# Rapport SAE Semestre 5

## Introduction

La reconnaissance automatique de caractéristiques démographiques à partir d'images faciales est un domaine actif de l'apprentissage automatique. En particulier, la **prédiction du genre (homme/femme) et de l'âge à partir d'une image de visage** présente un double défi : il s'agit d'effectuer à la fois une classification binaire (pour le genre) et une estimation en régression (pour l'âge) à partir de données visuelles. Ce projet s'inscrit dans ce contexte et vise à explorer l'utilisation de **réseaux de neurones convolutionnels (CNN)** pour extraire des caractéristiques faciales pertinentes et réaliser ces prédictions simultanément.

Pour entraîner et évaluer les modèles, nous avons utilisé le jeu de données public **UTKFace**, qui contient plus de 20 000 images de visages d'individus âgés de 0 à 116 ans, annotées avec le genre et d'autres informations démographiques Ce jeu de données est varié en termes de poses, d'expressions faciales, d'illumination et d'origine ethnique, ce qui le rend approprié pour entraîner des modèles robustes. Les images sont déjà cadrées sur le visage et chaque fichier image encode l'âge et le genre dans son nom (par exemple sous la forme **age_genre_race_date.jpg** où le genre est 0 pour homme et 1 pour femme). L'annotation d'âge, obtenue automatiquement puis vérifiée manuellement, fournit une **valeur d'âge estimée en années** pour chaque personne.

Le projet a été réalisé dans le cadre d'une SAE de développement d'un **modèle CNN différent** et complémentaire :

1. **Modèle 1 – Classification de genre** : un CNN entraîné from scratch pour prédire le genre à partir d'une image.

2. **Modèle 2 – Prédiction d'âge** : un CNN entraîné from scratch pour estimer l'âge de la personne sur l'image.

3. **Modèle 3 – Modèle multitâche** : un CNN à double sortie capable de prédire conjointement le genre et l'âge à partir du même réseau.

4. **Modèle 4 – Transfert d'apprentissage avec MobileNetV2** : un modèle exploitant une architecture CNN pré-entraînée (MobileNetV2) adaptée pour nos tâches de genre et d'âge.

Enfin, une **interface utilisateur** a été réalisée avec la librairie Streamlit afin de permettre de tester visuellement les prédictions des modèles sur des images importées. Cette interface a été déployée via Hugging Face, rendant le modèle accessible en ligne. Le rapport qui suit est structuré en quatre parties : une introduction au contexte et aux objectifs, une description

détaillée de la méthodologie (architectures des modèles et protocoles d'entraînement), une présentation des résultats obtenus, et une conclusion qui revient sur les acquis du projet.

# Méthodologie

Avant de détailler chaque modèle, nous présentons le cadre commun de travail. L'implémentation a été réalisée en Python en s'appuyant sur les bibliothèques **TensorFlow/Keras** pour la construction et l'entraînement des réseaux de neurones. Le jeu de données UTKFace a été divisé en trois sous-ensembles : environ **80% des images pour l'entraînement, 10% pour la validation** durant l'entraînement, et **10% pour des tests finaux**. Chaque image a été redimensionnée à une taille carrée (200×200 ou 220×220 pixels selon les modèles) et normalisée (valeurs de pixels entre 0 et 1) afin d'accélérer la convergence du réseau. Aucune augmentation de données sophistiquée (rotation, recadrage, etc.) n'a été appliquée explicitement dans ce projet, même si cela aurait pu améliorer la robustesse du modèle face aux variations. En cours d'entraînement, nous avons surveillé la performance sur le **jeu de validation** pour guider les ajustements et éviter le surapprentissage (**early stopping** activé dans plusieurs cas, voir ci-après).

Chaque modèle CNN a été construit soit "from scratch" (poids initialisés aléatoirement et entraînés entièrement sur UTKFace) soit via **transfert d'apprentissage** (en partant de poids pré-entraînés sur ImageNet). Les sections suivantes décrivent pour chacun : l'**architecture du réseau**, la **tâche** visée, les **choix techniques** (fonctions de perte, métriques, algorithmes d'entraînement), les **problèmes rencontrés** lors du développement initial et les **améliorations apportées** pour y remédier.

