Projet de fin d'études - Rapport de stage CNPEN/CCNE

Enjeux d'éthiques liés aux algorithmes d'apprentissage machine et mise en œuvre technique de recommandations

Mélanie Gornet

Catherine Tessier

Maître de recherche, référente intégrité scientifique
et éthique de la recherche à l'ONERA
membre du CNPEN

Claude Kirchner

Directeur de recherche émérite Inria
membre du CCNE
directeur du CNPEN

Olivier Grisel

Chercheur en apprentissage machine
équipe PARIETAL Inria-CEA

Septembre 2021

Remerciements

Derrière le formalisme de cette section se cache une réelle volonté de dire merci à toutes celles et ceux qui m'ont accompagnée tout au long de ce stage et même avant.

Merci à Catherine, sans qui ce stage n'aurait pas eu lieu. Mais plus encore que pour cette opportunité, merci pour ton accompagnement sans faille tout au long de ces six mois. Merci, bien sûr, tout autant à Claude, qui m'a accueillie au CNPEN et s'est assuré régulièrement que le stage se déroulait au mieux. Je voulais également vous remercier tous les deux pour tout ce que j'ai pu apprendre avec vous et qui ne saurait se résumer en quelques phrases (ni même en un rapport de stage).

Merci aussi à Olivier qui m'a guidée dans mon apprentissage des techniques de reconnaissance faciale et a pris le temps de relire et debugger mon code avec moi lorsque j'étais bloquée.

Merci à tous les membres du CNPEN avec qui j'ai pu échanger, sur mon travail ou le leur, sur l'éthique ou sur l'apprentissage machine, j'ai beaucoup grandi de ces discussions.

Merci à Camille, Hanna et Alexia d'avoir été plus que des collègues de travail et de m'avoir initiée à vos disciplines respectives.

Je voulais également remercier Dennis, d'avoir fait de cette année de SDD un souvenir mémorable.

Merci enfin à ma famille, à mes ami.e.s, à Alice. Merci à mes camarades de classe de SciencesPo et de l'ISAE-SUPAERO ainsi qu'à mes professeur.e.s, d'ici et d'ailleurs.

Résumé

Du 6 avril au 5 octobre 2021, j'ai eu la chance de pouvoir effectuer mon stage de fin d'études au Comité National Pilote d'Éthique du Numérique (CNPEN), placé aujourd'hui sous l'égide du Comité Consultatif National d'Éthique pour les sciences de la vie et de la santé (CCNE).

Durant ces six mois, je me suis penchée sur les enjeux d'éthique de l'apprentissage machine et la mise en œuvre technique de recommandations exprimées dans les textes internationaux relatifs à l'éthique de l'apprentissage machine. Ces textes parlent tour à tour de « valeurs », « principes » ou « exigences » de façon interchangeable et sans donner d'indication pour les prendre en compte de manière concrète au sein des systèmes numériques. La démarche du stage s'inscrit dans la volonté de converger vers des exigences opérationnelles pour les logiciels d'authentification par reconnaissance faciale. Le cas d'usage a été développé dans le cadre d'une autosaisine du CNPEN relative aux enjeux d'éthique de la reconnaissance faciale, posturale et comportementale.

Ainsi, chaque choix de conception comporte des implications, des enjeux et est porteur de certaines valeurs. En choisissant de construire un système de telle ou telle façon, le concepteur privilégie des valeurs par rapport à d'autres. Notre travail consiste à identifier où ces choix s'effectuent et tenter d'identifier les directions à prendre pour respecter certaines valeurs. Pour chaque choix, nous avons, autant que possible, analysé son impact et argumenté la solution retenue.

L'étude a abouti à une analyse de différents paramètres, de leur dépendance les uns envers les autres (cf figure 42) et de leur impact sur la performance et l'équité du modèle. L'analyse post-traitement n'a pas révélé de biais manifeste, ne signifiant pas que le système en soit dénué mais que ces études ne les exposent pas toujours. Il convient toutefois de toujours ramener la réflexion au contexte dans lequel sera déployé le système et de questionner avant tout sa raison d'être.

J'ai été suivie dans ce travail par Catherine Tessier, maître de recherche, référente intégrité scientifique et éthique de la recherche à l'ONERA et membre du CNPEN ; ainsi que par Claude Kirchner, directeur de recherche émérite Inria, membre du CCNE et directeur du CNPEN. Ce travail a été effectué avec le support technique d'Olivier Grisel, chercheur en apprentissage machine dans l'équipe PARIETAL, commune entre l'Inria et le CEA.

Table des matières

Remerciements **1**

Introduction **1**

I Contexte du stage **5**

I.1 L'éthique du numérique et de l' « intelligence artificielle » **5**
I.1.1 Qu'est-ce que l'éthique du numérique et pourquoi en a-t-on besoin ? . . . 5
I.1.2 Qu'entendons-nous par « IA » ? . . . 5
I.1.3 Un consensus international . . . 5
I.1.4 La nécessité d'une réflexion transversale et multidisciplinaire . . . 6

I.2 Le Comité National Pilote d'Éthique du Numérique **6**
I.2.1 Qu'est-ce que le CNPEN ? . . . 6
I.2.2 Organisation du Comité . . . 7

I.3 Groupe de travail sur la reconnaissance automatique **7**
I.3.1 Autosaisine . . . 7
I.3.2 Mode de fonctionnement du groupe de travail . . . 8

I.4 Descriptif du stage **8**
I.4.1 Objectifs . . . 8
I.4.2 Activités . . . 8

II Activités au sein du Comité **9**

II.1 Contributions **9**
II.1.1 Gestion et organisation des réunions . . . 9
II.1.2 Secrétariat des réunions et auditions . . . 9
II.1.2.1 Comptes rendus . . . 9
II.1.2.2 Préparer les auditions . . . 10
II.1.3 Questionnaire de consultation et d'appel à contributions . . . 10
II.1.3.1 L'exemple du GT chatbots . . . 10
II.1.3.2 Rédaction du questionnaire . . . 10
II.1.4 Aide complémentaire . . . 11
II.1.5 Travail de recherche . . . 11

II.2 Présentations et suivis du travail **12**
II.2.1 Suivis des encadrant.e.s . . . 12
II.2.2 Autres retours . . . 12

II.3 Activités annexes **12**
II.3.1 Séminaires . . . 12
II.3.2 Colloque . . . 12
II.3.3 Royaumont . . . 12

III Un travail sur les valeurs morales **13**

III.1 Cas choisi : authentification par reconnaissance faciale **13**

III.2 Critique de l'état de l'art en matière d'« éthique de l'IA » **13**
III.2.1 Un vocabulaire à éclairer . . . 13
III.2.2 Des recommandations abstraites . . . 14

III.3 Démarche globale et objectif du travail de recherche **14**
III.3.1 Formuler des exigences opérationnelles . . . 14
III.3.2 Identifier les parties prenantes . . . 15
III.3.3 Recenser les valeurs . . . 16
III.3.4 Relation entre les valeurs . . . 16
III.3.5 Identifier les choix de conception . . . 17
III.3.6 Exemple fictif . . . 18

IV Développement du cas école **20**

IV.1 Fonctionnement du code **20**
IV.1.1 Utilisation normale . . . 20
IV.1.2 Données . . . 21
IV.1.3 Gabarits d'images . . . 21
IV.1.4 Objectif de l'entraînement . . . 21
IV.1.5 Générer des triplés . . . 23
IV.1.6 Déterminer la valeur seuil . . . 23
IV.1.7 Éléments de sortie . . . 24

IV.2 Organisation pratique **24**
IV.2.1 Valeurs et critères sélectionnés . . . 24
IV.2.2 Capacités computationnelles . . . 24
IV.2.3 Librairies Python . . . 25
IV.2.4 Suivi des expériences . . . 25
IV.2.5 Organisation des notebooks et des fichiers Python . . . 25
IV.2.6 Rendus . . . 27

IV.3 Choix effectués **27**
IV.3.1 Choix initiaux et postulats . . . 28
IV.3.1.1 Base de données utilisée . . . 28
IV.3.1.2 Attributs utilisés . . . 29
IV.3.1.3 Choix d'un réseau convolutionnel . . . 30
IV.3.1.4 Perte par triplés . . . 30
IV.3.2 Choix d'entraînement : recherche de l' *overfitting* . . . 31
IV.3.2.1 Fonction de perte . . . 31
IV.3.2.2 Prétraitement et régularisation . . . 31
IV.3.2.3 Valeur de marge . . . 32
IV.3.2.4 Structure du réseau . . . 33
IV.3.2.5 *Learning rate* . . . 34
IV.3.2.6 *Optimizer* et *scheduler* . . . 35
IV.3.2.7 Structure d'une epoch . . . 36
IV.3.2.8 Nombre de batchs . . . 37
IV.3.2.9 Nombre d'epochs . . . 37
IV.3.3 Choix de validation : généraliser les résultats . . . 37
IV.3.3.1 Augmentation . . . 37
IV.3.3.2 Séparation entre test et entraînement . . . 38
IV.3.3.3 Images négatives . . . 38
IV.3.3.4 *Dropout* et *Earlystopping* . . . 39
IV.3.4 Choix de test : déterminer le seuil et évaluer l'équité des résultats . . . 39
IV.3.4.1 Paires de test pour l'histogramme des distances . . . 40
IV.3.4.2 Valeur seuil . . . 40

IV.4 Analyse post-traitement **43**
IV.4.1 Critères de performance . . . 43
IV.4.1.1 Métriques de précision . . . 43
IV.4.1.2 Taux d'apprentissage (*triplet accuracy*) . . . 43

IV.4.1.3 Analyse des extrêmes et nouvelles données . . . 43
IV.4.1.4 Rapidité . . . 44
IV.4.2 Critère d'équité . . . 45
IV.4.2.1 Données utilisées pour l'étude de l'équité . . . 45
IV.4.2.2 Analyse des disparités . . . 46
IV.4.3 Pour aller plus loin . . . 46
IV.4.3.1 Comment approfondir l'étude ? . . . 46
IV.4.3.2 Quelles questions se poser ? . . . 48

V Conclusion **49**

V.1 Résultats du cas école **49**

V.2 Bilan du stage au Comité **50**

V.3 Perspectives **50**

Annexe A Réseau de neurones artificiels et convolutionnels **51**

Annexe B Apprentissage d'un réseau de neurones **53**

Annexe C Résultats des notebooks : *Train* **54**

Annexe D Résultats des notebooks : *Determine Threshold* **55**

Annexe E Résultats des notebooks : *Various tests* **56**

Annexe F Résultats des notebooks : *Fairness Study* **60**

Références **63**

Première partie
Contexte du stage

I.1 L'éthique du numérique et de l' « intelligence artificielle »

I.1.1 Qu'est-ce que l'éthique du numérique et pourquoi en a-t-on besoin ?

Depuis quelques années, le développement des outils numériques s'est accéléré, bouleversant notre rapport à l'autre et au monde, interrogeant notre vision de la propriété, du travail, de la santé, de la mémoire ou de la connaissance. Les données de masse nourrissent ces outils : elles sont collectées, traitées, stockées avec un volume et une vitesse sans précédent. Ces nouveaux enjeux sont sources d'opportunités mais comportent aussi des risques qui ne doivent pas être sous-estimés. Cultiver son autonomie et son libre-arbitre, garantir la démocratie, lutter contre la fracture sociale... Nombreux sont les points de vigilance qu'il incombe à la société et aux individus de surveiller.

La réflexion éthique est centrée sur des choix de valeurs pour éclairer nos choix individuels ou collectifs. Mais l'omniprésence du numérique et son évolution rapide dans le temps oblige à adapter cet effort de réflexion et font toute la spécificité de l'éthique du numérique. L'anthropomorphisation des machines ou encore l'impact des technologies d'apprentissage machine s'inscrivent dans cette singularité et doivent nous pousser à l'expression d'une réflexion citoyenne critique [19].

I.1.2 Qu'entendons-nous par « IA » ?

Aujourd'hui, on parle communément d'« intelligence artificielle » (IA), terme vague et inapproprié [29], pour qualifier une branche de l'informatique qui vise à doter une machine de capacités de simulation de comportements s'inspirant du vivant.

L'IA est définie au niveau international comme une liste d'approches :

— Les approches d'apprentissage automatique, y compris l'apprentissage supervisé, non supervisé et par renforcement, en utilisant une grande variété de méthodes, y compris l'apprentissage profond.

— Les approches fondées sur la logique et la connaissance, y compris la représentation des connaissances, la programmation inductive (logique), les bases de connaissances, les moteurs d'inférence et déductifs, le raisonnement (symbolique) et les systèmes experts.

— Les approches statistiques, l'estimation bayésienne, les méthodes d'optimisation. [7]

Pour ce stage et dans le cadre du programme de la filière SD-SDD (Sciences des Données et de la Décision) de l'ISAE-SUPAERO, il a été décidé de se focaliser sur la première catégorie d'approches.

I.1.3 Un consensus international

Avec la popularité croissante des outils d'apprentissage automatique, il devient de plus en plus essentiel de reconnaître leur rôle dans la transformation de la société et d'inciter, d'une part les chercheur.e.s et acteurs.trice.s de l'IA, et de l'autre les autorités publiques, à la prise en compte des enjeux d'éthique de ces technologies [11].

De nombreux textes ont ainsi vu le jour ces cinq dernières années, listant des valeurs, principes et critères à étudier lors du développement et plus généralement du cycle de vie d'un système d'apprentissage automatique. L'UNESCO recommande ainsi d'établir des mécanismes de supervision adaptés permettant d'évaluer les algorithmes sur la base de trois principes : la vérifiabilité, la traçabilité et l'explicabilité [22].

Au niveau Européen, la Commission européenne a identifié, dans son Livre Blanc, des risques pour les droits fondamentaux et la sécurité [6]. Le *High-Level Expert Group on Artificial Intelligence* (AI HLEG) a ainsi présenté en juillet 2020 une liste de critères d'exigence pour l'IA dite « de confiance », parmi lesquels on retrouve notamment la robustesse et la sécurité des systèmes, la transparence, la non-discrimination et l'équité [34]. En France, en 2017, la CERNA introduisait déjà des préconisations, telles que la prévention des attaques des systèmes numériques, la vigilance face aux données comportant un risque de discrimination ou encore l'explicabilité [11].

I.1.4 La nécessité d'une réflexion transversale et multidisciplinaire

Le 30 juin 2021, j'ai été invitée à la cérémonie de remise de diplômes de la *Paris School of International Affairs* (PSIA) de Sciences Po Paris. Je me souviens avoir été marquée par le discours de M. Enrico Letta, homme politique italien, nouveau secrétaire du Parti Démocrate italien, aujourd'hui ancien doyen de la PSIA et invité spécial de cette graduation 2021. Interrogé sur les enseignements de la crise de la Covid-19, M. Letta a répondu :

« Cette crise de la Covid nous a posé un sacré défi : le défi que je décris est quelque chose qui était en émergence depuis quelques années [...], c'est le besoin d'avoir des activités d'enseignement et de recherche, qui permettent de croiser les sciences dures et les sciences sociales et humaines. »

Prenant en exemple la pandémie, le changement climatique et le rôle des *Big Data* dans le monde, il a ainsi expliqué devant les élèves que les grands défis de notre temps ne se résoudraient pas sans une approche multidisciplinaire, avant de conclure : « Nous devons être plus ouverts aux autres disciplines. ».

Cette approche est fondamentale également pour le domaine numérique. Comme expliqué en I.1.1, l'éthique du numérique est une réflexion nécessaire et singulière. En particulier, les enjeux d'éthique autour de l'apprentissage machine ne peuvent se concevoir sans une connaissance technique approfondie de ces technologies. Lorsque nous cherchons une solution aux défis qu'elles soulèvent, nous ne pouvons nous limiter à proposer une solution technique, nous nous devons de regarder leurs conséquences sur l'être humain et sur la société dans son ensemble, car ce que nous manipulons aura une force d'impact très importante que les probabilités de sorties de notre algorithme ne sauraient correctement représenter.

Par exemple, les concepts d'équité et de non-discrimination ne peuvent pas simplement être restreints à des critères statistiques, ils doivent être également être regardés sous un prisme légal et social.

Néanmoins, considérer ces enjeux comme étant purement socio-politiques n'avancerait pas davantage le problème, qui est également technique : comment rendre un réseau de neurones transparent, sécurisé, ou comment réduire ses biais sont également des questions pertinentes pour l'éthique de l'apprentissage machine. Ainsi, sciences formelles et sciences humaines et sociales se doivent de collaborer pour adresser ces défis. Ces disciplines ne sont pas seulement complémentaires mais se nourrissent également des avancées de l'autre pour progresser. C'est cette approche pluridisciplinaire qui a été mise en avant pour ce stage, attaquant la problématique sous différents angles pour mieux la résoudre.

I.2 Le Comité National Pilote d'Éthique du Numérique

I.2.1 Qu'est-ce que le CNPEN ?

Le Comité National Pilote d'Éthique du Numérique (CNPEN) a été mis en place à la fin de l'année 2019 à la demande du Premier ministre, suite, entre autres choses, à la recommandation du rapport *Donner un sens à l'Intelligence artificielle* du député Cédric Villani [40]. Ce Comité Pilote est alors placé sous l'égide du Comité Consultatif National d'Ethique (CCNE) pour les sciences de la vie et de la santé. Le CCNE est une autorité administrative indépendante rattachée aux services du Premier ministre et chargée de « donner des avis sur les problèmes éthiques et les questions de société soulevés par les progrès de la connaissance » [1]. Dans le cas du CNPEN, ces activités s'effectuent dans le domaine numérique. Constitué de 27 membres issu.e.s d'horizons différents, le rôle du CNPEN est d'élaborer des avis sur les saisines qui lui sont adressées ainsi que de mener un travail de veille sur les enjeux d'éthique du numérique.

La crise sanitaire de la Covid-19, qui a suivi de peu sa création, a intensifié le débat citoyen sur les outils numériques et poussé le nouveau Comité Pilote à s'interroger sur ces enjeux. Il a publié durant cette période de crise sanitaire différents bulletins, sur la fraternité numérique, le suivi numérique des personnes [16], la lutte contre la désinformation [13] ou encore la télémédecine [14]. Saisi par messieurs Olivier Véran, ministre des solidarités et de la santé, et Cédric O, secrétaire d'État chargé du numérique, il a rendu son premier avis sur les outils numériques pour le déconfinement en mai 2020 [12]. Depuis, il continue de travailler sur les trois saisines initiales du Premier ministre : « véhicule autonome », « agents conversationnels », « diagnostic médical et intelligence artificielle ». L'avis « Le véhicule autonome : enjeux d'éthique » est disponible depuis mai 2021 [17].

En parallèle, il s'est autosaisi sur d'autres sujets, comme celui de la reconnaissance faciale, posturale et comportementale ou celui des plateformes de données de santé.

I.2.2 Organisation du Comité

Les 27 membres du Comité Pilote sont nommé.e.s par le directeur, lui-même choisi (pour cette phase pilote) par le président du CCNE. Les membres du CNPEN sont tous bénévoles et continuent à mener leur activité professionnelle en parallèle de leur travail au Comité. Le Bureau du Comité, chargé d'aider la direction dans les prises de décision, est constitué de différents membres qui se réunissent une fois par mois. L'organisation complète est donnée en figure 1.

Les réunions plénières se tiennent également tous les mois. Tous les membres, personnels et stagiaires sont convié.e.s. Lors de ces réunions, ils et elles reviennent sur les événements organisés, discutent des avancées des différents groupes de travail, votent les textes d'avis avant publication ou encore travaillent sur la proposition de pérennisation du Comité Pilote.

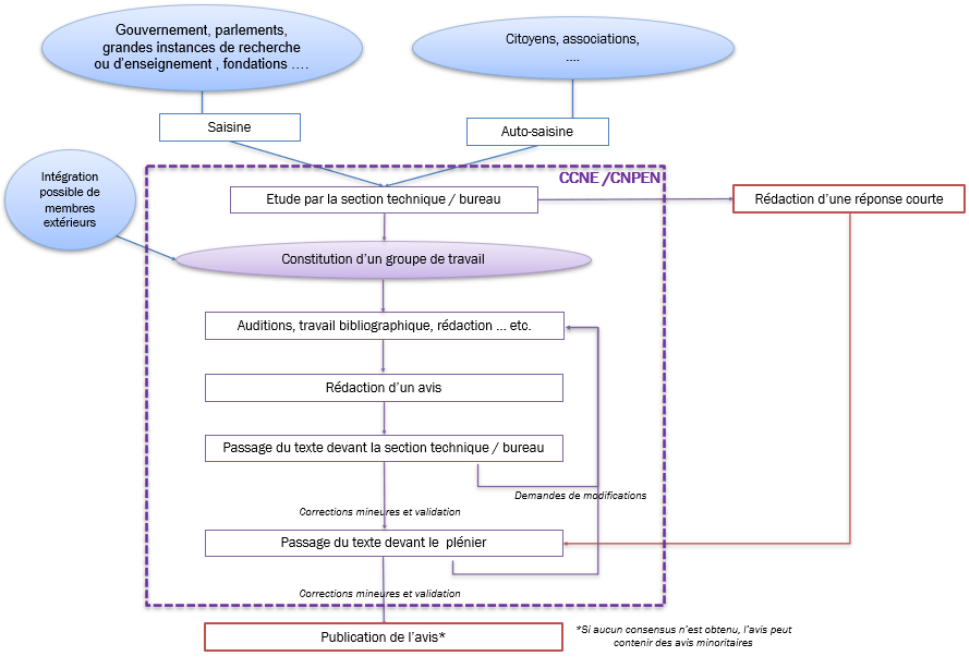


FIGURE 1 – Organisation du CCNE / CNPEN

I.3 Groupe de travail sur la reconnaissance automatique

I.3.1 Autosaisine

Le 2 avril 2021 est présenté au bureau du CNPEN un texte d'autosaisine nommé « Reconnaissance faciale, posturale et comportementale : entre questionnements et enjeux d'éthiques ». Il est décidé que c'est ce groupe de travail que je serai amenée à suivre durant mon stage (voir I.4.2).

L'autosaisine, comme de nombreux documents traitant des mêmes sujets [4, 10], classe la reconnaissance automatique en trois catégories :

— **l'authentification** : à savoir la confrontation des données biométriques enregistrées avec celles que présente une personne. C'est un système qui fonctionne en « 1 parmi 1 », sa sortie est binaire : si la sortie est « oui », le système accepte l'authentification, sinon, il la refuse. Les exemples les plus courants sont les systèmes de reconnaissance de visages pour déverrouiller un téléphone, entrer dans un lieu public ou privé ou passer un contrôle aux frontières (PARAFE) [9].

— **l'identification** : qui consiste à repérer une personne dans un ensemble de personnes. C'est un système qui fonctionne en « 1 parmi n ». Il peut s'utiliser pour rechercher une personne disparue ou en fuite, reconnaître des client.e.s régulier.e.s ou, au contraire, interdire l'accès d'un bâtiment ou d'un stade par exemple à certaines personnes.

— **la catégorisation** : qui vise à classer les individus, leur comportement ou leurs émotions à partir de leurs traits de visage ou de leur posture corporelle. C'est un système « p parmi n ». Certains systèmes font alors la corrélation (potentiellement non fondée) entre ces informations et des caractéristiques de l'individu comme son genre, son orientation sexuelle, politique, ou religieuse, son état émotionnel, etc.

Nous pourrions alors imaginer utiliser de tels systèmes pour reconnaître un comportement suspect ou détecter les mensonges par exemple.

Les deux rapporteur.e.s, responsables du groupe travail en charge de préparer un projet d'avis à partir de l'autosaisine, sont Karine Dognin-Sauze, ex-membre du Conseil National du Numérique (CNNum), présidente des Interconnectés, directrice générale de l'Institut Français des Administrateurs ; et Jean-Gabriel Ganascia, professeur d'informatique à Sorbonne Université, chercheur au LIP6 (équipe ACASA), président du comité d'éthique du CNRS, président du comité d'orientation du Cycle des Hautes Études de la Culture.

Le groupe de travail (GT) compte en tout onze membres du Comité et aucune personnalité extérieure. Il est aidé dans son travail par deux stagiaires.

Il doit rendre une proposition d'avis qui sera examinée en réunion plénière, puis, si elle est validée, l'avis sera publié fin 2021.

I.3.2 Mode de fonctionnement du groupe de travail

Le groupe de travail se réunit environ toutes les deux semaines par visioconférence. Les dates et horaires sont susceptibles de changer selon les disponibilités de chacun.e des membres (voir partie II.1.1).

Lors de ces réunions, les questions d'éthique posées par la reconnaissance faciale, posturale ou comportementale sont discutées. Ces réunions permettent également de s'accorder sur des textes à paraître, comme un chapitre [5], coécrit par les membres du GT, du futur livre du CNPEN, ou le questionnaire de consultation relatif à la reconnaissance automatique [20] (voir partie II.1.3.2).

Enfin, ces réunions peuvent être l'occasion d'auditions, consistant à rencontrer des personnes extérieures au Comité et travaillant sur un sujet relatif à la thématique du groupe de travail (ici, la reconnaissance faciale ou comportementale). Ces personnes peuvent être issues du milieu industriel, académique, ou de la fonction publique. L'échange commence souvent par une présentation de la personne auditionnée ainsi que de son travail, puis se poursuit par des questions des membres du GT. Une liste de personnes à auditionner est préparée par le GT, selon les besoins du groupe et les contacts de chacun.e. Une grille de questions, conçue par le GT, aide à orienter l'audition, même si régulièrement, d'autres questions plus spécifiques émergent et deviennent prioritaires.

Au vu de la crise sanitaire actuelle, toutes les réunions et auditions se sont tenues en ligne, via la plateforme de vidéoconférence Zoom.

I.4 Descriptif du stage

I.4.1 Objectifs

En s'intégrant dans le travail d'élaboration d'un avis du CNPEN mettant en jeu des techniques d'apprentissage machine et particulièrement de reconnaissance automatique, ce stage a pour objectif d'analyser les critères préconisés au niveau national, européen et international pour l'éthique de l'apprentissage machine et de les traduire en matière technique et critères de vérifiabilité. Dans un premier temps, il convenait de choisir le ou les critères les plus pertinents, puis d'étudier leurs implications éthiques et leur faisabilité technique. Le stage combinait ainsi à la fois de la recherche scientifique sur le sujet, ainsi qu'une réflexion politique et sociale. Il vise à développer des scénarios visant à la certification et régulation de ces technologies et sur la manière de garantir qu'elles soient fiables et respectent les critères précités.

I.4.2 Activités

Les activités menées durant le stage ont été les suivantes :

a le suivi d'un groupe de travail du CNPEN consacré à l'élaboration d'un avis relatif à l'utilisation d'algorithmes d'apprentissage machine pour la reconnaissance faciale ;

b Des recherches sur la signification et la mise en œuvre technique des recommandations nationales, européennes et internationales relatives à l'IA dite « de confiance » ;

c l'explicitation et la mise œuvre de quelques critères mentionnés dans les recommandations ;

d la participation à la rédaction de documents préparatoires à l'avis du CNPEN ;

e la rédaction du présent rapport dont la visée est d'éclairer la faisabilité technique des différents critères choisis.

Deuxième partie
Activités au sein du Comité

II.1 Contributions

II.1.1 Gestion et organisation des réunions

Le groupe de travail s'est réuni pour la première fois au mois d'avril 2021. Depuis, il se réunit environ toutes les deux semaines, souvent le vendredi matin. Ce rythme fut décidé par les rapporteur.e.s, en fonction du calendrier et des échéanciers fixés. Après une réunion préliminaire à la première réunion du GT avec les rapporteur.e.s, ces dernier.e.s m'ont demandé de proposer cet échéancier, notamment pour déterminer deux périodes principales jusqu'à la publication de l'avis : une première période, de réflexion et de conduite d'auditions, et une seconde de rédaction. En prenant en compte leur volonté et le calendrier effectif du Comité, je leur ai proposé une première ébauche d'échéancier.

Une fois les échéances fixées, il fallait trouver des dates de réunion où une majorité de membres soit disponible. Les membres étant bénévoles, ils et elles ont souvent un emploi du temps très rempli, consacré à leurs activités professionnelles. Pour trouver des créneaux pour se rassembler, il convenait d'organiser régulièrement des sondages à envoyer aux membres du groupe, via l'outil Framadate. Les réunions étaient quant à elles organisées sur Zoom.

Une fois les dates fixées, je devais les ajouter au calendrier du Comité (hébergé sur l'outil Framagenda), gérer les liens Zoom, créer des invitations agenda à envoyer aux membres pour qu'ils et elles puissent avoir accès à l'événement sur leur agenda personnel, envoyer des rappels réguliers, etc.

Enfin, il fallait tenir à jour l'espace de travail commun, partagé sur l'outil MyBox, géré par Inria. Il s'agit d'un espace de stockage et d'édition de documents en ligne, comme les *Drive* d'entreprises concurrentes. Il m'incombait d'ajouter les documents demandés dans leur dernière version, d'organiser cet espace en dossiers thématiques etc.

II.1.2 Secrétariat des réunions et auditions

II.1.2.1 Comptes rendus

Il m'a été demandé de rédiger un compte rendu de quelques pages de chaque réunion ou audition.

Ils permettent de souligner les moments forts de la réunion, de réunir les idées évoquées, etc. Ils servent autant aux membres absent.e.s, pour rattraper le travail effectué, qu'aux personnes présentes pour se rappeler ce qui a été dit. La difficulté de rédaction tient à ce qu'aucune réunion ou audition n'est enregistrée. Il convient donc de noter lors des échanges les informations importantes pour pouvoir les résumer correctement.

Par souci de structuration, je peux distinguer trois types de réunions qui mènent à trois façons de concevoir les comptes rendus. Bien entendu, cette classification est personnelle et officieuse, s'inspirant des tendances que j'ai pu remarquer et que je tenais à souligner.

Le premier correspond aux réunions « ouvertes » ou « débats » que les membres ont autour de l'avis : de quoi parler ou ne pas parler, quels sont les enjeux principaux, les points d'attention ou de vigilance, etc. Ces réunions sont passionnantes, en termes de contenu, mais sont difficiles à retranscrire lors de la rédaction du compte rendu car il convient de trier les idées par catégories et de structurer le débat alors même que la discussion était généralement très morcelée.

Ensuite, il y a les réunions « pratiques » où les membres se concertent sur qui auditionner, quelles questions poser, relisent les documents produits par le groupe, discutent sur la structure du questionnaire... Il y a souvent moins de débats, quoique certaines discussions sur la sémantique des termes lors de la relecture des textes peuvent s'y apparenter. Les comptes rendus sont plus courts et résument les changements principaux opérés, les échéanciers fixés ou l'avancement des auditions par exemple. Bien sûr, les réunions peuvent mixer des aspects « débats » et des aspects « pratiques », mais il me semble avoir repérer une tendance à les séparer.

Enfin, les auditions sont des réunions très particulières puisque le GT demande à une personne extérieure d'intervenir pendant environ une heure, pour éclairer le groupe dans son travail de réflexion. La personne auditionnée se présente, présente son parcours et les projets de reconnaissance automatique auxquels elle a pris ou prend part. Les membres peuvent alors écouter et poser des questions mais le temps imparti ne

laisse généralement pas la place pour le débat. Les comptes rendus sont alors souvent plus longs et déroulent l'audition presque chronologiquement.

II.1.2.2 Préparer les auditions

Outre ces comptes rendus, il m'a été demandé de rédiger une fiche résumé des parcours des membres du GT à envoyer aux personnes auditionnées en amont de la rencontre pour qu'elles sachent à qui elles s'adressaient.

Inversement, à chaque audition, j'ai eu à rédiger des fiches explicatives sur le parcours de la personne auditionnée ainsi qu'un résumé des projets de reconnaissance faciale et comportementale auxquels elle a pris part. Cette fiche était ensuite envoyée aux membres du GT en amont de l'audition.

Enfin, il a fallu dresser une liste des personnes à auditionner selon la volonté des membres du GT. J'ai tenu cette liste à jour au fur et à mesure que les auditions avançaient et que d'autres s'organisaient. De même, j'ai aidé à la rédaction de la liste des questions à poser en audition, revue par le GT durant sa deuxième réunion.

