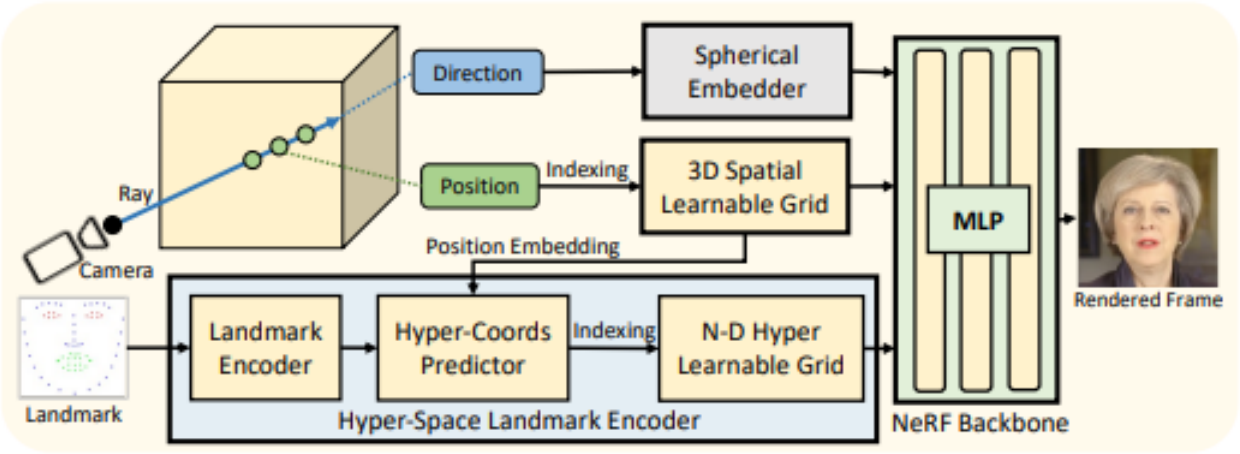


FIG. 3.15 : Module instantané de conversion de mouvement en vidéo [76].

Puisque nous obtenons une cartographie audio-à-mouvement expressive, cohérente dans le temps et robuste grâce au module audio-à-mouvement sensible à la hauteur et à la méthode de post-traitement *Landmark* LLE. GeneFace++ utilise un moteur de rendu NeRF basé sur une grille pour projeter le point de repère facial d'entrée dans une coordonnée ambiante N-dimensionnelle conditionnée par les caractéristiques spatiales basées sur la grille, ce qui permet une fusion efficace de l'information spatiale et de la

condition des points de repère (figure 3.15).
Une fois la coordonnée ambiante obtenue, au lieu d'interroger les caractéristiques des points de repère avec un MLP (Perceptron multicouche) dense, nous utilisons une grille N-D supplémentaire apprenable pour améliorer l'efficacité. Empiriquement, nous avons fixé $N = 3$ via une recherche par grille, comme compromis entre performance et efficacité. Les caractéristiques spatiales et les caractéristiques des points de repère interrogées sont concaténées et alimentées dans le backbone NeRF (un MLP peu profond) pour générer la densité et la couleur. Plus précisément, la fonction implicite peut être formulée comme suit :

$$F : (f_x, f_l, d) \rightarrow c, \sigma \tag{3.10}$$

où f_x et f_l sont respectivement les caractéristiques spatiales/points de repère interrogées à partir de la grille. d est la direction de vue.

3.2.6 Retrieval Based Voice Conversion (RVC)

La Conversion de Voix Basée sur la Recherche (ci-après dénommée RVC) est un nouveau type d'outil open-source de conversion de voix développé en juin 2023 et il est librement accessible sur GitHub [77] sous la licence logicielle MIT.
Comparé aux programmes traditionnels d'imitation de timbre par entraînement, la Conversion de Voix Basée sur la Recherche fonctionne en remplaçant les caractéristiques de la source d'entrée par les caractéristiques du jeu de données d'entraînement à l'aide de la récupération top1. En même temps, RVC présente les caractéristiques suivantes :

- Même sur des cartes graphiques relativement peu performantes, l'entraînement peut être rapide.
- Un entraînement avec une petite quantité de données peut également donner de meilleurs résultats (il est recommandé de collecter au moins 10 minutes de données vocales à faible bruit).

3.2.6.1 Principe de fonctionnement de RVC

Les techniques traditionnelles de conversion de voix reposent généralement sur des modèles statistiques, tels que les modèles de Markov cachés (HMM) [78] , pour apprendre la correspondance entre les voix source et

cible. Ces modèles sont entraînés sur un large corpus de données parallèles, composé de paires d'enregistrements audio des locuteurs source et cible. RVC, en revanche, ne nécessite pas de données parallèles. Au lieu de cela, il utilise une approche basée sur la recherche pour trouver les segments audio les plus similaires dans la voix du locuteur cible, puis utilise ces segments pour synthétiser la voix convertie. Cette approche est plus efficace et moins gourmande en données que les techniques traditionnelles de conversion de voix.

3.2.6.2 Processus de RVC :

Dans la conversion de voix basée sur la recherche (RVC), le modèle apprend à cloner une voix en récupérant et en combinant des segments audio de la voix du locuteur cible. Ce processus comprend plusieurs étapes :

- **Pré-entraînement** : Un large corpus de données vocales provenant de divers locuteurs est utilisé pour entraîner un modèle de réseau neuronal. Ce modèle apprend à encoder la parole en une représentation de haute dimension qui capture les caractéristiques acoustiques de la voix de chaque locuteur.
- **Extraction de caractéristiques** : Le modèle extrait les caractéristiques acoustiques de la voix des locuteurs source et cible. Ces caractéristiques représentent les propriétés du signal de la parole, telles que la hauteur, l'intensité et le contenu spectral.
- **Embeddings de locuteurs** : Le modèle apprend des embeddings de locuteurs, qui sont des représentations vectorielles capturant les caractéristiques uniques de la voix de chaque locuteur. Ces embeddings sont utilisés pour identifier les segments audio les plus similaires dans la voix du locuteur cible.
- **Récupération de segments** : Pour chaque énoncé dans la voix source, le modèle récupère les segments audio les plus similaires dans la voix du locuteur cible. La similarité est déterminée en fonction des caractéristiques acoustiques et des embeddings de locuteurs.
- **Combinaison de segments** : Les segments récupérés sont ensuite combinés à l'aide d'une technique de synthèse de forme d'onde, telle que le recouvrement-addition ou le vocodeur de phase, pour générer la

voix convertie. Ce processus garantit que la voix convertie maintient l'intonation et le rythme globaux du locuteur source tout en adoptant les caractéristiques vocales du locuteur cible.

- **Ajustement «Fine-tuning»** : Le modèle est affiné en utilisant une fonction de perte qui mesure la similarité entre la voix convertie et la voix du locuteur cible. Ce processus d'ajustement fin contribue à améliorer la qualité de la voix convertie.

À travers ce processus, le modèle apprend à ajuster les caractéristiques acoustiques et les embeddings de locuteurs de la voix du locuteur source pour les faire correspondre à ceux du locuteur cible. Cela permet au modèle de cloner efficacement la voix du locuteur cible.

3.3 Protocole de génération des vidéos

3.3.1 *Faceswap*

3.3.1.1 DeepFaceLab

Pour générer des *deepfakes* avec DFL, on a utilisé un modèle pré-entraîné sur les visages de plusieurs personnes pour non seulement gagner du temps d'apprentissage mais aussi augmenter la capacité du modèle à généraliser et générer des visages plus réalistes. Ce modèle pré-entraîné peut ensuite être utilisé pour démarrer n'importe quel *deepfake* avec les mêmes paramètres de modèle de base.
DeepFaceLab inclut l'ensemble d'images faciales pré-entraîné Flickr Faces HQ (FFHQ)[79] avec un masque de visage complet générique déjà appliqué aux images alignées. Les versions antérieures de DFL utilisaient le dataset CelebA [80]. On peut trouver les images dans le dossier intitulé -internal/pretrain-faces sous forme de fichier faceset.pak.

Entraînement du modèle On a utilisé un modèle SAEHD de résolution 512, avec un masque de visage complet (WF), pré-entraîné pendant 485 583 époques d'apprentissage sur plusieurs visages. Ensuite, on a affiné l'entraînement sur nos vidéos pendant 30 000 époques d'apprentissage (11 heures d'entraînement) sur un GPU «RTX 3060» avec 12 GB de vidéo RAM.

3.3.2 Synthèse vocale

Pour faire une évaluation des outils de génération de *deepfake* audio expliqué dans la section précédente, nous avons généré des audios par 3 méthodes différents (TTS, RVC, TTS + RVC).
Pour chaque outil, nous avons utilisé 5 personnes pour les audios (Hafid Derraji, Éric Zemmour, Kylian Mbappé, Emmanuel Macron, Le Président Abdelmadjid Tebboune), ce qui permet de comparer les performances de chaque méthodes de manière cohérente.

3.3.2.1 XTTS

Pour faire la synthèse vocale nous avons utilisé XTTS.
XTTS est un système TTS de bout en bout(End to End) qui utilise un modèle VQ-VAE [81] pour discrétiser l'audio en jetons audio. Ensuite, il emploie un modèle GPT pour prédire ces jetons audio en fonction du texte d'entrée et des vecteurs latents de locuteur. Les vecteurs latents de locuteur sont calculés par un empilement de couches d'auto-attention. La sortie du modèle GPT est ensuite transmise à un modèle de décodage qui génère le signal audio.
Pour bien utiliser le modèle de XTTS, nous avons utilisé XTTS-webui [82] (figure 3.16). Ainsi, pour générer les audios, nous avons calculé les vecteurs latents des locuteurs, qui sont une représentation vectorielle capturant les caractéristiques acoustiques uniques de la voix d'un locuteur, encodant des informations sur la hauteur, le ton, le timbre et d'autres qualités vocales.

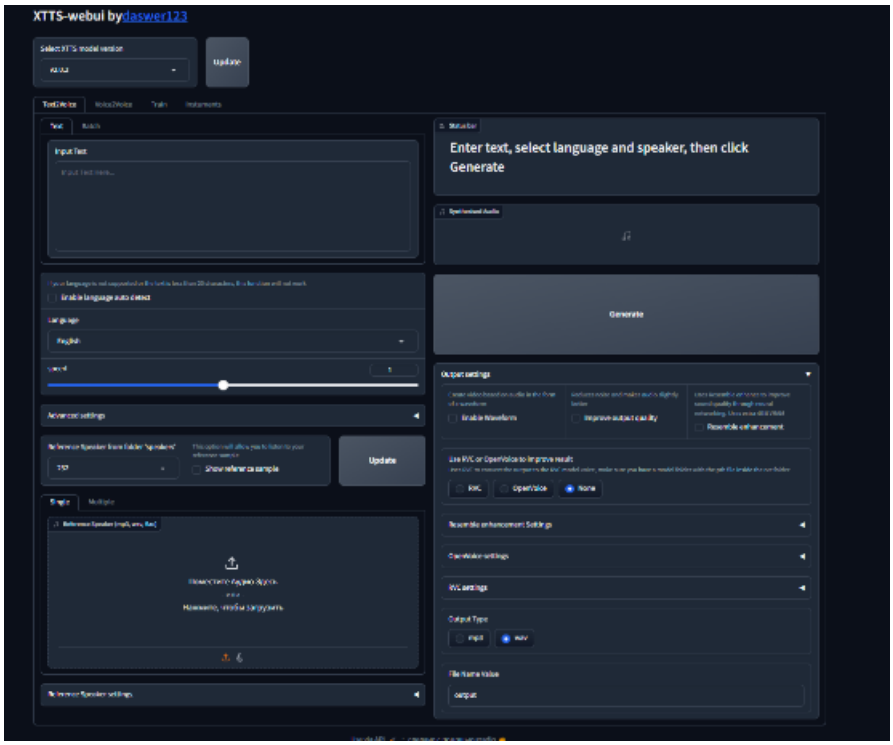


FIG. 3.16 : Interface web de XTTS-Webui [82].

3.3.2.2 RVC

RVC [77] est un outil open source pour la conversion voix-voix. Nous avons utilisé l'interface Applio [83] (figure 3.17) pour exploiter cet outil.