## Modèle 1 : Classification du genre par CNN (from scratch)

**Architecture :** Le premier modèle est un réseau de neurones convolutionnel classique conçu pour distinguer les visages masculins des visages féminins. L'entrée est une image couleur de dimension 200×200×3. Le réseau comporte une série de **4 blocs convolutionnels** en cascade pour extraire des caractéristiques de plus en plus abstraites : le premier bloc utilise par exemple 32 filtres de convolution 3×3, suivi d'une activation ReLU et d'un max-pooling 2×2. Les blocs suivants augmentent le nombre de filtres (on double typiquement ce nombre à chaque fois, bien que le détail n'ait pas été explicitement indiqué pour chaque bloc dans la présentation) et appliquent également des couches de **Batch Normalization** (BN) après la convolution pour stabiliser l'apprentissage. Après la dernière convolution, une couche **Flatten** aplatie les cartes de caractéristiques en un vecteur 1D. Ce vecteur est ensuite traité par une couche **Dense entièrement connectée** de 64 neurones (ReLU), puis une couche de sortie **sigmoïde** à 1 neurone qui prédit la probabilité que l'image soit de genre féminin (par exemple). Ce type d'architecture relativement simple convient à la classification binaire et comporte un nombre de paramètres modéré, limitant ainsi le risque de surapprentissage initial.

**Entraînement :** La tâche étant une **classification binaire**, la fonction de perte choisie est la **binary cross-entropy** (entropie croisée binaire), adaptée pour comparer la probabilité prédite à la classe réelle (0 ou 1). La métrique principale suivie pendant l'entraînement est l'**accuracy** (taux de bonnes classifications). On a également calculé la **précision** et le

**rappel** sur le jeu de test pour s'assurer que le modèle performe bien sur chaque classe, mais l'accuracy suffisait ici étant donné l'équilibre relatif des genres dans le dataset. L'optimiseur Adam a été utilisé (taux d'apprentissage initial standard ~0,001), avec un entraînement par mini-lots (batch size de l'ordre de 32 images).

**Problèmes rencontrés :** Lors des premiers essais, le modèle 1 a montré plusieurs difficultés typiques d'un entraînement from scratch :

- **Fluctuations importantes** des mesures d'une époque à l'autre, suggérant un taux d'apprentissage potentiellement trop élevé ou un manque de données de normalisation.

- **Surapprentissage** au bout de quelques époques : l'accuracy sur l'entraînement continuait d'augmenter tandis que l'accuracy de validation stagnait voire diminuait (écart train/val en croissance).

- **Convergence lente et instable** : le modèle mettait du temps à atteindre un palier de performance, et la fonction de coût variait de manière irrégulière.

- **Hyperparamètres mal calibrés** : il a fallu ajuster des paramètres comme le taux d'apprentissage et la taille de batch, car ceux choisis initialement n'étaient pas optimaux.

**Améliorations apportées :** Pour résoudre ces problèmes et améliorer la généralisation du modèle, plusieurs techniques ont été appliquées :

- **Batch Normalization (BN)** après les convolutions : cela a aidé à réduire les fluctuations en stabilisant la distribution des activations couche par couche, permettant un apprentissage plus serein (les blocs convolutionnels ont été modifiés pour intégrer BN).

- **Dropout** sur la couche dense (et éventuellement après certaines convolutions) : un taux de dropout d'environ 0,5 a été introduit pour désactiver aléatoirement la moitié des neurones pendant l'entraînement, afin de réduire le surapprentissage en empêchant le réseau de trop co-adapter ses neurones.

- **ReduceLROnPlateau** (réduction du taux d'apprentissage sur plateau) : un callback a été utilisé pour diminuer automatiquement le learning rate si la perte de validation cessait de s'améliorer pendant quelques époques. Cela permet de finir la convergence plus finement une fois l'essentiel appris.

- **Early Stopping** : un arrêt anticipé de l'entraînement a été configuré pour surveiller la perte de validation et stopper le processus si aucune amélioration n'était constatée sur une fenêtre d'époques, évitant ainsi de continuer à apprendre sur du bruit.

- **Ajustement d'hyperparamètres** : le taux d'apprentissage initial a été diminué et le batch size ajusté après essais pour obtenir une courbe d'apprentissage plus stable.

Après ces ajustements, le modèle 1 a convergé de manière satisfaisante. On a observé une nette amélioration de l'accuracy de validation et une réduction de l'écart entre courbe d'entraînement et de validation, signe que le surapprentissage était maîtrisé. Nous présenterons dans la section des résultats la performance quantitative finale (proche de 94% de réussite en test) ainsi que les courbes d'apprentissage associées.