II.1.3 Questionnaire de consultation et d'appel à contributions

II.1.3.1 L'exemple du GT chatbots

Au fur et à mesure que les réunions et auditions avançaient, les discussions convergeaient vers l'idée de l'élaboration d'un questionnaire de consultation.

Cette idée de questionnaire provient d'un cas précédent : celui du GT « agents conversationnels ». Ce GT, dont l'avis devrait être disponible à la parution de ce rapport [18], avait décidé en avril 2020 de lancer un appel à contributions sur les enjeux d'éthiques liés aux *chatbots*, ces « programmes informatiques interagi[ssant] avec l'utilisateur dans la langue naturelle de celui-ci » [15]. Cette consultation s'inscrivait dans la volonté du CNPEN « d'engager une discussion collective pour développer une approche partagée des innovations présentes et futures ». En outre, elle a servi à éclairer l'avis de ce GT ainsi qu'à amorcer le débat citoyen et éduquer sur ces enjeux d'éthiques. Parmi ses nombreux avantages, nous pouvons également mentionner que le questionnaire de consultation a permis au GT de construire et structurer sa réflexion, gagnant du temps lors de la rédaction de l'avis. Cette consultation publique a aussi permis de préparer l'arrivée de l'avis auprès du grand public et des médias, en mettant en lumière le travail effectué. Néanmoins, la rédaction du questionnaire, sa diffusion ainsi que la collecte des résultats ont demandé beaucoup de temps, retardant la rédaction et la publication de l'avis final.

Pour autant, de nombreux membres du CNPEN considèrent ce questionnaire comme un exemple à suivre. Il fut donc décidé que le GT reconnaissance en proposerait également un. À l'heure où j'écris ces lignes, il n'est pas encore décidé si ce questionnaire sera diffusé ou non. Sa rédaction permettra au GT de structurer sa réflexion mais le publier est un pas supplémentaire qui n'a pas encore été sauté.

II.1.3.2 Rédaction du questionnaire

Il fut décidé de confier la tâche de rédaction d'un premier jet aux deux stagiaires chargées de suivre le GT de reconnaissance faciale, posturale et comportementale, à savoir Alexia Pronesti, élève à l'École des Hautes Études en Sciences Sociales (EHESS), et moi-même.

Les difficultés principales que nous avons rencontrées furent d'une part que le sujet de la reconnaissance faciale possède un aspect politique que le sujet *chatbots* n'avait pas. La formulation est donc primordiale : il ne faut pas lui donner un ton alarmiste, ni banaliser l'usage de ces technologies. Notamment lorsque nous en venons à parler de systèmes de reconnaissance utilisés par la police pour scanner des foules lors de la recherche d'une personne en fuite, lors d'enquêtes pour détecter les mensonges, ou encore pour surveiller les manifestations, il convient d'exposer les faits tout en ne laissant transparaître aucune opinion personnelle.

D'autre part, il faut rester le plus neutre possible. Les termes utilisés, les cas d'usage mentionnés ou encore l'ordre des questions peuvent biaiser les réponses des personnes sondées. Par exemple, il a fallu éviter d'utiliser le terme « discrimination » en parlant des biais algorithmiques, mentionnant simplement que les bases de données ne sont pas toujours équilibrées et laissant les répondants arriver à leurs propres conclusions. Ici, toute la difficulté est de ne pas orienter trop la question mais de ne pas non plus la laissez trop ouverte pour que le lecteur ne passe pas à côté des enjeux fondamentaux soulevés par les systèmes de reconnaissance

automatique. Ainsi, nous posons régulièrement la question des « bénéfices ou risques potentiels », ce qui est une façon d'ouvrir le débat de façon neutre.

Les cas d'usage doivent également être bien choisis. Pour rester sur la question des biais, mentionner directement que la reconnaissance peut moins bien marcher pour les femmes noires que pour les hommes blancs amènerait quasi exclusivement à un sentiment d'injustice. Nous avons donc choisi de parler du cas du système qui classifiait les images comme loup lorsque l'animal était sur de la neige et comme chien lorsqu'il était sur de l'herbe.

Nous espérons que ce questionnaire [20] permettra de structurer l'avis du groupe de travail et aux répondants de réfléchir autour des enjeux d'éthique soulevés.

II.1.4 Aide complémentaire

Mon aide au groupe de travail ne s'est pas limitée à ce qui m'était strictement et clairement demandé puisque j'ai également pris, encouragée par les rapporteur.e.s ainsi que par mes encadrant.e.s (Mme Tessier et M. Kirchner), un certain nombre d'initiatives.

Régulièrement, les rapporteur.e.s ou les membres du GT m'envoyaient par mail des documents à ajouter à leur bibliographie mais il fallait trouver un moyen de structurer cette bibliographie. L'espace de travail Mybox n'était pas adapté car, comme tout espace *drive*, il est impossible d'y classifier les documents autrement que dans des dossiers, ce qui rend la navigation difficile. J'ai donc créé un groupe sur le site Zotero, permettant à chacun d'avoir accès à tous les documents à tout moment, de les rechercher par catégories (ou *tags*), ainsi que d'en rajouter à leur guise.

J'ai également recensé les usages possibles de la reconnaissance faciale, posturale et comportementale. Il me semblait intéressant d'avoir une vision globale sur les domaines où elle était utilisée ou en cours de développement. La liste que j'en ai tirée n'est bien sûr pas complète, elle sera à compléter au fur et à mesures de mes recherches (ou des évolutions de la société et de ces technologies). J'ai choisi pour faire cette classification de reprendre les trois types de reconnaissance, à savoir authentification, identification et catégorisation, puis de les décliner par catégories d'usages (accéder à un lieu, utiliser un service, tracer une personne...) et enfin par secteurs d'activités (sécurité, santé, divertissement...). Un dernier indicateur donne l'état de développement de la technologie dans ce secteur ainsi que des liens vers des entreprises travaillant sur ces systèmes.

Finalement, j'ai également dressé une liste des comités d'éthique des autres pays ainsi que la liste de toutes les institutions qui ont déjà publié des rapports sur la reconnaissance faciale, triés par pays. Un bref résumé venait expliquer le contenu des documents que j'avais pu lire.

J'ai également été amenée à aider brièvement un deuxième groupe de travail, sur les agents conversationnels. Les rapporteur.e.s m'ont demandé de contribuer à l'élaboration d'« encadrés » pour ce texte, visant à éclairer le lecteur sur certains aspects techniques. J'ai rédigé des paragraphes qui ont ensuite été relus et améliorés. Ces encadrés portent sur « les tests d'intelligence pour les machines », « l'apprentissage profond pour la reconnaissance de la parole », « l'apprentissage par renforcement dans les chatbots », « un chatbot adaptatif TAY » et « Lil Miquela, influence virtuelle ».

II.1.5 Travail de recherche

J'ai mené, en parallèle de mon aide au GT, mon travail de recherche. Lors des premières semaines, j'ai rassemblé de la bibliographie en vue de la rédaction d'un état de l'art, me renseignant sur les technologies de reconnaissance automatiques existantes, la façon dont ces systèmes sont conçus et les enjeux éthiques qu'ils soulèvent. J'ai également dû me renseigner sur les définitions données de l'IA et sur les textes internationaux concernant « l'éthique de l'IA ». J'ai ensuite conçu un code de reconnaissance faciale, en analysant les choix que j'étais amenée à faire et la manière dont ils influencent les valeurs que je souhaitais transmettre à travers eux. Cette recherche a été au cœur de mon travail durant ce stage : sa description complète est donnée dans la partie IV.

II.2 Présentations et suivis du travail

II.2.1 Suivis des encadrant.e.s

Durant le stage, j'ai pu travailler à la fois en distanciel et en présentiel, occupant un bureau dans le bâtiment du CCNE. Les premiers mois, j'ai pu venir un jour par semaine en présentiel, puis environ trois en fin de stage. Toutefois, les membres étant à distance, les réunions ont été organisées en ligne ainsi que les points de suivis avec les tuteur.trice.s. Nous organisions ainsi environ un point par semaine avec Mme Tessier et M. Kirchner où je leur présentais mon travail de la semaine ainsi que les perspectives pour la semaine suivante. De même, nous faisions un point technique avec M. Grisel environ toutes les deux semaines, pour qu'il puisse regarder mes avancées sur le code et me débloquer si besoin.

II.2.2 Autres retours

J'ai également eu d'autres retours sur mon travail. J'ai ainsi pu le présenter devant Mme Camille Blanchard, doctorante à l'ONERA et supervisée par Mme Tessier, ainsi que devant son comité de suivi de thèse. Mme Blanchard travaille sur la modélisation des scénarios de prospective et a besoin pour son système de définir des acteurs avec un certain nombre de valeurs et qui interagissent entre eux. Nous avons donc échangé sur cette notion de valeurs et ce qu'elle et moi considérons comme tel pour nos travaux respectifs.

J'ai également pu présenter ma démarche de recherche devant les membres du CNPEN durant le séminaire organisé à Royaumont (voir II.3.3) et échanger avec eux sur mon travail autour du cas école (développé en IV).

II.3 Activités annexes

II.3.1 Séminaires

Durant mon stage, j'ai été amenée à réaliser de nombreuses activités, en marge de mes travaux ordinaires.

J'ai par exemple pu assister à différents séminaires en ligne, encouragée par mes encadrant.e.s. Ces séminaires portaient sur une grande variété de thématiques en lien avec « l'éthique de l'IA ».

J'ai ainsi pu assister, le 8 avril, à l'édition 2021 du séminiaire Perspectives et Défis de l'IA (PDIA'21) de l'Association française pour l'Intelligence Artificielle (AfIA) qui portait sur « l'Explicabilité », ou encore à certains séminaires du GDR IA (CNRS).

Enfin, du 28 juin au 2 juillet 2021, se tenait la Plate-Forme Intelligence Artificielle (PFIA), une manifestation annuelle et, cette année, virtuelle, organisée par l'AfIA et hébergeant de nombreuses conférences sur des thèmes variés. Les conférences se tenant en parallèle, il était impossible d'assister à tout, mais j'ai pu, entre autres, assister à certaines présentations de la Conférence Nationale sur l'Intelligence Artificielle (CNIA) ou de la journée thématique « Défense et IA ».

II.3.2 Colloque

Le 11 juin 2021, le CNPEN a organisé en collaboration avec l'institut Covid-AdMemoriam un colloque de neuf tables rondes autour du thème « Numérique et pandémie », auquel j'ai pu participer. En amont du colloque, il m'a été demandé de rédiger et de mettre en forme les biographies des participants. Pendant l'événement, j'ai aidé à accueillir les invités, à modérer les questions et à m'assurer que tout fonctionnait comme prévu (notamment l'enregistrement des tables rondes, la diffusion en direct, etc.).

II.3.3 Royaumont

Du 21 au 23 juin 2021, les membres du CNPEN se sont réunis en présentiel à l'abbaye de Royaumont, comme ils et elles le font environ deux fois par an. J'ai eu la chance de pouvoir assister à leurs échanges qui ont portés principalement sur une réflexion autour des notions de « Valeurs et Principes » ainsi que la façon dont ces notions se transcrivent dans leurs travaux présents et passés. Dans ce cadre-ci, j'ai pu présenter les avancées de mon travail aux membres du groupe et échanger avec elles et eux sur les améliorations possibles. Rejoins par les membres CCNE (Comité Consultatif National d'éthique pour les sciences de la vie et de la santé) lors de la dernière journée, les membres des deux comités ont discuté de l'évolution des deux structures et de la possible pérennisation du CNPEN sous la forme d'un CCNE « numérique ».

Troisième partie
Un travail sur les valeurs morales

III.1 Cas choisi : authentification par reconnaissance faciale

Le sujet de l'autosaisine était trop large pour être couvert en entier. Il fallait donc se restreindre à un système particulier. Il a donc été décidé de traiter le cas de la reconnaissance faciale par authentification.

En effet, ce type d'application ne requiert pas d'enregistrements vidéo et peut être traité facilement sur un ordinateur peu puissant (ou un système virtuel tel que Google Colab ou Kaggle Kernel). Pour les données d'entraînement, il existe des bases de données accessibles librement sur internet (même si cela pose des questions, soulevées en IV.3.1.1). Le code une fois entraîné devra prendre deux images de deux visages en entrée et dire en sortie s'ils appartiennent (1) ou non (0) à la même personne. Une application simple de ce système pourrait-être le déverrouillage d'un téléphone ou le passage aux frontières.

Enfin, il a été proposé assez rapidement de développer un code fonctionnant par *triplet loss* car cette technique est à la fois récente et assez répandue. Encore une fois, ce choix est critiqué dans la partie IV.3.1.4. Le papier initial introduisant la *triplet loss* a été publié par des ingénieurs de chez Google en 2015 [36]. Nous nous baserons dessus pour construire notre modèle.

III.2 Critique de l'état de l'art en matière d'« éthique de l'IA »

III.2.1 Un vocabulaire à éclairer

Alors que des dizaines de textes d'experts internationaux appellent à une « éthique de l'IA », il convient de se demander de quoi il est question.

D'abord, lorsque ces textes parlent d'« IA », il est souvent sous-entendu qu'il est question d'apprentissage machine. Ce n'est en effet qu'un des pans de l'IA (voir la définition donnée en I.1.2), mais c'est peut-être celui qui pose le plus de soucis de régulation car les résultats de ces algorithmes sont difficilement interprétables. Mais avant tout, c'est un problème technique, et finalement d'utilité - à quoi sert un objet dont on ne sait pas ce qu'il fait ?

L'apprentissage est souvent réalisé à partir de données massives, sans contrôle humain, et il n'est pas rare de lire que ces systèmes ne se comportent pas comme souhaité par les développeur.se.s [1].

Pour le côté « éthique », il n'est pas clair non plus de quoi il est question. Ainsi le HLEG [2] parlera de « composantes », « principes » ou « exigences » éthiques ; l'OCDE [3] de « principes », « recommandations » ou « normes » ; le GEAH [4] de « valeurs » et de « principes » ; sans que ces termes soient définis correctement. Tous ces termes sont proches mais ont pourtant des sens quelque peu différents.

Ainsi, la seule définition donnée dans ces textes peut se trouver à l'article 10 du texte préliminaire rédigé par les experts de l'UNESCO [22] :

« Les valeurs ont un rôle important en tant qu'idéaux motivant l'orientation des mesures politiques et des normes juridiques. Alors que l'ensemble des valeurs énoncées ci-dessous inspirent ainsi des comportements souhaitables et constituent les fondements des principes, les principes quant à eux explicitent les valeurs de manière plus concrète, de façon à faciliter l'application de ces dernières dans les déclarations et actions politiques. » [5]

Nous voyons ici que le terme de valeur est plus abstrait que celui de principe. Nous pouvons ainsi penser le principe comme une propriété à respecter pour que la valeur le soit également.

Nous choisirons ici le terme de « recommandations » pour décrire les conseils des experts internationaux en matière d'IA. Mais quelles sont donc ces recommandations ? Elles sont nombreuses, diffèrent d'un

1. Citons ici par exemple le cas du *chatbot* Tay de Microsoft qui, après avoir été nourri de tweets d'internautes, avait tenu des propos racistes et sexistes [38] ; du logiciel de recrutement d'Amazon qui, après avoir été entraîné sur les CVs du personnel ne sélectionnait plus que des profils masculins [35] ; enfin du système COMPAS utilisé aux États-Unis pour prédire la récidive de criminels qui est fortement biasé envers les personnes noires [31].
2. Hight Level Expert Group de la Commission européenne en matière d'IA
3. Organisation de Coopération et de Développement Économiques
4. Groupe d'Expert Ad Hoc de l'UNESCO
5. Ce document est un avant-projet, les négociations en cours pourrait changer cette définition

texte à l'autre et il serait inutile de tenter ici d'en dresser une liste exhaustive. Néanmoins, parmi les plus courantes, nous retrouvons la « transparence » ou « explicabilité » des systèmes d'IA ; « l'équité », la « non-discrimination » ou la « diversité » ; la « sécurité », « sûreté », « robustesse » ; la « prévention de toute atteinte », « bienveillance », « non-malveillance » ; le « respect de l'autonomie humaine », le « bien-être », la « protection des droits de l'homme », la « protection de la vie privée », le « développement durable » ou encore le concept de « responsabilité ».

Cette liste partielle est ici mentionnée pour montrer la variété des recommandations. Certaines relèvent plus de ce que nous pourrions appeler « valeurs », d'autres plus des « principes ». Il n'est même pas clair pour certains textes dans quelle catégorie elles tombent. Ainsi le terme de « transparence » est tantôt utilisé comme une valeur, tantôt comme un principe ou une exigence.

III.2.2 Des recommandations abstraites

Ces recommandations énoncées par les textes internationaux sont souvent très vagues et loin d'un caractère opérationnel.

Prenons le critère d'équité par exemple. Qu'entendons-nous lorsque nous disons qu'un algorithme doit être « équitable » ? Qu'il doit traiter tous les individus de la même façon ? Cette définition, trop simpliste, ne reflète pas la complexité philosophique, juridique et même, mathématique du problème. Pour un état de l'art sur les questions d'équité dans l'apprentissage machine, voir [33].

Il y a un gouffre aujourd'hui entre les recommandations des experts internationaux et la réalité technique qui doit être considérée pour réguler ces systèmes. Il y a un besoin urgent de combler ce gouffre et de définir des critères techniques d'évaluation des algorithmes en accord avec les valeurs prônées par les textes internationaux. Mener des audits sur les algorithmes doit permettre de se reposer sur des critères précis, des exigences concrètes qui devront figurer dans les cahiers des charges.

Or pour définir ces critères, nous devons regarder l'intérieur de la machine. Nous nous pencherons donc sur le fonctionnement des algorithmes d'apprentissage machine pour mieux identifier comment ces critères peuvent se traduire.

Dans le cadre de l'autosaisine du CNPEN, nous travaillons sur des systèmes de reconnaissance. Un cas école sera donc développé, avec l'objectif d'étudier les étapes de construction du code et d'identifier les choix de conception (voir partie IV.3).

III.3 Démarche globale et objectif du travail de recherche

III.3.1 Formuler des exigences opérationnelles

Nous allons ici garder les termes de « valeurs » et « principes », tels que décrits en III.2.1. Nous pouvons même aller plus loin dans le caractère opérationnel et parler d'« exigences de conception », propriétés plus précises, évaluables, qui pourraient figurer dans un cahier des charges. Nous obtenons ainsi un arbre, représenté en figure 2, qui n'est pas une hiérarchie mais un graphe de dépendances, organisé en trois niveaux d'abstraction, du plus vague (la valeur), au plus concret (l'exigence).

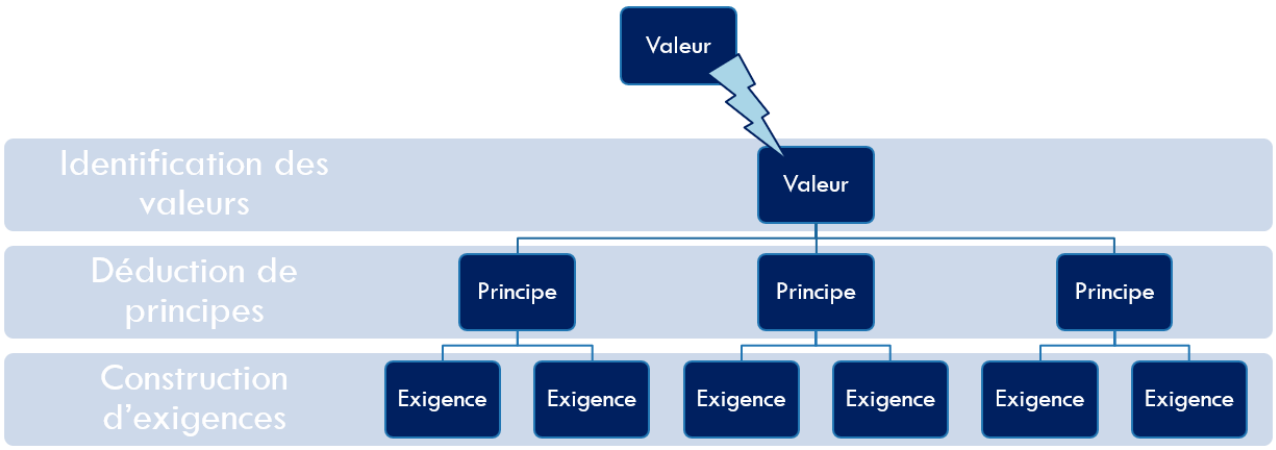


FIGURE 2 – Déclinaison d'une valeur en principes puis en exigences

Pour respecter un principe, il faudrait alors respecter les exigences associées. Et pour considérer que nous respectons une valeur, il faudrait respecter les principes qui lui sont associés. Tout notre travail consiste à définir ces exigences. C'est un travail long car de nombreux critères existent et il faudrait pouvoir tous les regarder. Ici nous en choisirons certains que nous étudierons plus en détail IV.2.1.

Pour exemple, nous donnons ci-dessous la figure 3 représentant l'exigence d'explicabilité déclinée en plusieurs principes puis exigences. Ces exigences ne se veulent pas exhaustives mais servent à illustrer concrètement notre démarche.

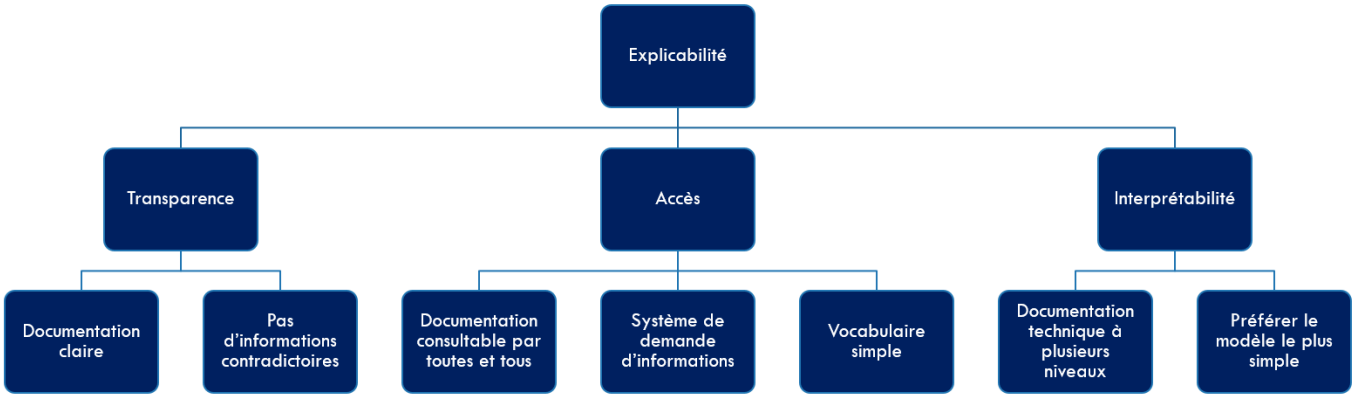


FIGURE 3 – Exemple de déclinaison de la valeur d'explicabilité

Il est à noter également que les valeurs, principes ou exigences peuvent être en tension les uns avec les autres. Nous y reviendrons en III.3.4.

III.3.2 Identifier les parties prenantes

Lors du développement d'un système, qu'il soit ou non informatique, il est nécessaire de se demander qui va l'utiliser et interagir avec lui, directement ou indirectement (voir la carte des acteurs figure 4). Identifier ces acteurs permet de dresser une liste de leur valeurs et ainsi analyser la façon dont ces dernières affectent le développement et le déploiement du système. Il faut par exemple se demander qui sont les concepteur.trice.s, opérateur.trice.s et utilisateur.trice.s du système, mais aussi dans quelle société (avec quel gouvernement, quelles agences de régulation, etc.) ils interagissent.

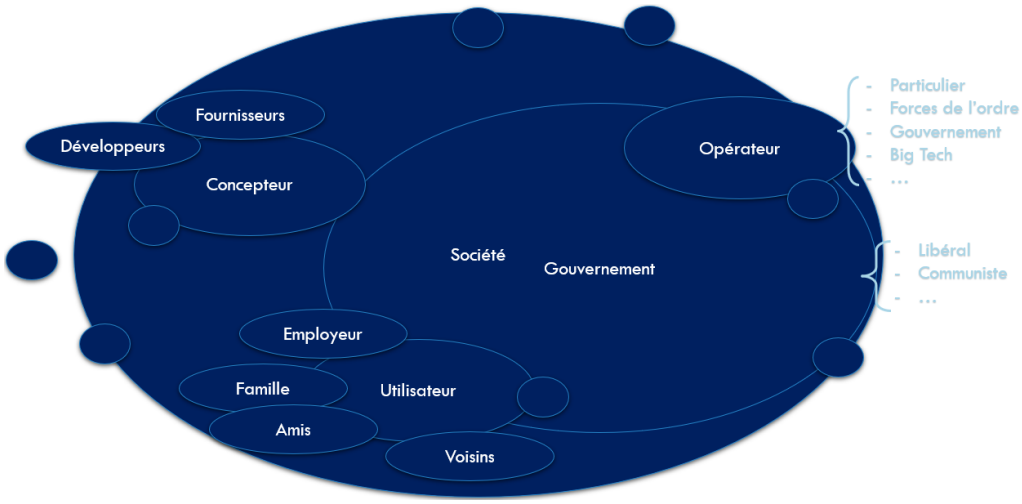


FIGURE 4 – Carte (non exhaustive) des parties prenantes

Par exemple dans le cas de la reconnaissance faciale, si l'opérateur est un particulier voulant poser une caméra dans son jardin ou dans son magasin, nous n'aurons pas les mêmes enjeux que si ce sont les forces de l'ordre qui déploient ces systèmes dans la rue, le gouvernement, ou encore une entreprise américaine comme les GAFAMI [6]. De même, selon le gouvernement (libéral, communiste...), les valeurs mises en avant ne seront pas les mêmes.

6. Google Amazon Facebook Apple Microsoft IBM, les grandes entreprises du numérique

Dans notre cas école précis, il est difficile de lister ces acteurs puisque le logiciel n'a pas vocation d'être déployé. Toutefois, si nous prenons le cas d'un déverrouillage de téléphone par reconnaissance faciale, nous pourrions alors imaginer que l'entreprise à l'origine du téléphone est le **concepteur**. Le ou la propriétaire du téléphone est à la fois **opérateur.trice** et **utilisateur.trice** du système puisqu'il ou elle le déploie et l'utilise pour son usage personnel.

Dans le cas d'un passage aux frontières, le concepteur est l'entreprise qui développe le logiciel et le vend aux aéroports et gares qui en font usage, l'opérateur est le service douanier de l'aéroport ou gare en question, et l'utilisateur.trice est le ou la voyageur.se souhaitant passer la frontière.

III.3.3 Recenser les valeurs

Une fois que ces acteurs sont identifiés, il convient de recenser leurs valeurs. Bien sûr, les valeurs promues par un acteur ne seront pas les mêmes que celles qu'il respectera en réalité et il faut être prudent et garder un œil critique sur ce que nous pouvons trouver. Rappelons à titre d'exemple que la démocratie et la liberté sont deux valeurs du parti communiste chinois. Toutefois, il est difficile d'imaginer quelles pourraient être les valeurs réelles d'un acteur, sans compter que notre réponse serait biaisée par notre propre jugement. Nous n'avons, par exemple, peut-être pas tous la même définition de la valeur d'intégrité.

Ce travail de recensement est, pour les mêmes raisons que précédemment, difficile à réaliser pour notre cas école car le logiciel n'a pas vocation à être déployé. Néanmoins, nous ferons comme si notre système devait être commercialisé et regarder comment certaines valeurs peuvent être traduites (ou non) de manière opérationnelle.

Nous pouvons aller regarder les valeurs de certains acteurs pour nous donner une idée de la forme qu'aurait un tel recensement.

Nous pouvons commencer à séparer ces valeurs en trois catégories : les **valeurs promues** en général par les acteurs, quel que soit le domaine ou le secteur dont ils et elles font partie ; les **valeurs à respecter** lors de la conception, qui sont des valeurs orientées vers le cas d'usage général qu'est l'IA, comme par exemple un certain nombre des recommandations des textes internationaux ; puis enfin les **valeurs contextuelles**, spécifiques au cas de la reconnaissance faciale. Cette classification est inspirée de [39]. Un exemple de quelques valeurs que nous pouvons ainsi classifier est donné en figure 5 [7].

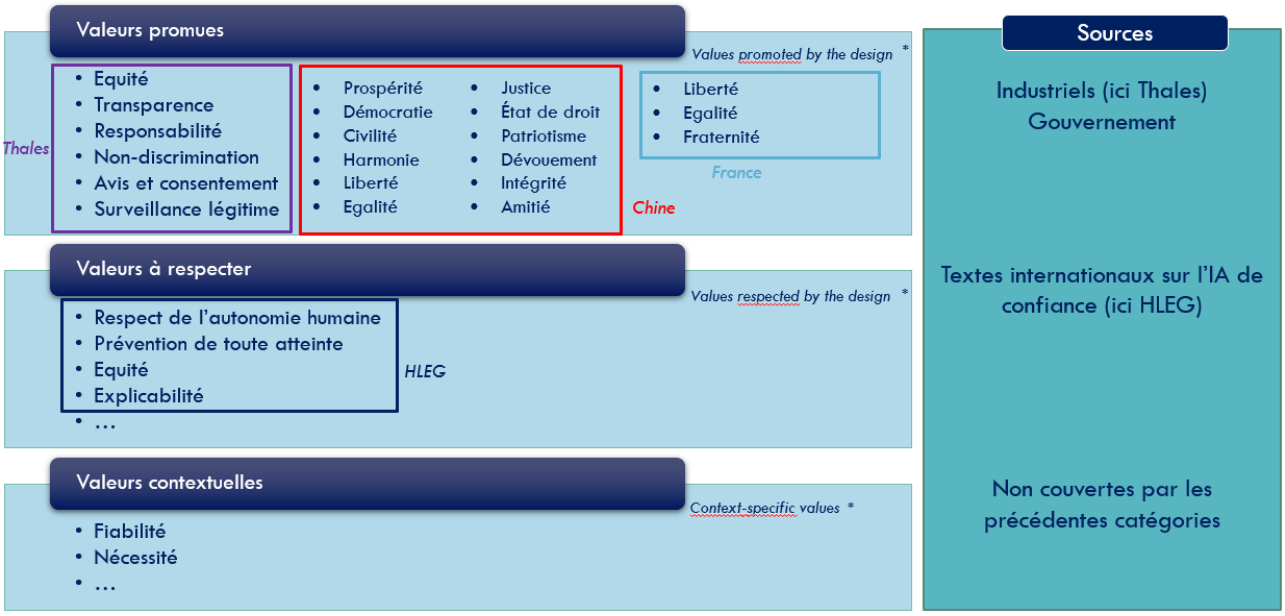


FIGURE 5 – Exemple de tableau de recensement des valeurs

III.3.4 Relation entre les valeurs

Comme indiqué précédemment (cf III.3.1), les valeurs ne sont pas indépendantes les unes des autres. Ainsi, il existe entre elles des relations, comme des frictions, pouvant dans certains cas opposer ces valeurs, ou les faire converger, comme représenté figure 6.