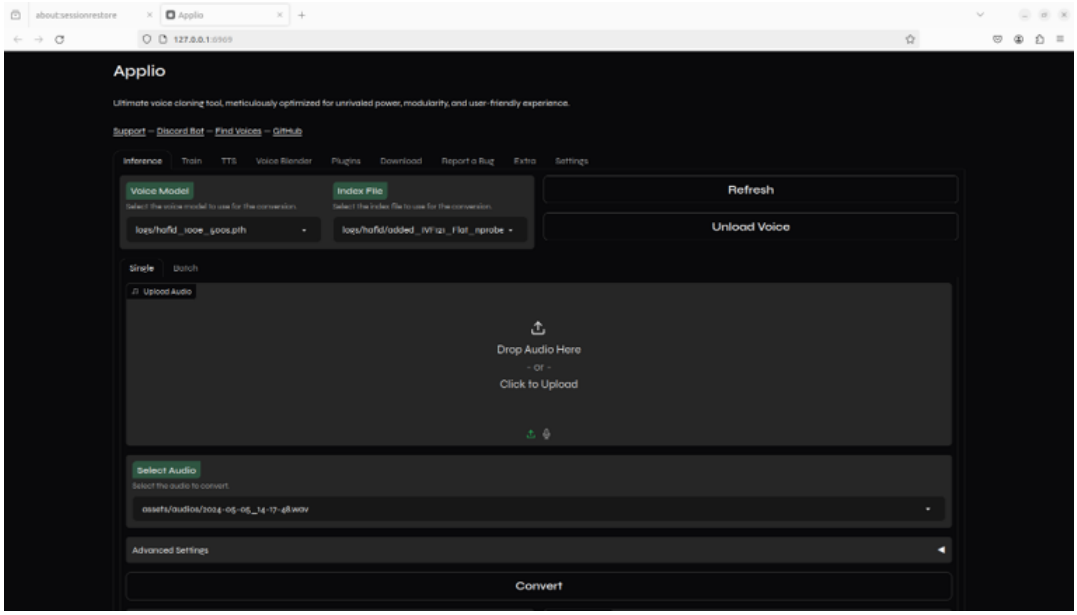


FIG. 3.17 : Interface web d'Applio [83].

Entraînement des modèles : Nous avons utilisé le modèle pré-entraîné publié pour (Kyliane, Macron et Zemmour). Par contre, pour Hafid Derraji et Le Président Abdelmadjid Tebboune, nous avons entraîné les modèles de RVC.

- Prétraitement : Lors du traitement des données, nous avons collecté des audios d'une dureé de 1 heure pour Hafid Derraji et 15 min pour le Président Abdelmadjid Tebboune.
- Entraînement : sur un GPU (RTX 3060 et 12GB de vidéo RAM)
  - Le modèle de «Le Président Abdelmadjid Tebboune» a été entraîné pendant 500 époques d'apprentissage (3 heures).
  - Le modèle de «Hafid Derraji» a été entraîné pendant 1000 époques d'apprentissage (9 heures 20 minutes).

3.3.3 Lip-Sync

Pour faire une évaluation des outils de génération de *deepfake* expliqué dans la section précédente, nous avons généré 15 vidéos par 3 outils différents (Wav2lip, DINet, GeneFace++), soit 5 vidéos par outil. Pour chaque outil, nous avons utilisé les mêmes 5 personnes pour les vidéos (Hafid Derraji, Éric Zemmour, Kylian Mbappé, Emmanuel Macron, le Président Abdelmadjid Tebboune), ce qui permet de comparer les performances de chaque outil de manière cohérente.

3.3.3.1 Wav2lip

Prétraitement : Lors du traitement des données, toutes les vidéos sont rééchantillonnées à 25 fps. et la résolution des vidéos d'entrée a été transformée a 1920x1080.

Modèle utilisé et inférence : Nous avons utilisé le modèle pré-entraîné publié qui est entraîné sur le dataset LRS2 [84].

3.3.3.2 DINet

Prétraitement : Lors du traitement des données, toutes les vidéos sont rééchantillonnées à 25 fps. Ainsi que on a transformer la résolution des videos d'entreé en 1920x1080.

Modèle utilisé et inférence : Nous avons utilisé le modèle pré-entraîné publié qui est entraîné sur le dataset HDTF [85] avec 363 vidéos d'entraînement collectées avec une résolution de 720p ou 1080p.
Ensuite nous avons utilisé OpenFace figure 3.18 pour détecter les points de repère faciaux lisses des vidéos (2d *Landmarks*).

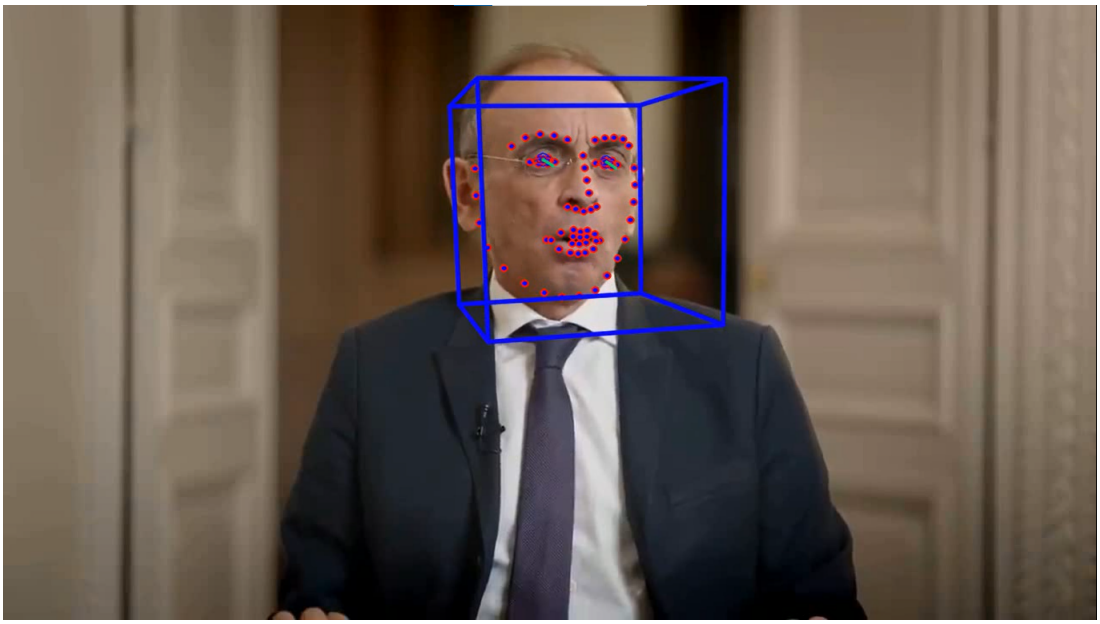


FIG. 3.18 : détection des points de repère faciaux avec OpenFace.

3.3.3.3 GeneFace++

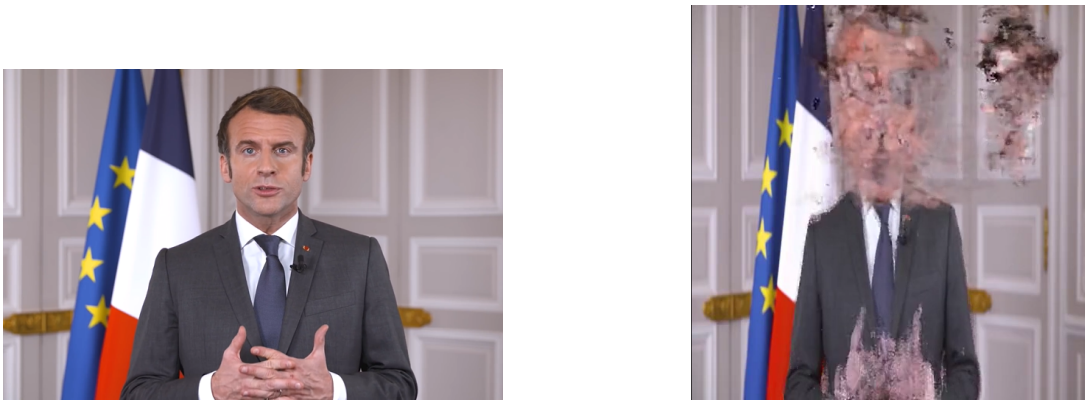
***Audio-to-motion* :** Nous avons utilisé le modèle publié pré-entraîné sur VoxCeleb2 [86], un ensemble de données de lecture labiale de 2000 heures.

***motion-to-vidéo* :** Nous avons entrainé le modèle pour les 5 personnes suivantes :

FIG. 3.19 : Préparation des 5 vidéos pour l'entraînement de GeneFace++.

- Lors du traitement des données, toutes les vidéos sont rééchantillon-

nées à 25 fps. Nous découpons ensuite la région du visage (figure 3.19) et redimensionnons tous les visages à une résolution de 512 × 512. Nous choisissons des vidéos qui garantissent la présence d'un visage dans chaque image pour éviter les problèmes de détection de visage lors de l'apprentissage et l'apparition d'artefacts lors de l'étape d'inférence (figure 3.20) a causé des problèmes de détection des points de repère faciaux (figure 3.21).



(a) Vidéo utilisée pour l'apprentissage du modèle. (b) Résultat d'inférence

FIG. 3.20 : Illustration des problèmes liés à l'utilisation des vidéos non zoomées sur le visage (corps complet) comme données d'apprentissage du modèle *audio-to-motion*.

- Entraînement : Nous avons entraîné le modèle de conversion de mouvement en vidéo en deux parties : le NeRF de la tête, puis le NeRF du torse .

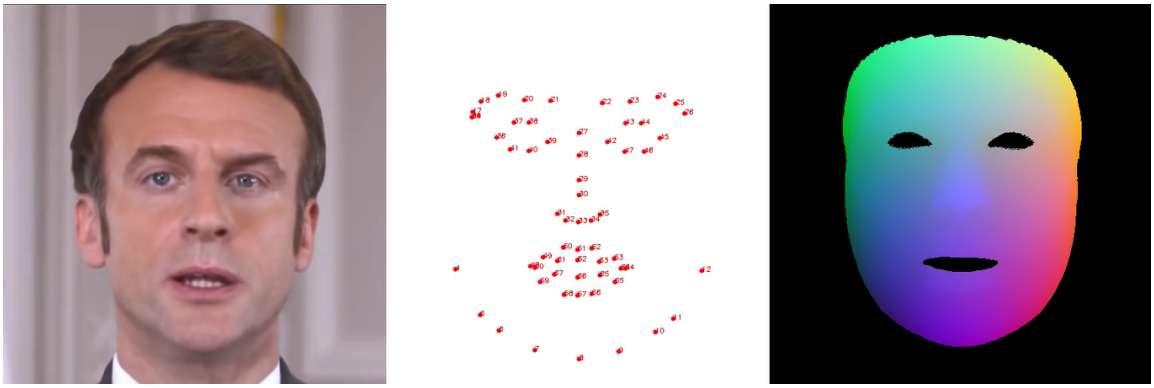


FIG. 3.21 : Détection des points de repère faciaux.

Montage des vidéos : Après la phase de génération des vidéos de la tête, nous avons pu monter manuellement cette tête générée dans la vidéo originale à l'aide d'outils de montage open source shotcut [87] (figure 3.23).

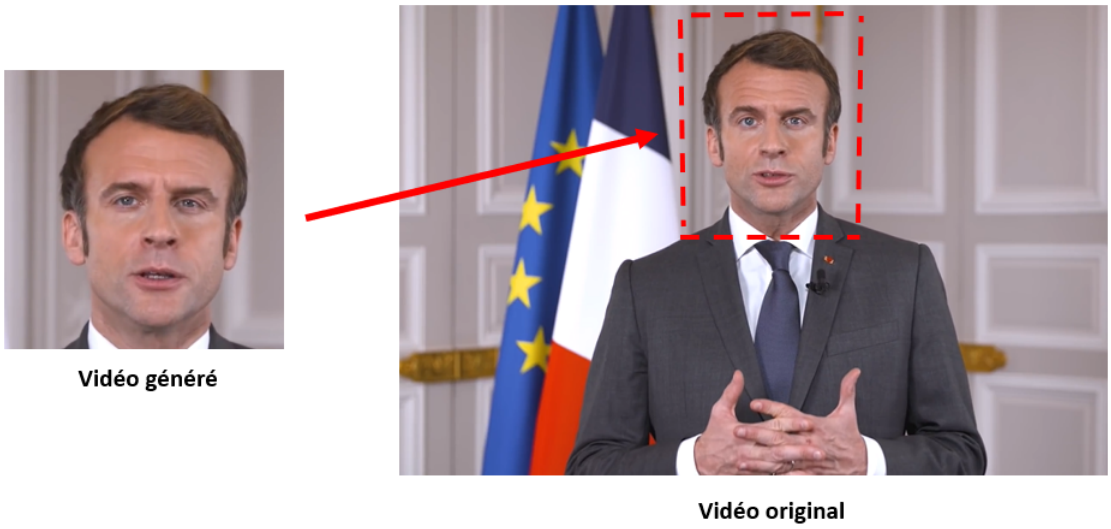


FIG. 3.22 : Remplacement des visages dans les vidéos originales par les vidéos générées.