7. Le vocabulaire originel du texte est marqué par des * dans la figure 5

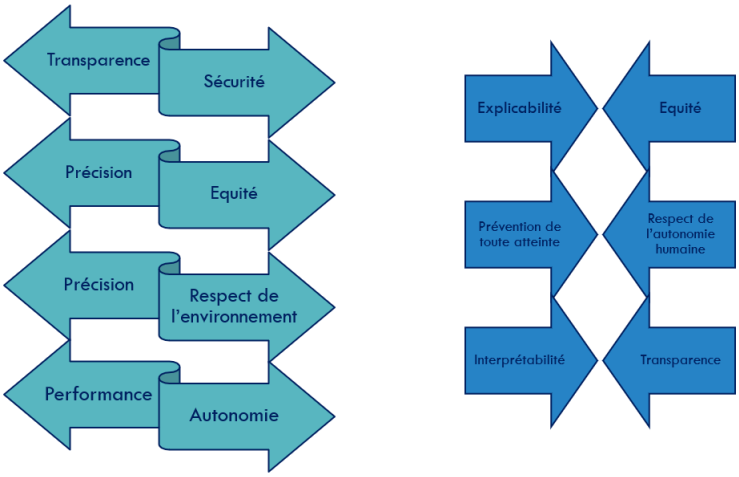


FIGURE 6 – Exemple de relations entre des valeurs - adapté de [37]

Bien sûr, les relations qu'elles entretiennent sont plus complexes que le laisse suggérer ce schéma. Par exemple, d'un point de vue juridique, la transparence et la sécurité fonctionnent plutôt conjointement : plus nous avons d'informations sur un cas et plus il est facile d'en assurer la sécurité. Au contraire, si nous allons trop loin, la connaissance approfondie d'un système permet à des entités mal intentionnées de l'attaquer plus facilement.

A trop vouloir favoriser la performance et la précision du système, il arrive que les concepteur.trice.s délaissent le critère d'équité (voir III.3.6). Mais parfois, ces mêmes valeurs peuvent évoluer dans le même sens et en améliorant l'équité nous pouvons également améliorer la précision globale, comme par exemple si nous améliorons la précision d'un système pour un sous-groupe sans la détériorer pour d'autres sous-groupes.

De même, à chercher à être toujours plus précis, nous engendrons des calculs de plus en plus sophistiqués qui demandent plus de puissance et nuise à l'environnement.

Enfin, garantir l'autonomie humaine est de plus en plus difficile au fur et à mesure que les systèmes deviennent de plus en plus performants et « autonomes ». Ces performances, toujours plus poussées, peuvent également dégrader l'expertise humaine [37].

Nous voyons ainsi que ces relations entre valeurs sont bien plus complexes que le laisse suggérer notre schéma [8]. Elles forment en réalité plutôt un réseau d'influence, corrélées positivement ou négativement selon le cas d'usage et la perspective de chacun. Pour exemple, la relation transparence-sécurité ne sera perçue de la même façon par un juriste que par un ingénieur en cybersécurité [9].

Ces relations affectent les exigences que nous souhaitons construire : il est probable que certaines exigences ne soient pas compatibles, c'est-à-dire que nous ne puissions pas les respecter en même temps. Il faudra alors choisir quel critère et donc quelle valeur respecter en priorité.

III.3.5 Identifier les choix de conception

Lors de la conception d'un algorithme, un.e développeur.se est amené.e à faire des choix de conception qui auront un impact sur le résultat et sur les valeurs qui lui sont associées. Tous les critères ne peuvent pas être respectés en même temps. Ce phénomène est nommé le « *no free lunch theorem* » [43], autrement dit « rien n'est gratuit ». Si un critère est amélioré, un autre sera dégradé. Le travail du ou de la concepteur.trice est donc de trouver un compromis entre tous ces critères.

Puisque aucun choix n'est neutre, faire ces choix amènera à préférer un critère ou une valeur à une autre. Chaque choix est ainsi un compromis entre chacune des valeurs (voir figure 7).

Ces choix, nous cherchons dans le travail qui suit à les identifier afin d'éclairer au maximum à quel moment du développement un.e concepteur.trice choisit en réalité une valeur plutôt qu'une autre. Nous pourrions ainsi définir des exigences de conception permettant de respecter une certaine hiérarchie de valeurs (voir le schéma récapitulatif de la démarche en figure 8).

8. Ce schéma a été très critiqué lors de sa présentation aux membres du CNPEN. J'ai néanmoins choisi de le conserver car, bien que trop simpliste, il me semble qu'il permet bien de représenter la dépendances des différentes valeurs les unes avec les autres.

9. Pour cet exemple, je me suis inspirée d'une discussion que j'ai eu avec des membres de professions différentes, qui ne percevaient effectivement pas ces valeurs de la même façon.

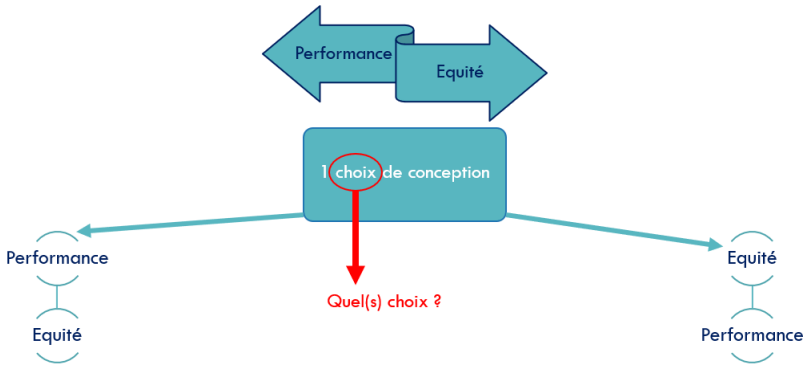


FIGURE 7 – Un choix de conception amène à favoriser une ou plusieurs valeurs par rapport à d'autres

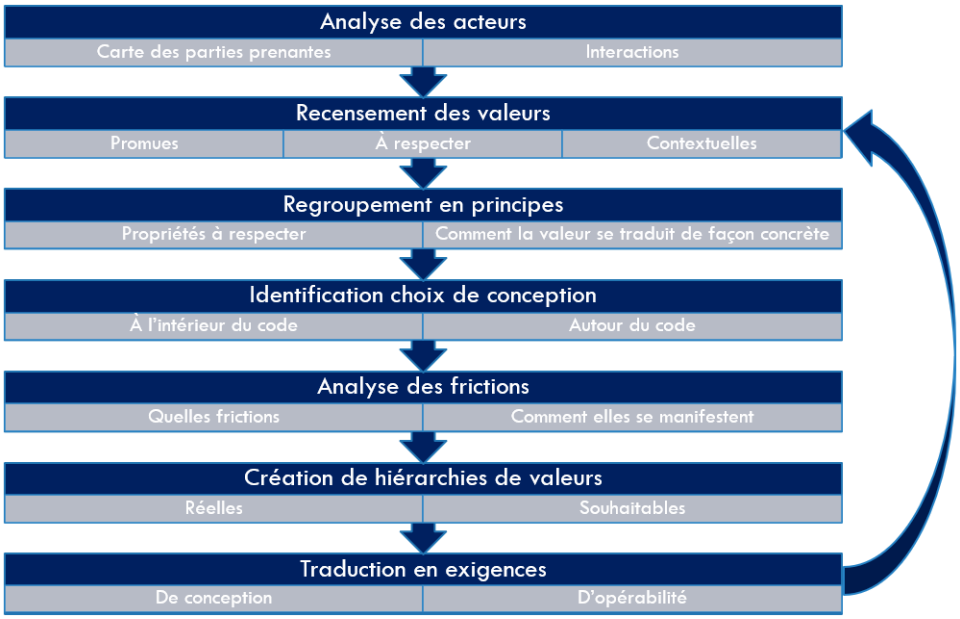


FIGURE 8 – Résumé de la démarche de travail

III.3.6 Exemple fictif

En reconnaissance faciale par authentification, un choix crucial et celui de savoir quel seuil donner à l'authentification.

Nous envoyons dans le réseau deux images dont nous souhaitons déterminer si elles sont de la même personne. Nous souhaitons alors un résultat binaire (1 : les images sont de la même personne, 0 : les images sont de personnes différentes). Pour cela, nous calculons la distance entre deux images et nous décidons d'un seuil en-dessous duquel deux images seront considérées semblables (1) et au-dessus différentes (0). Pour cela nous testons le système sur des images de différentes personnes.

Nous nous plaçons dans le cadre d'un système de déverrouillage de téléphone par reconnaissance faciale. Ici, nous cherchons à séparer les individus marqués en bleu, qui sont effectivement propriétaires du téléphone, des personnes en orange qui ne le sont pas (cf figure 9). Nous déterminons un seuil entre les deux histogrammes, permettant de les séparer au mieux. Mais cela ne fonctionne pas aussi bien dans tous les cas.

Lorsque les deux histogrammes se superposent (figure 34), il y a un choix à faire. Dans l'histogramme de la figure 34 à gauche, nous choisissons de réduire au maximum le taux d'erreur, c'est-à-dire de faux positifs et de faux négatifs. Or nous savons que ces systèmes de reconnaissance fonctionnent moins bien pour des femmes noires que pour des hommes blancs par exemple [3]. Il y a donc plus de chance de trouver une répartition où les femmes noires se trouveraient davantage dans ces zones d'erreurs que les hommes blancs. Si une femme noire souhaitait déverrouiller son téléphone, elle aurait plus de chance d'échouer que l'homme blanc. Sachant que nous avons cette répartition inégale, en plaçant le seuil ainsi, nous avons favorisé une valeur de performance par rapport à une valeur d'équité. Si nous souhaitions traiter tous les utilisateur.trice.s de la même façon, il faudrait placer le seuil dans le cas de l'histogramme de la figure 34 à droite, mais nous aurions alors des problèmes de sécurité évidents car bien que toutes les personnes propriétaires peuvent maintenant

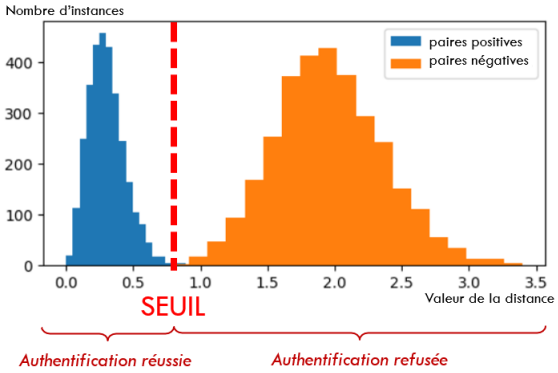


FIGURE 9 – Histogramme des distances pour des paires positives et négatives : groupes séparés

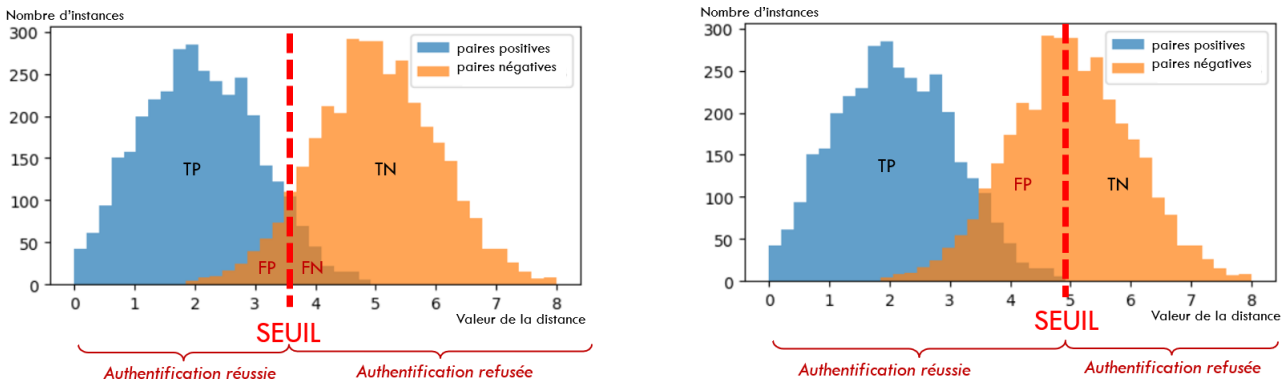


FIGURE 10 – Histogrammes des distances pour des paires positives et négatives : groupes superposés et seuil selon un critère de performance (à gauche) et d'équité (à droite)

déverrouiller leur téléphone, c'est également le cas de près de la moitié des personnes non-propriétaires - les utilisateur.trice.s en orange à gauche du seuil. La valeur d'équité aura été préférée à celle de performance et à celle de sécurité.

Cet exemple naïf est en réalité très représentatif des problèmes rencontrés. Nous ne pouvons pas considérer le critère du seuil indépendamment du critère d'équilibrage des données. Et lorsque nous le faisons varier pour une distribution fixée, nous favorisons une valeur ou une autre.

Le critère de seuil est un des critères que nous allons considérer par la suite (cf IV.3.4.2).

Quatrième partie
Développement du cas école

Nous avons souhaité que ce rapport soit compréhensible par toutes et tous. Ainsi, de nombreux termes techniques sont expliqués et le rapport s'arrête longuement sur le fonctionnement du système. Nous avisons nos lecteurs connaissant le principe de la *triplet loss* de passer directement à la partie IV.2.

IV.1 Fonctionnement du code

IV.1.1 Utilisation normale

Nous prenons deux images dont nous souhaitons savoir si elles sont de la même personne et nous les présentons à un modèle d'apprentissage machine en réseau de neurones artificiels (ou en anglais, *artificial neural networks* [10] ou ANNs) [11] préalablement entraîné.

Ce réseau va générer des gabarits pour chacune des deux images. Ces gabarits sont des vecteurs de dimension n représentant l'image. Nous calculons ensuite la distance euclidienne entre ces deux vecteurs dans $\mathbb{R}^+$ [12]. S'ils sont « éloignés », i.e. que la distance est élevée, nous renvoyons 0 : les images sont issues de personnes différentes. Si la distance est « faible », nous renvoyons 1 [13] : les images sont issues de la même personne. Pour quantifier cette notion d'éloignement, nous convenons d'un seuil au-delà duquel nous considérons que les images sont trop éloignées pour représenter la même personne. Voir le fonctionnement global en figure 11.

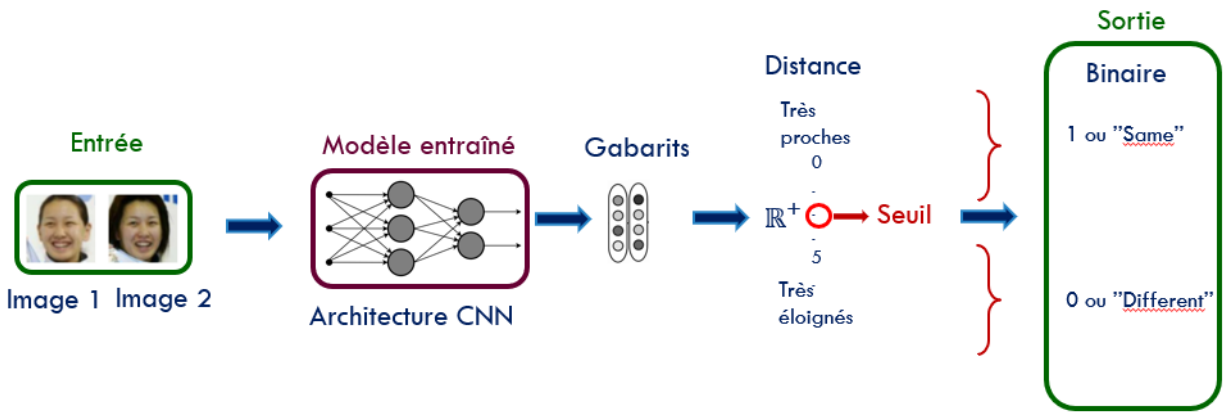


FIGURE 11 – Fonctionnement du code : utilisation normale

Dans le cas d'un déverrouillage de téléphone, le système compare le visage de la personne qui se présente devant l'écran avec les photos du ou de la propriétaire du téléphone stockées sur l'appareil et prises au moment de la configuration du déverrouillage par reconnaissance faciale.

Dans ce système, le téléphone a plusieurs photos pour comparer (une dizaine la plupart du temps), et renvoie une valeur moyenne de distance avec le visage actuel. Dans notre cas, nous ne faisons la comparaison qu'une fois. Pour être plus précis, nous pourrions également faire cette moyenne, il faudrait alors étudier ce que cela changerait en termes de résultats.

Notre cas ressemble plus au fonctionnement du logiciel PARAFE pour le passage de frontière où seulement la photo du passeport du voyageur ou de la voyageuse est comparée à son visage actuel.

10. Le vocabulaire d'apprentissage machine regorge d'anglicismes, jusqu'à son nom - le domaine est régulièrement appelé « *machine learning* ». Dans ce rapport, nous nous sommes efforcée d'utiliser un maximum de mots français, mais cela n'était pas toujours possible car les termes techniques ne sont souvent pas traduits et les acteurs du domaine les utilisent couramment directement en anglais.
11. Pour plus d'explications sur qu'est-ce qu'un réseau de neurones artificiels et quels type de réseaux sont utilisés pour la reconnaissance d'images, voir l'Annexe A
12. Pour plus d'explications sur les gabarits voir IV.1.3, sur le calcul de la distance, voir IV.3.2.1
13. C'est une sortie « binaire »

IV.1.2 Données

Avant de pouvoir se servir du système, nous devons entraîner le réseau de neurones sur des données [14], c'est la base de l'apprentissage machine. La base de données utilisée est importante. Il faut se demander d'où proviennent les images et si nous avons les droits nécessaires pour les utiliser. Dans le cadre de notre système, ces entrées sont des images de visages, mais de qui ? Les personnes ont-elles donné leur consentement pour que leurs données soient traitées ? Et comment ces images sont-elles annotées ? La diversité de la base de données est également importante mais faut-il encore pouvoir la mesurer.

Bien sûr, le format de l'image importe également : une image trop « pixélisée » pénalisera l'entraînement alors qu'une image trop haute définition le ralentira. Nous pouvons appliquer des filtres et transformations aux images, cela est appelé la *data augmentation* et permet, entre autres, de mieux généraliser pendant l'entraînement et de rendre le système plus robuste. Ces transformations vont avoir un impact sur le résultat qu'il convient d'estimer. Nous reviendrons sur ces choix dans les parties IV.3.1.1 et IV.3.3.1.

IV.1.3 Gabarits d'images

Les données d'entrée du réseau de neurones artificiels sont les images (de visages dans le cas de la reconnaissance automatique) et les gabarits en sont la sortie. Ces gabarits générés par le réseau, sont des vecteurs à n dimensions. Cette dimension n est la dimension de sortie du réseau, choisi par le ou la concepteur.trice de l'algorithme.

Prenons l'exemple d'un algorithme devant différencier des images de chiffres entre 0 et 9 (cf 12). En dimension 2, nous pouvons placer ces points dans un plan (x,y). Lorsque l'algorithme n'est pas entraîné, les points des différentes images se mélangent (cf figure 13). Le but de l'entraînement est de séparer les images correspondant aux différents chiffres en plusieurs *clusters*, ou groupes (cf figure 14).

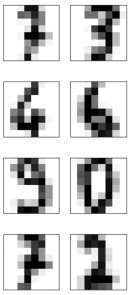


FIGURE 12 – Chiffres à classer

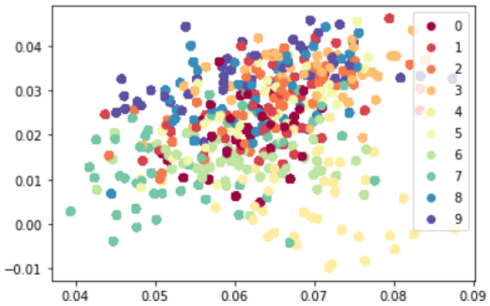


FIGURE 13 – Répartition des gabarits avant entraînement

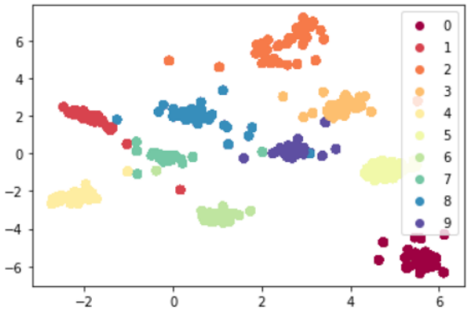


FIGURE 14 – Répartition des gabarits après entraînement

Ainsi, si nous passons une image d'un chiffre dans le réseau entraîné, la valeur de son vecteur gabarit nous permettra de le placer dans le plan et ainsi de dire à quel *cluster* il appartient.

Pour en revenir à la reconnaissance faciale, les différents *clusters* représenteront les identités des personnes. Pour les séparer, il faut augmenter le nombre de dimensions, rendant la visualisation impossible. Le principe reste toutefois le même. Choisir cette dimension n est un choix de conception. Il est intimement lié à la structure du réseau que nous étudierons en IV.3.2.4.

IV.1.4 Objectif de l'entraînement

Notre système fonctionne par *triplet loss*, une technique développée en 2015 par des ingénieurs de chez Google [36] pour entraîner efficacement un réseau de neurones pour la reconnaissance faciale. Pour réaliser cet entraînement, nous avons besoin d'une base de données contenant des photos de visages, avec plusieurs photos par personne. Le principe de l'entraînement est illustré en figure 15.

Nous organisons ces images par triplés. Chaque triplé est constitué de deux images de la même personne, ce sont les images dites « image ancre » (*anchor image*) et « image positive » (*positive image*), et une image d'une personne différente, c'est l'image dite « négative » (*negative image*).

Ces trois images sont ensuite passées dans le réseau de neurones qui aura été initialisé aléatoirement.

14. Pour plus d'explications sur comment se déroule l'entraînement d'un réseau de neurones, voir l'Annexe B

Le réseau renvoie trois vecteurs correspondant aux gabarits des images. Ces gabarits sont ensuite comparés grâce à la fonction de perte (voir IV.1.4), qui donne un résultat dans $\mathbb{R}^+$. Le réseau apprend [15] de cette fonction de perte pour optimiser les poids de ses neurones [16].

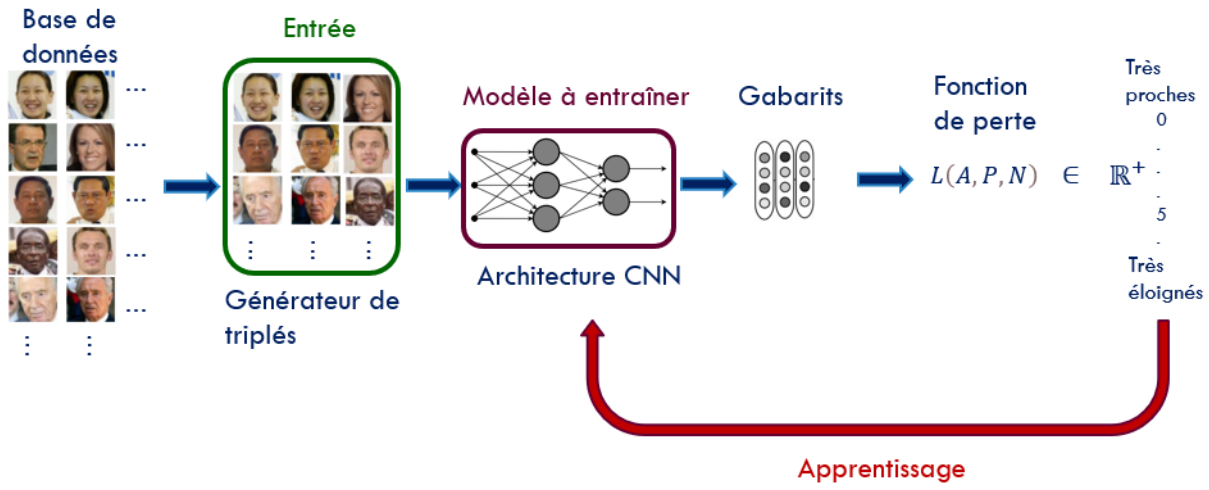


FIGURE 15 – Fonctionnement du code : entraînement

Afin de séparer les *clusters* comme expliqué dans la partie IV.1.3, nous cherchons à avoir la distance entre l'ancre et la positive plus petite que la distance entre l'ancre et la négative. C'est le but de l'entraînement (cf figure 16).

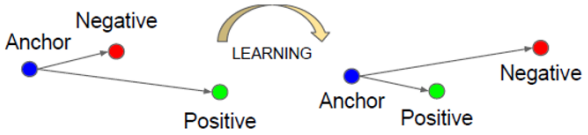


FIGURE 16 – Objectif de l'apprentissage

Nous choisissons ici la distance euclidienne. Nous avons alors :

$d(A,P) < d(A,N)$
$\alpha + \|f(A) - f(P)\|^2 < \|f(A) - f(N)\|^2$

Avec :

— α un paramètre nommé « marge »
— f la fonction de sortie du réseau
— A, P, N respectivement les images ancre, positive et négative

Nous définissons alors la fonction de perte comme :

$$L : (A,P,N) \rightarrow L(A,P,N) = max(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha, 0)$$

Cette fonction s'appelle la *triplet loss* [36], ou perte par triplés.

Nous avons fait ici plusieurs choix de conceptions. D'abord, nous avons utilisé la distance euclidienne alors que d'autres distances existent. Nous pouvons prendre la norme L2 [17] sans son carré, ou remplacer les normes par des cosinus par exemple. De nombreuses fonctions existent et influencent les résultats [42]. Nous nous poserons la question du choix de cette fonction de perte en IV.3.2.1.

Si nous fixons le choix de cette fonction à la *triplet loss* telle que décrite initialement par [36], il reste à déterminer un paramètre important : le choix de la marge. La valeur donnée par le papier originel [36] est

15. Dans la suite, nous utiliserons le terme d'« apprentissage » ou nous dirons que le système « apprend » malgré le fait que cela soit un anthropomorphisme, faute de meilleur terme pour décrire ce processus de mise à jour des paramètres.
16. cf Annexe B pour comprendre comment ces poids évoluent lors de l'entraînement
17. aussi appelé norme euclidienne ou norme de degré 2. Sa définition est donnée en IV.3.2.1.

de 0.2. Néanmoins, il a un impact important sur le déroulé de l'entraînement du modèle, il convient donc de critiquer ce choix, ce que nous ferons en IV.3.2.3.

IV.1.5 Générer des triplés

Pour accélérer les calculs lors de l'entraînement, il est souhaitable de paralléliser les opérations. Pour cela nous créons des paquets d'images, appelés *batchs*. Un *batch* est une liste de taille #batchs constitué de triplés d'images (ancre, positive, négative) passée directement dans le réseau de neurones. Nous répètons ensuite l'opération sur un certain nombre d'*epochs* (#epochs). Au total pendant l'entraînement nous aurons donc passé dans notre réseau 3 x #batchs x #epochs images. Ces opérations prendront du temps : c'est le temps d'entraînement. Ainsi lorsque nous disons qu'il faut entraîner le réseau pendant un certain temps, nous parlons en réalité du nombre de données qui lui sont fournies, ce qui dépend directement des paramètres #batchs et #epochs qui sont un choix de conception important.

Outre ces nombres, il faut également savoir comment nous construisons nos batchs et nos epochs, i.e. quelles identités sont choisies pour représenter les images ancres et positives. Par exemple, est-ce qu'elles sont choisies au hasard, ou est-ce que chaque personne doit apparaître au moins une fois par epoch ?

De façon encore plus précise, pour chaque triplé, comment choisir les images négatives sachant que nous avons une certaine image ancre ou positive ? Il semble évident que ce sera plus facile pour un algorithme de distinguer un homme noir d'une femme blanche car instinctivement nous pouvons penser que la distance entre les deux images sera plus grande que si nous comparons deux femmes blanches (cf figure 17). Mais quelle doit-être la difficulté de cette image négative ? Un négatif trop « facile » pourrait ne pas apprendre beaucoup à l'algorithme tandis qu'une image trop « dure » pourrait soit rendre l'entraînement plus efficace soit le bloquer complètement si l'algorithme n'arrive pas à distinguer les deux identités.

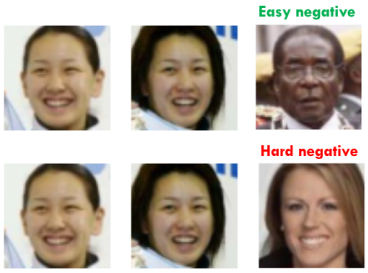


FIGURE 17 – Hard et easy negatives pour des visages

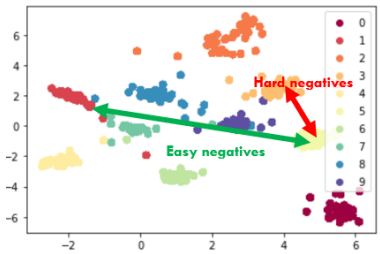


FIGURE 18 – Hard et easy negatives clusters

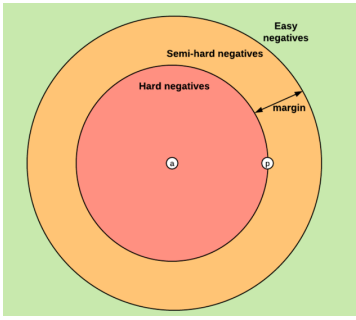


FIGURE 19 – Distance définissant les semi-hard negatives - adapté de [32]

L'article [36] distingue trois catégories d'images négatives : les *hard negatives*, très proches de l'image ancre, les *easy negatives* très éloignées, et les *semi-hard negatives* entre les deux (cf figure 19). Géométriquement, plus les identités sont dures à distinguer, et plus leur clusters respectifs seront proches (cf figure 18).

L'article recommande d'alterner les phases d'apprentissage avec ces différents négatifs, mais la façon et le temps de ces différentes phases est à nouveau un choix fort. Quel serait le résultat si nous abandonnions ce choix de négatifs et que nous les sélectionnions de façon aléatoire ? Si nous choisissons d'utiliser les *hard* ou *semi-hard negatives*, comment cela influence-t-il le temps d'entraînement ? Nous essayerons de répondre au mieux à ces questions dans la partie IV.3.3.3.

IV.1.6 Déterminer la valeur seuil

Une fois le modèle entraîné, nous cherchons à déterminer le seuil pour lequel le modèle classifiera au mieux les images. Pour cela, nous créons des paires d'images, soit positives (les deux images ont la même identité), soit négatives (il s'agit de personnes différentes). Nous les envoyons dans le réseau et nous comparons la distance entre leur gabarits. Les images positives devraient être plus proches que les images négatives. Voir le fonctionnement de l'évaluation figure 20.

En traçant ensuite l'histogramme, nous pouvons décider d'un seuil séparant au mieux les deux groupes d'images (voir figure 9).

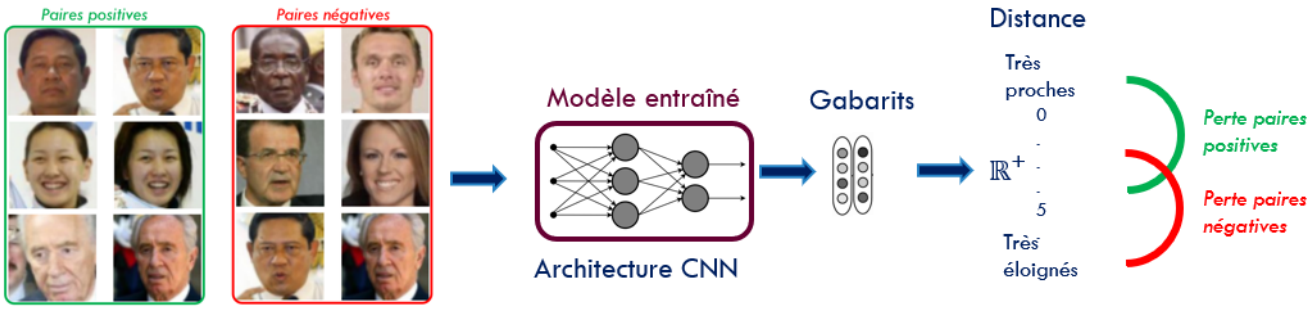


FIGURE 20 – Détermination de la valeurs seuil avec des paires positives et négatives

Le choix de ce seuil est déterminant dans le fonctionnement du système et est très sensible à d'autres paramètres, comme la distribution de population dans la base de données, comme expliqué précédemment (voir partie III.3.6). Nous discuterons du choix de ce seuil dans la partie IV.3.4.2.

IV.1.7 Éléments de sortie

Une fois le modèle entraîné, et le système conçu, il convient de tester ses sorties. Quelle est la précision - ou au contraire, le taux d'erreur - du système et selon quelles métriques ? Pouvons-nous mesurer son taux d'équité (sa *fairness*) et selon quelles métriques ? Ces critères d'équité se font-ils sur des sous-groupes de la base de données et, si oui, lesquels ? Il y a-t-il des techniques pour rendre le code plus « équitable » ? Nous essayerons d'aborder un maximum de ces aspects dans la partie IV.4.2.