Nous avons découpé les vidéos avant l'apprentissage et remonté les vidéos après l'inférence dans la même position des pixels, comme illustré dans la figure 3.22, pour obtenir des résultats plus réalistes. De plus, nous avons choisi les vidéos d'apprentissage de cinq personnes avec un arrière-plan statique.

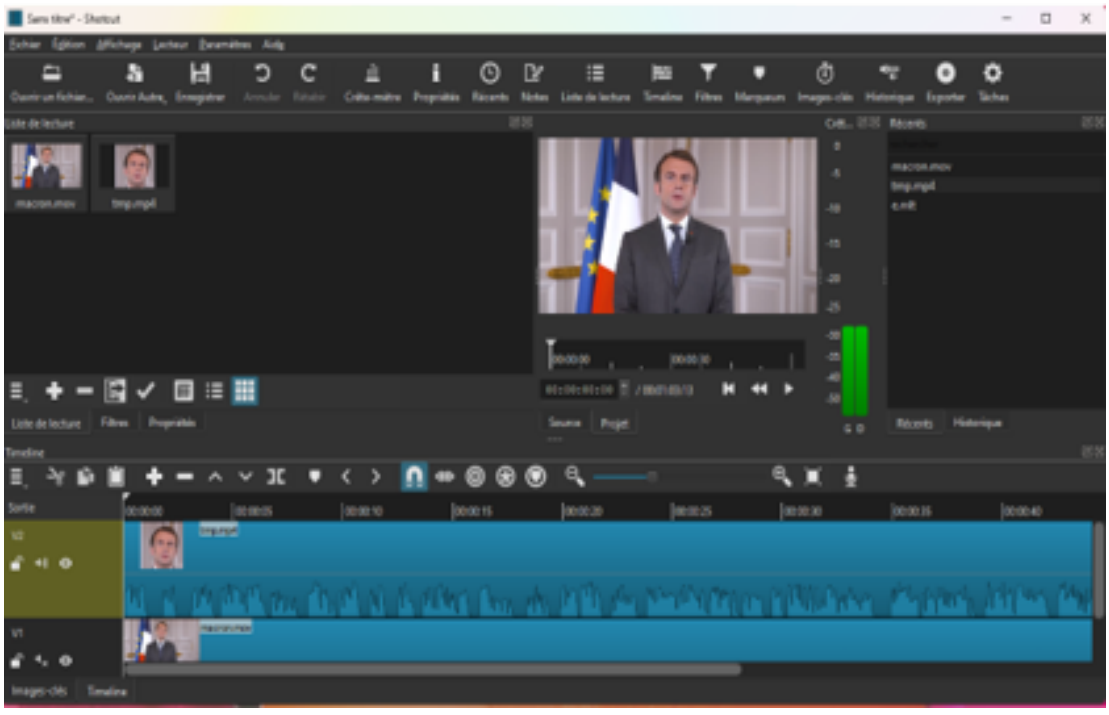


FIG. 3.23 : Montage des vidéos avec shotcut.

3.4 Conclusion

En conclusion, nos expérimentations ont permis de tester plusieurs méthodes pour créer des *deepfakes* audio et visuels, comme le remplacement de visage et la synchronisation labiale. L'objectif était de sélectionner les outils les plus efficaces pour produire des *deepfakes* réalistes. Grâce à ces tests, nous avons identifié les technologies les plus performantes pour obtenir des *deepfakes* de haute qualité, en prenant en compte la fluidité des mouvements, la précision des expressions faciales et la synchronisation audio-visuelle.

Chapitre 4

Résultats expérimentaux et évaluation

4.1 Introduction

Dans ce chapitre, nous allons présenter et discuter des résultats de nos expérimentations. Notre objectif principal était d'évaluer l'efficacité des différentes techniques et outils de génération de *deepfake* présentés dans les chapitres 2 et 3. Pour cela, nous avons effectué une série de tests en utilisant différentes méthodes d'évaluation objective, telles que le PSNR, le SSIM et le VMAF, ainsi que des techniques d'évaluation subjective. Pour l'évaluation subjective, nous avons sollicité 10 personnes pour noter une série de vidéos et d'audios en termes de précision de synchronisation labiale, qualité vidéo, réalisme vidéo. De plus, nous avons testé les vidéos générées avec ces outils à l'aide de différents détecteurs de *deepfake* afin de vérifier leur robustesse face à la détection.

Le but de cette section est de présenter de manière détaillée les résultats obtenus pour chaque outil, afin de déterminer leurs caractéristiques et performances, ainsi que d'identifier quels outils peuvent réaliser des *deepfakes* réalistes. Nous discutons également des limites de chaque outil et proposons des pistes d'amélioration possibles pour les futures recherches dans ce domaine.

4.2 Caractéristiques des deepfakes générés pour l'évaluation

Dans cette évaluation, nous évaluons des vidéos générées par les outils de génération présentés dans le chapitre 3. Nous générons cinq vidéos pour cinq personnes différentes, utilisant différentes langues pour les trois outils. Chaque personne a des caractéristiques spécifiques et chaque personne a une vidéo *deepfake* pour chaque outil, soit un total de trois outils par personne. Ces vidéos ont la même durée et la même résolution.

Ces vidéos sont comparées avec une vidéo originale de chaque personne (la vidéo utilisée pour la génération). Le tableau 4.1 présente les cinq vidéos utilisées, les personnes présentées, la durée des vidéos *deepfake* et originales, ainsi que leur résolution et la langue parlée dans la vidéo.

	Les personnes	Durée de la vidéo *deepfakes*	Résolution de la vidéo *deepfakes*	La langue	Durée de la vidéo originale	Résolution de la vidéo originale
Vidéo 1	Emmanuel Macron	20 sec	1920 x 1080	Français	4 min 3 sec	1920 x 1080
Vidéo 2	Kylian Mbappé	16 sec	1920 x 1080	Francais	28 sec	1920 x 1080
Vidéo 3	Hafid Derradji	21 sec	1920 x 1080	Arabe (darrija)	22 sec	1920 x 1080
Vidéo 4	Éric Zemmour	16 sec	1920 x 1080	Francais	11 min 2 sec	1920 x 1080
Vidéo 5	Le président Tebboune	22 sec	1920 x 1080	Arabe (darrija)	51 sec	1920 x 1080

TAB. 4.1 : Caractéristiques des vidéos utilisées pour l'évaluation des *deepfakes*

4.3 Évaluation objective des *deepfakes* générés

Dans cette évaluation, nous utilisons trois métriques (PSNR, SSIM et VMAF) pour évaluer les performances des 3 outils sur les 5 vidéos spécifiées dans le tableau 4.1. Pour mener à bien cette analyse, nous utilisons FFmetric, un logiciel open-source permettant de calculer ces métriques, présenté en section 4.3.4.

4.3.1 *Peak Signal to Noise Ratio* (PSNR)

Le *Peak Signal-to-Noise Ratio* (PSNR) est une métrique couramment utilisée pour mesurer la qualité des algorithmes de compression d'images [18]. Il quantifie la différence entre l'image originale et l'image compressée en évaluant le rapport signal/bruit en décibels. De plus, il peut également être utilisé pour évaluer la qualité des vidéos *deepfake*. Le PSNR est calculé à l'aide de la formule suivante :

$$PSNR = 20 \cdot \log_{10}(\text{MAX}) - 10 \cdot \log_{10}(\text{MSE}) \tag{4.1}$$

Où :

- MAX est la valeur maximale des pixels de l'image (par exemple, 255 pour une image de 8 bits).
- MSE (Mean Squared Error, ou erreur quadratique moyenne) est la moyenne des différences au carré entre les images originale et les images générées par deepfake.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{4.2}$$

Une valeur PSNR plus élevée indique une plus petite différence entre les images d'origine et les images générées par *deepfake*, ce qui correspond à

une meilleure qualité de la génération. En général, un PSNR plus élevé est préférable, car il indique moins de distorsion dans les *deepfakes*. Cependant, il est important de noter que le PSNR est une métrique limitée pour évaluer la qualité des *deepfakes*. D'autres métriques telles que l'indice de similarité structurelle (SSIM) ou l'indice de qualité perceptuelle (PQI) peuvent fournir une évaluation plus complète de la qualité perceptuelle des *deepfakes*, en prenant en compte des facteurs comme la perception visuelle humaine.

4.3.2 Structural Similarity Index Measure (SSIM)

L'Indice de Similarité Structurale (SSIM) est un modèle basé sur la perception qui évalue la dégradation des vidéos deepfake comme un changement dans l'information structurelle perçue. Ce modèle intègre également des facteurs de perception importants tels que le masquage de luminance et le masquage de contraste. L'information structurelle se réfère aux pixels fortement interdépendants ou spatialement proches, fournissant des informations cruciales sur les objets visuels dans une vidéo. Le masquage de luminance désigne le fait que les distorsions sont moins visibles sur les bords de l'image, tandis que le masquage de contraste indique que les distorsions sont également moins perceptibles dans la texture de l'image. Le SSIM évalue la qualité perçue des vidéos en mesurant la similarité entre une vidéo originale et une vidéo générée par deepfake [88]. La méthode de l'Indice de Similarité Structurale peut être exprimée à travers ces trois termes comme suit

$$\mathrm{SSIM}(x, y) = [l(x, y)]^{\alpha} \cdot [c(x, y)]^{\beta} \cdot [s(x, y)]^{\gamma} \tag{4.3}$$

Ici, $\boldsymbol{l}$ représente la luminance (utilisée pour comparer la luminosité entre deux images), $\boldsymbol{c}$ représente le contraste (utilisé pour différencier les plages entre les régions les plus claires et les plus sombres de deux images), et $\boldsymbol{s}$ représente la structure (utilisée pour comparer les motifs de luminance locale entre deux images afin de trouver les similitudes et les dissemblances entre les images). Les paramètres α, β et γ sont des constantes positives.

La luminance, le contraste et la structure d'une image peuvent être exprimés séparément de la manière suivante :

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{4.4}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{4.5}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{4.6}$$

où μ_x et μ_y sont les moyennes locales, σ_x et σ_y sont les écarts-types, et σ_{xy} est la covariance croisée pour les images x et y respectivement. Si $\alpha = \beta = \gamma = 1$, alors l'indice est simplifié sous la forme suivante en utilisant les équations 4.4, 4.5, 4.6 :

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)\,(2\sigma_x\sigma_y + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{4.7}$$

Une valeur SSIM plus élevée indique une plus grande similarité structurelle entre les vidéos d'origine et les vidéos générées par *deepfake*, ce qui correspond à une meilleure qualité de la génération. En général, un SSIM plus élevé est préférable, car il indique une moins grande dégradation perçue dans les *deepfakes*.

4.3.3 *Video Multimethod Assessment Fusion* (VMAF)

Le *Video Multi-Method Assessment Fusion* (VMAF) est une métrique de qualité vidéo développée par Netflix pour fournir une évaluation plus précise et fiable de la qualité perçue par les spectateurs. VMAF combine plusieurs méthodes de mesure de la qualité vidéo, en fusionnant leurs résultats pour obtenir une seule note de qualité [89].
VMAF utilise des métriques existantes de qualité d'image et d'autres caractéristiques pour prédire la qualité vidéo .

4.3.3.1 *Visual Information Fidelity* (VIF)

La métrique *Visual Information Fidelity* (VIF) permet d'évaluer la qualité d'une image ou d'une vidéo en évaluant la fidélité de l'information visuelle par rapport à une image ou vidéo de référence. Le concept de VIF repose sur le fait que l'information visuelle est transmise à travers des canaux de communication imparfaits, et elle évalue la quantité d'information préservée dans l'image ou la vidéo dégradée [90].

Visual Information Fidelity (VIF) permet d'évaluer la fidélité de l'information visuelle par rapport à une image ou vidéo de référence afin d'évaluer la qualité d'une image ou d'une vidéo. Le VIF repose sur l'idée que

l'information visuelle est transmise par des canaux de communication imparfaits, et elle évalue la quantité d'information conservée dans l'image ou la vidéo dégradée. Elle analyse l'image à diverses échelles spatiales pour obtenir des informations à différentes échelles de résolution. L'image de référence est considérée comme une source d'information et la version dégradée comme un signal transmis par un canal de communication bruyant. Ensuite, elle évalue la quantité d'information partagée entre l'image de référence et l'image dégradée, ce qui permet de mesurer la fidélité de l'information visuelle [90].

4.3.3.2 *Detail Loss Metric* (DLM)

Detail Loss Metric (DLM) est une mesure qui évalue à la fois la perte d'informations visuelles et les effets de perturbation qui pourraient être distrayants pour un observateur. Il est extrêmement important pour la compréhension des effets de la distorsion par compression et des distorsions par transformation sur la perception visuelle globale. Son utilisation est de trouver la région d'une image ou d'une vidéo où les détails fins sont écrasés et les structures visuelles qui vont attirer les yeux et ruiner le genre de qualité que l'on veut. La mesure est obtenue en comparant les changements locaux entre l'image de référence et l'image qui a de tels changements indésirables, et en donnant une attention particulière aux régions de haute fréquence spatiale qui contiennent beaucoup de détails [91].

4.3.3.3 *Mean Co-Located Pixel Difference* (MCPD)

Mean Co-Located Pixel Difference (MCPD) est une mesure de la différence temporelle entre les images consécutives dans une séquence vidéo et est donnée par la luminance. Le changement de luminosité d'une image à l'autre est un attribut très important pour la qualité de la fluidité visuelle et la cohérence temporelle. C'est la valeur moyenne du décalage de pixel co-localisé entre les images consécutives d'une séquence vidéo. Basée sur la luminance, la variation la plus significative de cette mesure est celle de la luminance qui, dans une large mesure, détermine la décision de la fluidité et de l'information continue. Il est conçu pour déceler les distorsions météorologiques qui produisent des artefacts visuels tels que le scintillement et le flou de mouvement, pour fournir ainsi un moyen objectif de mesurer la stabilité temporelle et la qualité perçue de la vidéo de manière fiable

[91].

4.3.4 FFMpeg

FFMpeg, un logiciel *open-source*, peut être utilisé pour calculer différentes métriques de qualité visuelle (PSNR, SSIM, VMAF). FFMetrics est une interface graphique pour FFMpeg dont le but est de visualiser les métriques de qualité calculées par FFMpeg. Le programme vous permet de sélectionner des fichiers sans avoir à utiliser la ligne de commande, de calculer et de visualiser les métriques de qualité PSNR, SSIM et VMAF pour tous les fichiers en une seule fois [92]. La figure 4.1 ci-dessous illustre l'interface du logiciel lors de l'évaluation des vidéos.

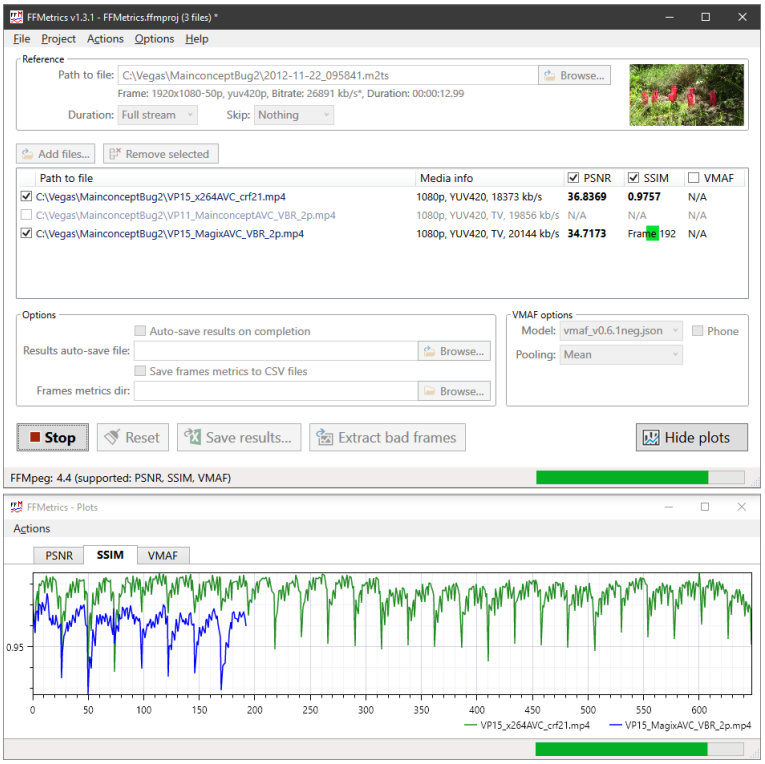


FIG. 4.1 : Interface graphique du logiciel.

- FFMpeg fonctionne en prenant en entrée la vidéo originale ainsi que les vidéos générées par les outils DINET, GENFACE++ et WAVE2LIP. Il divise ces vidéos en images individuelles (frames). Ensuite, pour chaque frame de chaque vidéo générée, FFMpeg calcule les valeurs de PSNR, SSIM et VMAF, qui mesurent respectivement la fidélité, la similarité structurelle et la qualité visuelle par rapport à la vidéo originale. Enfin, le logiciel agrège ces résultats pour obtenir des mesures moyennes globales de chaque métrique, fournissant ainsi une évaluation complète de la qualité des vidéos générées par ces outils.

4.4 Résultat de l'évaluation objective

Pour évaluer les outils de génération de vidéos *deepfake* (WAVE2LIP, DINET, GENFACE++), nous examinons les résultats obtenus pour chaque individu à l'aide de trois métriques : *Peak Signal-to-Noise Ratio (PSNR)*, *Structural Similarity Index Measure* (SSIM), et *Video Multi-method Assessment Fusion* (VMAF). Ces métriques permettent d'évaluer la qualité des vidéos générées par ces outils. Nous avons généré 15 vidéos avec les trois outils, en utilisant différentes personnes. Dans ces vidéos, chaque personne prononce un discours spécifique écrit par nous-mêmes, conçu pour montrer que c'est un *deepfake*. Chaque personne a trois vidéos de même durée (qui dépend du discours donné) et de même résolution.

	Métrique utilisée	**Outils de génération**		
		WAVE2LIP	DINET	GENFACE++
Vidéo 1	PSNR ↑	39.90	40.24	26.22
	SSIM ↑	0.9903	0.9907	0.9397
	VMAF ↑	86.46	85.86	19.09

Tab. 4.2 : Résultat de l'évaluation objective sur la première vidéo.

Pour la première vidéo (tableau 4.2), l'outil DINET montre la meilleure qualité d'image (PSNR de 40,24) et la plus grande similarité structurelle (SSIM de 0,9907). Cependant, GENFACE++ affiche une faible performance globale en VMAF (19,09), indiquant une moins bonne qualité vidéo perçue. WAVE2LIP, quant à lui, performe globalement bien, en particulier en VMAF (86,46).

	Métrique utilisée	Outils de génération		
		WAVE2LIP	DINET	GENFACE++
Vidéo 2	PSNR ↑	29.73	36.99	25.97
	SSIM ↑	0.9528	0.9828	0.9387
	VMAF ↑	39.22	74.93	40.40

Tab. 4.3 : Résultat de l'évaluation objective sur la deuxième vidéo.

En ce qui concerne la deuxième vidéo (tableau 4.3), l'outil DINET domine avec les meilleurs scores en PSNR (36.99), SSIM (0.9828), et VMAF (74.93), suggérant une qualité d'image et vidéo supérieure.

	Metric utilisée	Outils de génération		
		WAVE2LIP	DINET	GENFACE++
Vidéo 3	PSNR ↑	32.78	34.88	39.22
	SSIM ↑	0.9941	0.9944	0.9780
	VMAF ↑	93.22	83.56	82.73

Tab. 4.4 : Résultat de l'évaluation objective sur la troisième vidéo.

Pour la troisième vidéo (tableau 4.4), l'outil GENFACE++ obtient le meilleur score PSNR (39.22), mais WAVE2LIP montre les meilleures performances en SSIM (0.9941) et VMAF (93.22), ce qui indique une meilleure qualité vidéo globale perçue.

	Métrique utilisée	Outils de génération		
		WAVE2LIP	DINET	GENFACE++
Vidéo 4	PSNR ↑	24.78	25.88	24.22
	SSIM ↑	0.9290	0.9381	0.9294
	VMAF ↑	23.22	88.56	43.73

Tab. 4.5 : Résultat de l'évaluation objective sur la quatrième vidéo .

En ce qui concerne Éric Zemmour (tableau 4.5), l'outil DINET excelle avec les meilleures valeurs en PSNR (25.88), SSIM (0.9381) et VMAF (88.56), démontrant ainsi une qualité d'image et de vidéo nettement supérieure.

	Métrique utilisée	Outils de génération		
		WAVE2LIP	DINET	GENFACE++
Vidéo 5	PSNR ↑	33.22	41.16	33.86
	SSIM ↑	0.9713	0.9853	0.9716
	VMAF ↑	56.22	83.41	70.73

Tab. 4.6 : Résultat de l'évaluation objective sur la cinquième vidéo.

Pour la cinquième vidéo (tableau 4.6), l'outil DINET montre les meilleurs résultats dans toutes les métriques, avec un PSNR de 41.16, un SSIM de 0.9853 et un VMAF de 83.41, ce qui indique une excellente qualité d'image et de vidéo.

4.4.1 Synthèse globale moyenne de toutes les vidéos

	Métrique utilisée	Outils de génération		
		WAVE2LIP	DINET	GENFACE++
Toutes les vidéos	PSNR ↑	33.22	37.90	33.42
	SSIM ↑	0.9801	0.9872	0.9514
	VMAF ↑	69.46	83.67	56.35

Tab. 4.7 : Récapitulatif des scores moyens sur les cinq vidéos.

le tableau 4.7 représente les moyennes des métriques de performance pour les outils de génération WAVE2LIP, DINET et GENFACE++, Les résultats présentés dans le tableau seront discutés en détail dans la sous-section 4.4.1.1.

4.4.1.1 Discussion des résultats

DINET se distingue comme l'outil le plus performant, obtenant les meilleures valeurs en PSNR (37.90), SSIM (0.9872) et VMAF (83.67). Cela suggère que DINET est le plus efficace pour générer des vidéos de haute qualité.

GENFACE++ montre des performances variables. Bien qu'il excelle parfois en PSNR (31.30), il est souvent moins performant en VMAF (56.35), indiquant une qualité vidéo perçue inférieure.

WAVE2LIP affiche de bons résultats en SSIM (0.9801) et en VMAF

(69.46), mais il est généralement surpassé par DINET. Avec une moyenne de PSNR de 33.42, WAVE2LIP reste compétitif mais inférieur à DINET.

En résumé, DINET est le plus efficace pour générer des vidéos de haute qualité, suivi par WAVE2LIP, tandis que GENFACE++ montre des performances moins consistantes.

4.5 Évaluation subjective de la qualité des *deepfakes*

4.5.1 Protocole d'évaluation

Afin de compléter l'évaluation objective de la qualité des *deepfakes* par une évaluation "subjective" portant sur la qualité perçue, nous avons mis en place un protocole d'évaluation structuré et rigoureux, impliquant des utilisateurs réels. Ce protocole est détaillé dans les sections suivantes.

4.5.1.1 Préparation des vidéos

Dans le cadre de cette analyse subjective, nous avons utilisé les mêmes vidéos que dans l'analyse objective, en conservant les mêmes individus, enregistrements audio et durée. Les cinq vidéos et leurs caractéristiques sont présentées dans le tableau 4.1. Cette partie de l'évaluation sera plus approfondie et exhaustive. Pour évaluer les performances des outils de génération de vidéos synchronisées avec les lèvres et de génération d'audio, nous procédons en deux étapes : d'abord, nous étudions les audios seuls, puis les vidéos avec leurs audios.

4.5.1.2 Processus de recrutement pour des évaluateurs

Nous avons recruté 10 évaluateurs pour cette tâche d'évaluation. Afin de garantir que l'évaluation soit efficace et pertinente, les évaluateurs ont été choisis parmi les candidats qui connaissent les personnes dont ils vont évaluer les vidéos et les audios.

4.5.1.3 Critères d'évaluation

Trois aspects sont pris en compte pour évaluer la qualité de l'audio.

- **L'évaluation de la clarté** est de 1 à 5, avec un score de 1 pour l'audio très brouillé et un score de 5 pour l'audio très cristallin.