IV.2 Organisation pratique

IV.2.1 Valeurs et critères sélectionnés

Notre démarche s'inscrit dans la volonté de développer des exigences de conception à partir de valeurs morales, promues notamment par les textes internationaux sur l'« éthique de l'IA » (voir partie I.1.3). Or nous ne pouvions regarder toutes les valeurs qu'ils mentionnent de par leur nombre important. Nous nous étions initialement focalisée sur les quatre valeurs recommandées par le groupe d'experts de haut niveau de la Commission Européenne [34], à savoir le respect de l'autonomie humaine, la prévention de toute atteinte, l'équité et l'explicabilité. Puis, le travail avançant, nous avons réalisé la tâche importante que cela représentait et avons décidé de nous restreindre à l'équité, celle-ci étant normalement facilement analysable dans le cadre de la reconnaissance faciale. C'est sous ce prisme que nous avons disséqué tous nos choix de conception, jusqu'à l'analyse des résultats.

IV.2.2 Capacités computationnelles

Les services du Premiers ministre ne disposant que d'ordinateurs servant aux tâches administratives et ne possédant pas de GPU [18], il convenait de trouver un moyen de faire tourner notre algorithme. En effet, la reconnaissance faciale faisant partie des techniques de traitement des images (*image processing*), elle est très gourmande en calculs et nécessite l'utilisation de cartes graphiques.

Plusieurs solutions s'offraient à nous. Le code pouvait être développé en local, via des fichiers « .py », et l'entraînement lancé sur des machines virtuelles (VMs) comme celles de GCP [19], AWS [20] ou autre. Il fallait alors se procurer des crédits. L'autre solution était d'utiliser les interfaces de notebooks en ligne, ce qui permet d'avoir accès à des VMs gratuitement mais atteint vite ses limites pour des utilisations prolongées ou importantes, l'accès aux accélérateurs matériels étant limité en temps (quelques heures ou dizaines d'heures selon les interfaces) et en nombre (généralement une seule session peut être ouverte en même temps avec accès à un seul GPU et un ou deux CPU [21]).

18. *Graphics Processing Unit* est une technologie de carte graphique permettant à l'ordinateur d'effectuer du traitement d'images efficacement
19. Google Cloud Plateform
20. Amazon Web Services
21. *Central Processing Unit*

Olivier Grisel m'a suggéré d'utiliser l'interface Google Colab, régulièrement utilisée à des fins éducatives. Mais cette dernière limite l'usage du GPU à quelques heures à l'utilisateur.trice avant de lui demander d'acheter une version payante. Il n'y avait, de plus, aucun moyen de savoir combien d'heures d'utilisation il nous restait avant de nous voir refuser l'accès au GPU, détruisant le kernel et stoppant net l'entraînement du modèle. Nous avons donc décidé de nous pencher vers une autre interface de machines virtuelles également très répandue : Kaggle Kernel. Cette dernière permet d'avoir accès à une machine dotée d'un GPU pendant au minimum 30 heures (et régulièrement plus) par semaine, ce qui suffisait à notre utilisation. Les sessions étaient détruites après chaque utilisation et nous veillions à télécharger en local les fichiers nécessaires (notamment les modèles entraînés) avant de quitter.

IV.2.3 Librairies Python

Comme souvent en apprentissage machine, le code de l'algorithme a été écrit en langage Python, avec utilisation de nombreuses librairies classiques comme NumPy, ou Pandas. Le réseau de neurones a été créé et entraîné sous PyTorch, une des deux librairies les plus répandues en apprentissage profond avec Tensorflow, pour la simple raison que j'étais plus à l'aise avec cette librairie, ayant plus eu l'occasion de développer dessus. De plus, PyTorch est connue pour être plus personnalisable que Tensorflow. L'avantage de Tensorflow aurait été l'ajout d'un *framework* comme Keras permettant d'entraîner facilement des modèles. En l'occurrence, aucun *framework* pour l'entraînement de notre algorithme n'a été rajouté. En effet, il existe de nombreuses extensions à PyTorch, comme PyTorch-Ignite que je connais bien l'ayant pratiqué en cours, mais aucune ne permet, à ma connaissance, d'entraîner un modèle apprenant par *triplet loss*. Elles se concentrent souvent sur des boucles d'apprentissage pour la classification et ne permettent pas assez de modulation pour notre cas d'usage.

IV.2.4 Suivi des expériences

Le suivi des différents entraînements du modèle (ou *run*), d'abord effectué à la main (en affichant notamment les courbes de l'erreur d'entraînement et de validation directement à la suite dans le notebook) a vite été remplacé par une automatisation de l'enregistrement de ces courbes via l'outil Weights & Biases - appelé par la suite « wandb ».

IV.2.5 Organisation des notebooks et des fichiers Python

Tous les fichiers qui ont servi au développement de ce cas école sont trouvables à l'adresse suivante :

https://github.com/mgornet/CNPEN

Un fichier README explique le contexte du stage et comment déployer le code soi-même.

Différents notebooks ont été créés pour notre cas d'usage, ils sont trouvables dans le dossier « notebooks ». De nombreuses itérations ont été faites dessus, pour essayer de nouvelles choses ou vérifier des hypothèses. La version finale du travail en comporte quatre : un pour l'entraînement du modèle (train), un pour l'évaluation et la détermination du seuil (determine_threshold), un servant de bac à sable pour tester des propriétés du réseau, pour l'exploration de données, etc (various_tests) et enfin, un pour l'étude d'équité (fairness_study).

Un certains nombre de modèles pré-entraînés et réutilisables sont trouvables dans le dossier « models ». Ils peuvent être chargés directement dans les notebooks various_tests, determine_threshold ou fairness_study.

Les fichiers nécessaires au fonctionnement du code sont trouvables dans le dossier « files ». On y trouve un fichier « lfw_att_73.mat », contenant les attributs des personnes présentes dans la base de données. Pour plus d'information sur son origine, voir IV.3.1.1. Les autres fichiers sont des fichiers « .py » contenant un certain nombre de fonctions et de classes utilisées par la suite dans les notebooks. En voici un aperçu :

- **prints.py** : contient toutes les fonctions d'affichage, permettant de tracer des graphiques, afficher des images issues de la base de données, etc.
- **builder.py** : contient entre autres, les fonctions permettant de créer le tableau de données et de lui ajouter les attributs des personnes.
- **triplet.py** : contient les classes de différents objets nécessaires au modèle comme une classe TripletLoss représentant la perte, une classe TripletGenerator permettant de créer des triplés d'images comme expliqué en IV.1.5 ou une classe TripletLearner représentant le réseau de neurones.

— **test_train_loops.py** : contient les différentes boucles d'entraînement et de test.

— **classification.py** : contient les fonctions nécessaires à la détermination du seuil de classification et à l'étude d'équité.

Toutes les fonctions et classes comportent une documentation (*docstring*) décrivant leur utilité, leurs paramètres d'entrées ou de sorties et permettant la reprise du code par une tierce personne.

Les notebooks sont déroulés ainsi :

– **train.ipynb** : notebook servant à l'entraînement du modèle. Après avoir installé puis importé les modules nécessaires, cloné notre repo git et téléchargé la base de données LFW (voir partie IV.3.1.1), nous définissons plusieurs paramètres et objets :

- df, all_imgs, df_train, df_train, df_train : nous créons en premier lieu le tableau de données « df » (*dataframe* de la librairie pandas) avec les noms et attributs de chaque personne. Toutes les images sont concaténées dans un tenseur (torch.tensor) « all_imgs ». Ces opérations sont effectuées grâce aux fonctions create_dataframe et extend_dataframe du fichier builder.py. Nous séparons ensuite df en trois : « df_train » qui contiendra les données utilisées pour l'entraînement, « df_valid » pour l'évaluation et « df_test » pour le test. Cette séparation est discutée en IV.3.3.2.
- batch_size : le nombre de batchs, discuté en IV.3.2.8. Il est également possible de définir un batch_valid_size et un batch_test_size si l'on souhaite une taille de batch différente pour les différents jeux de données.
- device : le GPU utilisé, ou s'il n'y en a pas, le CPU.
- model : le réseau de neurones utilisé pour l'apprentissage. Il est d'abord initialisé de façon aléatoire avant de commencer l'entraînement. Nous définissons ses paramètres base_channels (discuté en IV.3.2.4) et dropout (discuté en IV.3.3.4). Il est créé via la classe TripletLearner du fichier triplet.py.
- lr : le *learning rate*, discuté en IV.3.2.5.
- optimizer : discuté en IV.3.2.6.
- scheduler : discuté en IV.3.2.6.
- margin : la marge, discutée en IV.3.2.3.
- criterion : la fonction d'erreur servant au modèle à apprendre. Elle est instanciée par la classe TripletLoss du fichier triplet.py. Elle est discutée en IV.3.2.1.
- epochs : le nombre d'epochs à réaliser pendant l'entraînement si aucun *earlystopping* n'est implémenté. Il est discuté en IV.3.2.9.
- gen_train, gen_valid et gen_test : les générateurs chargés de scanner les données de df_train, df_valid et df_test pour créer des tenseurs de taille (batch_size, 3, h, w) correspondant aux images ancres, positives et négatives. Concrètement, les générateurs sont des listes de taille $n = \lfloor \frac{\#epochs}{\#batch} \rfloor$. Chacun des éléments de la liste contient un tuple de trois éléments correspondant à ces trois tenseurs, que nous pouvons récupérer via l'opération $xa, xp, xn = gen[id_batch]$. Notons que nous pouvons également définir si les générateurs appliquent de la *data augmentation* sur les données (discutée en IV.3.3.1), ou quelle technique de *mining* ils mettent en place pour créer les triplés et choisir notamment les images positives et négatives (discuté en IV.3.3.3). Ils sont des instances de la classe TripletGenerator du fichier triplet.py.
- train_loader, valid_loader, test_loader : de la classe DataLoader de torch.utils.data. Ils servent à créer un itérable à partir des générateurs ainsi qu'à mélanger les données.

Une fois tous les objets créés, l'entraînement peut commencer. Nous commençons par nous connecter à notre compte wandb (fonction login), puis à initialiser un *run* (fonction init). Cela aura pour conséquence d'envoyer nos données directement au site lors de l'entraînement. Enfin nous lançons l'entraînement de notre modèle avec la fonction training du fichier test_and_train_loops.py. Une fois l'entraînement fini, nous pouvons sauvegarder notre modèle sous forme d'un fichier « .pth » et arrêter le *run* wandb (via la fonction finish). L'analyse des choix faits dans ce notebook est développée en partie IV.3.2 et IV.3.3 correspondant respectivement à la façon dont nous obtenons l'*overfitting* puis la façon dont nous essayons de rendre le modèle généralisable sur de nouvelles données.

- **determine_threshold.ipynb** : notebook permettant de déterminer le seuil à partir d'un modèle entraîné. Nous téléchargeons les données et créons les mêmes éléments que précédemment. Cette fois, au lieu d'entraîner un modèle, nous chargeons un fichier « .pth » récupéré à l'étape précédente. Un nouveau générateur et dataloader nommés « gen » et « loader » sont instanciés sur les données de validation. Le loader est alors concaténé plusieurs fois (discuté en IV.3.4.1), puis l'analyse peut commencer. Nous ne redétaillerons pas ici les différentes façons d'obtenir le seuil, et avisons nos lecteurs de lire la partie IV.3.4 contenant les détails des calculs et les courbes tracées.
- **various_tests.ipynb** : notebook servant à réaliser différents tests. Bien que non nécessaire à la création de notre système de reconnaissance faciale, il sert à vérifier, tout au long de la démarche, que les hypothèses émises sont correctes. Il commence par le téléchargement des données et les mêmes instanciations que les notebooks précédents. Puis se succèdent des affichages des dimensions de différents objets, d'images issues de la base de données, avec différentes augmentations (discutés en IV.3.3.1) ou selon certains attributs. Nous y avons également fait figurer les tests réalisés pour la concaténation du valid_loader (discuté en IV.3.4.1), du biais inductif du réseau de neurones (discuté en IV.3.1.3) et de l'erreur initiale du réseau (discuté en IV.3.2.3), des *schedules* de lr (discuté en IV.3.2.6). Nous y avons également implémenté les tests pour la classification, une fois le seuil déterminé. Cela permet, entre autres, d'afficher les images qui ont plus de chances d'être mal classifiées (les paires positives avec une distance importante ou négatives avec une distance faible). Enfin, nous y testons notre système (modèle et seuil), sur des images inédites – comme des photos de l'autrice du rapport.
- **fairness_study.ipynb** : notebook servant à réaliser l'étude sur l'équité des sorties de l'algorithme. Après avoir téléchargé les données, défini les objets et paramètres, chargé un modèle et déterminé son seuil, le notebook commence sur la création d'un nouveau dataframe : df_fairness, réalisée par la fonction build_df_fairness du fichier classification.py. Une exploration des données est alors menée ainsi qu'une étude par sous-groupes, décrites toute deux dans la partie IV.4.2.

IV.2.6 Rendus

Le code et sa documentation techniques sont disponibles librement sur internet à l'adresse :

https://github.com/mgornet/CNPEN

Le présent rapport sera disponible le 15 septembre 2021 et rendu aux tuteur.trice.s, au référent école et au jury. Il sera accompagné d'une fiche de synthèse de deux pages et fera l'objet d'une soutenance le 30 septembre.

Il est envisagé que ce travail fasse l'objet d'un article scientifique.

IV.3 Choix effectués

Nous avons choisi de séparer cette partie en quatre, selon les catégories de choix que nous considérons. Ces catégories ont été créées pour représenter de façon presque chronologique le processus de développement du code dans un cas idéal. En aucun cas elles ne représentent une supériorité hiérarchique d'un choix sur un autre en termes d'implications sur le résultat final. Il est clair que le choix de la base de données (partie choix initiaux), le choix du *learning rate* (partie entraînement), le choix du *preprocessing* (partie validation) ou le choix de la valeur seuil (partie test) sont tous des exemples de choix déterminants dans le processus de développement et ne sauraient être hiérarchisés.

Le processus de développement s'effectue dans l'ordre suivant :

- Les choix effectués initialement sont ceux qui déterminent la structure globale de la technologie. Ils ont des conséquences très importantes puisque les changer reviendrait à changer du tout au tout notre système. C'est le cas de la base de données utilisée, du choix d'utiliser un ConvNet ou de développer un système qui fonctionne par triplet loss. Nous examinerons pourquoi nous avons fait ces choix, mais nous n'évaluerons leurs conséquences qu'hypothétiquement car nous n'avons pas suffisamment le temps de créer plusieurs systèmes différents.
- Les choix effectués pour l'entraînement, correspondent à ceux que nous faisons dans la première étape

de notre développement : provoquer l'*overfitting*[22]. En effet, lorsque l'on réalise un réseau de neurones, il convient de voir si le modèle choisi peut surapprendre sur les données. Ainsi, nous saurons si la taille du réseau et les hyperparamètres choisis permettent de converger vers un résultat. Pour cela, nous voulons que la courbe d'entraînement descende le plus bas possible (près de zéro). Nous savons que nous avons alors trouver les paramètres qui permettent en théorie d'apprendre sur les données.

– Les choix de validation sont ceux qui permettent de généraliser le modèle en faisant baisser la courbe de validation. Une fois que nous savons que le modèle peut apprendre correctement car nous avons réussi à le faire surapprendre, nous essayons d'éviter l'*overfitting* en augmentant les données, en rajoutant du *dropout* ou de l'*earlystopping* par exemple.
– Les choix de test sont ceux qui nous permettent de déterminer la valeur seuil. En effet, une fois que nous avons un modèle prêt à l'usage qui distingue correctement les identités nouvelles, nous voulons tester ce modèle et nous rapporter à un système binaire qui classifie deux images comme d'identités identiques ou différentes.

Nous pouvons remarquer que ces choix ne sont pas indépendants les uns des autres et nous avons représenté ces dépendances dans la figure 42.

IV.3.1 Choix initiaux et postulats

IV.3.1.1 Base de données utilisée

Plusieurs bases de données académiques existent et sont disponibles librement sur internet.

Parmi les plus connues, nous pouvons citer par exemple PubFig, CelebA, CASIA-WebFace ou VGGFace2. De nombreuses bases de données ont émergées suite aux accusations de biais retrouvés dans des logiciels commerciaux [3]. Ces bases sont plus équilibrées, notamment en termes de genre ou de race supposée. Toutefois elles sont rarement utilisées en pratique, sauf dans un cadre académique. Ici, il est de plus intéressant de prendre une base de données plus commune et moins équilibrée pour appuyer les choix en matière de *fairness*.

Nous voulions également une base de données qui ne soit pas trop lourde en termes de poids (pas trop d'images et pas de trop grande qualité) pour qu'elles puissent être traitées directement sur mon ordinateur personnel ou sur une interface en ligne (Google Colab ou Kaggle Kernel), sans avoir à déployer de clusters de calculs. Cela exclut par exemple les bases IJB plus utiles pour l'analyse de vidéos.

Nous voulions également que les données soient labellisées pour pouvoir étudier les critères de *fairness* selon différents sous-groupes (à minima selon le label homme-femme). Ces labels, parfois très problématiques car extrêmement subjectifs et arbitraires (cf IV.4.2.1) sont toutefois nécessaires pour étudier la balance d'une base de données. Une meilleure approche aurait sans doute été de récolter nous-mêmes les données et de demander à chaque personne quel label elle souhaitait porter. Néanmoins, cela aurait été bien trop chronophage et éloigné de la réalité industrielle. Nous avons donc opté pour une base de données annotée. Il n'en restait alors que deux qui répondaient aux critères précités : PubFig et LFW. Or PubFig ne contient que 200 identités différentes ce qui est trop peu pour faire de la reconnaissance faciale.

Nous avons finalement choisi le set de données Labelled Faces in the Wild (LFW), attribué à Gary B. Huang et al [25]. C'est une base de données contenant des images de personnalités célèbres, prises dans un environnement quotidien et n'ayant pas spécialement posé pour l'occasion. L'article et la base de données sont trouvables sur le site internet des auteur.trice.s. Pour remonter à comment cette base de données a été créée, il faut remonter à une étude plus ancienne, puisque les articles récents comme notre article de référence [36] ne commentent que rarement ce choix et son origine. Les études citent souvent le travail de Gary B. Huang et son équipe [25] comme origine de la base de données, mais cette dernière est en fait une reconstitution mieux organisée d'un travail initié par Tamara Berg à l'université de Berkeley [2] en 2004. La chercheuse avait alors appliqué un détecteur de visages à environ un demi-million d'images d'actualité sous-titrées, trouvées sur le site internet Yahoo ! News, pour en extraire à chaque fois un visage associé à un nom. Gary B. Huang et son équipe ont ensuite repris ces quelque quarante-quatre mille images pour les trier en passant dessus le détecteur de visages de Viola-Jones [41], très connu dans le domaine de la reconnaissance faciale. Pour chaque image, s'il n'y avait pas détection, que le nom de la personne ne pouvait pas être retrouvé, ou que l'image était en doublon, elle était omise de la base de données. Enfin, chaque

22. Surapprentissage en français. Phénomène où le réseau de neurones n'est plus capable de généraliser car il utilise des paramètres spécifiques au set de données d'entraînement.

image était recadrée et redimensionnée. La base de données finale (que nous avons utilisée), contient 13233 images de 5749 personnalités différentes, dont 1680 ont au moins deux images. Chaque image est étiquetée par le nom de la personne qu'elle représente. Les images sont de format 250 x 250 pixels et la plupart sont en couleurs.

Il convient également de mentionner que dans le papier original de la *triplet loss*[36], le dataset LFW n'est utilisé que pour les tests mais pas pour l'entraînement du modèle. Sauf erreur de notre part, l'article ne dévoile pas sur quelles données FaceNet a été entraîné, malgré le fait que ce choix est crucial pour pouvoir critiquer les résultats. Il est probable que les données soient des images de visages appartenant à Google et non issues d'une base de données publique. Nous ne connaissons donc ni le nombre de données, ni leur distribution, seule leur résolution en pixels est mentionnée. Le test sur des données publiques permet la comparaison à d'autres systèmes dans un but marketing.

Pour notre travail, nous souhaitions que les données soient labélisées pour étudier les problèmes de *fairness* du code. Nous avons donc utilisé la même base de données pour l'entraînement et pour les tests, tout en veillant à séparer les données initialement pour ne pas fausser les résultats (voir la séparation des train et test sets en partie IV.3.3.2).

IV.3.1.2 Attributs utilisés

Les attributs que nous avons utilisés proviennent d'un article de Ziwei Liu et al [30]. Le site internet des auteur.trice.s contient l'article et les fichiers attributs, téléchargeables en bas de la page. Pour prédire ces attributs, les auteur.trice.s utilisent un réseau convolutionnel, autrement dit, l'étiquetage n'est pas fait à la main mais de façon automatique. Cette méthode comporte nécessairement des erreurs qu'il conviendra d'évaluer par la suite (voir IV.4.2.1).

La classification des individus selon certains attributs soulève la question de l'étiquetage, régulièrement posée en apprentissage machine. Comment étiqueter les données de sorte à respecter les volonté des individus ? Il faut se demander ce que nous devons annoter, comment vérifier que cette annotation est correctement réalisée et quelles instances peuvent être en capacité de mener cette vérification. La qualité de l'annotation doit également être regardée. Souvent, une classification binaire ne suffit pas à inclure tous les individus. L'exemple le plus couramment rencontré vis-à-vis de cette classification binaire est celui du genre. Alors que, dans la société actuelle, les voix se multiplient pour faire reconnaître que le genre ne saurait se résoudre à une binarité du l'ordre du masculin et du féminin et devrait davantage être considéré comme un spectre dont toutes les nuances seraient valides, l'étiquetage pour l'apprentissage machine n'a que peu évolué et il est quasiment impossible de trouver actuellement des annotations qui vont plus loin que la limitation 0-1. De plus, cette classification dans des cases restreintes se fait bien souvent sans le consentement ou même la simple approbation de la personne, amenant à des situations où les individus ne considèrent pas que l'étiquette qu'on leur a assignée soit la bonne. Bien qu'ici la classification soit réalisée automatiquement, il convient de mentionner que, le plus souvent en apprentissage machine, l'annotation est faite à la main par des micro-travailleur.se.s dont l'exploitation par des grandes entreprises est régulièrement soulevée dans les débats. Pour contrer cela, le monde académique a poussé la recherche dans le domaine de la classification automatique des données. Mais cette méthode ne résout pas tout. Il faudrait garder un contrôle par l'humain des résultats, or cela est tout autant chronophage que d'annoter à la main puisque chaque donnée doit être regardée. De plus, l'annotation automatique pourrait devenir une excuse des ingénieur.e.s et chercheur.e.s chargé.e.s du système d'étiquetage automatique, permettant de se dédouaner de leur responsabilité d'annotateur.trice.s. Nous souhaitons rappeler ici que cette responsabilité doit être maintenue dans tous les cas et que le système automatique ne saurait être « responsable » d'une mauvaise annotation.

Les attributs que nous avons récupérés sont au nombre de 73. Ils sont fournis sous un format « .mat » qui pourra être transformé en un dictionnaire Python (voir fonction extend_dataframe de builder.py). Nous utiliserons principalement les attributs « Male », « Asian », « White », « Black » et « Indian ». Nous pourrons également regarder des attributs comme « Child », « Bald » ou « Eyeglasses » par exemple, pour étudier les problèmes de déséquilibre des données sans manipuler d'attributs sensibles.

Nous avons déjà parlé des problèmes soulevés par la classification binaire du genre, mais ici il convient également de critiquer cette classification en ethnies ou races supposées, très courante dans la culture américaine mais ici utilisée de façon extrêmement abusive, les personnes ayant été classées sans leur accord dans ces cases restreintes. Une autre classification de plus en plus répandue et plus éthiquement correcte consiste à classifier sur la couleur de peau selon l'échelle de Fitzpatrick, mise au point en 1975 par le dermatologue du même nom. Cette échelle vise à classer le teint d'une personne en fonction de sa tolérance à la lumière

du soleil et inclut six types et couleurs de peau différents [21]. Malheureusement, nous n'avons pas réussi à trouver une base de données avec attributs publique qui utilise cette classification. Nous ne nous attarderons pas sur d'autres attributs de la classification de Ziwei Liu qui nous semblent absurdes, comme les étiquettes « Attractive Man » ou « Attractive Woman » qui sont totalement déplacées et que nous ne pouvons cautionner.

Malgré ces (très) nombreux problèmes, nous avons choisi de continuer à utiliser ce fichier d'attributs, par manque de solution alternative. Notons tout de même que les auteur.trice.s de l'article fournissant la base de données initiale [25] ont également généré des attributs par une méthode automatique [27]. Néanmoins, ils ont laissé la sortie de leur réseau telle quelle, ne donnant pas un résultat binaire (exemple : la personne possède des lunettes ou n'en possède pas) mais un résultat dans $\mathbb{R}$ difficile à interpréter. De plus, les attributs utilisés sont les mêmes que ceux du fichier de Ziwei Liu, ne réglant pas les problèmes éthiques qu'ils soulèvent.

IV.3.1.3 Choix d'un réseau convolutionnel

L'architecture utilisée dans l'article de référence est un réseau convolutionnel (ConvNet) [23]. Or ce genre de structure très profonde peut capter des invariances et similarités locales correspondant aux identités des personnes.

Cela signifie qu'avant même que le réseau soit entraîné, il pourrait reconnaître quelques personnes, alors même que les poids sont initialisés aléatoirement. C'est ce qu'on appelle un biais inductif. Ce phénomène est visualisable figure 21.

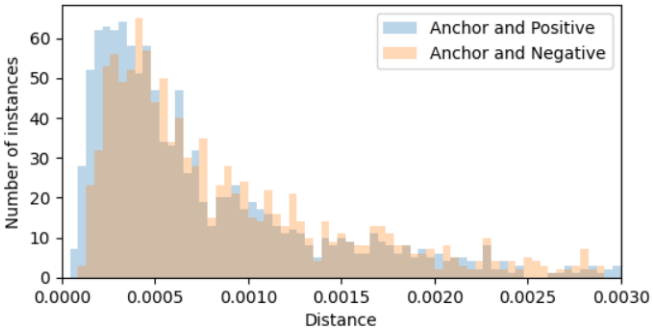


FIGURE 21 – Histogramme des distances avant entraînement

Sur la figure 21, alors que les poids du réseaux sont aléatoirement choisis, nous voyons que les histogrammes de distances ne se superposent pas exactement. Ce biais peut influencer les résultats des critères de *fairness*.

IV.3.1.4 Perte par triplés

Il existe plusieurs techniques pour développer des systèmes de reconnaissance faciale. Les plus connues sont les réseaux siamois et la perte par triplés. Néanmoins, ce domaine de recherche n'en est qu'à ses débuts et des architectures sont régulièrement développées et mériteraient d'être testées. Voir une frise de l'évolution de ces techniques figure 22.

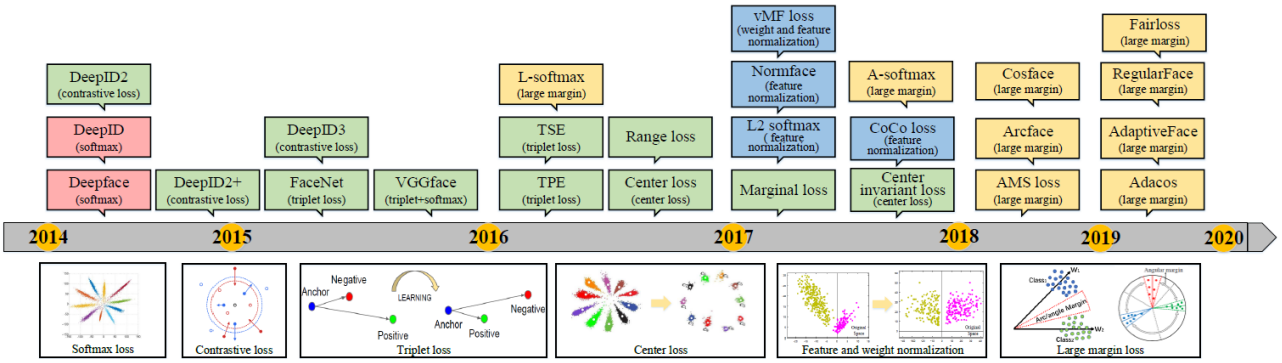


FIGURE 22 – Etat de l'art des fonctions de perte pour reconnaissance faciale - adapté de [42]

23. Voir Annexe A pour une explication sur la structure d'un ConvNet

Ici nous souhaitions une technique qui soit un compromis entre une étude récente et couramment utilisée. En effet, plus les techniques sont récentes et moins elles sont répandues dans le monde industriel car il y a une certaine inertie à les mettre en œuvre. De plus, de nombreuses méthodes, bien que prometteuses sur le plan académique, ne franchissent jamais le pas de l'industrialisation car elles représentent trop de contraintes pour le domaine.

Ainsi, il nous a semblé judicieux de partir sur un système fonctionnant par perte par triplés car l'article original n'est pas trop ancien (2015) [36], la technique a été développée par le géant du numérique Google et semble bien s'implémenter sur le marché industriel de la reconnaissance faciale, la recherche autour de cette technique étant très prospère.

IV.3.2 Choix d'entraînement : recherche de l' *overfitting*

IV.3.2.1 Fonction de perte

Dans le papier sur lequel nous nous basons [36], la fonction de perte utilise la distance euclidienne carrée :

$L(A, P, N) = max(\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha, 0)$ [24]

Certains utilisent également la distance euclidienne standard (sans le carré) :

$L(A, P, N) = max(\|f(A) - f(P)\|_2 - \|f(A) - f(N)\|_2 + \alpha, 0)$

Ou le cosinus :

$L(A, P, N) = max(cos(f(A) - f(N)) - cos(f(A) - f(P)) + \alpha, 0)$

Ici il a d'abord été choisi la distance euclidienne carrée pour mieux pouvoir comparer les résultats au papier initial. La perte par la distance euclidienne standard a été écartée car trop difficile à surveiller à cause de la racine carrée introduite par la norme euclidienne. En effet, si l'on porte la norme au carré, on supprime le problème en zéro.

Selon certaines études l'utilisation de la perte en cosinus peut être avantageuse [42] mais requiert différents ajustages des paramètres de marge et de *learning rate* (lr). Nous avons essayé d'entraîner notre modèle de la sorte mais pour une raison inconnue, l'entraînement ne démarrait pas : le système ne semblait pas apprendre des données et nous obtenions une courbe en plateau (comme celle représentée figure 25).

Il existe des milliers de fonction de perte possibles, c'est un domaine de recherche à part entière [42]. Néanmoins, toutes ces fonctions sont plus complexes et il est difficile de saisir leur comportement et leur implication sur le modèle final.

Ici, nous pourrons faire le choix de prendre la fonction de perte qui permet d'arriver au résultat le plus précis (*accuracy* élevée) ou à l'entraînement le plus court. Mais pour la compréhension du système et de son comportement nous en resterons à la distance euclidienne carrée.

IV.3.2.2 Prétraitement et régularisation

Le prétraitement (*preprocessing*) correspond à la dimension et qualité que nous donnons à l'image, c'est-à-dire comment nous la recadrons (*croping*) puis comment nous la redimensionnons (*resizing*). Enfin, c'est aussi déterminer si nous normalisons les valeurs des pixels (soit en soustrayant la moyenne et en divisant par la variance, soit en soustrayant le minimum et en divisant par le *range*), si nous passons l'image en nuance de gris, etc. Nous n'avons ici cité que les prétraitements les plus communs, mais cela comprend bien entendu tout ce qu'un.e développeur.se peut inclure dans son code qui transformerait les images avant que celles-ci soient traitées par l'algorithme.

Ces transformations, en apparence inoffensives, peuvent s'avérer être des choix cruciaux qui influencent le résultat et la précision de l'algorithme. La normalisation a par exemple l'avantage de remettre les images sur une même échelle mais en retour, il peut être plus difficile de distinguer les nuances.