- **L'évaluation de l'intelligibilité** s'élève de 1 à 5, avec un score de 1 pour un audio difficile à comprendre et un score de 5 pour un audio très courant .

- **Les émotions** sont finalement évaluées de 1 à 5, avec un score de 1 pour l'audio monotone et un score de 5 pour l'audio très expressif.

En ce qui concerne la vidéo :

- **Le réalisme des vidéos** on attribue une note de 1 à 5 , avec 1 pour la vidéo peu réaliste et 5 pour la vidéo très réaliste.

- **La qualité des vidéos** est notée de 1 à 5, avec un score de 1 pour une qualité médiocre et un score de 5 pour une qualité très élevée.

- **la synchronisation entre la vidéo et l'audio** on attribue une note de 1 à 5 , avec 1 pour une synchronisation très décalée et 5 pour une synchronisation parfaite.

4.5.1.4 Processus d'évaluation

Chaque participant commence par écouter et évaluer les audios en fonction de critères spécifiques définis pour l'audio. Les audios avec les meilleurs scores seront ensuite utilisés pour générer des deepfakes vidéos synchronisées sur les lèvres. Les participants visionneront ensuite ces vidéos et les évalueront selon des critères distincts définis pour la vidéo. Les notes seront compilées pour obtenir une moyenne par critère. Une analyse comparative est effectuée pour évaluer la performance des outils de génération de vidéos synchronisées sur les lèvres et d'audio.

4.5.2 Résultat de l'évaluation subjective

4.5.2.1 Évaluation des enregistrements audio

Pour déterminer quel outil est le plus fiable pour la génération d'audios, nous avons mené un test en évaluant trois critères : la clarté, l'intelligibilité, et les émotions et l'intonation. La moyenne des scores pour ces trois critères a été calculée pour les 10 évaluateurs. Les résultats de cette évaluation sont présentés dans le tableau 4.8.

	Outils de génération		
	TTS	RVC	TTS + RVC
Audio 1	1.85	3.42	4.21
Audio 2	2.28	2.42	4
Audio 3	1.92	3.92	3
Audio 4	3.07	3.07	3.85
Audio 5	1.5	4.14	3.07

TAB. 4.8 : Résultat de l'évaluation subjective de l'audio.

Interprétation des résultats

Les résultats de ce tableau fournissent des informations précieuses sur l'efficacité relative des outils de synthèse vocale (TTS) et de conversion de voix (RVC), ainsi que leur combinaison, dans l'amélioration de la qualité perçue de l'audio. Voici une analyse plus approfondie pour chaque personne :

Pour la première vidéo, le score le plus élevé (4.21) est obtenu avec la combinaison TTS + RVC, suggérant que l'amélioration apportée par chaque outil est complémentaire lorsqu'ils sont utilisés ensemble. L'outil RVC seul (3.42) offre une amélioration significative par rapport au TTS seul (1.85).

Concernant la deuxième vidéo, la combinaison TTS + RVC (4) donne également le meilleur résultat, confirmant que les deux outils ensemble fournissent une meilleure performance que chacun séparément. Les scores des outils individuels ((2.28) pour TTS et (2.42) pour RVC) sont assez proches, indiquant une efficacité similaire mais limitée individuellement.

Pour la troisième vidéo, l'outil RVC seul (3.92) surpasse la combinaison TTS + RVC (3). Cela pourrait indiquer que, pour cette vidéo spécifique, l'ajout de TTS n'apporte pas de valeur ajoutée et pourrait même diminuer la qualité perçue. Le score du TTS seul (1.92) est le plus bas, suggérant une performance inférieure de cet outil pour cette vidéo.

En ce qui concerne quatrième vidéo, la combinaison TTS + RVC (3.85) produit le meilleur score, soulignant l'avantage de l'utilisation combinée des deux outils. Les scores individuels de TTS et RVC sont identiques (3.07), indiquant que chaque outil seul apporte une amélioration significative mais

égale.

Pour la cinquième vidéo, l'outil RVC seul (4.14) obtient le score le plus élevé, ce qui pourrait indiquer que cet outil est particulièrement bien adapté à cette voix ou ce contenu spécifique. Le score du TTS seul (1.5) est le plus bas, suggérant une performance inférieure de cet outil pour cette vidéo. La combinaison TTS + RVC (3.07) est moins efficace que RVC seul, ce qui pourrait indiquer une possible interférence ou incompatibilité entre les deux outils pour cette vidéo.

4.5.2.2 Évaluation des vidéos générées

	Critères d'évaluation	Outils de génération		
		WAVE2LIP	DINET	GENFACE++
Vidéo 1	*video realness* (0-5)	3.14	3.21	1.71
	video visual quality (0-5)	3.21	3.71	2.85
	video synchronization with audio (0-5)	2.57	2.71	2.57

Tab. 4.9 : Résultat de l'évaluation subjective de la vidéo d'Emmanuel Macron

Le tableau 4.9 présente les résultats d'une évaluation subjective de la vidéo d'Emmanuel Macron, utilisant divers outils. Les colonnes représentent les différents outils ou méthodes utilisés, et les lignes représentent les différents critères d'évaluation.

Interprétation des résultats

Pour la réalité de la vidéo, DINET obtient le score le plus élevé (3.21), suggérant que cet outil est perçu comme le plus réaliste pour cette vidéo particulière. WAVE2LIP obtient un score proche (3.14), indiquant également une bonne performance, tandis que GENFACE++ a le score le plus bas (1.71).

Concernant la qualité visuelle de la vidéo, DINET se distingue nettement avec un score de (3.71), indiquant la meilleure performance visuelle. WAVE2LIP suit avec un score de 3.21, et GENFACE++ obtient (2.85), ce qui est inférieur mais encore relativement bon.

Pour la synchronisation vidéo avec audio, DINET obtient encore le meilleur score (2.71), suivi de WAVE2LIP et GENFACE++ avec des scores égaux de (2.57). Cela suggère que DINET est légèrement meilleur pour synchroniser la vidéo avec l'audio, bien que les trois outils aient des per-

formances similaires.

	Critères d'Évaluation	Outils de génération		
		WAVE2LIP	DINET	GENFACE++
Vidéo 2	*video realness* (0-5)	1.85	1.5	3.14
	video visual quality (0-5)	2.57	2.64	3.35
	video synchronization with audio (0-5)	2.64	3	3.07

Tab. 4.10 : Résultat de l'évaluation subjective de la deuxième vidéo.

Le tableau 4.10 présente les résultats d'une évaluation subjective de la deuxième vidéo, utilisant divers outils. Les colonnes représentent les différents outils ou méthodes utilisés, et les lignes représentent les différents critères d'évaluation.

Interprétation des résultats

Pour la réalité de la vidéo, GENFACE++ obtient le score le plus élevé (3.14), suggérant que cet outil est perçu comme le plus réaliste pour cette vidéo particulière. WAVE2LIP (1.85) et DINET (1.65) ont des scores significativement plus bas, indiquant une performance moins convaincante en termes de réalisme.

Concernant la qualité visuelle de la vidéo, GENFACE++ se distingue avec un score de (3.15), indiquant la meilleure performance visuelle. WAVE2LIP suit avec un score de (2.87), et DINET obtient (2.64), ce qui est inférieur mais encore relativement bon.

Pour la synchronisation vidéo avec audio, GENFACE++ obtient encore le meilleur score (3.07), suivi de près par DINET (3.00). WAVE2LIP a une note légèrement plus basse à 2.64, suggérant une synchronisation acceptable mais moins bonne que celle des deux autres outils.

	Critères d'Évaluation	Outils de génération		
		WAVE2LIP	DINET	GENFACE++
Vidéo 3	*video realness* (0-5)	2.57	2.92	1.42
	video visual quality (0-5)	3	3.07	1.87
	video synchronization with audio (0-5)	1.98	2.64	2

Tab. 4.11 : Résultat de l'évaluation subjective sur la troisième vidéo.

Le tableau 4.11 présente les résultats d'une évaluation subjective de la troisième vidéo, utilisant divers outils. Les colonnes représentent les différents outils ou méthodes utilisés, et les lignes représentent les différents

critères d'évaluation.

Interprétation des résultats

Pour la réalité de la vidéo, DINET obtient le score le plus élevé (2.92), suggérant que cet outil est perçu comme le plus réaliste pour cette vidéo particulière. WAVE2LIP suit avec un score de (2.57), indiquant également une bonne performance, tandis que GENFACE++ a le score le plus bas (1.42).

Concernant la qualité visuelle de la vidéo, DINET se distingue avec un score de (3.07), indiquant la meilleure performance visuelle. WAVE2LIP suit de près avec un score de (3.00), et GENFACE++ obtient (1.87), ce qui est nettement inférieur.

Pour la synchronisation vidéo avec audio, DINET obtient encore le meilleur score (2.64), suivi de GENFACE++ avec un score de (2.00). WAVE2LIP a une note de (1.98), suggérant une synchronisation acceptable mais inférieure à celle de DINET.

	Critères d'Évaluation	**Outils de génération**		
		WAVE2LIP	DINET	GENFACE++
Vidéo 4	*video realness* (0-5)	3.07	3.14	1.92
	video visual quality (0-5)	3	3.64	3.14
	video synchronization with audio (0-5)	1.92	3.07	2.14

Tab. 4.12 : Résultat de l'évaluation subjective sur la quatrième vidéo.

Le tableau 4.12 présente les résultats d'une évaluation subjective de la quatrième vidéo, utilisant divers outils. Les colonnes représentent les différents outils ou méthodes utilisés, et les lignes représentent les différents critères d'évaluation.

Interprétation des résultats

Pour la réalité de la vidéo, DINET obtient le score le plus élevé (3.14), suggérant que cet outil est perçu comme le plus réaliste pour cette vidéo particulière. WAVE2LIP suit de près avec un score de (3.07), indiquant également une bonne performance, tandis que GENFACE++ a le score le plus bas (1.92).

Concernant la qualité visuelle de la vidéo, DINET se distingue nettement avec un score de (3.64), indiquant la meilleure performance visuelle.

WAVE2LIP suit avec un score de (3.00), et GENFACE++ obtient (3.14), ce qui est inférieur à DINET mais supérieur à WAVE2LIP.

Pour la synchronisation vidéo avec audio, DINET obtient encore le meilleur score (3.07), suivi de GENFACE++ avec un score de (2.14). WAVE2LIP a une note de (1.92), suggérant une synchronisation acceptable mais inférieure à celle de DINET et GENFACE++.

	Critères d'Évaluation	**Outils de génération**		
		WAVE2LIP	DINET	GENFACE++
Vidéo 5	*video realness* (0-5)	4.35	4.45	3.22
	video visual quality (0-5)	4.07	4.14	3.93
	video synchronization with audio (0-5)	3.92	4	3.07

Tab. 4.13 : Résultat de l'évaluation subjective sur la cinquième vidéo.

Le tableau 4.13 présente les résultats d'une évaluation subjective de la cinquième vidéo, utilisant divers outils. Les colonnes représentent les différents outils ou méthodes utilisés, et les lignes représentent les différents critères d'évaluation.

Interprétation des résultats

Pour la réalité de la vidéo, DINET obtient le score le plus élevé (4.45), suggérant que cet outil est perçu comme le plus réaliste pour cette vidéo particulière. WAVE2LIP suit de près avec un score de (4.35), indiquant également une bonne performance, tandis que GENFACE++ a le score le plus bas (3.22).

Concernant la qualité visuelle de la vidéo, DINET se distingue avec un score de (4.14), indiquant la meilleure performance visuelle. WAVE2LIP suit avec un score de (4.07), et GENFACE++ obtient (3.93), ce qui est inférieur mais encore relativement bon.

Pour la synchronisation vidéo avec audio, DINET obtient encore le meilleur score (4.00), suivi de près par WAVE2LIP avec un score de (3.92). GENFACE++ a une note de (3.07), suggérant une synchronisation acceptable mais inférieure à celle des deux autres outils.

4.5.3 Synthèse globale moyenne de toutes les vidéos

Afin de procéder à une évaluation approfondie de ces outils, nous calculons la moyenne des résultats de l'évaluation subjective pour les cinq vidéos

pour chaque outil. Les résultats sont détaillés dans le tableau4.14.

	Critères d'Évaluation	Outils de génération		
		WAVE2LIP	DINET	GENFACE++
Toutes les vidéos	*video realness* (0-5)	2.03	3.75	1.78
	video visual quality (0-5)	2.53	3.84	3.82
	video synchronization with audio (0-5)	3.26	3.99	3.88

Tab. 4.14 : Résultat de l'évaluation subjective (score moyen) sur les cinq vidéos.