Outre le prétraitement, nous pouvons également mentionner la régularisation qui peut être implémentée sur ces systèmes, par exemple sur la première couche du réseau de neurones. La régularisation est très

24. La notation $\|.\|_2$ désigne la norme euclidienne obtenue à partir du produit scalaire : $\|x\|_2 = \sqrt{|x|_1^2 + ... + |x|_n^2}$

utilisée en apprentissage machine car elle permet, entre autres, de réduire davantage l'erreur de test. Or dans certains cas, elle peut empêcher de voir la couleur des images et favoriser la luminosité, ce qui peut avoir pour conséquence de durcir l'apprentissage de certains visages à la peau foncée. Ce choix, comme beaucoup d'autres cités ici, est représentatif des choix de valeurs que nous faisons sans même en avoir conscience. En mettant une couche de régularisation « parce que cela améliore les performances », le ou la développeur.se se dédouane de sa responsabilité, prétextant n'avoir pas mesuré l'impact en termes d'équité. Ces démarches d'évitement sont nombreuses et il convient de réaffirmer la responsabilité des acteurs et d'éviter de réduire les problèmes de *fairness* à « un problème de données ».

Ici, nous avons choisi de redimensionner les images en 100px par 100px puis de les recadrer en prenant les 60px centraux (de 20 à 80). Cela permet de zoomer sur l'image pour ôter certains signes distinctifs du décor et forcer le modèle à se concentrer sur le visage mais cela peut néanmoins pénaliser certaines images qui seraient déjà prises de près.

Nous pourrions imaginer augmenter cette résolution pour améliorer la précision de notre algorithme mais le traitement des images serait alors plus long et l'entraînement durerait bien plus longtemps. Ici nous nous sommes contentée de vérifier que les identités étaient reconnaissables à œil humain, c'est-à-dire que la qualité des images après ce découpage soit suffisante pour distinguer correctement les visages.

Une meilleure méthode aurait sans doute été d'utiliser en amont un détecteur de visage qui zoome pour chaque image juste assez pour que le visage occupe la majeure partie de la photo. Néanmoins cette technique était assez difficile à mettre en œuvre car elle supposait de déployer un système supplémentaire. De plus, il y aurait alors une possibilité que les biais potentiels du premier système se répercutent sur l'ensemble. Par exemple si une certaine catégorie de population était moins bien détectée qu'une autre, les images seraient mieux prétraitées pour certaines personnes que pour d'autres. Ce point pourrait-être intéressant à creuser. À notre connaissance, il existe énormément de recherches sur l'implication du *preprocessing* sur la performance du système et notamment sur la généralisation des résultats, mais très peu sur ses implications en termes de *fairness*.

Nous avons choisi de ne pas avoir recours à la normalisation. Le système semblait ne pas apprendre lorsque les valeurs des pixels se trouvaient entre 0 et 1. Cela est sans doute dû à un problème d'initialisation du réseau, notre réseau s'initialisant dans $\mathbb{R}$ et non dans $[\![0, 1]\!]$.

Nous avons également choisi de stocker les images et les poids du réseau de neurone en float32, moins lourd qu'en float64. Cela permet d'avoir un temps de traitement beaucoup moins long mais peut jouer sur la précision des calculs.

IV.3.2.3 Valeur de marge

La valeur de marge correspond à l'endroit d'où « part » le modèle initialement. Si nous affichons l'histogramme de la perte après avoir initialisé le réseau, toutes les valeurs sont proches de la marge (cf figure 23). Si nous regardons les courbes d'entraînement, nous voyons qu'elles partent de la valeur de marge.

Nous comprenons que cela aura une influence sur l'entraînement : plus nous partons de loin, plus il faudra longtemps pour arriver à zéro. Mais ce paramètre est à considérer en parallèle du *learning rate* (lr) par exemple : plus nous prenons une marge importante et plus nous pourrons nous permettre de choisir un lr plus grand car les variations stochastiques du gradient auront moins d'importance.

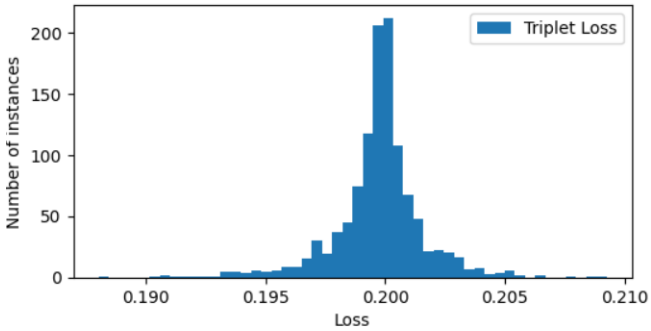


FIGURE 23 – Histogramme de la perte avant entraînement (pour une marge de 0,2)

La marge est également la valeur avec laquelle est comparée la distance entre les images. Selon la façon dont est initialisé le réseau et la façon dont nous avons prétraité les images, la distance entre les images avant apprentissage ne sera pas la même et donc cette comparaison ne sera pas effectuée de la même façon,

rendant l'apprentissage potentiellement plus difficile. Couplé au biais inductif, nous pourrions imaginer que cela puisse avoir une influence sur l'équité du modèle.

Ici nous avons choisi la valeur de 0.2 car c'est celle de l'article initial [36]. Néanmoins, nous pouvons faire des tests pour des valeurs plus ou moins élevées et nous rendre compte que, en apparence du moins, elle va simplement influencer la façon dont nous devons choisir les autres paramètres comme le lr. Cela nécessiterait tout de même une étude plus poussée, notamment en termes de *fairness*. Notons toutefois que, dans le papier initial, aucune information n'est donnée quant au choix de cette valeur de 0.2.

IV.3.2.4 Structure du réseau

Il y aurait des milliers de façon d'agencer un réseau convolutionnel. En général, on le rend de moins en moins large, tout en le rendant plus profond. Nous choisissons par exemple de diviser par deux sa taille à chaque nouvelle couche et de multiplier par deux sa profondeur.

Ici, nous pourrions critiquer ce choix et analyser différents réseaux et comparer leur résultats. Cela a été fait dans le papier original [36], bien qu'il s'agisse simplement d'une analyse en termes de précision des résultats et en aucun cas en termes de *fairness*. Ici, nous nous contenterons de cette structure élémentaire.

Une fois que nous savons que nous voulons rester sur cet agencement de couches de plus en plus profondes, nous cherchons quelle profondeur finale doit atteindre le réseau. Or, les couches les plus importantes (car étant les plus profondes, avec le plus de neurones) sont celles de la fin du réseau, il n'est donc pas nécessaire de mettre toutes les premières couches. Par exemple, les images d'entrée sont de dimension (3, 60, 60). Dès la première couche convolutionnelle, nous les forçons à avoir une dimension (base_channels, 60, 60) avec un *kernel* de 3 et un *padding* de 1 pour conserver la résolution 60x60. Le paramètre base_channels est rentré par l'utilisateur.trice. Cela permet d'éviter de devoir rajouter et enlever des couches au réseau à chaque fois que nous étudions l'implication de sa structure. Nous n'aurons sans doute pas exactement les mêmes résultats pour base_channels=24 – i.e. une seule première couche (32, 60, 60), que si nous prenions un réseau avec des couches montant jusqu'à 24 – i.e. une première couche (6, 60, 60) puis une (12, 60, 60), puis une (24, 60, 60). Néanmoins nous estimons que cette approximation ne devrait pas affecter le résultat même si cela resterait à prouver.

Après avoir introduit ce paramètre base_channels, qui contrôle indirectement la taille des gabarits de sortie à un nombre de couches convolutionnelles fixées, nous le faisons varier pour déterminer quelle serait sa valeur idéale. Nous comparons ici les courbes d'entraînement pour plusieurs valeurs de base_channels.

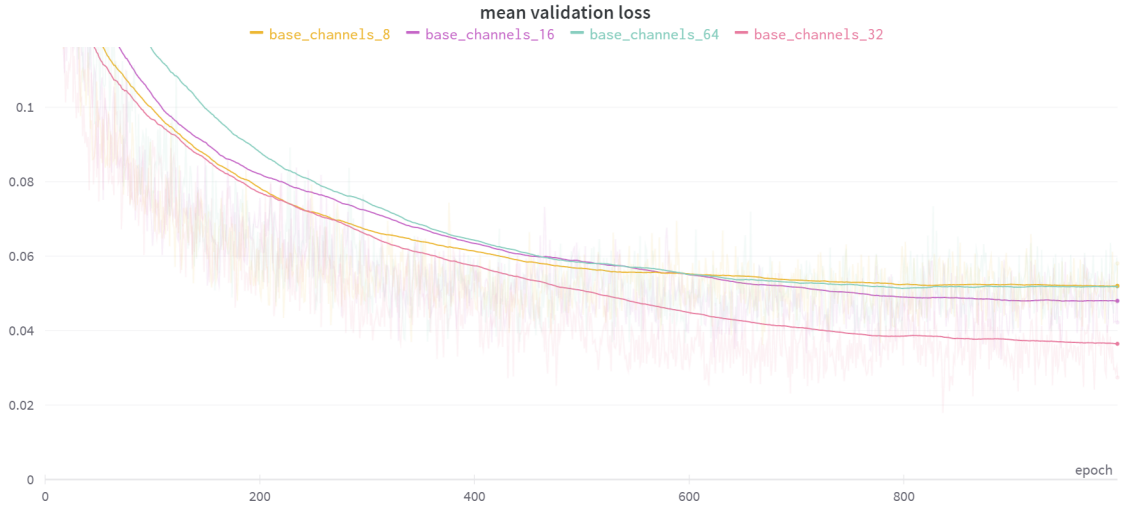


FIGURE 24 – Courbes de validation lissées pour différentes valeurs de base_channels

Dans la figure 24, nous voyons que la validation est légèrement meilleure pour base_channels=32. Notons que selon les *runs*, parfois base_channels=64 donnait aussi de bon résultats mais convergeait en 2h plutôt que 80min pour base_channels=32, c'est pourquoi nous avons finalement choisi base_channels=32.

Puisque la profondeur d'un réseau peut influencer ses paramètres de décision, il pourrait être intéressant de mener l'étude d'équité (de la partie IV.4.2) sur plusieurs modèles plus ou moins profonds.

IV.3.2.5 *Learning rate*

Le *learning rate* (lr) - ou « taux d'apprentissage » en fançais, est peut-être le paramètre qui influence le plus l'entraînement. La tâche d'apprentissage par triplés étant très complexe, s'il est mal adapté, le modèle pourrait ne pas converger du tout.

FIGURE 25 – Courbes des erreurs d'entraînement et de validation : échec de l'apprentissage

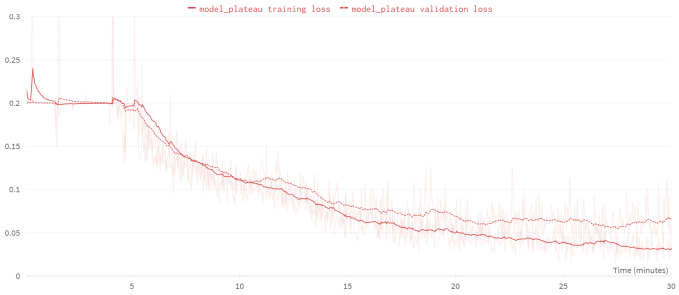


FIGURE 26 – Courbes des erreurs d'entraînement et de validation : problème d'apprentissage

Dans la figure 25, le modèle n'apprend pas. En voyant une telle courbe, nous pouvons nous demander si cela est dû simplement au lr, à une mauvaise combinaison de paramètres ou à un manque de chance sur cet essai. Nous n'avons pas la réponse. Bien sûr, si nous prenons un lr beaucoup trop élevé par rapport aux autres choix de paramètres (selon la marge, la fonction de perte, la profondeur du réseau, etc), cela aggrave nos chances de ne pas converger. Mais parfois, les comportements du gradient ne s'expliquent pas. Ainsi, nous avons remarqué que régulièrement, avant de baisser, la courbe d'entraînement présentait un plateau, comme sur la figure 26. Parfois même, ce plateau ne s'arrêtait jamais et le modèle n'apprenait pas (c'est le cas de la figure 25).

Ici, soulignons que les deux modèles dont les courbes sont affichées en figures 25 et 26 ont été instanciés avec exactement les mêmes paramètres, seulement pour l'un, l'entraînement a pu démarrer tandis que l'autre oscille indéfiniment autour de la valeur de marge. Un gradient trop élevé renforce les chances de voir apparaître ce genre de comportement. Mais quelle que soit la valeur de lr choisie, ce comportement pourra toujours apparaître, peut être moins fortement ou fréquemment, mais il y a toujours un risque.

Cela est dû au fait que nous ne manipulons pas des modèles déterministes et nous devons nous en rappeler chaque fois que nous faisons un choix, écrivons un guide de bonnes pratiques ou formulons une exigence. La reproductibilité de l'expérience est limitée car même avec exactement les mêmes valeurs de paramètres, l'initialisation aléatoire ou la stochasticité du gradient amènent à des résultats différents. Il faudrait alors reprendre exactement chaque étape de calcul pour la reproduire à l'identique (via une *random seed* par exemple), mais cela n'expliquerait pas en termes compréhensibles par des êtres humains le comportement du système. De plus, nous ne pouvons faire des recommandations différentes pour chaque initialisation du gradient, il y en aurait, par définition, une infinité. Nous pourrions vérifier la norme du gradient pour voir si cela est effectivement lié à nos problèmes d'optimisation.

Ici nous nous contenterons de regarder le comportement global du système et essayerons d'en tirer des conclusions les plus générales possibles, mais nous avisons nos lecteurs de garder en tête que cela ne saurait représenter la complexité du problème.

Pour en revenir à notre lr, les comportements en plateau sont très dérangeants pour l'entraînement, nous essayerons donc de les éviter au possible. Pour cela, nous réglons notre lr à une valeur suffisamment faible pour ne pas trouver trop régulièrement ce genre de comportement. Ici, il avait été initialement fixé à 10^{-3} mais observant trop souvent ces plateaux, nous avons décidé de le descendre à 5.10^{-4}. Le modèle en 1.10^{-4} convergeant beaucoup plus lentement, cela nous a paru être un bon compromis. Nous avons alors observé beaucoup moins régulièrement ce comportement en plateau, ce qui semble signifier qu'il y avait une corrélation entre les deux, même si le problème semble plus complexe que cela. Notons tout de même que l'article original fait démarrer lr à la valeur 5.10^{-2}, expliquant simplement qu'ils le faisaient baisser par la suite (voir le *scheduling* de lr en IV.3.2.6). Mais lorsque nous avons essayé cette valeur pour notre réseau, le système ne semblait pas apprendre (comme en figure 25). Cela peut-être dû à une différence d'architecture, de données, d'*optimizer*, ou de réglage d'autres paramètres.

Bien sûr, pour déterminer notre valeur initiale de lr, il faut tenir compte de la profondeur du réseau, définie en IV.3.2.4. Á chaque fois que nous changons ce paramètre, ou le paramètre de marge par exemple, il convient de re-tester quel lr est le plus pertinent. Les réseaux plus profonds (en base_channel=64 par

exemple) semblent présenter plus souvent un comportement en plateau avec un gradient de 5.10^{-4}, si nous cherchions à optimiser le lr pour cette taille de réseau, il faudrait tester de nouvelles valeurs de lr.

Il est généralement recommandé en apprentissage machine, et c'est également ce qui est dit dans notre article référence [36], de baisser le lr au fur et à mesure de l'entraînement. Nous commençons donc l'entraînement avec une valeur en 5.10^{-4} et nous la baissons au fur et à mesure, en la divisant par deux, ou dix par exemple. Ici nous remarquons qu'avec un lr constant de 5.10^{-4}, la courbe de validation commence à atteindre un plateau vers l'epoch 300. Nous la prendrons donc comme première étape de division (*milestone*). Ainsi, en choisissant un lr de 5.10^{-4} jusqu'à l'epoch 300 puis en le divisant par deux toutes les cents epochs, nous voyons que nous arrivons à une erreur d'entraînement quasiment nulle, c'est-à-dire que nous arrivons bien à de l'*overfitting* sur les données, comme représenté figure 27.

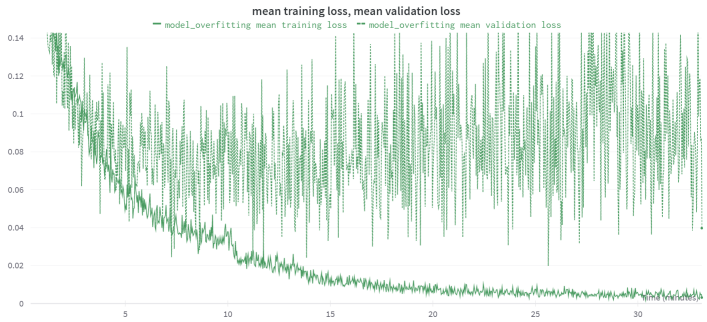


FIGURE 27 – Courbe d'entraînement caractéristique d'un surapprentissage

Il est d'ailleurs possible que le problème de plateau évoqué plus haut puisse être réglé avec un *schedule* de lr adapté (voir notre piste dans la partie suivante).

IV.3.2.6 *Optimizer* et *scheduler*

Nous prenons donc un lr initial de 5.10^{-4} jusqu'à l'epoch 300 puis nous le divisons par deux toutes les cents epochs. Il y a plusieurs façons de réaliser cette variation de lr durant l'entraînement. Nous pouvons redéfinir un *optimizer* avec un lr différent à chaque fois, mais nous perdons alors l'état interne de l'*optimizer* contenant notamment son *momentum*, permettant l'efficacité de la convergence.

Nous pouvons, pour éviter cela, utiliser un *scheduler*. Ce dernier nous permet de faire évoluer le lr de l'*optimizer* à chaque *step* donné. Par exemple, celui que nous avons utilisé était de type « *multistep* » : il multipliait à chaque étape (pour nous aux epochs 300, 400, 500 etc.) le lr par un facteur γ (pour nous égal à 0.5 car nous souhaitons diviser par deux). Nous pourrions aussi prendre un *scheduler* qui permette de faire évoluer le lr en forme exponentielle, cosinus ou toute sorte de fonction (cf figure 28). Le *scheduling* de lr est d'ailleurs un domaine de recherche conséquent. Néanmoins, ici nous arrivons au résultat souhaité (l'*overfitting*) avec un simple *scheduler multistep*, ce qui nous suffit (figure 28, courbe tout à droite).

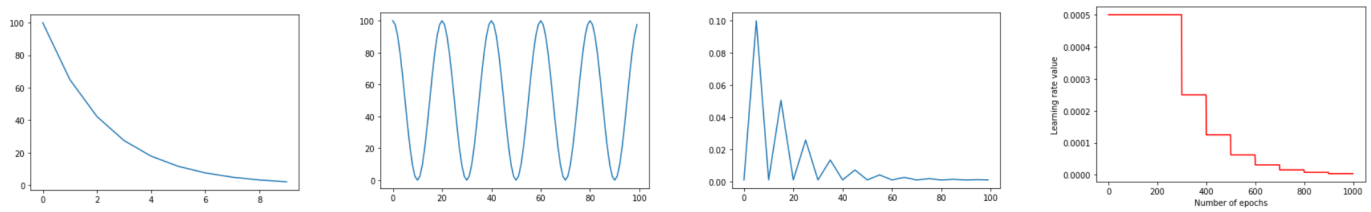


FIGURE 28 – Différents scheduling possibles de lr (en bleu) - adaptés de [28] - et le notre (en rouge)

Il est également probable que certaines méthodes de *scheduling* puissent régler notre problème d'optimisation conduisant à une courbe en plateau (comme le *scheduling* affiché en figure 29). En effet, ce phénomène connu également des *data scientists* travaillant avec des *transformers* pour le traitement automatique du langage (une autre branche de l'apprentissage automatique), peut souvent être atténué par une croissance du lr sur les quelques premières epochs. Malheureusement, torchvision, qui est une librairie intégrée à PyTorch et que nous utilisons pour les *optimizers* et *schedulers* ne comporte pas de classes permettant de réaliser une montée puis une descente de lr. Il faudrait pour cela écrire notre propre classe dérivant de la classe *scheduler* de base, mais cela est un travail difficile qui ne nous a pas paru être une priorité, les problèmes d'optimisation ne survenant pas assez souvent pour handicaper le travail de recherche.

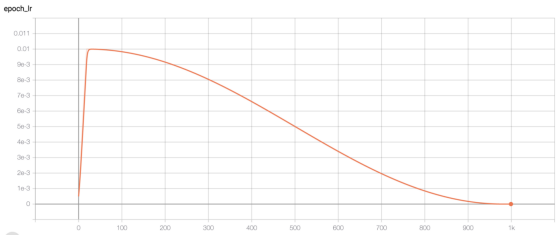


FIGURE 29 – Scheduling de lr en rampe linéaire pouvant régler certains problèmes d'optimisation - adapté de [8]

L'*optimizer* utilisé est également important. Nous avons expérimenté principalement avec Adam [26], mais en avons essayé d'autres (AdaGrad, etc.) qui donnaient des performances moindres et n'ont donc pas été retenus.

IV.3.2.7 Structure d'une epoch

Au début, la technique utilisée pour construire une epoch consistait à construire un certain nombre de paires d'images positives aléatoirement parmi les paires possibles grâce à l'outil itertools de Python. Nous nous sommes alors rendu compte que cette méthode favorisait les identités qui possédaient le plus d'images dans la base de données : elles avaient plus de chances d'être tirées au sort. Cela se traduit par une variation importante lors de l'entraînement sur la courbe de validation. En effet, certaines paires sont bien plus « dures » à reconnaître pour le modèle qui a très peu vu ces identités.

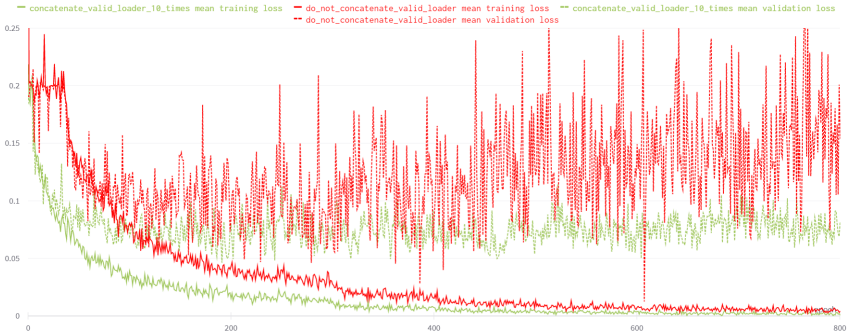


FIGURE 30 – Oscillations des courbes de validation avec ou sans concaténation du valid_loader

Finalement, nous avons choisi de construire une epoch en faisant une passe sur toutes les identités présentes dans le set de données d'entraînement (et de même pour les sets de validation/test). Ainsi, chaque identité est représentée une fois et une seule dans une epoch. Cela conduit à avoir des identités qui ont toujours les mêmes images (car elles sont peu représentées) tandis que d'autres identités sont représentées à chaque epoch par des images différentes car la base de données possède beaucoup de photos différentes de ces mêmes personnes. Ceci peut conduire à des problèmes de *fairness* car le modèle surapprend plus facilement sur les identités moins représentées et ne généralise pas suffisamment. En effet, certaines identités ont plus de 500 images dans la base de données alors que d'autres n'en ont que deux, nous comprenons alors que le modèle pourrait apprendre à mieux reconnaître celle qui a le plus d'images dans des positions différentes car il aura « vu » plus de données différentes la concernant. Mais il ne faut pas pour autant faire de conclusion hâtives, décider de représenter chaque identité dans une epoch est un choix sans doute plus judicieux en termes de *fairness* que de prendre des paires indépendamment comme nous le faisions initialement et cela réduit de beaucoup le biais initial de la base de données.

Nous remarquons que même si le problème de variation sur la courbe d'entraînement a diminué, il reste présent (figure 30). En effet, choisir de représenter chaque identité une fois par epoch limite notre choix de la taille du batch car nous n'avons qu'un nombre d'identités fixé. Dans le validation set avec la séparation que nous avons choisie en IV.3.3.2, nous n'avons que 79 identités différentes comportant plus de deux images dans la base de données. Ce sont donc ces 79 personnes seulement sur lesquelles notre modèle peut être validé. La *batch size* de validation sera au maximum de 79 (cf IV.3.2.8). Nous avons donc décidé de représenter plusieurs fois les personnes par batch pour augmenter artificiellement ce nombre sans avoir l'inconvénient de prendre des paires aléatoirement comme nous le faisions au début. Cette fois, nous prenons le batch initial

que nous concaténons 10 fois. Ainsi, nous n'avons toujours que 79 identités mais nous prenons pour chacune d'elles 10 fois une paire d'images aléatoirement, ce qui permet de moyenner sur plus d'images et donc de réduire la variabilité.

Cette fois, nous remarquons effectivement que la variabilité moyenne pour une epoch diminue, mais la variabilité sur chaque batch reste semblable (la taille du batch est toujours la même). Nous pouvons alors essayer de jouer sur le nombre de batchs (partie suivante).

IV.3.2.8 Nombre de batchs

Nous avons tenté plusieurs taille de batch. Nous commençons par 128, taille de batch classique, et nous regardons ce qu'il se passe si nous l'augmentons, et si nous le diminuons. La variabilité des courbes d'entraînement et de validation diminue avec la taille du batch et la rapidité de calcul augmente. Néanmoins, nous finissons par causer des problèmes de mémoire car la taille du batch est directement proportionnelle à la taille des *feature maps*.

Nous avons pris la même taille pour le *train set*, le *validation set* et le *test set* mais il est également possible de faire varier le nombre de ces batchs. La valeur 128 a finalement été adoptée pour tous les sets.

IV.3.2.9 Nombre d'epochs

Il convient d'arrêter l'entraînement après un certain nombre d'epochs car le modèle n'apprend plus et ne conduit qu'à l'*overfitting*. Cela se voit notamment car la courbe de validation forme un plateau, voire remonte. Ici nous cherchons à provoquer l'*overfit* en premier lieu, nous regardons donc le nombre d'epochs pour lesquelles l'erreur d'entraînement est proche de zéro. Cela arrive après environ 1000 epochs selon les paramètres. Ce nombre est amené à changer lorsque nous passons à l'étape suivante de généralisation car il faut alors absolument éviter l'*overfitting* et arrêter l'entraînement avant. Pour cela, nous pouvons vérifier à la main quand la courbe de validation semble stagner ou utiliser du *earlystopping* (voir partie IV.3.3.4)

IV.3.3 Choix de validation : généraliser les résultats

IV.3.3.1 Augmentation

L'augmentation des données, ou plus couramment *data augmentation*, est la technique la plus répandue pour augmenter la généralisation du modèle. Il suffit simplement de prendre les images d'entraînement et de leur appliquer des changements aléatoires pour tromper le modèle et faire comme si les images étaient bien plus nombreuses qu'elles ne le sont en réalité.

Le plus élémentaire est de réaliser un retournement horizontal (*horizontal flip*) avec une probabilité p. Par exemple pour $p = 0.5$, le modèle aura une chance sur deux de voir l'image originale et une autre de voir l'image miroir. Nous pouvons également retourner l'image verticalement (*vertical flip*), la zoomer ou la redimentionner (*crop and resize*), la dézoomer (*pad*), la tourner (*rotation*), jouer sur les couleurs (*color jitter*)... Un exemple de ces différentes augmentations est disponible en figure 31.

Il convient de toujours vérifier quelles sont les modifications appliquées à l'image en les affichant avant et après transformation. Par exemple si nous ne reconnaissons plus du tout la personne car l'image est trop déformée, cela peut dégrader l'entraînement. Malgré tout, il y a des conséquences à cette augmentation des données sur les résultats qui ne sont pas forcément visibles à l'œil humain. Mais s'assurer que les identités sont reconnaissables est une première précaution.

FIGURE 31 – Différentes *data augmentations*. De gauche à droite : image originelle, *horizontal flip*, zoom 20%, rotation 5°, *color jitter*, image finale.

Dans la figure 32, nous voyons que la rotation introduit un artefact mais à petite dose, la rotation peut être bénéfique pour l'entraînement.

Nous nous limitons donc à des transformations minimes comme le retournement horizontal, la rotation de -5 à 5° ou le zoom de 20%. Cela permet de bouger légèrement les pixels d'une image à l'autre pour que le modèle les considère comme différentes sans pour autant détruire l'identité de la personne.

Nous pouvons tester pour différentes augmentations de données les résultats sur les courbes de validation.

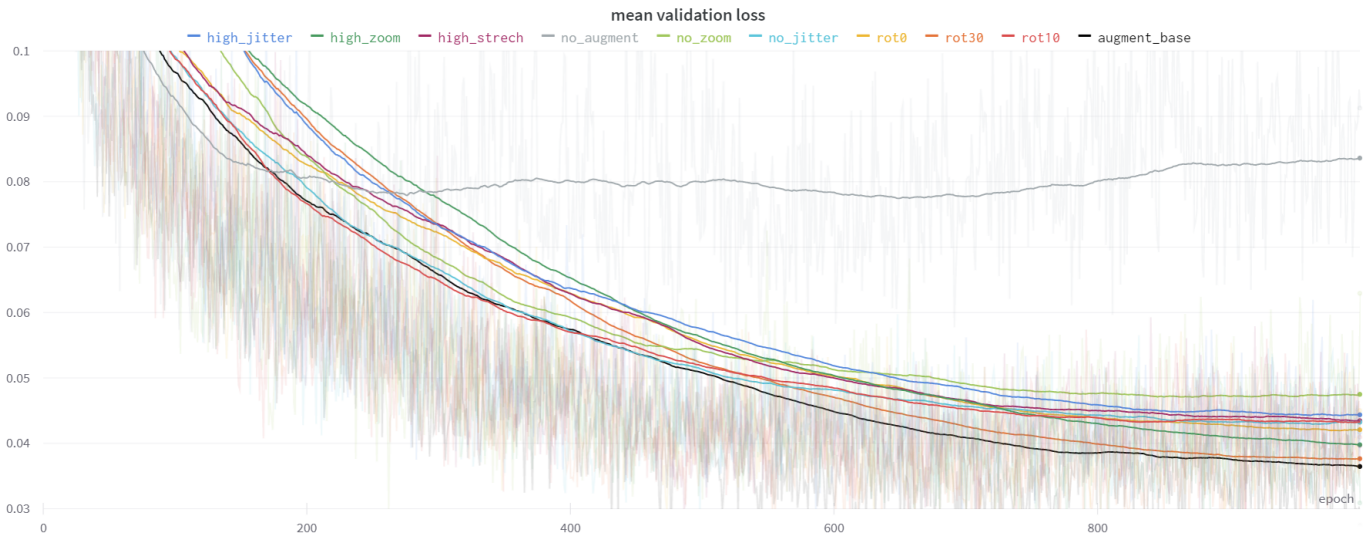


FIGURE 32 – Courbes de validation lissées pour différentes *data augmentation*

Il faut aussi tester l'impact de cette augmentation en termes de *fairness*. Pour cela, une première étape est de vérifier l'impact de l'augmentation pour différents groupes de personnes. Ainsi par exemple, nous nous rendons compte que lorsque que nous jouons sur la luminosité, nous avons plus de chances de détruire l'image de personnes blanches lorsque la luminosité augmente beaucoup et celle de personnes noires lorsque la luminosité est beaucoup trop réduite. Bien sûr, cet exemple caricatural ne saurait représenter toute la complexité du problème mais permet d'expliquer l'impact que peut avoir cette augmentation en termes d'équité. Une étude approfondie de plusieurs modèles ayant appris sur différentes données augmentées permettrait de quantifier cet impact.

IV.3.3.2 Séparation entre test et entraînement

Il convient de séparer les données d'entraînement, de validation et de test en amont et de n'entraîner le modèle QUE sur les données d'entraînement afin de garder quelques données « nouvelles » pour l'algorithme, permettant de vérifier que le modèle généralise bien.