Les résultats présentés dans le tableau seront discutés en détail dans la sous-section 4.5.3.1

4.5.3.1 Discussion des résultats

Pour le réalisme vidéo, l'outil DINET obtient le score moyen le plus élevé avec (3,75), indiquant que ses vidéos sont perçues comme les plus réalistes. Cela montre que DINET est particulièrement efficace pour simuler des détails et des mouvements naturels, contribuant ainsi à l'authenticité perçue des vidéos. En revanche, l'outil GENFACE++ a la moyenne la plus basse de (1,78), suggérant qu'il a des difficultés à reproduire des détails convaincants, rendant ses vidéos moins crédibles. L'outil WAVE2LIP se situe entre les deux avec une moyenne de (2,03), montrant que bien qu'il puisse générer des vidéos quelque peu réalistes, il ne parvient pas à atteindre le niveau de DINET.

En termes de qualité visuelle, l'outil DINET se distingue également avec la meilleure moyenne de (3,84), suggérant que ses vidéos sont visuellement les plus attrayantes avec une bonne résolution et clarté. Cela marque une amélioration par rapport à une valeur précédente de (3,61), soulignant l'attention aux détails de DINET. L'outil GENFACE++ suit de près avec une moyenne de (3,82), montrant qu'il produit des vidéos de haute qualité visuelle. L'outil WAVE2LIP, avec une moyenne de (2,53), est perçu comme offrant une qualité visuelle moindre, probablement en raison de problèmes de netteté ou de résolution qui réduisent leur attrait.

Pour la synchronisation labiale avec l'audio, l'outil DINET obtient la moyenne la plus élevée de (3,99), ce qui indique une excellente synchronisation entre les mouvements labiaux et l'audio, essentielle pour une perception réaliste des vidéos. L'outil GENFACE++ a une moyenne de (3,88), également très bonne, montrant une performance solide en synchronisation labiale. L'outil WAVE2LIP, avec une moyenne de (3,26), bien que bonne,

n'est pas aussi précise que DINET et GENFACE++, ce qui peut affecter l'authenticité perçue de ses vidéos.

En conclusion, les moyennes des évaluations subjectives montrent que l'outil DINET excelle globalement, offrant la meilleure qualité visuelle (3,84), le plus grand réalisme (3,75), et une excellente synchronisation labiale (3,99). L'outil GENFACE++ est compétitif en qualité visuelle (3,82) et en synchronisation labiale (3,88), mais moins performant en termes de réalisme (1,78). L'outil WAVE2LIP, bien qu'ayant une bonne synchronisation labiale (3,26), affiche des scores inférieurs en réalisme (2,03) et en qualité visuelle (2,53), indiquant des domaines à améliorer. Ces résultats offrent un aperçu précieux des capacités de chaque outil en termes de génération de vidéos, mettant en avant leurs points forts et faiblesses.

4.6 Évaluation de l'authenticité des vidéos à l'aide de détecteurs

Pour évaluer les outils de génération de vidéos *deepfake* (DINET, GENFACE++, WAVE2LIP) en fonction de leur authenticité, nous analysons les scores des détecteurs LipFD et SBI+RECCE pour chaque outil. Nous déterminerons si les vidéos générées sont détectées comme authentiques ou fausses, en nous basant sur un seuil de 0,5. Les résultats des tests sont présentés dans le tableau 4.15.

Outils de génération	DINET		GENFACE++		WAVE2LIP	
Détecteur	LipFD	SBI+RECCE	LipFD	SBI+RECCE	LipFD	SBI+RECCE
Vidéo 1	0.0000	0.3252	0.0000	0.1891	0.0000	0.6933
Vidéo 2	0.3675	0.1448	0.1766	0.1201	0.5652	0.5085
Vidéo 3	0.0002	0.4298	0.0001	0.4409	0.0002	0.8056
Vidéo 4	0.0754	0.5151	0.074	0.1826	0.0949	0.7529
Vidéo 5	0.0151	0.2986	0.0156	0.2669	0.0696	0.4871

TAB. 4.15 : Évaluation de l'authenticité des vidéos générées par différents outils de génération à l'aide de détecteurs (LipFD [93], SBI [94]+RECCE [95])

Remarque : Les valeurs en vert indiquent que le *deepfake*a été détecté avec succès, tandis que les valeurs en orange indiquent une vidéo suspecte (score très proche de 0,5).

Interprétation des résultats

Pour l'outil DINET, les résultats sont les suivants. Les vidéos 1, 2, 3 et 5 sont considérées comme authentiques avec des scores respectifs de 0.0002, 0.0151, 0.3675 et 0.0000 (LipFD) et 0.4298, 0.2986, 0.1448 et 0.3252 (SBI+RECCE). La vidéo 4 est considérée authentique avec un score de 0.0754 (LipFD) mais suspecte avec un score de 0.5151 (SBI+RECCE).

Pour GENFACE++, toutes les vidéos 1, 2, 3, 4 et 5 sont considérées comme authentiques. Les scores sont respectivement 0.0001, 0.0150, 0.1766, 0.0747 et 0.0000 (LipFD) et 0.4409, 0.2669, 0.1201, 0.4707 et 0.3076 (SBI+RECCE).

Pour WAVE2LIP, les résultats sont plus variés. Les vidéos 1 et 2 sont considérées comme authentiques avec des scores de 0.0002 et 0.0006 (LipFD) mais la vidéo 1 est détectée comme *deepfake* avec un score de 0.8056 (SBI+RECCE) et celle de la vidéo 2 est suspecte avec un score de 0.5696 (SBI+RECCE). La vidéo 3 est suspecte avec des scores de 0.5652 (LipFD) et 0.5085 (SBI+RECCE). Les vidéos 4 et 5 sont considérées comme authentiques avec des scores de 0.0152 et 0.0000 (LipFD) mais détectées comme *deepfake* avec des scores de 0.7529 et 0.6933 (SBI+RECCE).

4.6.1 Synthèse globale moyenne de toutes les vidéos

Afin de procéder à une évaluation approfondie de ces outils, nous calculons la moyenne des sorties des détecteurs pour chaque outil. Les résultats sont détaillés dans le tableau 4.16.

Outils de génération	DINET		GENFACE++		WAVE2LIP	
Détecteur	LipFD	SBI+RECCE	LipFD	SBI+RECCE	LipFD	SBI+RECCE
Toutes les vidéos	0.09164	0.3427	0.05326	0.23992	0.14596	0.64946

Tab. 4.16 : Résumé des scores moyens d'authenticité pour les vidéos générées par différents outils de génération.

Les résultats présentés dans le tableau seront discutés en détail dans la sous-section 4.6.1.1.

4.6.1.1 Discussion des résultats

Pour l'outil DINET, LipFD donne un score moyen de 0,0875, ce qui indique une forte authenticité perçue des vidéos. Toutefois, SBI+RECCE affiche un score moyen de 0,2392, montrant que ce détecteur trouve da-

vantage de signes de manipulation, suggérant que DINET est modérément performant en générant des vidéos qui semblent authentiques.

Pour l'outil GENFACE++, les scores moyens sont encore plus faibles : 0,0687 pour LipFD et 0,1936 pour SBI+RECCE. Cela signifie que GENFACE++ génère des vidéos qui paraissent très authentiques, surpassant DINET, particulièrement sur LipFD.

En revanche, l'outil WAVE2LIP obtient des scores moyens plus élevés de 0,1334 sur LipFD et de 0,4522 sur SBI+RECCE. Ces résultats montrent que les vidéos générées par WAVE2LIP sont perçues comme les moins authentiques parmi les trois outils, particulièrement selon SBI+RECCE, qui détecte plus fréquemment des signes de manipulation. En conclusion, l'outil GENFACE++ se démarque par la meilleure performance en termes d'authenticité perçue, suivi de l'outil DINET, tandis que l'outil WAVE2LIP est moins performant, produisant des vidéos plus souvent détectées comme suspectes ou manipulées.

4.6.2 Synthèse globale sur les performances des outils de génération de deepfake

D'après les résultats obtenus dans les sous-sections 4.4.1.1 4.5.3.1 et 4.6.1.1, DINET est l'outil le plus performant pour générer des vidéos deepfake de haute qualité, offrant la meilleure combinaison de réalisme, qualité visuelle et synchronisation labiale. GENFACE++ présente des performances variables mais excelle en termes d'authenticité perçue. WAVE2LIP est compétitif mais généralement inférieur à DINET et GENFACE++ en termes de qualité vidéo et d'authenticité perçue. Ces évaluations fournissent une vue d'ensemble des capacités et des limites de chaque outil, soulignant leurs points forts et domaines d'amélioration.

4.7 Conclusion

Dans ce chapitre, nous avons approfondi l'analyse des résultats de nos expérimentations visant à évaluer l'efficacité des techniques et outils de génération de *deepfake* présentés dans les chapitres 2 et 3. Notre objectif principal était de déterminer leur capacité à produire des vidéos et des audios réalistes, en utilisant à la fois des mesures objectives comme le PSNR, le SSIM et le VMAF, ainsi que des évaluations subjectives impliquant un

panel de 10 personnes pour juger la qualité et le réalisme des contenus générés. Parallèlement, nous avons soumis ces productions à des détecteurs de *deepfake* (LipFD, SBI+RECCE) pour évaluer leur résistance à la détection.

Cette analyse détaillée nous a permis de caractériser les performances spécifiques de chaque outil, d'identifier ceux capables de générer des deepfakes convaincants, et de discuter des limites rencontrées.

Conclusion et perspectives

En conclusion, cette étude a exploré en profondeur le domaine des *deepfakes*, en abordant d'abord les généralités sur ce phénomène pour en établir les bases conceptuelles pour créer des outils de génération des *deepfakes* comme les GAN et les autoencondeurs.

Nous avons ensuite détaillé les différentes méthodes de génération des *deepfakes*, mettant en lumière les algorithmes et techniques les plus couramment utilisés.

À travers l'approche et le protocole d'expérimentation des outils logiciels de génération, nous avons fourni une méthodologie structurée pour tester ces technologies.

Enfin, les résultats expérimentaux obtenus ont été analysés, offrant une évaluation subjective et objective de la qualité des *deepfakes* produits. Cette étude souligne l'importance des *deepfakes* qui sont en constante évolution dans divers domaines tout en mettant en évidence les défis techniques et éthiques associés à leur création et utilisation.

En résumé, cette étude apporte une contribution importante en maîtrise et compréhension des différentes méthodes et outils de génération des *deepfakes* audio et vidéo. Parvenir à générer des *deepfakes* audio réalistes en différentes langues en combinant différentes méthodes de génération audio (TTS + RVC) et entraîner des modèles de conversion de voix en voix de personnes parlant arabe (darija), comme Hafid Derradji et le président Tebboune. Réaliser une évaluation complète (subjective et objective) des différents *deepfakes* générés.

En répondant à la problématique du projet, nous avons franchi une étape importante. Toutefois, il est essentiel de reconnaître qu'il existe encore des possibilités d'amélioration à explorer. Dans cette section, nous exposons quelques pistes d'amélioration possibles qui s'inscrivent dans la continuité de nos travaux.

1 Conception d'un *workflow* (pipeline) qui automatise le processus de génération des *deepfakes* de bout en bout, en transformant une vidéo

originale avec un texte en entrée en une vidéo de *deepfakes* réaliste.

2 Amélioration des architectures de réseaux neuronaux des différentes techniques de génération pour améliorer la qualité et le réalisme des *deepfakes*.

3 Automatisation du remplacement de la tête dans les méthodes de synchronisation labiale basées sur les Nerf.

4 Exploration de la possibilité de génération de *deepfakes* pour des personnes en mouvement dans une vidéo ou bien dont les visages apparaissent de profil.

Bibliographie

1. ELLUL, Jacques. *Propaganda : The Formation of Men's Attitudes*. Trad. par KELLEN, Konrad ; LERNER, Jean. New York : Vintage Books, 1973. ISBN 978-0-394-71874-3.
2. POMPONIK.PL. *Andrzej Kozera zniknął z Dziennika Telewizyjnego z dnia na dzień*. 2017. Consulté le 25 juin 2024.
3. KING, David. *The Commissar Vanishes : The Falsification of Photographs and Art in Stalin's Russia*. 1re éd. New York : Metropolitan Books, 1997. ISBN 978-0-8050-5294-7.
4. IMGUR. *Stalinist Photographic Manipulation : Original photo from 1926 Leningrad*. n.d. Consulté le 25 juin 2024.
5. THIES, Justus ; ZOLLHOFER, Michael ; STAMMINGER, Marc ; THEOBALT, Christian ; NIESSNER, Matthias. Face2face : Real-time face capture and reenactment of rgb videos. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 2387-2395.
6. TOLOSANA, Ruben ; VERA-RODRIGUEZ, Ruben ; FIERREZ, Julian ; MORALES, Aythami ; ORTEGA-GARCIA, Javier. DeepFakes and Beyond : A Survey of Face Manipulation and Fake Detection. *Information Fusion*. 2020, t. 64, p. 131-148. Disp. à l'adr. DOI : 10.1016/j.inffus.2020.06.014.
7. LI, Yuezun ; CHANG, Ming-Ching ; QI, Honggang ; CHEN, Jingjing ; SOLEYMANI, Somayeh ; LYU, Siwei. In the Face of Deception : A Survey on Face Manipulation Detection. *IEEE Transactions on Information Forensics and Security*. 2020, t. 15, p. 2718-2741.
8. GÜERA, David ; DELP, Edward J. Deepfake video detection using recurrent neural networks. In : *2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS)*. IEEE, 2018, p. 1-6.
9. GUPTA, Surbhi ; MOHAN, Neeraj ; KAUSHAL, Priyanka. Passive image forensics using universal techniques : a review. *Artificial Intelligence Review*. 2022, t. 55, no 3, p. 1629-1679.
10. PAVAN KUMAR, MR ; JAYAGOPAL, Prabhu. Generative adversarial networks : a survey on applications and challenges. *International Journal of Multimedia Information Retrieval*. 2021, t. 10, no 1, p. 1-24.

11. CHOI, Yunjey ; CHOI, Minje ; KIM, Munyoung ; HA, Jung-Woo ; KIM, Sunghun ; CHOO, Jaegul. Stargan : Unified generative adversarial networks for multi-domain image-to-image translation. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, p. 8789-8797.

12. SHARMA, Mridul ; KAUR, Mandeep. A review of Deepfake technology : an emerging AI threat. *Soft Computing for Security Applications : Proceedings of ICSCS 2021*. 2022, p. 605-619.

13. BREGLER, Christoph ; COVELL, Michele ; SLANEY, Malcolm. Video rewrite : Driving visual speech with audio. In : *Seminal Graphics Papers : Pushing the Boundaries, Volume 2*. 2023, p. 715-722.

14. NGUYEN, TT ; NGUYEN, CM ; NGUYEN, DT ; NGUYEN, DT ; NAHAVANDI, S. Deep learning for deepfakes creation and detection. arXiv 2019. *arXiv preprint arXiv :1909.11573*. 2019.

15. JOHNSON, Deborah G ; DIAKOPOULOS, Nicholas. What to do about deepfakes. *Communications of the ACM*. 2021, t. 64, n° 3, p. 33-35.

19. SUWAJANAKORN, Supasorn ; SEITZ, Steven M ; KEMELMACHER-SHLIZERMAN, Ira. Synthesizing obama : learning lip sync from audio. *ACM Transactions on Graphics (ToG)*. 2017, t. 36, n° 4, p. 1-13.

20. CHAN, Caroline ; GINOSAR, Shiry ; ZHOU, Tinghui ; EFROS, Alexei A. Everybody dance now. In : *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, p. 5933-5942.

21. SIAROHIN, Aliaksandr ; LATHUILIÈRE, Stéphane ; TULYAKOV, Sergey ; RICCI, Elisa ; SEBE, Nicu. *First order motion model for image animation*. T. 32. 2019.

22. ZHOU, Hang ; SUN, Yasheng ; WU, Wayne ; LOY, Chen Change ; WANG, Xiaogang ; LIU, Ziwei. Pose-controllable talking face generation by implicitly modularized audio-visual representation. In : *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, p. 4176-4186.

23. KIM, Hyeongwoo ; GARRIDO, Pablo ; TEWARI, Ayush ; XU, Weipeng ; THIES, Justus ; NIESSNER, Matthias ; PÉREZ, Patrick ; RICHARDT, Christian ; ZOLLHÖFER, Michael ; THEOBALT, Christian. Deep video portraits. *ACM transactions on graphics (TOG)*. 2018, t. 37, n° 4, p. 1-14.

24. LU, Yuanxun ; CHAI, Jinxiang ; CAO, Xun. Live speech portraits : real-time photorealistic talking-head animation. *ACM Transactions on Graphics (TOG)*. 2021, t. 40, n° 6, p. 1-17.

25. LAHIRI, Avisek ; KWATRA, Vivek ; FRUEH, Christian ; LEWIS, John ; BREGLER, Chris. Lipsync3d : Data-efficient learning of personalized 3d talking faces from video using pose and lighting normalization. In : *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, p. 2755-2764.

26. WESTERLUND, Mika. The emergence of deepfake technology : A review. *Technology innovation management review*. 2019, t. 9, n° 11.

27. GREENGARD, Samuel. Will deepfakes do deep damage ? *Communications of the ACM*. 2019, t. 63, n° 1, p. 17-19.

28. LEE, YoungAh ; HUANG, Kuo-Ting ; BLOM, Robin ; SCHRINER, Rebecca ; CICCARELLI, Carl A. To believe or not to believe : framing analysis of content and audience response of top 10 deepfake videos on youtube. *Cyberpsychology, Behavior, and Social Networking*. 2021, t. 24, n° 3, p. 153-158.

29. VAN DEN OORD, Aaron ; DIELEMAN, Sander ; ZEN, Heiga ; SIMONYAN, Karen ; VINYALS, Oriol ; GRAVES, Alex ; KALCHBRENNER, Nal ; SENIOR, Andrew ; KAVUKCUOGLU, Koray et al. Wavenet : A generative model for raw audio. *arXiv preprint arXiv :1609.03499*. 2016, t. 12.

30. WANG, Yuxuan ; SKERRY-RYAN, RJ ; STANTON, Daisy ; WU, Yonghui ; WEISS, Ron J ; JAITLY, Navdeep ; YANG, Zongheng ; XIAO, Ying ; CHEN, Zhifeng ; BENGIO, Samy et al. Tacotron : Towards end-to-end speech synthesis. *arXiv preprint arXiv :1703.10135*. 2017.

31. ARIK, Sercan Ö ; CHRZANOWSKI, Mike ; COATES, Adam ; DIAMOS, Gregory ; GIBIANSKY, Andrew ; KANG, Yongguo ; LI, Xian ; MILLER, John ; NG, Andrew ; RAIMAN, Jonathan et al. Deep voice : Real-time neural text-to-speech. In : *International conference on machine learning*. PMLR, 2017, p. 195-204.

32. WANG, Run ; JUEFEI-XU, Felix ; HUANG, Yihao ; GUO, Qing ; XIE, Xiaofei ; MA, Lei ; LIU, Yang. Deepsonar : Towards effective and robust detection of ai-synthesized fake voices. In : *Proceedings of the 28th ACM international conference on multimedia*. 2020, p. 1207-1216.

33. ARIK, Sercan ; CHEN, Jitong ; PENG, Kainan ; PING, Wei ; ZHOU, Yanqi. Neural voice cloning with a few samples. *Advances in neural information processing systems*. 2018, t. 31.

38. IPROOV. *Deepfakes : statistiques, solutions, protection biométrique*. 2022.

40. KINGMA, Diederik P ; WELLING, Max. Auto-encoding variational bayes. *arXiv preprint arXiv :1312.6114*. 2013.

41. MAKHZANI, Alireza ; SHLENS, Jonathon ; JAITLY, Navdeep ; GOODFELLOW, Ian ; FREY, Brendan. Adversarial autoencoders. *arXiv preprint arXiv :1511.05644*. 2015.

42. SUTSKEVER, Ilya ; VINYALS, Oriol ; LE, Quoc V. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*. 2014, t. 27.

43. SIDDIQUE, Nahian ; PAHEDING, Sidike ; ELKIN, Colin P ; DEVABHAKTUNI, Vijay. U-net and its variants for medical image segmentation : A review of theory and applications. *Ieee Access*. 2021, t. 9, p. 82031-82057.

44. NIELSEN, Michael A. *Neural networks and deep learning*. T. 25. Determination press San Francisco, CA, USA, 2015.

45. CUNNINGHAM, James D ; SHU, Dule ; SIMPSON, Timothy W ; TUCKER, Conrad S. A sparsity preserving genetic algorithm for extracting diverse functional 3D designs from deep generative neural networks. *Design Science*. 2020, t. 6, e11.

46. BOTTOU, Léon et al. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*. 1991, t. 91, n° 8, p. 12.

47. LE, Quoc V ; NGIAM, Jiquan ; COATES, Adam ; LAHIRI, Abhik ; PROCHNOW, Bobby ; NG, Andrew Y. On optimization methods for deep learning. In : *Proceedings of the 28th international conference on international conference on machine learning*. 2011, p. 265-272.

48. GUO, Xifeng ; LIU, Xinwang ; ZHU, En ; YIN, Jianping. Deep clustering with convolutional autoencoders. In : *Neural Information Processing : 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part II 24*. Springer, 2017, p. 373-382.

50. KINGMA, Durk P ; MOHAMED, Shakir ; JIMENEZ REZENDE, Danilo ; WELLING, Max. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*. 2014, t. 27.

51. GOODFELLOW, Ian ; BENGIO, Yoshua ; COURVILLE, Aaron. *Deep learning*. MIT press, 2016.

53. DI SIPIO, Riccardo ; GIANNELLI, Michele Faucci ; HAGHIGHAT, Sana Ketabchi ; PALAZZO, Serena. DijetGAN : a generative-adversarial network approach for the simulation of QCD dijet events at the LHC. *Journal of high energy physics*. 2019, t. 2019, n° 8.

54. WALCZYNA, Tomasz ; PIOTROWSKI, Zbigniew. Quick Overview of Face Swap Deep Fakes. *Applied Sciences*. 2023, t. 13, n° 11, p. 6711.

55. LU, Xin ; KANG, Xin ; NISHIDE, Shun ; REN, Fuji. Object detection based on SSD-ResNet. In : *2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)*. IEEE, 2019, p. 89-92.

56. ZHANG, Kaipeng ; ZHANG, Zhanpeng ; LI, Zhifeng ; QIAO, Yu. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*. 2016, t. 23, n° 10, p. 1499-1503.

57. ZHANG, Shifeng ; ZHU, Xiangyu ; LEI, Zhen ; SHI, Hailin ; WANG, Xiaobo ; LI, Stan Z. S3fd : Single shot scale-invariant face detector. In : *Proceedings of the IEEE international conference on computer vision*. 2017, p. 192-201.

58. BULAT, Adrian ; TZIMIROPOULOS, Georgios. How far are we from solving the 2d & 3d face alignment problem ?(and a dataset of 230,000 3d facial landmarks). In : *Proceedings of the IEEE international conference on computer vision*. 2017, p. 1021-1030.

59. FENG, Yao ; WU, Fan ; SHAO, Xiaohu ; WANG, Yanfeng ; ZHOU, Xi. Joint 3d face reconstruction and dense alignment with position map regression network. In : *Proceedings of the European conference on computer vision (ECCV)*. 2018, p. 534-551.

60. UMEYAMA, Shinji. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*. 1991, t. 13, n° 04, p. 376-380.

62. IGLOVIKOV, Vladimir ; SHVETS, Alexey. Ternausnet : U-net with vgg11 encoder pre-trained on imagenet for image segmentation. *arXiv preprint arXiv :1801.05746*. 2018.

63. MASOOD, Momina ; NAWAZ, Mariam ; MALIK, Khalid Mahmood ; JAVED, Ali ; IRTAZA, Aun ; MALIK, Hafiz. Deepfakes generation and detection : State-of-the-art, open challenges, countermeasures, and way forward. *Applied intelligence*. 2023, t. 53, n° 4, p. 3974-4026.

64. ZEN, Heiga ; SENIOR, Andrew ; SCHUSTER, Mike. Statistical parametric speech synthesis using deep neural networks. In : *2013 ieee international conference on acoustics, speech and signal processing*. IEEE, 2013, p. 7962-7966.

65. FAN, Yuchen ; QIAN, Yao ; XIE, Feng-Long ; SOONG, Frank K. TTS synthesis with bidirectional LSTM based recurrent neural networks. In : *Fifteenth annual conference of the international speech communication association*. 2014.