Ici nous pouvons séparer en termes de pourcentage d'images ou d'identités. Nous préférons séparer sur les identités pour être sûre que des personnes que l'algorithme a déjà « vues », ne se retrouvent pas dans la partie où nous mesurons s'il peut généraliser. En effet, cela donnerait l'impression que le système généralise mieux qu'il ne le fait vraiment.

Nous choisissons un découpage classique 75-5-20 (dans l'ordre *train-valid-test*), mais bien sûr il est possible de faire varier ces proportions et de mesurer leur influence sur le résultat. Intuitivement, plus nous mettons de données dans le *train set* et meilleure sera la précision de l'algorithme car il se sera entraîné sur plus de personnes différentes. Nous pouvons aussi imaginer qu'ainsi il verrait plus de visages différents et serait également meilleur en critère de *fairness*. Pour autant, plus nous fournissons le *train set*, et moins il devient facile de l'évaluer à travers le *validation* ou le *test set*. Il faut donc faire un compromis et ces proportions nous ont semblées pertinentes. Les détails de ce découpage sont donnés en annexe C.

Pour clarifier ce choix, il faudrait bien sûr tester les implications de ce découpage en termes de performance et *fairness*.

IV.3.3.3 Images négatives

Nous améliorons le système en commençant par un choix aléatoire des images négatives. Or si nous souhaitons que l'entraînement soit le plus efficace possible à reconnaître les identités, il ne faut pas choisir les images négatives au hasard. L'article initial introduit deux autres façons de choisir ces images négatives : le *hard mining* (*positive* ou *negative hard mining*) et le *semi-hard mining*, comme présentées en IV.1.5.

Lors d'un *hard mining*, nous prenons comme négative l'image du batch dont la distance avec l'ancre $\|f(A) - f(N)\|^2$ est la plus petite. Nous pouvons aussi *hard-mine* les images positives en prenant comme positive l'image qui a la distance avec l'ancre $\|f(A) - f(P)\|^2$ la plus grande. Pour les *semi-hard*, nous prenons une des images qui vérifient le critère $\|f(A) - f(P)\|^2 < \|f(A) - f(N)\|^2 < \alpha$.

Il est recommandé par l'article [36] de commencer l'entraînement par du *random* ou du *semi-hard mining* puis d'augmenter en difficulté par la suite.

Mais miner autrement qu'aléatoirement les images requiert un entraînement beaucoup plus long car la distance entre chaque A-P et A-N doit être mesurée à chaque batch.

Pour cette raison, nous avons beaucoup travaillé avec du *random mining*, avec l'idée de revenir plus tard sur ce *mining* et d'étudier son impact sur les performances et la *fairness*. Cette partie restera malheureusement en suspens, mais sa possibilité sera discutée en partie IV.4.3.1.

IV.3.3.4 *Dropout* et *Earlystopping*

Le *dropout* et l'*earlystopping* sont deux techniques pour faire baisser la courbe de validation et arrêter l'entraînement avant l'*overfitting*.

Le *dropout* consiste à ne pas entraîner certains poids à une étape de mise à jour donnée, ce qui permet une meilleure généralisation sur l'ensemble du réseau. Les activations des neurones sont « abandonnées » (*dropped out*), c'est-à-dire mises à zéro, avec une probabilité p. C'est comme si, pour une mise à jour donnée, seul un sous-réseau était entraîné.

Nous avons étudié l'impact d'une couche de *dropout* après les couches de convolution, avant la dernière couche *fully-connected*.

Nous pourrions regarder aussi l'impact de l'emplacement d'une telle couche dans le réseau, voire du nombre de couches. Ici, nous nous contenterons de cette simple étude car cet emplacement est classique des couches de *dropout*.

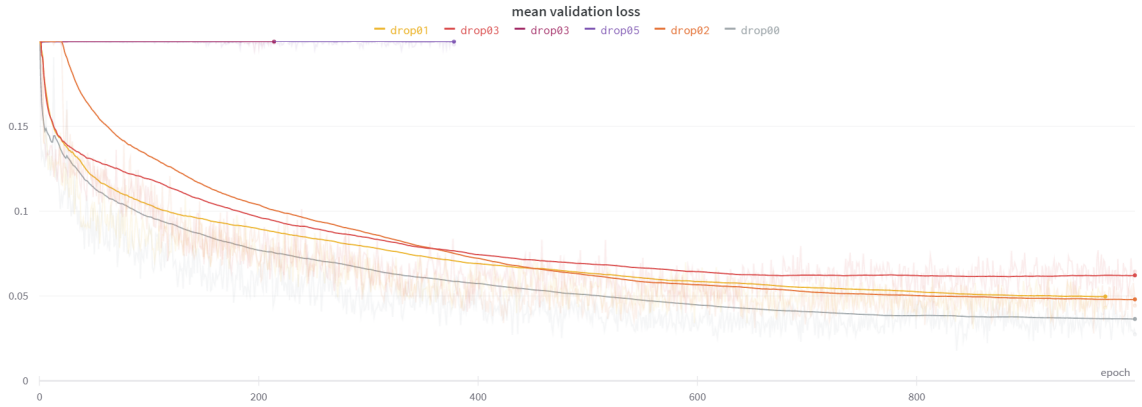


FIGURE 33 – Courbes de validation pour différentes valeurs de *dropout*

Nous voyons sur la figure 33 un résultat peu attendu : plus nous ajoutons du *dropout*, plus la courbe de validation peine à baisser. Pensant d'abord que cela signifiait que le modèle n'était pas assez profond, nous avons changé le paramètre base_channel pour l'élever à 64 et recommencer l'usage du *dropout*. Mais, même ainsi, le *dropout* ne fait que déteriorer l'entraînement. Nous nous sommes donc résolue à l'abandonner et à repasser par la même occasion à base_channel=32 qui paraissait donner de meilleurs résultats pour *dropout*=0.

L'*earlystoppping* quant à lui, arrête l'entraînement quand certains critères sont respectés. Classiquement, nous donnons comme condition d'arrêt que l'erreur de validation soit constante ou remonte depuis un certain nombre d'epochs (5 par exemple).

Ici nous ne l'avons pas implémenté, arrêtant l'entraînement manuellement lorsque la courbe de validation se stabilisait.

IV.3.4 Choix de test : déterminer le seuil et évaluer l'équité des résultats

Nous regarderons ici le modèle une fois entraîné et chercherons à déterminer le seuil de distance en dessous duquel nous pourrons classifier deux images comme semblables. Pour comprendre le processus menant à la création des histogrammes, voir la partie IV.1.6. Pour un exemple fictif sur l'influence du choix sur la *fairness*,

voir III.3.6. Ici nous expliquerons en détail comment nous avons procédé pour déterminer ce seuil et quelles auraient été les solutions alternatives, ainsi que leur influence sur la *fairness*. Pour le détail des calculs effectués, voir le notebook determine_threshold.

IV.3.4.1 Paires de test pour l'histogramme des distances

Pour tracer les histogrammes des distances, nous itérons sur le valid_loader qui charge les données de validation. Nous pourrions également réaliser cette itération sur le test_loader car les deux comportent des données inconnues au modèle. Néanmoins ici, nous préfèrerons utiliser les données de validation pour garder les données de test uniquement pour le test final et ainsi voir si le seuil déterminé sur les données de validation se généralise bien à de nouvelles données.

Une fois que nous savons que nous voulons utiliser les données de validation, reste à déterminer sur quelles paires tracer cet histogramme. Ici nous allons prendre comme paires « positives » (de la même personne) les images ancres et positives du valid_loader et comme images « négatives » (de personnes différentes) les images ancres et négatives du valid_loader. Bien sûr, comme d'habitude, ce choix peut avoir des conséquences sur les résultats – ici, sur la valeur du seuil et donc sur les critères de performance et de *fairness* qui en résultent.

Comme durant l'entraînement (voir partie IV.3.2.7), nous avons décidé de concaténer le valid_loader 10 fois, pour permettre d'avoir artificiellement plus de données. L'histogramme est ainsi plus rempli et plus facile à analyser. Toutefois, cela signifie que ce sont les 79 mêmes identités qui sont représentées 10 fois chacune. Cette augmentation artificielle ne saurait donc être représentative d'une généralisation des données.

Voici les histogrammes que nous obtenons :

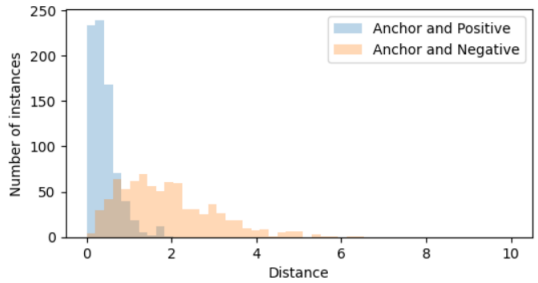


FIGURE 34 – Histogramme des distances pour des paires positives et négatives

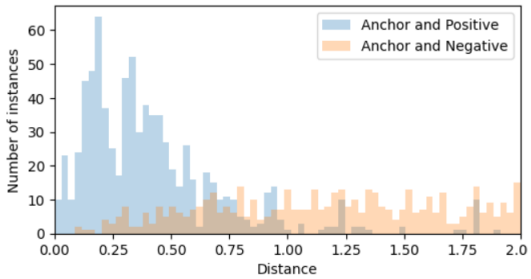


FIGURE 35 – Histogramme des distances pour des paires positives et négatives (zoomé)

Nous pouvons voir sur les figure 34 et 35 que les données se séparent assez bien, même si certaines paires positives et négatives se superposent. Nous devons alors nous rendre à l'évidence qu'aucune séparation ne sera parfaite, reste maintenant à déterminer un seuil tout en gardant cela en tête.

IV.3.4.2 Valeur seuil

Quelle valeur seuil choisir ? Cette question cruciale n'a pas une seule réponse mais une multitude de possibilités que nous essayerons d'examiner.

Utiliser la courbe ROC Nous possédons la distance entre chaque paire positive et négative, nous pouvons donc l'utiliser pour tracer la courbe ROC (*Receiver Operating Characteristic*), affichant le taux de faux positifs (*false positive rate* ou FPR) en fonction du taux de vrais positifs (*true positive rate* – ou TPR). Plus la courbe est éloignée de la diagonale et plus le classifieur (ici sur la distance) est bon. Au contraire, quand elle se rapproche de la diagonale, cela signifie que nous avons autant de faux positifs que de vrais positifs, notre classifieur n'est alors pas meilleur que le hasard. Cette courbe sera tracée grâce à la fonction roc_curve de la librairie scikit-learn (package sklearn.metrics).

Ici, la distance évoluant en sens inverse de la prédiction (plus la distance est faible, plus la paire a de chance de recevoir le label 1 – positif), nous prendrons soit l'inverse de la distance, soit son opposée. Ici l'opposée semble moins risquée pour éviter les problèmes en zéro. Nous avons donc comme y_true (véritables étiquettes binaires) des 1 ou des 0 pour les paires respectivement positives et négatives, et comme y_score (scores cibles) l'opposée des distances de chacune de ces paires.

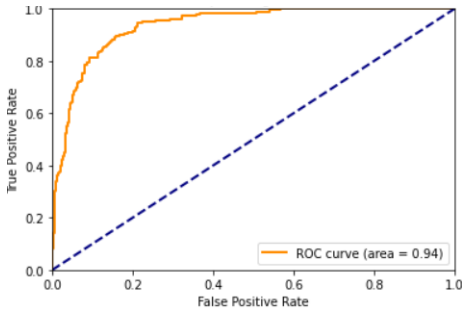


FIGURE 36 – Courbe ROC pour la distance

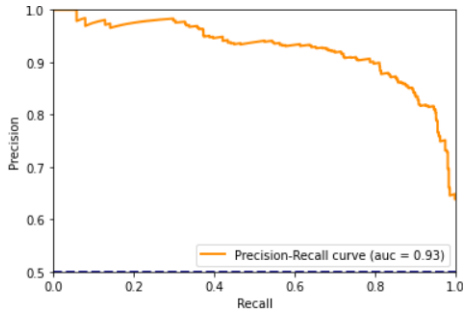


FIGURE 37 – Courbe *precision-recall* pour la distance

Nous voyons d'abord que l'aire sous la courbe figure 36 est de 0.94, ce qui est assez bon. Elle est, de plus, relativement symétrique.

A partir de cette courbe, nous pouvons déjà penser à plusieurs valeurs de seuil possibles, comme $max(TPR-FPR)$, $min(|TPR-(1-FPR)|)$, $min((1-TPR)^2+FPR^2)$. Les valeurs de ces seuils sont données en annexe D.

Mais la courbe ROC seule ne permet pas une analyse complète. Il convient également d'inclure les taux de faux négatifs (*false negative rate* – FNR) et de vrais négatifs (*true negative rate* – TNR). Pour cela, nous pouvons calculer le *recall* du modèle et tracer la courbe de *precision-recall* - ou précision-rappel (fonction precision_recall_curve de scikit-learn) - figure 37. Sur cette courbe, un « bon » classifier aura une courbe en haut à droite, et un mauvais proche de *precision*=0.5 (classification aléatoire).

Valeur de sigmoïde Notons que notre première idée pour tracer les courbes ROC et *precision-recall* était de placer une sigmoïde (figure 38) à une valeur de seuil déterminée (figure 39) pour évaluer la précision du modèle ainsi créé. Cette valeur sigmoïdale aurait évolué dans [0,1], créant une sorte de probabilité que la paire soit classée comme positive ou négative. Nous aurions alors pu évaluer ce modèle pour différentes valeurs de seuil et chercher laquelle donnait les meilleurs résultats en termes de précision.

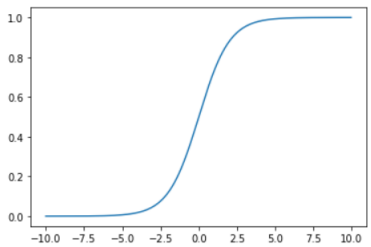


FIGURE 38 – Fonction sigmoïde

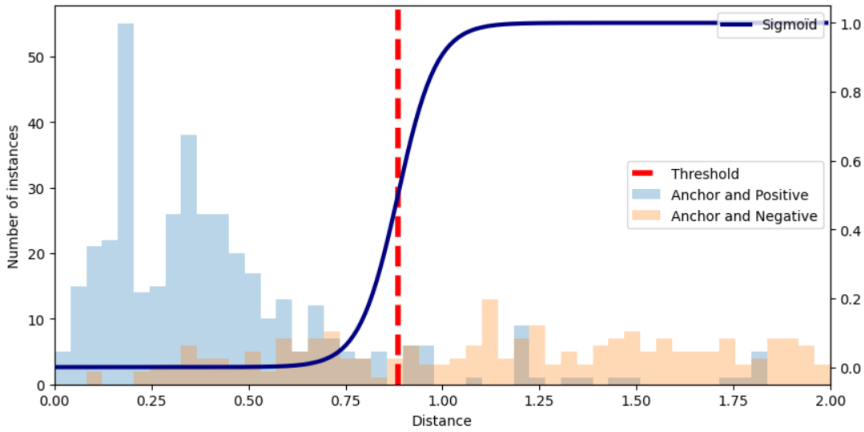


FIGURE 39 – Sigmoïde et histogramme des distances

Mais cette méthode aurait été bien trop gourmande en calcul, la précision devant être re-testée pour toutes les valeurs seuils envisagées. Nous avons préféré optimiser ce processus à l'aide de la régression logistique, reprenant ce même principe de sigmoïde.

Régression logistique La régression logistique est un classifieur, une version plus complexe de la régression linéaire, où l'on cherche des corrélations entre les variables telles que $y = \beta_0 + \sum_i \beta_i X_i$, avec y la variable expliquée – ici elle binaire et correspond à notre classification positive-négative –, et X nos variables explicatives qui se résument dans notre cas à une seule variable : la distance (ou plutôt son opposée). La fonction sigmoïde est alors utilisée pour transformer cette variable binaire y en une probabilité dans $[0, 1]$. Enfin, la régression logistique cherche le maximum de vraisemblance, c'est-à-dire qui trouve le point minimum de la différence entre ce que notre classifieur prédit et la réalité. Cette recherche de minimum s'effectue par descente de gradient.

Ici, nous utiliserons la fonction LogisticRegression de sklearn.linear_model. Notons que la régularisation L2 est appliquée par défaut au classifieur. Il conviendrait de jouer sur les paramètres de la régression logistique et d'évaluer tour à tour leurs impacts en termes de précision.

Á partir des prédictions de ce classifieurs, nous pouvons calculer le score (la précision [25] moyenne), ainsi que le f1-score [26] par exemple, qui donne une indication sur les performances du classifieur. En effet nous avons accès aux prédictions binaires (via la méthode predict) ainsi qu'aux probabilités de chacune des classes (via predict_proba).

Nous pouvons tracer les prédictions probabilistes pour les classes 0 et 1 et les superposer à notre histogramme (cf figure 40). Les courbes se croisent alors en un point correspondant à β_0/β_1 : c'est le seuil de la régression logistique que nous pouvons également utiliser comme valeur pour notre modèle.

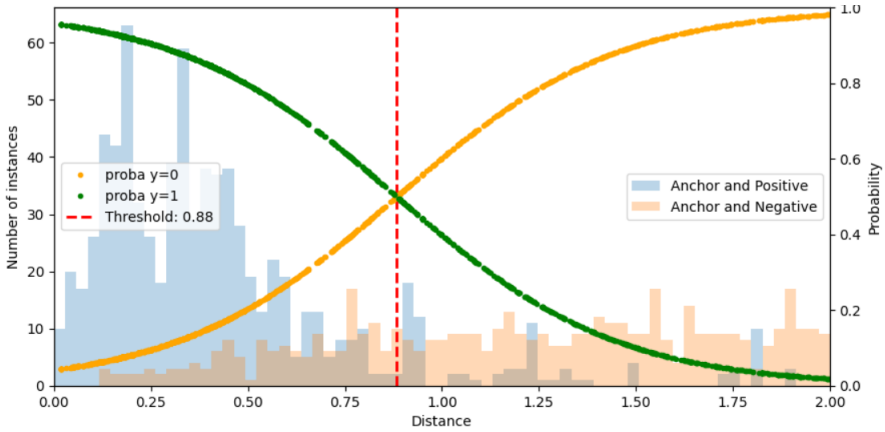


FIGURE 40 – Régression logistique et son seuil sur histogramme des distances

Une fois cette valeur seuil déterminée, nous pouvons afficher la matrice de confusion, donnant le nombre de vrais positifs (TP), faux positifs (FP), faux négatifs (FN) et vrais négatifs (TN). Nous pouvons également retracer les courbes ROC et *precision-recall* à partir des valeurs de probabilités fournies par le classifieur. Ces dernières devraient être sensiblement les mêmes que celles tracées à partir de la distance. Néanmoins, les valeurs de FPR et TPR étant légèrement différentes de celles obtenues à partir de la distance, nous pourrions, plutôt que de prendre β_0/β_1 en valeur seuil, prendre différentes valeurs comme précédemment.

Accuracy Pour calculer la précision du modèle, nous pouvons soit reprendre les prédictions de notre classifieur de régression logistique, soit réécrire une fonction qui, prenant en entrée une valeur seuil, calcule la distance entre deux images et renvoie 1 si cette valeur est inférieure au seuil, 0 sinon (fonction « authentification » de classification.py). En itérant sur tout le générateur de données (le valid_loader), nous obtenons un vecteur de 0 et de 1 correspondant aux prédictions du modèle avec ce seuil défini (fonction « predict » de classification.py).

Nous pouvons alors afficher la matrice de confusion du classifieur ainsi que sa précision moyenne ou son f1-score. Notons que si nous avons choisi de retenir la valeur β_0/β_1 comme valeur seuil, ces valeurs seront égales à celles affichées précédemment pour la régression logistique.

Si nous souhaitons maintenant trouver le seuil pour lequel la précision est maximale, nous pouvons tracer la courbe de précision en fonction du seuil (cf figure 41).

Malheureusement cette fonction n'est pas dérivable, il faudra donc utiliser une méthode alternative n'utilisant pas le gradient pour déterminer son maximum. Ici nous choisissons d'utiliser la méthode dite de « Brent », fonctionnant par dichotomie et accessible via la fonction optimize.brent de la librairie scipy.

Nous obtenons une valeur de seuil de 0.77 pour laquelle la précision est de 0.88 (cf annexe F).

25. Notons ici et dans la suite que la langue française utilise le terme de « précision » pour traduire le terme anglais « *accuracy* ». Attention à ne pas confondre le terme d'*accuracy* avec le terme « *precision* » qui lui est un faux ami. Pour éviter cette confusion, « *precision* » est régulièrement traduit par « valeur prédictive positive ». Nous redonnons ici leurs définitions mathématiques respectives :
$accuracy = \frac{TP+TN}{TP+TN+FP+FN}$; $precision = \frac{TP}{TP+FP}$
Dans la suite et sauf mention contraire, le terme de « précision » sera utilisé pour parler de l'*accuracy*.

26. $f1 - score = \frac{2*precision*recall}{precision+recall}$

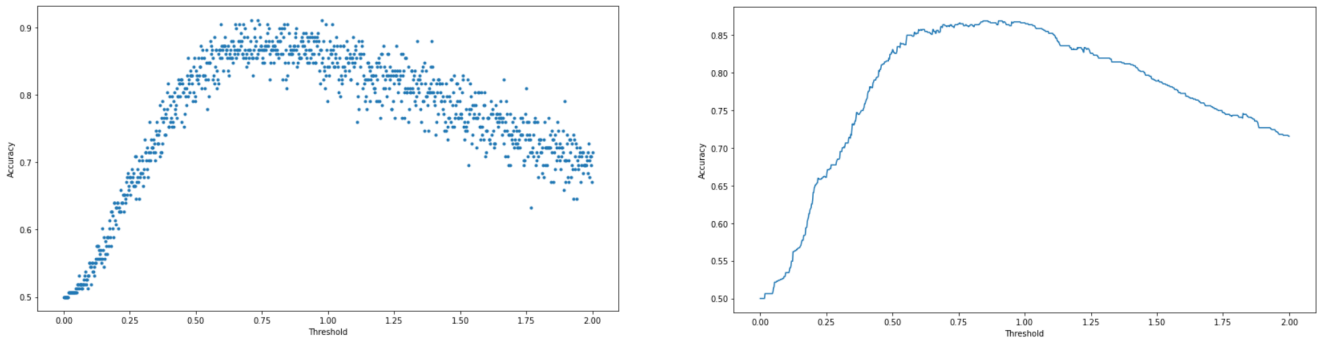


FIGURE 41 – Courbes de la précision en fonction du seuil en réinitialisant (gauche) ou non (droite) le valid_loader

Choix final de la valeur seuil De toutes les valeurs de seuils que nous avons mentionnées, nous devons maintenant en choisir une.

Notons que selon la figure 41, il y a une sorte de plafond d'*accuracy* entre, approximativement, 0.6 et 1. En prenant seulement un critère de précision, on pourrait alors argumenter que chaque choix entre ces valeurs serait à peu près équivalent. D'ailleurs, entre différents modèles que nous avons testés qui semblaient équivalents lorsque nous regardions les courbes d'entraînement, les histogrammes de distances, et même la courbe de précision, nous pouvions obtenir des résultats qui variaient dans cet intervalle.

Dans un cas réel, nous pourrions regarder les taux de faux positifs (FPR) et faux négatifs (FNR) et estimer quel choix nous paraît le plus adapté en fonction du contexte. En effet, selon les cas d'usage, nous allons préférer optimiser sur l'*accuracy*, sur le FPR, le FNR, ou faire des compromis entre tous (voir partie IV.4.3.2).

En l'absence de déploiement de notre système et donc de contexte permettant de déterminer quelle métrique optimiser, nous avons choisi de garder la valeur déterminée par régression logistique qui minimise à la fois le FPR et le FNR.

IV.4 Analyse post-traitement

IV.4.1 Critères de performance

IV.4.1.1 Métriques de précision

Puisque la détermination du seuil a été réalisée sur les données de validation, il convient de vérifier, maintenant que le système est entièrement fini, si nous obtenons des résultats similaires sur les données de test. Pour cela nous reprenons toutes les mesures de précision mentionnées depuis le début de IV.3.4 sur le set de test. Faute de place nous n'afficherons pas les résultats ici mais vous pourrez les retrouver dans le notebook various_tests.

Cette analyse inclut la précision (*accuracy*), les taux d'erreurs (FPR, FNR), le f1-score, etc.

Les résultats obtenus sont comparables à ceux obtenus sur le set de validation.

IV.4.1.2 Taux d'apprentissage (*triplet accuracy*)

La précision ne saurait représenter à elle seule toutes les données. Nous avons choisi de calculer des valeurs de précision alternatives comme celle que nous appellerons « *triplet accuracy* », ou « taux d'apprentissage », et qui est simplement le nombre de paires pour lesquelles $d(A, P) < d(A, N)$ sur le nombre de paires total.

Cette mesure de précision a l'avantage d'être indépendante du seuil.

Pour notre modèle actuel, nous trouvons une valeur entre 90 et 98% selon les initialisations du dataloader (résultats disponibles dans l'annexe D).

IV.4.1.3 Analyse des extrêmes et nouvelles données

Cas extrêmes Pour comprendre comment le modèle se comporte, nous pouvons regarder les cas extrêmes, c'est-à-dire ceux dont la distance est très faible ou très élevée. Des exemples d'images de ces cas extrêmes sont

donnés en annexe E. Pour plus d'images de cas extrêmes, voir le notebook various_tests. Ces cas extrêmes nous donnent une idée à la fois de la performance et de l'équité du système.

Alors que nous nous attendions à voir que le système marchait moins bien pour des femmes ou des personnes noires comme c'est régulièrement le cas des systèmes industriels [3], ici les paires de même identité qui ont une distance très élevée (faux négatifs) comportent beaucoup d'individus « hommes blancs ». Nous remarquons néanmoins que les enfants semblent être moins reconnus. Au contraire, dans les personnes correctement authentifiées, nous retrouvons une variété de profils. De même, dans les paires d'identités différentes qui ont une distance faible (faux positifs) ont retrouve quasi exclusivement des personnes blanches. Par contre, il semble que les personnes très jeunes ou plus âgées sont régulièrement mal classifiées par le système. Ces comportements, en l'absence d'une analyse quantitative et d'un système explicable, sont difficilement interprétables.

Enfin, concluons que ces *outliers* sont très difficiles à analyser et parfois nous ne comprenons pas pourquoi le système trouve une distance très élevée alors que les personnes semblent se ressembler à vue d'œil ou, inversement, réussit à authentifier une personne avec des poses très éloignées.

Images inédites Dans le même objectif d'analyser le comportement du système, nous avons pris plusieurs photos inédites (de l'autrice du rapport, ses encadrant.e.s, ami.e.s et famille... [27]). La prédiction du système sur ces nouvelles données permet d'analyser son comportement, même si nous aurions besoin de plus de données pour conduire une étude représentative. Néanmoins cela permet de tester le système pour des poses différentes (avec lunettes, chapeau, de face, de côté...). Quelques exemples peuvent être trouvés en annexe E et il est possible d'en tester d'autres dans le notebook various_tests.

Sur les quelques photos que nous lui avons présentées, le système arrive bien à reconnaître les identités lorsqu'elles sont de face sous une luminosité correcte. Les photos de profil semblent être classées comme semblables lorsqu'elles sont approximativement dans la même pose (voir les faux positifs de l'annexe E). Peut-être le système n'a-t-il pas assez appris sur des images de profil. Une solution pour contrer cela serait sans doute d'utiliser un détecteur de visage qui remette tous les visages dans la même position. Les lunettes semblent empêcher l'authentification, sauf lorsque sur les deux images la personne porte des lunettes. Cela peut représenter une faille de sécurité : si l'image de référence dans le système est une image avec lunettes, n'importe qui portant des lunettes pourrait s'authentifier... De plus, ici le système reconnaît mal lorsque la luminosité est élevée. Pour contrer cela, nous pourrions rendre le système plus robuste avec un peu plus de *color jitter* dans l'augmentation des données.

En conclusion sur ces quelques exemples, nous avons détecté quelques failles du système (comme les lunettes), ce qui serait passé inaperçu autrement. Une recommandation pourrait être de tester sur plusieurs critères (port de lunettes, chapeau, barbe...) ou d'utiliser une option de rejet de l'image : le système ne prend pas de décision sans un superviseur humain lorsque le résultat est incertain. Ces solutions comportent des avantages et inconvénients qu'il convient de peser lors de la conception du système.

Notons que vis-à-vis de la précision du système pour ces nouvelles images, les résultats sont bien moins bons que pour les images issues de LFW. Cela peut s'expliquer car nous avons fait exprès de prendre des poses difficiles à analyser pour le système (lunettes, chapeau, masque...). De plus, il serait nécessaire de disposer de bien plus de données pour réaliser une véritable étude statistique sur l'impact du masque, des lunettes et autres accessoires, ou de l'âge de la personne. Toutefois, le taux de faux positifs semble très élevé comparé à ce que nous pouvons obtenir sur LFW. Cela signifie peut-être qu'avec des images trop dures (lunettes, masque) le système a tendance à accepter l'authentification à tort trop souvent.

IV.4.1.4 Rapidité

Notre architecture telle qu'elle, sans *hard negative mining*, s'entraîne pendant environ 80min. Avant cela, il convient de télécharger le set de données et de les organiser, ce qui prend environ 5min. Déterminer le seuil prend également quelques minutes. En comptant tout cela, il faut environ 1h30 sur un ordinateur doté d'un GPU et de deux CPU - ou sur une plateforme en ligne comme Google Colab ou Kaggle Kernel, en ne partant de rien, pour avoir un modèle fonctionnel.

Ce temps, très court, s'explique par la simplicité de notre système, encore très loin d'un système comme celui de Google qui nécessite un entraînement pendant plusieurs jours sur plusieurs clusters de calcul, utili-

27. Toutes les photos qui ont été traitées par le système ou qui sont présentes dans ce rapport ont été utilisées avec le consentement desdites personnes.

sant plusieurs centaines de millions d'images, d'une dizaine de millions d'identités différentes, de résolution quasiment 4 fois supérieure et utilisant un détecteur de visages en amont [36].

Nos performances en termes de précision semblent bonnes pour un cas école entraîné pendant 1h30 (plus de 85% de précision en moyenne), mais largement insuffisantes s'il devait être déployé. Ceci est dû aux différences de moyens alloués à notre travail et à celui de l'équipe de Schroff.

IV.4.2 Critère d'équité

Jusqu'à maintenant, nous n'avons regardé que les choix que nous pourrions faire avec un critère de performance, tout en discutant si possible leur implication en termes d'équité.

Cette analyse n'est néanmoins pas complète et de nombreuses autres études seraient nécessaires pour comparer exhaustivement performance et équité (voir partie IV.4.3.1).

Toutefois, une première étape possible est d'analyser les résultats finaux de nos choix (cf IV.1.7) en termes d'équité : vérifier a minima quel est le comportement du système et s'il se comporte pour tous.tes de la même manière (cf IV.4.2.2).

IV.4.2.1 Données utilisées pour l'étude de l'équité

Pour mener l'étude de *fairness*, il convient de se demander sur quelles données tester. En effet, pour déterminer le seuil, nous avions regardé les données de validation puis vérifié que nos résultats se généralisaient sur les données de test. Ici, il est plus logique de prendre directement les données de test pour réaliser une étude a posteriori et ainsi rendre les données totalement indépendantes du seuil.

Nous avons choisi pour l'étude de concaténer plusieurs passes sur un générateur de données de test dans un tableau puis de faire nos statistiques directement sur ce tableau, nommé df_fairness. Chaque ligne de ce dernier correspond à une paire d'images (A et B) d'identités similaires ou différentes (annotées grâce à un attribut y_true qui vaut 1 si similaire, 0 sinon). Ces lignes sont construites à partir du générateur : la paire ancre-positive remplit une ligne et la paire ancre-négative en remplit une autre. Il aurait également été possible de sauvegarder la paire positive-négative mais nous aurions alors eu une surreprésentation des paires d'identités différentes. Nous récupérons alors la prédiction de notre système pour cette paire dans une colonne y_pred ainsi que certains attributs sensibles[28] des images A et B, notamment si A (respectivement B) est « Male », « White » ou les deux à la fois, qui sont ensuite transformés en colonnes de variables binaires.

Nous pouvons alors commencer par une analyse des données grâce à la fonction describe de pandas (cf annexe F).

Nous remarquons que les hommes blancs représentent environ 60% de nos données. Si nous affichons les percentiles 5, 25, 50, 75 et 95 pour les identités similaires et pour les identités différentes, nous remarquons qu'ils se superposent : si nous traçons l'histogramme, nous retombons bien sur une distribution similaire à celle de la figure 34. De même, nous pouvons retrouver l'*accuracy* moyenne d'environ 85%. Nous pouvons également calculer la *triplet accuracy*, les courbes ROC et *precision-recall* ainsi que la matrice de confusion avec les FPR et FNR. Toutes ces statistiques sont données dans le notebook ainsi qu'en annexe F. Notons également qu'un *bootstrapping*[29] peut être réalisé pour stabiliser les résultats.

Nous allons ensuite devoir vérifier que nos données sont bien annotées. Pour cela nous affichons plusieurs images selon une condition (par exemple les images annotées « Male »), et vérifions visuellement que la condition est vérifiée. Nous avons alors pu constater que dans les données annotées « Male », beaucoup d'individus semblent effectivement être des hommes, mais lorsque le paramètre Male vaut 0 nous retrouvons bien une majorité de femmes mais également une certaine quantité d'hommes pour qui le paramètre n'a pas été passé à 1. De même, pour « White », lorsque le paramètre vaut 1 nous retrouvons quasiment exclusivement des personnes blanches, mais lorsqu'il vaut 0 nous pouvons y trouver des personnes blanches ou non blanches. Nous devrons garder en tête ces erreurs d'annotations lors de notre analyse.

28. On parle couramment d'attributs sensibles pour décrire les attributs pouvant être source de discrimination. Ces critères sont définis par les conventions internationales et textes européens ou par le législateur français. Au sens plus large, le terme d'attribut sensible peut également désigner l'attribut selon lequel les sous-groupes d'étude sont séparés lors des études statistiques, sans être nécessairement un critère de discrimination selon la loi.

29. Technique consistant à re-échantilloner n fois sur les données et de réaliser la même opération pour obtenir n résultats et vérifier que ce dernier est fiable.

IV.4.2.2 Analyse des disparités

Nous pouvons commencer par regarder les distributions de différents sous-groupes. Pour commencer, nous regardons les cas où A et B sont étiquetées « White » et « Male » par rapport aux cas où au moins une ne l'est pas, ou les deux ne le sont pas.

Les histogrammes, bien que très ressemblants, semblent se chevaucher davantage pour le groupe des « White Male », ce qui est surprenant : nous nous attendions à ce que le système marche mieux pour cette catégorie sachant qu'elle est sur-représentée dans le set de données.

Si nous réalisons l'analyse statistique, les résultats montrent un léger biais : les « White Male » semblent plus difficilement distinguables que les autres, leur précision et f1-score sont plus bas, et leur FPR et FNR sont plus hauts que pour le groupe « autres » (cf annexe F). En réalisant une mesure de *bootstrapping* sur la précision et en affichant les quantiles, nous obtenons un intervalle où se trouvent 90% des données. Cet intervalle nous donne une indication de la proximité des échantillonnages. En effet, si nous reproduisons la procédure pour d'autres échantillons, dans 90% des cas, cet intervalle contiendra la vraie valeur moyenne. Ici, nous voyons alors que l'intervalle du groupe « White Male » et celui du groupe « Non White Male » ne se chevauchent pas : l'écart est significatif.

Cela est sans doute dû au fait que la distribution des « Non White Male » est plus hétérogène, menant à une distinction plus facile à réaliser.

Si nous regardons des groupes plus homogènes (« hommes » vs « non hommes » et « blancs » vs « non blancs ») les différentes mesures de précision semblent équivalentes. Il n'y a pas de disparité notable : en regardant les résultats du *bootstrapping*, nous voyons que les intervalles de confiance à 90% se chevauchent. L'écart entre les deux moyennes n'est donc pas significatif.

Pour conforter ces hypothèses, nous pourrions réaliser une étude bayesienne (qui se situe hors du camp actuel de notre étude).

Cela ne signifie pas que notre système est non biasé, mais prouve que malgré des données très déséquilibrées, les biais ne sont pas inévitables.

Néanmoins, si nous avons laissé apparaître ici notre cheminement, d'abord en considérant les groupes « White Male » vs « Non White Male » puis en revenant sur notre décision pour finalement considérer les groupes « Male » vs « Non Male » et « White » vs « Non White », c'est que nous voulions montrer la difficulté de sélection desdits groupes. En effet, choisir quels sous-groupes considérer lors d'une étude de *fairness* n'est pas anodin et les résultats peuvent changer du tout au tout. Plus nous prenons des groupes larges et plus les résultats seront moyennés. Au contraire, en comparant sur un pied d'égalité un groupe très homogène « hommes blancs » avec un groupe très hétérogène « hommes non blancs », nous pouvions faire une erreur d'interprétation. La baisse de performance est corrélée au fait que le groupe ne soit constitué que d'« hommes blancs », mais cela n'en est pas la cause. La cause vient d'un autre facteur : le fait que le groupe soit plus homogène et donc que les paires d'images à distinguer soient plus « dures ». En comparant des groupes dont l'homogénéité est équivalente, nous obtenons des résultats tout autres. Mais jusqu'où aller dans la définition de ces groupes ? En effet, il sera toujours possible de trouver des groupes plus petits et plus spécifiques, jusqu'à pousser le découpage pour obtenir des groupes ne contenant plus qu'un individu chacun. Le simple fait de diviser la population en groupes d'individus arbitraires est une décision très discutable. Cette habitude des ingénieur.e.s, vient du fait que des écarts de performance ont été remarqués sur des systèmes de reconnaissance automatique [3], il est donc devenu coutume de vérifier cet écart *a posteriori* du développement du système. Mais cette approche n'est pas parfaite. Plutôt que de chercher à corriger les biais d'un système avec plus de complexité algorithmique, peut-être faudrait-il penser à intégrer le critère d'équité dès leur conception.

IV.4.3 Pour aller plus loin

IV.4.3.1 Comment approfondir l'étude ?

Au delà de l'étude que nous avons réalisée, certains points techniques restent en suspens :

- Après avoir terminé le processus de conception, il convient de vérifier chacun des choix effectués à l'aide d'une *ablation study*[30] et de réévaluer le modèle ainsi tronqué en termes de performance et d'équité.

30. Une *ablation study* étudie les performances d'un algorithme en retirant certains composants, afin de comprendre la contribution du composant au système global.

Nous avons vérifié à la main quelques paramètres comme base_channels, la marge ou certaines *data augmentation*.

- Une étude comparative de divers modèles entraînés sur des données augmentées de différentes façons ou avec différentes techniques de *mining* permettrait d'évaluer leurs impact sur l'équité.
- Il aurait été intéressant de tenter d'augmenter la qualité des images et de quantifier son impact sur la performance et sur l'équité. Notre intuition est qu'augmenter la résolution rendra le modèle plus lent mais permettra peut-être une plus grande précision globale. Attention tout de même à ce que cela ne conduise pas plus vite à de l'*overfitting*, ou que cela n'augmente pas les écarts entre les groupes.
- Nous avons laissé le problème d'optimisation en plateau et d'échec d'apprentissage (cf IV.3.2.5) de côté, nous contentant de choisir un lr suffisamment petit pour que cela n'arrive que très peu. Il faudrait néanmoins creuser ce point si nous étions amenée à continuer ce projet, en surveillant notamment la norme du gradient.
- Comprendre pourquoi notre modèle entraîné sur une perte en cosinus ne réussi pas à apprendre pourrait être une amélioration notable puisque ces modèles sont réputés pour être plus avancés que ceux avec une simple norme euclidienne.
- Nous n'avons utilisé aucune régularisation. Puisque nous savons que cela peut détruire les identités dans certains sous-groupes (cf IV.3.2.2), comparer un modèle avec et sans en termes de performance mais surtout d'équité serait une étude pertinente.
- Nous pourrions également tester l'effet de la marge sur l'équité ou changer l'initialisation du réseau pour permettre la normalisation des données.
- Pour améliorer les résultats, nous pourrions entraîner sur plus de données. Pour cela, au lieu de séparer le set de données LFW, nous pourrions entraîner notre modèle sur des données non labelisées et garder LFW pour les tests d'équité et de performance.

D'autres points ne faisaient pas partie des objectifs de ce travail mais pourraient être intéressants à étudier lors d'une étude complémentaire :

- Les auteur.trice.s du papier original [36] ont mené une étude sur les performances selon l'architecture du réseau, comparant notamment une architecture type Zeiler&Fergus très profonde et gourmande en paramètres à une architecture GoogLeNet (modèle Inception), toutes deux adaptées pour le cas d'étude. Il pourrait être intéressant de reprendre exactement les mêmes architectures mais de mener l'étude en termes d'équité.
- En général, reprendre exactement les mêmes paramètres que dans l'article pour voir si nous pouvons nous rapprocher d'un résultat semblable puis étendre leur étude au critère d'équité serait le meilleur moyen de conclure ce travail. Cela aurait nécessité un cluster de calcul permettant d'entraîner des modèles plus lourds.
- Pour comparer notre modèle à d'autres, nous pourrions tester les performances et l'équité d'autres systèmes ne fonctionnant pas grâce à la perte par triplés, comme des réseaux neuronaux siamois, ou un modèle autre que convolutionnel comme un MLP classique [31]. Cela permettrait de généraliser notre cas d'étude pour sortir du cadre du simple réseau FaceNet [36].

Enfin, le modèle que nous avons réalisé n'est pas utilisable tel quel par l'industrie. Il faudrait pour cela rajouter un certain nombre d'éléments :

- Nous pourrions essayer de passer un détecteur de visages sur nos données pour que les images soient automatiquement recadrées ou utiliser plusieurs images de comparaison et moyenner la distance à la manière d'un FaceID (cf IV.1.1) pour que notre système soit plus fiable et proche de la réalité industrielle.
- Nous pourrions développer un système complet d'identification comme le cas d'un système permettant de rentrer dans un bâtiment, en comparant une image à une base de données de visages. L'analyse des résultats en termes d'équité pourrait alors être plus intéressante que la simple comparaison des taux d'erreurs binaires.

31. Perceptron multicouche. Pour comprendre le fonctionnement de ce réseau, voir l'annexe A

IV.4.3.2 Quelles questions se poser ?

Pour conclure, l'analyse statistique que nous avons menée, même si elle reste modeste, nous montre principalement que les chiffres sont interprétables de plusieurs façons. De façon provocatrice, nous pourrions nous arrêter à la simple comparaison entre les individus étiquetés « White Male » et les individus qui ne le sont pas. Nous pourrions alors affirmer que notre algorithme n'est pas biasé (il semblerait même qu'il fonctionne mieux pour les « Non White Male » !). Toutefois, la conclusion est toute autre lorsque l'on a connaissance de la façon dont sont étiquetées les données ou de l'hétérogénéité des personnes « Non White Male ». Si nous nous permettons d'insister sur la façon dont peuvent être présentés les chiffres, c'est qu'il n'est pas rare de voir des entreprises affirmer, pour raison publicitaire, que leur logiciel est sans biais, chiffres à l'appui. Mais en creusant, nous nous rendons vite compte que leurs études sont souvent sans fondements, présentant des résultats non justifiés et non reproduisibles (absence de protocole expérimental).

De même, les mesures statistiques ne doivent en aucun cas être sorties de leur contexte. Connaître le FPR et le FNR d'un système est un premier pas mais cela n'en dit pas long sur la réalité de l'expérience du système par des individus. Par exemple, dans le cas d'un système d'authentification pour accéder à un bâtiment, si une personne est rejetée de façon répétée, tous les matins en voulant accéder à son lieu de travail et devant ses collègues, si elle doit chaque fois se présenter à l'accueil du bâtiment pour prouver qu'elle y travaille bien, perdant du temps alors que ses collègues l'attendent pour rentrer, le poids de cette erreur régulière ne saurait être transmis à travers un pourcentage mathématique. Il faut bien sûr dans ce contexte réaliser l'importance de l'erreur sur les faux négatifs. Dans un autre contexte, si une personne est accusée à tort d'un crime qu'elle n'a pas commis car un logiciel de reconnaissance automatique a confondu son visage avec celui de quelqu'un d'autre [32], l'humiliation subie suite à cette accusation sera présente même si les développeur.se.s du système le jugent suffisamment fiable dans le cas général. Toutefois, cela n'exclut pas d'être vigilant au pourcentage de faux positifs.

Chaque cas étant unique, la conception du système doit inclure une réflexion sur son impact après déploiement, amenant le ou la développeur.se à faire un certain nombre de compromis.

Les choix que nous faisons pendant le développement influençant les taux d'erreur, il convient d'évaluer leur impact pour réduire par exemple davantage le FPR ou le FNR selon les cas d'usage, tout en gardant en tête que même si les pourcentages d'erreurs sont extrêmement faibles, ils n'en affectent pas moins les individus qui les subissent.

32. Ce cas n'est malheureusement pas fictif et nous invitons nos lecteurs à se renseigner notamment sur l'affaire Robert Williams [24].

Cinquième partie
Conclusion

V.1 Résultats du cas école

Le cas école développé n'avait pas pour but de créer un système de reconnaissance faciale parfait ou « non biasé », ceci étant très certainement impossible à obtenir. Notre vocation était plutôt d'éclairer la manière dont nous développons un tel système. L'apprentissage machine étant une discipline encore très récente, la recherche de solutions plus « performantes » ou plus « équitables » n'avance souvent qu'à tâtons, par essais erreurs, avant de choisir le modèle le plus performant. Ici, nous remettons ce fonctionnement en perspective. Notre thèse, maturée lors de ce travail, est de dire que chaque choix que fait le constructeur d'un algorithme ou d'un système automatique lors de sa phase de développement, est porteur de valeurs et de principes. Dans ce rapport, nous avons analysé un certain nombre de ces choix. Néanmoins, si nous devions revenir sur notre méthode, il y aurait certainement des moyens de la rendre plus rigoureuse à l'aide d'une recherche automatique de meilleurs paramètres, d'une *ablation study* pour vérifier que chacun de ces choix était pertinent, etc. (cf. IV.4.3.1)

Nous tenions à souligner dans ce travail l'incroyable dépendance de tous ces choix, chacun n'influençant pas simplement le résultat final mais également les autres paramètres. Nous avons tenté de résumer figure 42, la dépendance des quelques paramètres que nous avons étudiés ainsi que leur influence sur les critères de performance et d'équité.

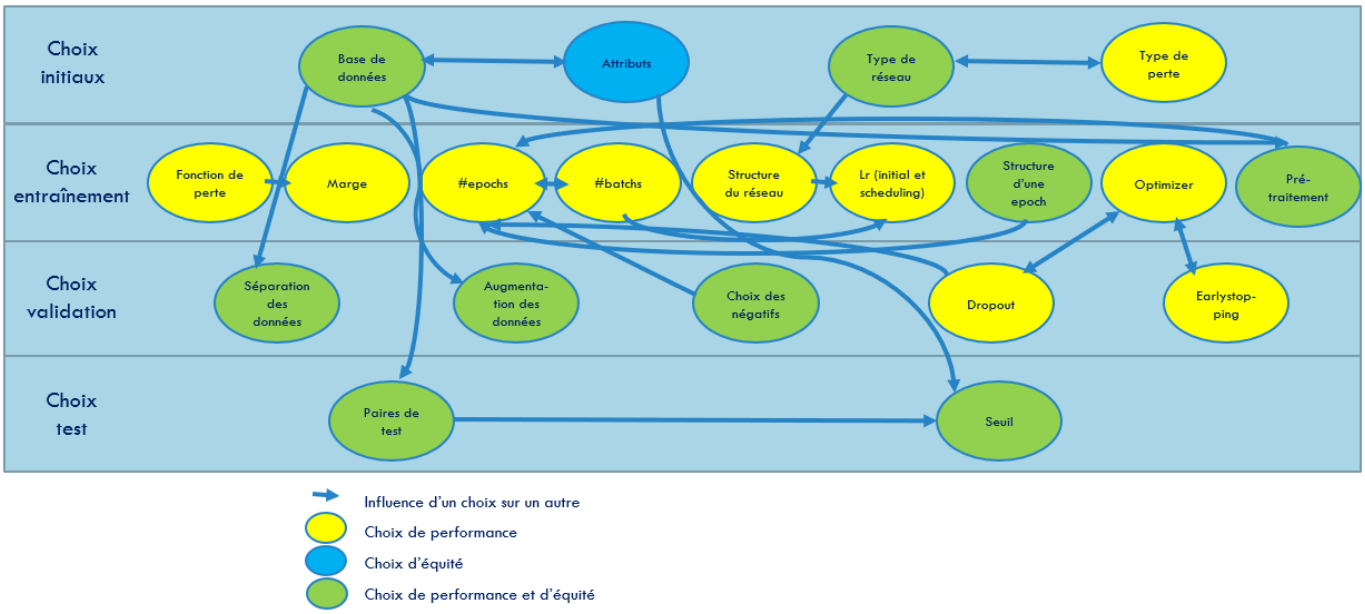


FIGURE 42 – Dépendances des paramètres entre eux et avec les valeurs

Notre propos est d'affirmer que ces choix ne sont qu'une goutte d'eau dans toutes les possibilités de développement d'un système d'authentification. Chacun d'entre eux peut être regardé avec une certaine perspective, une certaine hiérarchie de valeurs. Nous avons ici simplement gratté la surface d'une comparaison performance-équité mais les valeurs qui pourraient être étudiées sont bien plus nombreuses (cf III.2.2).

Comme nous l'avons mentionné, l'étude pourrait être continuée davantage (cf IV.4.3.1) et la route pour aboutir à des exigences de conception est encore longue. Nous espérons néanmoins avoir permis d'avancer la réflexion autour de nouvelles approches de conception.

Notons toutefois que l'analyse des résultats post-traitement du cas école n'a pas permis de mettre en évidence des biais évidents. Il n'est donc pas systématique d'obtenir un système biaisé lorsque l'on utilise des données déséquilibrées. De plus, il est commun d'entendre parler de compromis à faire entre performance et équité, comme si dès que nous essayons d'améliorer un paramètre, nous dégradons nécessairement l'autre. Nous avons nous-même, dans ce rapport, parlé de « compromis performance-équité », comme si les deux notions étaient irréconciliables. Bien que cela soit parfois le cas, il convient de ne pas aller trop vite dans

nos propos. Rien ne prouve que les deux critères soient exclusifs. Parfois il est possible de concilier les deux. L'équité est alors pensé comme un critère à considérer en plus lors du développement du système, venant s'ajouter à la performance sans nécessairement la dégrader.

Enfin, nous tenions à rappeler que les questions d'« éthique » autour d'un système ne sauraient se limiter à sa conception. Les points que nous avons abordés dans ce rapport doivent l'être mais seulement après une discussion préalable sur l'usage de la technologie. Les systèmes dits d'IA ne sont pas inéluctables, leur utilisation doit restée facultative et seulement si elle est appropriée. Il est nécessaire, avant de déployer tout système, d'avoir un questionnement relatif à sa raison d'être. Finalement, nous avons besoin, en tant que société, d'un débat sur ce qui est souhaitable. Particulièrement concernant la reconnaissance automatique, il faut nous poser la question de l'acceptabilité d'une société basée sur la surveillance, les conséquences sur notre comportement collectif, l'impact sur les individus, ainsi que les bénéfices et risques qui en découlent.

V.2 Bilan du stage au Comité

Le stage au Comité m'a permis d'approcher de plus près le domaine de « l'éthique ». Simplement assister aux débats entre les membres du CCNE et celles et ceux du CNPEN sur leurs sujets respectifs a avancé ma compréhension de leur démarche et raisonnement. Si je devais les simplifier en une phrase, je reprendrais les paroles d'un.e membre qui disait « nous faisons de l'humain avant tout ». Ce point de vue est crucial lorsque l'on s'intéresse aux sujets numériques, pour ne pas les réduire à des questions techniques ou de régulations.

En outre, le meilleur souvenir que j'en garderai sera d'avoir pu discuter avec des personnes passionnées et passionnantes sur des sujets essentiels à notre société. J'y ai bien sûr appris un nombre incalculable de choses, sur l'« IA » et l'apprentissage machine, et particulièrement la reconnaissance faciale. Mais j'y ai appris également une pensée, une façon de raisonner, sur comment nous questionner, en tant que société ou individu, sur nos valeurs et nos principes moraux.

V.3 Perspectives

Je souhaite poursuivre le travail initié avec ce stage en thèse, car il me semble que l'éthique de l'apprentissage machine a encore bien du chemin à parcourir et je souhaite contribuer à son épanouissement. Je continuerai de travailler sur la reconnaissance faciale et comportementale, sous un prisme plus réglementaire. Comment aboutir, par un échange et une discussion entre plusieurs disciplines, notamment l'informatique et le droit, à des critères de régulation des technologies de reconnaissance automatique ?

Ce travail sera encadré par Florence d'Alché-Buc, Winston Maxwell et Stéphan Clemençon des laboratoires i3 et LTCI de l'Institut Polytechnique de Paris (Télécom Paris), et se déroulera dans l'équipe *Operational AI Ethics*.

Merci encore à Catherine et Claude pour m'avoir aidée à obtenir cette thèse, pour le temps passé à rédiger des lettres de motivation, dénicher des contacts ou simplement discuter de mon parcours professionnel.

Annexe A Réseau de neurones artificiels et convolutionnels

Nous rappelons ici brièvement le fonctionnement d'un réseau de neurones artificiels, qui est une structure s'inspirant du cerveau humain.

Les réseaux neuronaux reçoivent une entrée (*input*) sous forme de vecteur, et la transforment à travers une série de couches cachées (*hidden layers*). Chaque couche cachée est constituée d'un ensemble de neurones, où chaque neurone est entièrement connecté à tous les neurones de la couche précédente, et où les neurones d'une même couche fonctionnent de manière totalement indépendante et ne partagent aucune connexion. La dernière couche entièrement connectée est appelée « couche de sortie », dans le cas de la classification c'est un vecteur de scores de classe, dans le cas de la reconnaissance faciale, c'est le gabarit de la photo du visage [23].

Chaque neurone possède un vecteur de poids w, un biais b et une fonction d'activation σ. Il réalise l'opération $z = \sigma(w^T x + b)$, avec x et z respectivement l'entrée et la sortie du neurone.

Il est possible de démontrer qu'en enchaînant un très grand nombre de ces opérations élémentaires, avec des poids bien choisis, nous pouvons reproduire n'importe quelle fonction mathématique.

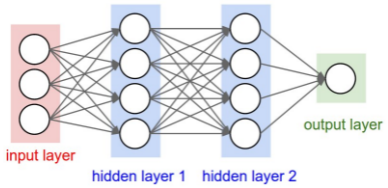


FIGURE 43 – Réseau de neurones - adapté de [23]

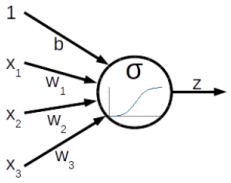


FIGURE 44 – Neurone artificiel - adapté de [?]

Néanmoins, les réseaux neuronaux ordinaires ne s'adaptent pas bien aux images complètes. Si les images sont de taille 200x200x3 (200 pixels de large, 200 pixels de haut, 3 canaux de couleur), alors un seul neurone entièrement connecté dans une première couche cachée d'un réseau neuronal ordinaire aurait 200*200*3 = 120 000 poids. Or nous aurions besoin de plusieurs neurones de ce type, de sorte que les paramètres s'additionneraient rapidement. Il est clair que cette connectivité complète est un gaspillage et que le nombre énorme de paramètres conduirait rapidement à un surajustement (couramment appelé *overfitting*) [23].

Les réseaux neuronaux convolutifs (ConvNet ou CNNs pour *convolutional neural network*) sont des volumes de neurones, répartis en trois dimensions : largeur, hauteur et profondeur. Les neurones d'une couche ne sont connectés qu'à une petite région de la couche qui la précède, au lieu que tous les neurones soient entièrement connectés. La couche de sortie finale a généralement des dimensions de (1,1,n) : l'image complète est réduite en un seul vecteur disposés le long de la dimension de profondeur [23], c'est le gabarit.

L'architecture d'un ConvNet est constitué d'alternance de couches convolutionnelles (*convolutional layers*), de *pooling layers* et de couches entièrement connectées (*fully-connected*). Nous décrivons ici la structure d'un ConvNet classique, comme CIFRAR-10 par exemple :

- L'entrée contiendra les valeurs brutes des pixels de l'image. Elle est de dimensions (w,h,3) dans ce cas une image de largeur w, hauteur h, et avec trois canaux de couleur R,G,B.
- La couche convolutionnelle calcule la sortie des neurones qui sont connectés à des régions locales dans l'entrée, chacun calculant un produit scalaire entre leurs poids et une petite région à laquelle ils sont connectés dans le volume d'entrée. Cela donne un volume (w,h,k) si nous avons décidé d'utiliser k filtres.
- Nous pouvons intercaler des fonctions d'activation entre les couche comme la fonction RELU : $x \to max(0, x)$. Cela laisse la taille du volume inchangée.
- La couche de *pooling* effectue une opération de sous-échantillonnage le long des dimensions spatiales (largeur, hauteur). Avec les bon paramètres, nous pouvons ainsi diviser les dimensions par deux et obtenir un volume (w/2,h/2,k).
- La couche entièrement connectée calcule les scores de classe, ce qui donne un volume de taille (1,1,n). Comme pour les réseaux neuronaux ordinaires chaque neurone de cette couche sera connecté à tous les chiffres du volume précédent [23].

La structure de ce réseau est un choix de conception fort car il impacte énormément les résultats. Plus ce réseau possède de paramètres (poids et biais des neurones), plus on dit qu'il est profond, d'où le terme

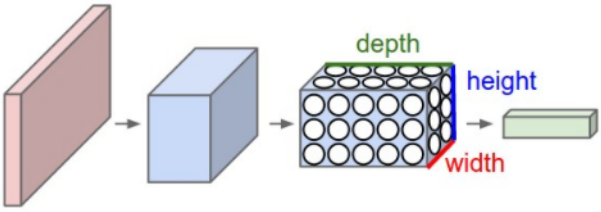


FIGURE 45 – Réseau de neurones convolutionnel - adapté de [23]

anglais de *deep learning*. De façon instinctive : si un réseau ne possède pas assez de paramètres, converger vers un résultat trop complexe sera impossible. Si au contraire, le réseau de neurones est trop profond, au mieux l'apprentissage sera très long, au pire le résultat sera inatteignable. En général, nous chercherons donc à avoir le réseau le plus simple pour lequel nous arrivons à converger vers des résultats « convenables » (ce terme reste bien sûr à définir selon les exigences et valeurs de chacun). L'organisation des différentes couches et le choix de leur taille et agencement a également un impact sur les résultats. La recherche sur les différentes architectures de réseaux de neurones est un domaine vaste dans lequel beaucoup reste à explorer. Ici nous ne nous attarderons que peu dessus mais essayerons de faire varier quelques paramètres pour mesurer leur impact (voir partie IV.3).

Annexe B Apprentissage d'un réseau de neurones

Ce qu'on appelle communément « apprentissage » d'un réseau de neurones est en réalité un ajustement de ses paramètres (poids et biais) pour améliorer son score final. Ce score est calculé par une fonction de perte (*loss function*) : L.

Avant de commencer l'entraînement, nous séparons les données en 3 : les données d'apprentissage (*training set*), les données de validation (*validation set*), et les données de test (*test set*). Nous n'entraînerons le réseau QUE sur les données d'apprentissage. Les données de validation serviront, quant à elles, à vérifier que le réseau généralise bien durant l'apprentissage et permettrons notamment de sélectionner le meilleur modèle : celui qui généralise mieux. Les données de test servirons à vérifier que le modèle final fonctionne comme attendu.

Nous passons donc nos entrées d'entraînement (qui sont ici des images de visages) dans le réseau, ce qui nous renvoie un vecteur de sortie (ici nos gabarits) et nous calculons la perte grâce à la fonction L, vue comme une sorte d'éloignement au résultat souhaité : c'est le *feed-forward*. Nous réalisons ensuite la *back propagation* : nous mettons à jour les poids du réseau à l'aide du gradient de la perte selon la formule $w \leftarrow w - \eta \nabla L(\Theta)$. Où η est le *learning rate* dont nous discuterons l'influence en IV.3.2.5. Ce processus est appelé « descente de gradient stochastique ».

Sur combien de données apprendre et pendant combien de temps, quel *learning rate* choisir ou encore quelle fonction de perte utiliser ? L'apprentissage d'un réseau de neurones est une phase délicate que beaucoup de paramètres peuvent influencer. Il convient de faire ces choix consciencieusement et de prêter attention à leurs conséquences.

Annexe C Résultats des notebooks : *Train*

Création du dataframe

```
Number of individuals:  5749
Number of total images:  13233
images weigh  0.57 GB
DataFrame creation: 2.4 min

DataFrame extention: 0.2 min
```

Exploration du dataframe

	Classid	Name	Path	Male	Asian	White	Black	Baby	Child	Youth	...	Pale Skin	5 o Clock Shadow	Strong Nose-Mouth Lines	Wearing Lipstick	Flushed Face
0	0	AJ_Cook	AJ_Cook/AJ_Cook_0001.jpg	0.0	0.0	1.0	0.0	0.0	0.0	1.0	...	1.0	0.0	1.0	1.0	0.0
1	1	AJ_Lamas	AJ_Lamas/AJ_Lamas_0001.jpg	1.0	0.0	1.0	0.0	0.0	0.0	1.0	...	0.0	0.0	1.0	0.0	1.0
2	2	Aaron_Eckhart	Aaron_Eckhart/Aaron_Eckhart_0001.jpg	1.0	0.0	1.0	0.0	0.0	0.0	0.0	...	1.0	1.0	0.0	0.0	0.0
3	3	Aaron_Guiel	Aaron_Guiel/Aaron_Guiel_0001.jpg	1.0	0.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0
4	4	Aaron_Patterson	Aaron_Patterson/Aaron_Patterson_0001.jpg	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	1.0	1.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
13228	5744	Zorica_Radovic	Zorica_Radovic/Zorica_Radovic_0001.jpg	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	1.0	0.0
13229	5745	Zulfiqar_Ahmed	Zulfiqar_Ahmed/Zulfiqar_Ahmed_0001.jpg	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	1.0	0.0	0.0	0.0
13230	5746	Zumrati_Juma	Zumrati_Juma/Zumrati_Juma_0001.jpg	0.0	0.0	1.0	1.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0
13231	5747	Zurab_Tsereteli	Zurab_Tsereteli/Zurab_Tsereteli_0001.jpg	1.0	0.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0
13232	5748	Zydrunas_Ilgauskas	Zydrunas_Ilgauskas/Zydrunas_Ilgauskas_0001.jpg	1.0	0.0	1.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0

13233 rows × 76 columns

	Classid	Male	Asian	White	Black	Baby	Child	Youth	Middle Aged	Senior	...	Pale Skin	5 o C Sha
count	13233.000000	13143.000000	13143.000000	13143.000000	13143.000000	13143.000000	13143.000000	13143.000000	13143.000000	13143.000000	...	13143.000000	13143.00
mean	2828.386005	0.774633	0.076771	0.747927	0.042608	0.004489	0.049304	0.163357	0.101803	0.213726	...	0.507038	0.40
std	1642.531642	0.417840	0.266238	0.434220	0.201980	0.066853	0.216510	0.369705	0.302401	0.409951	...	0.499969	0.49
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00
25%	1507.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00
50%	2688.000000	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	1.000000	0.00
75%	4243.000000	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	1.000000	1.00
max	5748.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...	1.000000	1.00

8 rows × 74 columns

Séparation des données d'entraînement

```
Train set from indiv 0 to 4310
Valid set from indiv 4311 to 4598
Test set from indiv 4599 to 5748

Number of training images:  10060
Number of validation images:  586
Number of testing images:  2587
Number of total images:  13233
```

```
Number of individuals in the training set:  4311
Number of individuals in the validation set:  288
Number of individuals in the testing set:  1150

Number of individuals with more than one image in the training set:  1267
Number of individuals with more than one image in the validation set:  79
Number of individuals with more than one image in the testing set:  334
```

Annexe D Résultats des notebooks : *Determine Threshold*

Seuils avec la distance

À partir de la courbe ROC

```
max(TPR-FPR): 0.7772988677024841
min(abs(TPR-(1-FPR))): 0.822944164276123
min((1-TPR)²+FPR²): 0.7772988677024841
```

Seuils avec la régression logistique

Précision de la régression

```
Score (mean accuracy): 0.8645569620253165
f1 score: 0.8682266009852218
```

```
THRESHOLD with logistic regression: 0.9316837146356096
```

```
Confusion Matrix Total
array([[661, 129],
       [ 85, 705]])
```

```
Accuracy score: 0.8645569620253165
```

À partir de la courbe ROC

```
max(TPR-FPR): 0.64612483733356
min(abs(TPR-(1-FPR))): 0.6044513400530664
min((1-TPR)²+FPR²): 0.64612483733356
```

À partir de la mesure de précision (*Accuracy*)

```
Maximum of accuracy (0.882) found for threshold = 0.7783457183275462
```

```
Accuracy: [0.8879746835443038, 0.8734177215189873, 0.8683544303797468]
Threshold:  [0.8184843212500865, 0.8262620346831784, 0.5333776315685422]
Mean accuracy: 0.8765822784810127
Mean threshold: 0.726041329167269
```

Autres mesures

Taux d'apprentissage (*triplet accuracy*)

```
Triplets where d(A,P) < d(A,N):  94.1 %
```

```
Mean triplet accuracy:  0.9408607594936709
Min-Max triplet accuracy:  [0.920253164556962, 0.9582278481012658]
```

Annexe E Résultats des notebooks : *Various tests*

Affichage des extrêmes [33]

Identités semblables

SAME IDENTITY WITH HIGH DISTANCES
MORE LIKELY TO BE FALSE NEGATIVES
Distance: 2.185533285140991

Distance: 2.0913643836975098

Distance: 1.973333477973938

Distance: 1.914870023727417

Distance: 1.914870023727417

SAME IDENTITY WITH LOW DISTANCES
MORE LIKELY TO BE TRUE POSITIVES
Distance: 0.03050186298787594

Distance: 0.032420992851257324

Distance: 0.032420992851257324

Distance: 0.03724568337202072

Distance: 0.03724568337202072

33. Ces images ne représentent qu'un échantillon des données. Les conclusions que nous avons pu en tirer IV.4.1.3 sont le fruit d'une étude réalisée sur un plus large échantillon. Néanmoins, aucune analyse quantitative de ces cas n'a été faite et le résultat est donc sujet à interprétation. Cette remarque est également valable pour les images inédites.