66. ALMUTAIRI, Zaynab ; ELGIBREEN, Hebah. A review of modern audio deepfake detection methods : Challenges and future directions. *Algorithms*. 2022, t. 15, n° 5, p. 155.

68. VAN DEN OORD, Aäron ; KALCHBRENNER, Nal ; KAVUKCUOGLU, Koray. Pixel recurrent neural networks. In : *International conference on machine learning*. PMLR, 2016, p. 1747-1756.

69. PRAJWAL, KR ; MUKHOPADHYAY, Rudrabha ; NAMBOODIRI, Vinay P ; JAWAHAR, CV. A lip sync expert is all you need for speech to lip generation in the wild. In : *Proceedings of the 28th ACM international conference on multimedia*. 2020, p. 484-492.

70. MILDENHALL, Ben ; SRINIVASAN, Pratul P ; TANCIK, Matthew ; BARRON, Jonathan T ; RAMAMOORTHI, Ravi ; NG, Ren. Nerf : Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*. 2021, t. 65, n° 1, p. 99-106.

72. ZHANG, Zhimeng ; HU, Zhipeng ; DENG, Wenjin ; FAN, Changjie ; LV, Tangjie ; DING, Yu. Dinet : Deformation inpainting network for realistic face visually dubbing on high resolution video. In : *Proceedings of the AAAI Conference on Artificial Intelligence*. 2023, t. 37, p. 3543-3551. N° 3.

73. HANNUN, Awni ; CASE, Carl ; CASPER, Jared ; CATANZARO, Bryan ; DIAMOS, Greg ; ELSEN, Erich ; PRENGER, Ryan ; SATHEESH, Sanjeev ; SENGUPTA, Shubho ; COATES, Adam et al. Deep speech : Scaling up end-to-end speech recognition. *arXiv preprint arXiv :1412.5567*. 2014.

74. YE, Zhenhui ; JIANG, Ziyue ; REN, Yi ; LIU, Jinglin ; HE, Jinzheng ; ZHAO, Zhou. Geneface : Generalized and high-fidelity audio-driven 3d talking face synthesis. *arXiv preprint arXiv :2301.13430*. 2023.

75. HSU, Wei-Ning ; BOLTE, Benjamin ; TSAI, Yao-Hung Hubert ; LAKHOTIA, Kushal ; SALAKHUTDINOV, Ruslan ; MOHAMED, Abdelrahman. Hubert : Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. 2021, t. 29, p. 3451-3460.

76. YE, Zhenhui ; HE, Jinzheng ; JIANG, Ziyue ; HUANG, Rongjie ; HUANG, Jiawei ; LIU, Jinglin ; REN, Yi ; YIN, Xiang ; MA, Zejun ; ZHAO, Zhou. Geneface++ : Generalized and stable real-time audio-driven 3d talking face generation. *arXiv preprint arXiv :2305.00787*. 2023.

78. YOSHIMURA, Takayoshi ; TOKUDA, Keiichi ; MASUKO, Takashi ; KOBAYASHI, Takao ; KITAMURA, Tadashi. Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In : *Sixth European conference on speech communication and technology*. 1999.

80. LIU, Ziwei ; LUO, Ping ; WANG, Xiaogang ; TANG, Xiaoou. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*. 2018, t. 15, n° 2018, p. 11.

81. YAN, Wilson ; ZHANG, Yunzhi ; ABBEEL, Pieter ; SRINIVAS, Aravind. Videogpt : Video generation using vq-vae and transformers. *arXiv preprint arXiv :2104.10157*. 2021.

84. CHUNG, J. S. ; ZISSERMAN, A. Lip Reading in the Wild. In : *Asian Conference on Computer Vision*. 2016.

86. CHUNG, Joon Son ; NAGRANI, Arsha ; ZISSERMAN, Andrew. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv :1806.05622*. 2018.

88. SARA, Umme ; AKTER, Morium ; UDDIN, Mohammad Shorif. Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study. *Journal of Computer and Communications*. 2019, t. 7, n° 3, p. 8-18.

89. LI, Zhi ; AARON, Anne ; KATSAVOUNIDIS, Ioannis ; MOORTHY, Anush ; MANOHARA, Megha et al. Toward a practical perceptual video quality metric. *The Netflix Tech Blog*. 2016, t. 6, n° 2, p. 2.

90. SHEIKH, Hamid R ; BOVIK, Alan C. Image information and visual quality. *IEEE Transactions on image processing*. 2006, t. 15, n° 2, p. 430-444.

91. LI, Songnan ; ZHANG, Fan ; MA, Lin ; NGAN, King Ngi. Image quality assessment by separately evaluating detail losses and additive impairments. *IEEE Transactions on Multimedia*. 2011, t. 13, n° 5, p. 935-949.

93. WU, Pengfei ; WANG, Zheng ; PAN, Zhiming ; WANG, Weilun. LIPFD-NPU : Low-overhead Instruction-driven Permanent Fault Detection for Neural Processing Unit. In : *2022 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA)*. 2022, p. 22-23. Disp. à l'adr. DOI : 10.1109/ICTA56932.2022.9963136.

94. SHIOHARA, Kaede ; YAMASAKI, Toshihiko. Detecting deepfakes with self-blended images. In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, p. 18720-18729.

95. CAO, Junyi ; MA, Chao ; YAO, Taiping ; CHEN, Shen ; DING, Shouhong ; YANG, Xiaokang. End-to-end reconstruction-classification learning for face forgery detection. In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, p. 4113-4122.

Webographie

16. FAKEAPP. *FakeApp 2.2.0* [https://www.malavida.com/en/soft/fakeapp/]. [s. d.]. Accessed September 18, 2020.

17. DEEPFAKES. *Faceswap : Deepfakes software for all* [https://github.com/deepfakes/faceswap]. [s. d.]. Accessed September 08, 2020.

18. DEEPFACELAB. *DeepFaceLab* [https://github.com/iperov/DeepFaceLab]. [s. d.]. Accessed August 18, 2020.

34. JOURNAL, Search Engine. *How Deepfake Technology Is Impacting Digital Marketing*. 2024. Aussi disponible à l'adresse : https://www.searchenginejournal.com/deepfake-technology-digital-marketing/454395/.

35. YOUTUBE. *Deepfake | Cara is Leia / Rogue One / side by side comparison*. n.d. Aussi disponible à l'adresse : https://www.youtube.com/watch?v=gvzqfoB9R3o. Consulté le 25 juin 2024.

36. MONDAQ. *Deepfakes in Advertising : Who's Behind the Camera ?* 2024. Aussi disponible à l'adresse : https://www.mondaq.com/uk/advertising-marketing--branding/1433508/deepfakes-in-advertising--whos-behind-the-camera.

37. SENTINELONE. *What are Deepfakes ?* 2024. Aussi disponible à l'adresse : https://www.sentinelone.com/cybersecurity-101/deepfakes/.

39. NEOSOFT. *Techniques d'augmentation de dataset avec les Variational Autoencoders (VAE)* [https://www.neosoft.fr/nos-publications/blog-tech/techniques-augmentation-dataset-vae/]. n.d. Aussi disponible à l'adresse : https://www.neosoft.fr/nos-publications/blog-tech/techniques-augmentation-dataset-vae/. Consulté le 25 juin 2024.

49. SCRATCH, Deep Learning from. *Autoencoders* [https://deeplearningfromscratch.wordpress.com/2018/07/30/autoencoders/]. 2018. Aussi disponible à l'adresse : https://deeplearningfromscratch.wordpress.com/2018/07/30/autoencoders/. Consulté le 25 juin 2024.

52. MATHWORKS. *Generative Adversarial Networks (GAN)* [https://nl.mathworks.com/discovery/generative-adversarial-networks.html]. n.d. Aussi disponible à l'adresse : https://nl.mathworks.com/discovery/generative-adversarial-networks.html. Consulté le 25 juin 2024.

61. COMMUNITY, Stack Overflow. *How to align first and extract second face alignment using RetinaFace without deep learning in Python ?* [https://stackoverflow.com/questions/76729320/how-to-align-first-and-extract-second-face-alignment-using-retinaface-without]. 2023. [visité le 2024-06-25]. Disp. à l'adr. : https://stackoverflow.com/questions/76729320/how-to-align-first-and-extract-second-face-alignment-using-retinaface-without. Stack Overflow.

67. BLACK, Alan ; TAYLOR, Paul ; CALEY, Richard ; CLARK, Rob. *The festival speech synthesis system*. 1998.

71. SAUNDERS, Jack. *Wav2Lip : Generalized Lip Sync Models* [https://medium.com/@jacksaunders909/wav2lip-generalized-lip-sync-models-e0effc4e8ed3]. n.d. Aussi disponible à l'adresse : https://medium.com/@jacksaunders909/wav2lip-generalized-lip-sync-models-e0effc4e8ed3. Consulté le 25 juin 2024.

77. PROJECT, RVC. *Retrieval-based Voice Conversion WebUI* [https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI]. n.d. GitHub.

79. NVLABS. *FFHQ-Dataset* [https://github.com/NVlabs/ffhq-dataset]. n.d. GitHub.

82. DASWER123. *XTTS-Webui* [https://github.com/daswer123/xtts-webui?tab=readme-ov-file]. n.d. GitHub.

83. IAHISPANO. *Applio* [https://github.com/IAHispano/Applio]. n.d. GitHub.

85. MRZZM. *HDTF* [https://github.com/MRzzm/HDTF]. n.d. GitHub.

87. MLTFRAMEWORK. *Shotcut* [https://github.com/mltframework/shotcut]. n.d. GitHub.

92. FIFONIK. *FFMetrics — yet another program for video Visual Quality Metrics visualization* [https://github.com/fifonik/FFMetrics]. [s. d.]. 2022.

Annexes

Définitions

Divergence de Kullback-Leibler :

En théorie des probabilités et en théorie de l'information, la divergence de Kullback-Leibler (divergence K-L ou encore entropie relative) est une mesure de dissimilarité entre deux distributions de probabilités.
Considérons deux distributions de probabilités P et Q. Typiquement, P représente les données, les observations, ou une distribution de probabilités calculée avec précision. La distribution Q représente typiquement une théorie, un modèle, une description ou une approximation de P. La divergence de Kullback-Leibler s'interprète comme la différence moyenne du nombre de bits nécessaires au codage d'échantillons de P en utilisant un code optimisé pour Q plutôt que le code optimisé pour P.

Spectrogramme Mel :

Un spectrogramme mel est une variante du spectrogramme couramment utilisée dans les tâches de traitement de la parole et d'apprentissage automatique. Il est similaire à un spectrogramme en ce sens qu'il montre le contenu en fréquence d'un signal audio au fil du temps, mais sur un axe de fréquence différent. Dans un spectrogramme standard, l'axe de fréquence est linéaire et est mesuré en hertz (Hz). Cependant, le système auditif humain est plus sensible aux changements dans les basses fréquences que dans les fréquences plus élevées, et cette sensibilité diminue logarithmiquement à mesure que la fréquence augmente. L'échelle mel est une échelle perceptuelle qui se rapproche de la réponse en fréquence non linéaire de l'oreille humaine. Pour créer un spectrogramme mel, le STFT est utilisé comme auparavant, divisant l'audio en segments courts pour obtenir une séquence de spectres de fréquence. De plus, chaque spectre est envoyé à travers un

ensemble de filtres, appelé mel filterbank, pour transformer les fréquences à l'échelle mel.

HMM (*Hidden Markov Model*) :

Un modèle statistique appelé Modèle de Markov Caché (HMM) est utilisé pour analyser des processus séquentiels etdécrire des systèmes avec des états non observables changeant au fil du temps. Il repose sur l'idée qu'il existe un processus sous-jacent avec des états cachés, chacun ayant un résultat connu. Le modèle définit les probabilités de transition entre les états cachés et d'émission de symboles observables.
En raison de leur capacité supérieure à capturer l'incertitude et les dépendances temporelles. Les HMM sont utiles pour modéliser des systèmes dynamiques et prévoir les états futurs basés sur les séquences observées grâce à leur flexibilité. .

Umeyama (*Kabsch algorithm*) :

L'algorithme de Kabsch, également connu sous le nom d'algorithme de Kabsch-Umeyama, est une méthode pour calculer la matrice de rotation optimale qui minimise la déviation quadratique moyenne (RMSD) entre deux ensembles appariés de points.
L'algorithme ne calcule que la matrice de rotation, mais nécessite également le calcul d'un vecteur de translation.