Identités différentes

DIFFERENT IDENTITIES WITH LOW DISTANCES
MORE LIKELY TO BE FALSE POSITIVES
Distance: 0.1515355110168457

Distance: 0.15384261310100555

Distance: 0.15991473197937012

Distance: 0.1844446212053299

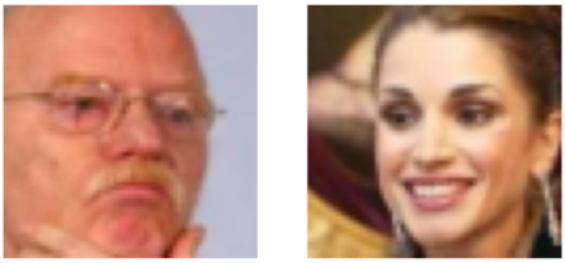
Distance: 0.19451779127120972

DIFFERENT IDENTITIES WITH HIGH DISTANCES
MORE LIKELY TO BE TRUE NEGATIVES
Distance: 6.3498215675354

Distance: 6.211137771606445

Distance: 5.838595390319824

Distance: 5.6426005363464355



Distance: 5.588470458984375

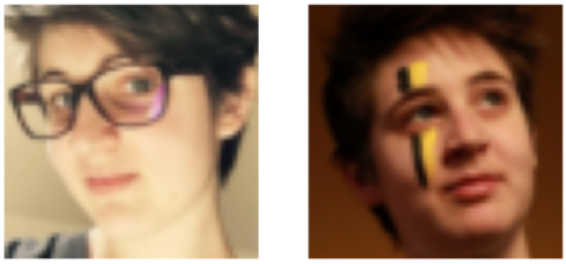
Images inédites

True Positives et *True Negatives*

Distance: 0.44
Prediction: Same person

Distance: 0.77
Prediction: Same person

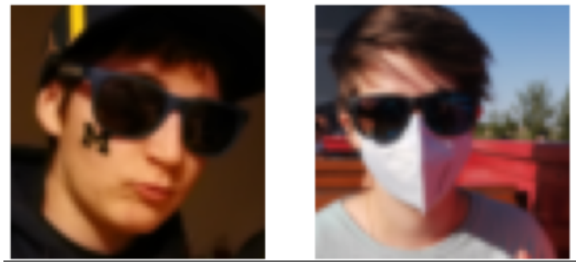
Distance: 0.45
Prediction: Same person



Distance: 0.51
Prediction: Same person

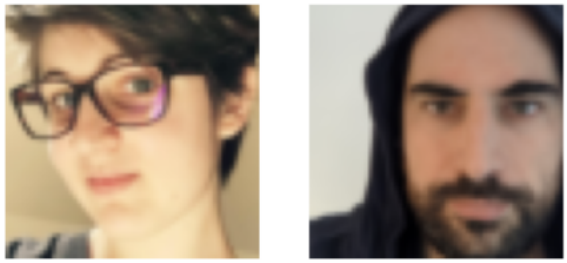
Distance: 0.33
Prediction: Same person

Distance: 0.48
Prediction: Same person



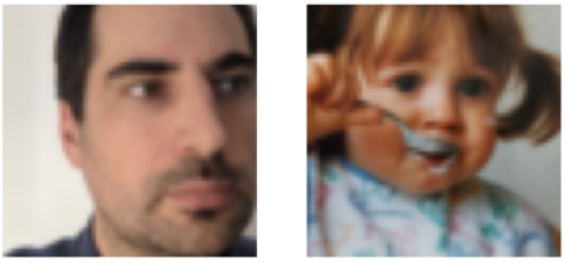
Distance: 1.77
Prediction: Different people

Distance: 0.95
Prediction: Different people

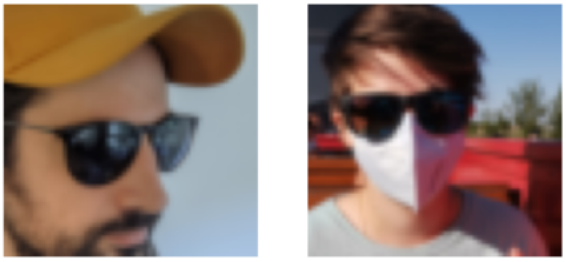


Distance: 0.98
Prediction: Different people

Distance: 1.74
Prediction: Different people



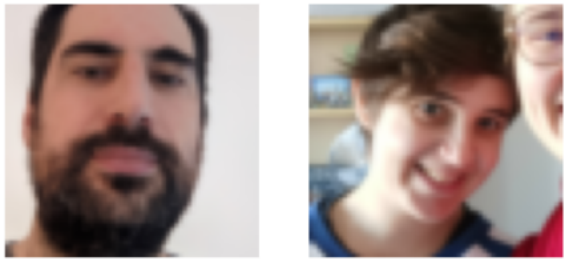
Distance: 1.7
Prediction: Different people



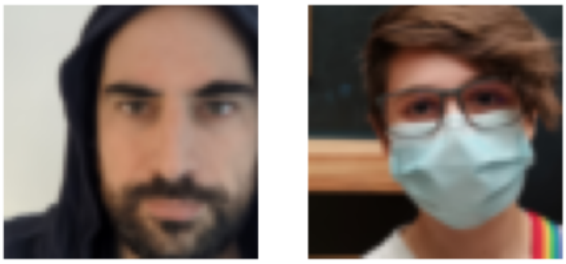
Distance: 1.05
Prediction: Different people

False Positives et *False Negatives*

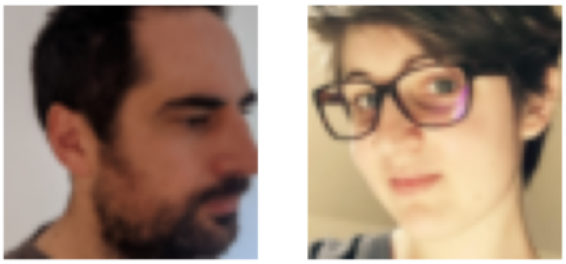
Distance: 0.6
Prediction: Same person



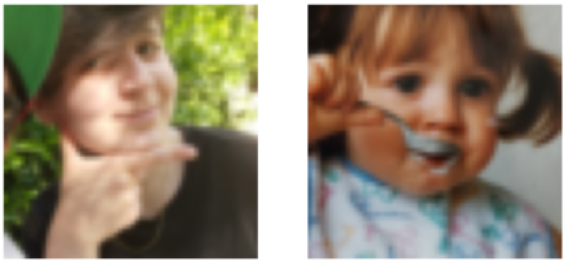
Distance: 0.79
Prediction: Same person



Distance: 0.92
Prediction: Same person

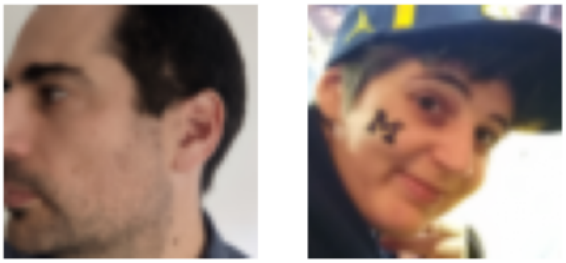


Distance: 0.12
Prediction: Same person



Distance: 0.71
Prediction: Same person

Distance: 0.19
Prediction: Same person

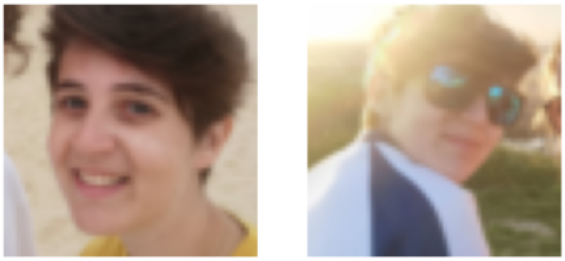


Distance: 1.61
Prediction: Different people

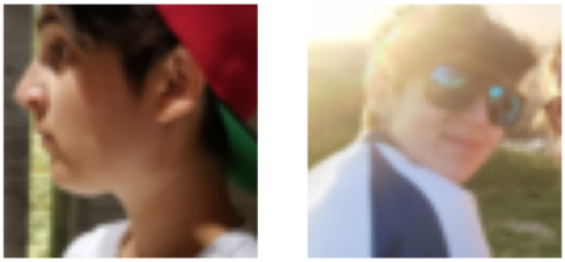
Distance: 2.04
Prediction: Different people

Distance: 1.31
Prediction: Different people

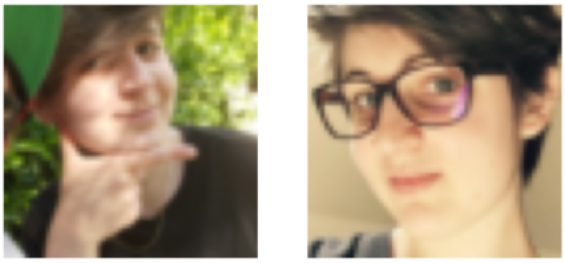
Distance: 2.53
Prediction: Different people



Distance: 1.93
Prediction: Different people



Distance: 1.25
Prediction: Different people



Annexe F Résultats des notebooks : *Fairness Study*

Création et exploration du dataframe pour l'étude de l'équité

```
DataFrame creation: 1.5 min
```

	id_A	id_B	y_true	y_pred	Distance	A_Male	B_Male	A_White	B_White	A_WhiteMale	B_WhiteMale	AB_WhiteMale	AB_NoWhiteMale	correct_predict
0	10914	10913	1	1	0.294181	1.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0
1	10708	10710	1	1	0.462511	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0
2	11096	11099	1	1	0.400648	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0
3	12981	12985	1	1	0.360847	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0
4	11013	10985	1	1	0.089605	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10235	11186	12321	0	0	2.180452	1.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0
10236	11305	10947	0	0	2.350889	0.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0
10237	12235	11016	0	0	6.006941	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0
10238	12144	13221	0	1	0.601629	1.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0
10239	11240	11890	0	0	1.994690	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0

10240 rows × 14 columns

	id_A	id_B	y_true	y_pred	Distance	A_Male	B_Male	A_White	B_White	A_WhiteMale	B_WhiteMale	AB_WhiteMale	AB_NoWhiteMale	correct_predict
count	10240.000000	10240.000000	10240.000000	10240.000000	10240.000000	10152.000000	10169.000000	10152.000000	10169.000000	10152.000000	10169.000000	10081.000000	10240.000000	10240.000000
mean	11843.282617	11894.703418	0.500000	0.546875	1.195921	0.787037	0.797424	0.737392	0.715311	0.597715	0.584423	0.425454	0.238672	0.863086
std	776.779564	762.998941	0.500024	0.497822	1.093090	0.409422	0.401939	0.440073	0.451288	0.490383	0.492845	0.494436	0.426293	0.343773
min	10648.000000	10647.000000	0.000000	0.000000	0.010067	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	11174.000000	11237.000000	0.000000	0.000000	0.350236	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
50%	11790.000000	11857.000000	0.500000	1.000000	0.822480	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	1.000000
75%	12579.750000	12586.000000	1.000000	1.000000	1.782595	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000
max	13227.000000	13232.000000	1.000000	1.000000	9.055379	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

```
Same identity - mean distance:  0.4567046639751425
Same identity - std distance:  0.3509625274314767
Same identity - percentiles:  [0.09292177 0.19956444 0.36061332 0.61639988 1.1492126 ]


Different identities - mean distance:  1.9351364014146384
Different identities - std distance:  1.083354777184975
Different identities - percentiles:  [0.50635051 1.12297061 1.76408696 2.55505311 4.00851719]


Mean accuracy:  0.8630859375
Bootstrapping mean accuracy:  [0.8578125  0.86113281 0.86347656 0.86572266 0.86894531]


Triplet accuracy:  0.9365234375


Confusion Matrix Total
[[4179  941]
 [ 461 4659]]

 Accuracy score: 0.8630859375

 f1 score: 0.8692164179104478

TPR:  0.8162109375
FPR:  0.1837890625
TNR:  0.9099609375
FNR:  0.0900390625
```

Analyse des données par sous-groupes

Histogrammes des distances par sous-groupes

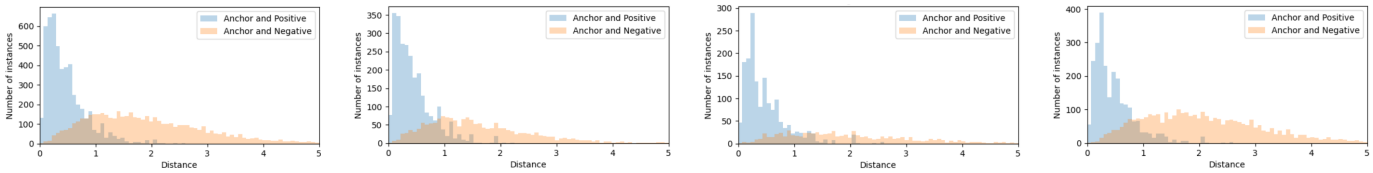


FIGURE 46 – Histogrammes de la distance, de gauche à droite : pour les données complètes, lorsque A et B sont « White Male », lorsque A et B ne sont pas « White Male », lorsque A ou B ne sont pas « White Male »

Statistiques par sous-groupes

Précision, taux d'apprentissage et f1-score

```
A and B White Male - mean accuracy:  0.8507664562669072
A and B White Male - bootstrapping mean accuracy:  [0.84151262 0.84732867 0.85076646 0.85437331 0.859344  ]
A and B White Male - triplet accuracy:  0.8480613165013525
A and B White Male - f1 score: 0.8781744571218255


A and B Non White Male - mean accuracy:  0.8756668034468609
A and B Non White Male - bootstrapping mean accuracy:  [0.86540829 0.87115306 0.8756668  0.88059089 0.88676652]
A and B Non White Male - triplet accuracy:  0.8021346469622331
A and B Non White Male - f1 score: 0.9053420805998126


A or B Non White Male - mean accuracy:  0.8850149674238422
A or B Non White Male - bootstrapping mean accuracy:  [0.87849974 0.88237366 0.88519105 0.88800845 0.89171509]
A or B Non White Male - triplet accuracy:  0.8904543853469532
A or B Non White Male - f1 score: 0.871583087512291
```

Matrice de confusion et taux d'erreurs

```
A and B White Male - Confusion Matrix
[[1388  425]
 [ 237 2386]]


A and B White Male - TPR:  0.7655819084390513
A and B White Male - FPR:  0.2344180915609487
A and B White Male - TNR:  0.9096454441479223
A and B White Male - FNR:  0.09035455585207777

A and B Non White Male - Confusion Matrix
[[ 685  140]
 [ 163 1449]]


A and B Non White Male - TPR:  0.8303030303030303
A and B Non White Male - FPR:  0.1696969696969697
A and B Non White Male - TNR:  0.8988833746898263
A and B Non White Male - FNR:  0.1011166253101737

A or B Non White Male - Confusion Matrix
[[2810  429]
 [ 224 2216]]


A or B Non White Male - TPR:  0.8675517134918185
A or B Non White Male - FPR:  0.13244828650818155
A or B Non White Male - TNR:  0.9081967213114754
A or B Non White Male - FNR:  0.09180327868852459
```

Étude de sous-groupes plus homogènes

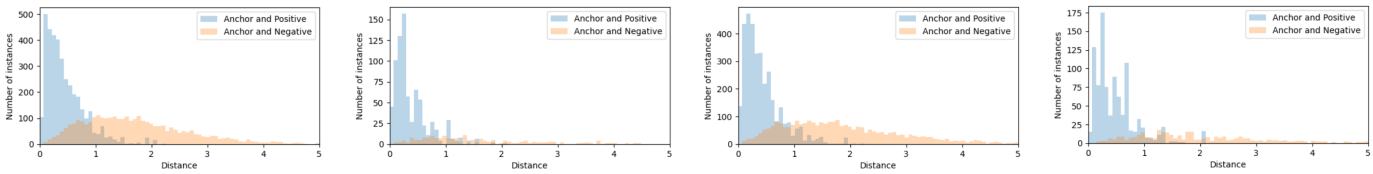


FIGURE 47 – Histogrammes de la distance, de gauche à droite : A et B sont « Male », A et B ne sont pas « Male », A et B sont « White », A et B ne sont pas « White »

```
A and B Male - mean accuracy:  0.8462987107054903
A and B Male - bootstrapping mean accuracy:  [0.83875851 0.84325656 0.84644358 0.84934087 0.85354194]
A and B Male - triplet accuracy:  0.8939437844103159
A and B Male - f1 score: 0.8626181535672666
A and B Male - Confusion Matrix
[[2511  705]
 [ 356 3331]]
A and B Male - TPR:  0.7807835820895522
A and B Male - FPR:  0.21921641791044777
A and B Male - TNR:  0.9034445348521833
A and B Male - FNR:  0.09655546514781665
```

```
A and B Non Male - mean accuracy:  0.8594377510040161
A and B Non Male - bootstrapping mean accuracy:  [0.84337349 0.85140562 0.85943775 0.86646586 0.87650602]
A and B Non Male - triplet accuracy:  0.6506024096385542
A and B Non Male - f1 score: 0.9118387909319899
A and B Non Male - Confusion Matrix
[[132  69]
 [ 71 724]]
A and B Non Male - TPR:  0.6567164179104478
A and B Non Male - FPR:  0.34328358208955223
A and B Non Male - TNR:  0.9106918238993711
A and B Non Male - FNR:  0.08930817610062892
```

```
A and B White - mean accuracy:  0.8542615484710475
A and B White - bootstrapping mean accuracy:  [0.84596617 0.85100846 0.85426155 0.85735198 0.86141835]
A and B White - triplet accuracy:  0.8822381262199089
A and B White - f1 score: 0.8738738738738738
A and B White - Confusion Matrix
[[2148  575]
 [ 321 3104]]
A and B White - TPR:  0.7888358428204186
A and B White - FPR:  0.21116415717958134
A and B White - TNR:  0.9062773722627737
A and B White - FNR:  0.09372262773722628
```

```
A and B Non White - mean accuracy:  0.874337623012869
A and B Non White - bootstrapping mean accuracy:  [0.85844058 0.8682816  0.87433762 0.88039364 0.88947767]
A and B Non White - triplet accuracy:  0.7151515151515152
A and B Non White - f1 score: 0.9126315789473683
A and B Non White - Confusion Matrix
[[288  68]
 [ 98 867]]
A and B Non White - TPR:  0.8089887640449438
A and B Non White - FPR:  0.19101123595505617
A and B Non White - TNR:  0.8984455958549222
A and B Non White - FNR:  0.10155440414507771
```

Références

[1] Loi n° 2004-800 du 6 août 2004 relative à la bioéthique, August 2004. URL : https://www.legifrance.gouv.fr/loda/id/LEGITEXT000005821809/.

[2] T.L. Berg, A.C. Berg, J. Edwards, M. Maire, R. White, Yee-Whye Teh, E. Learned-Miller, and D.A. Forsyth. Names and faces in the news. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages 848–854, Washington, DC, USA, 2004. IEEE. URL : http://ieeexplore.ieee.org/document/1315253/, doi:10.1109/CVPR.2004.1315253.

[3] Joy Buolamwini and Timnit Gebru. Gender Shades : Intersectional Accuracy Disparities in Commercial Gender Classification. In *Conference on Fairness, Accountability and Transparency*, pages 77–91. PMLR, January 2018. ISSN : 2640-3498. URL : https://proceedings.mlr.press/v81/buolamwini18a.html.

[4] Davide Castelvecchi. Is facial recognition too biased to be let loose ? *Nature*, 587(7834) :347–349, November 2020. URL : https://www.nature.com/articles/d41586-020-03186-4, doi:10.1038/d41586-020-03186-4.

[5] R. Chatila, L. Devillers, K. Dognin-Sauze, J.-G. Ganascia, M. Gornet, A. Pronesti, and C. Tessier. Pourquoi la reconnaissance faciale, posturale et comportementale soulève-t-elle des questionnements éthiques ? In E Germain, Cl. Kirchner, and C. Tessier, editors, *L'éthique du numérique : pour quoi faire ?* PUF, 2022.

[6] European Commission. White Paper On Artificial Intelligence - A European approach to excellence and trust, February 2020. URL : https://ec.europa.eu/info/sites/default/files/commission-white-paper-artificial-intelligence-feb2020_en.pdf.

[7] European Commission. Proposal for a Regulation of the European Parliament and of the Council laying down harmonised rules on Artificial Intelligence (Artificial Intelligence Act) and Amending certain Union legislative acts. COM/2021/206 final - Annex I : Artificial Intelligence techniques and approaches, April 2021. URL : https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0206.

[8] Sebastian Correa. Cosine Learning rate decay. *Medium*, September 2019. URL : https://scorrea92.medium.com/cosine-learning-rate-decay-e8b50aa455b.

[9] Ministère de l'Intérieur. Le passage rapide aux frontières extérieures PARAFE, May 2020. URL : https://www.immigration.interieur.gouv.fr/Europe-et-International/La-circulation-transfrontiere/Le-passage-rapide-aux-frontieres-exterieures-PARAFE.

[10] Commission de l'éthique en science et en technologie (CEST). Les enjeux éthiques soulevés par la reconnaissance faciale. *Gouvernement du Québec*, April 2020. URL : https://www.ethique.gouv.qc.ca/fr/publications/reconnaissance-faciale/.

[11] Commission de réflexion sur l'Éthique de la Recherche en sciences et technologies du Numérique d'Allistene (CERNA). Éthique de la recherche en apprentissage machine - Avis de la CERNA. Technical report, June 2017. URL : http://cerna-ethics-allistene.org/digitalAssets/53/53991_cerna__thique_apprentissage.pdf.

[12] Comité National Pilote d'Ethique du Numérique (CNPEN). Enjeux d'éthique concernant des outils numériques pour le déconfinement - Avis 1. Technical report, May 2020. URL : https://www.ccne-ethique.fr/sites/default/files/a_la_une/cnpen-ethique-numerique-deconfinement-2020-05-14.pdf.

[13] Comité National Pilote d'Ethique du Numérique (CNPEN). Enjeux d'éthique dans la lutte contre la désinformation et la mésinformation - Bulletin de veille n°2. Technical report, July 2020. URL : https://www.ccne-ethique.fr/sites/default/files/cnpen-desinformation-v2020-10-01.pdf.

[14] Comité National Pilote d'Ethique du Numérique (CNPEN). Enjeux d'éthique liés aux outils numériques en télémédecine et télésoin dans le contexte de la COVID-19 - Bulletin de veille n°3. Technical report, July 2020.

[15] Comité National Pilote d'Ethique du Numérique (CNPEN). Les enjeux éthiques des agents conversationnels - Appel à contributions, June 2020. URL : https://www.ccne-ethique.fr/sites/default/files/agents-conversationnels-appel_a_contributions-cnpen-2020-11-10.pdf.

[16] Comité National Pilote d'Ethique du Numérique (CNPEN). Réflexions et points d'alerte sur les enjeux d'éthique du numérique en situation de crise sanitaire aiguë - Bulletin de veille n°1. Technical report, April 2020. URL : https://www.ccne-ethique.fr/sites/default/files/publications/bulletin-1-ethique-du-numerique-covid19-2020-04-07.pdf.

[17] Comité National Pilote d'Ethique du Numérique (CNPEN). Le « véhicule autonome » : enjeux d'éthique - Avis 2. Technical report, April 2021. URL : https://www.ccne-ethique.fr/sites/default/files/cnpen-avis-vehicule-autonome-avril-2021.pdf.

[18] Comité National Pilote d'Ethique du Numérique (CNPEN). Les enjeux éthiques des agents conversationnels - Avis 3. Technical report, 2021.

[19] Comité National Pilote d'Ethique du Numérique (CNPEN). Manifeste pour une éthique du numérique. Á paraître, 2021.

[20] Comité National Pilote d'Ethique du Numérique (CNPEN). Reconnaissance faciale, posturale et comportementale - Appel à contributions. Á paraître, 2021.

[21] Thomas B. Fitzpatrick. The Validity and Practicality of Sun-Reactive Skin Types I Through VI. *Archives of Dermatology*, 124(6) :869, June 1988. URL : http://archderm.jamanetwork.com/article.aspx?doi=10.1001/archderm.1988.01670060015008, doi:10.1001/archderm.1988.01670060015008.

[22] Ad Hoc Expert Group (AHEG) for the Preparation of a Draft text of a Recommendation the Ethics of Artificial Intelligence. Avant-projet de Recommandation sur l'éthique de l'intelligence artificielle, 2020. URL : https://unesdoc.unesco.org/ark:/48223/pf0000373434_fre/PDF/373434fre.pdf.multi.

[23] Yosefa Gilon. CS231n : Convolutional Neural Networks for Visual Recognition - Stanford course, 2021. URL : https://cs231n.github.io/convolutional-networks/.

[24] Kashmir Hill. Wrongfully Accused by an Algorithm. *The New York Times*, June 2020. URL : https://www.nytimes.com/2020/06/24/technology/facial-recognition-arrest.html.

[25] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled Faces in the Wild : A Database for Studying Face Recognition in Unconstrained Environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. URL : http://vis-www.cs.umass.edu/lfw/.

[26] Diederik P. Kingma and Jimmy Ba. Adam : A Method for Stochastic Optimization. *arXiv :1412.6980 [cs]*, January 2017. arXiv : 1412.6980. URL : http://arxiv.org/abs/1412.6980.

[27] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. Attribute and simile classifiers for face verification. In *2009 IEEE 12th International Conference on Computer Vision*, pages 365–372, Kyoto, September 2009. IEEE. URL : http://ieeexplore.ieee.org/document/5459250/, doi:10.1109/ICCV.2009.5459250.

[28] Bhargav Lad. Guide to Pytorch Learning Rate Scheduling, 2020. URL : https://kaggle.com/isbhargav/guide-to-pytorch-learning-rate-scheduling.

[29] B. Levin and P. Kunz. ERCIM Workshop on Quality in AI. *ERCIM News*, 123 :4–5, 2020. URL : https://ercim-news.ercim.eu/en123/joint-ercim-actions/ercim-workshop-on-quality-in-ai.

[30] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. *arXiv :1411.7766 [cs]*, September 2015. arXiv : 1411.7766. URL : http://arxiv.org/abs/1411.7766.

[31] Lauren Kirchner Surya Jeff Larson Mattu, Julia Angwin. How We Analyzed the COMPAS Recidivism Algorithm. *ProPublica*, May 2016. URL : https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm?token=nD-X136_tDm0nh1l4XtvOLbpjY_BSO3u.

[32] Oliver Moindrot. Triplet Loss and Online Triplet Mining in TensorFlow, March 2018. URL : https://omoindrot.github.io/triplet-loss.

[33] Deirdre K. Mulligan, Joshua A. Kroll, Nitin Kohli, and Richmond Y. Wong. This Thing Called Fairness : Disciplinary Confusion Realizing a Value in Technology. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW) :1–36, November 2019. URL : https://dl.acm.org/doi/10.1145/3359221, doi:10.1145/3359221.

[34] Independent Hight-Level Expert Group on Artificial Intelligence set up by the European Commission (HLEG). Assessment List for Trustworthy Artificial Intelligence (ALTAI) for self-assessment, July 2020. URL : https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment.

[35] Reuters. Amazon ditched AI recruiting tool that favored men for technical jobs. *the Guardian*, October 2018. URL : http://www.theguardian.com/technology/2018/oct/10/amazon-hiring-ai-gender-bias-recruiting-engine.

[36] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet : A Unified Embedding for Face Recognition and Clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, June 2015. arXiv : 1503.03832 version : 1. URL : http://arxiv.org/abs/1503.03832, doi:10.1109/CVPR.2015.7298682.

[37] Catherine Tessier. Éthique et IA : analyse et discussion. *Conférence nationale en intelligence artificielle (CNIA)*, 2021. URL : https://pfia2021.fr/conferences/cnia/actes_CNIA_CH_PFIA2021.pdf.

[38] Morgane Tual. A peine lancée, une intelligence artificielle de Microsoft dérape sur Twitter. *Le Monde.fr*, March 2016. URL : https://www.lemonde.fr/pixels/article/2016/03/24/a-peine-lancee-une-intelligence-artificielle-de-microsoft-derape-sur-twitter_4889661_4408996.html.

[39] Steven Umbrello and Ibo Van de Poel. Mapping value sensitive design onto AI for social good principles. *AI and Ethics*, February 2021. URL : http://link.springer.com/10.1007/s43681-021-00038-3, doi:10.1007/s43681-021-00038-3.

[40] Cédric Villani. Donner un sens à l'Intelligence Artificielle - Pour une Stratégie Nationale et Européenne. Technical report, March 2018. URL : https://cdn2.nextinpact.com/medias/rapport-villani-ia.pdf.

[41] Paul Viola and Michael J Jones. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2) :137–154, 2004. URL : https://www.face-rec.org/algorithms/Boosting-Ensemble/16981346.pdf.

[42] Mei Wang and Weihong Deng. Deep Face Recognition : A Survey. *Neurocomputing*, 429 :215–244, March 2021. arXiv : 1804.06655. URL : http://arxiv.org/abs/1804.06655, doi:10.1016/j.neucom.2020.10.081.

[43] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1) :67–82, April 1997. URL : http://ieeexplore.ieee.org/document/585893/, doi:10.1109/4235.585893.