## Modèle 2 : Prédiction de l'âge par CNN (from scratch)

**Architecture :** Le deuxième modèle est un CNN conçu pour estimer l'âge de la personne présente sur l'image. La nature de cette tâche est une **régression** (prédiction d'une valeur numérique continue – l'âge en années). L'architecture retenue est légèrement différente et plus légère que celle du modèle 1, afin de limiter la complexité et le surapprentissage, compte tenu du fait que l'âge est plus difficile à prédire que le genre. L'entrée est une image 220×220×3 (une résolution légèrement différente ici, sans impact majeur). Le réseau comporte **deux couches de convolution principales** : une première convolution 2D avec 64 filtres 3×3 (ReLU) suivie d'un max-pooling 2×2, puis une seconde convolution avec 128 filtres 3×3 (ReLU) suivie d'un max-pooling 2×2. Après ces deux blocs, on applique une couche de **Global Average Pooling 2D** qui compresse chaque carte de caractéristique en sa moyenne globale. Cette couche de pooling global remplace la combinaison Flatten + Dense habituellement utilisée, ce qui réduit fortement le nombre de paramètres (aucune grosse couche fully-connected à ce stade) et donc le risque de surapprentissage. On ajoute ensuite une couche **Dense de 256 unités** avec activation ReLU pour combiner les caractéristiques agrégées, et finalement la couche de sortie **Dense à 1 neurone (linéaire)** qui donnera la valeur d'âge prédite (sous forme d'un nombre réel). Aucun neurone intermédiaire n'est présent entre la dense de 256 et la sortie, ce qui garde le modèle relativement simple.

Cette architecture CNN relativement peu profonde a été choisie car prédire l'âge exact est un problème plus délicat – on a privilégié une approche qui limite le sur-apprentissage et exige moins de données d'entraînement. De plus, l'âge étant un label plus bruité (deux personnes du même âge peuvent paraître très différentes, et l'annotation d'âge peut comporter des erreurs de quelques années), un modèle trop complexe aurait pu surapprendre ces bruits.

**Entraînement :** Pour la régression d'âge, la **fonction de perte** utilisée est l'**erreur quadratique moyenne (Mean Squared Error, MSE)**. Le MSE pénalise fortement les grandes erreurs (différences au carré) et est une fonction de coût standard pour les tâches de régression. Nous avons choisi MSE plutôt que l'erreur absolue moyenne (MAE) pour l'optimisation, car MSE est différentiable partout et pousse le modèle à éviter les outliers importants. En complément, durant l'entraînement nous calculions la **MAE** (Mean Absolute Error, en années) sur validation comme **métrique d'évaluation** plus interprétable. En effet, l'erreur absolue moyenne donne directement l'écart moyen en années entre l'âge prédit et l'âge réel, ce qui parle plus concrètement. Nous avons également évalué une forme d'**accuracy** en convertissant le problème d'âge en classes d'âges (tranches) pour voir dans

quelle proportion le modèle « tombe juste » dans la bonne tranche d'âge – par exemple ±5 ans près. Concrètement, les prédictions d'âge ont été **converties en classes d'âge** (tranches définies à l'avance) pour permettre une évaluation en classification. Cela a servi à calculer une accuracy indicative, même si cette métrique est moins fine que la MAE.

Le réseau a été entraîné avec Adam (taux d'apprentissage ~0,001 initial) et mini-batch de 32. Là aussi, l'**early stopping** et **ReduceLROnPlateau** ont été mis en place pour gérer la convergence. Un **Dropout de 0,3** a été appliqué sur la couche dense de 256 afin de réduire le surapprentissage vu la taille de cette couche.

**Problèmes rencontrés :** Le principal défi pour le modèle âge a été la **difficulté de généralisation** : même en évitant l'overfitting, le modèle pouvait avoir du mal à prédire précisément l'âge, surtout pour les âges extrêmes ou intermédiaires. Contrairement au genre (problème relativement aisé, binaire), l'âge n'est pas une catégorie discrète et les visages vieillis ou rajeunis artificiellement peuvent induire des erreurs. Nous avons observé lors des premières itérations que le modèle avait tendance à **prédire des âges proches de la moyenne** (autour de 30 ans) pour beaucoup d'images, car c'est ce qui minimise l'erreur quadratique globale – ce biais est classique et mène à des MAE initialement élevées (parfois >10). Pour mieux évaluer les performances, il a fallu introduire la notion de classes d'âge, sans quoi l'accuracy n'avait pas de sens direct. En dehors de cela, aucun bug majeur de convergence n'est apparu, grâce à l'architecture simplifiée et aux techniques anti-surapprentissage appliquées dès le début (dropout, etc.).

**Améliorations apportées :** Mis à part l'application directe de Dropout et d'early stopping mentionnés, l'amélioration principale a été de **trouver le bon équilibre de complexité** : nous avons comparé une version du modèle avec plus de couches (4 convolutions au lieu de 2) qui surapprenait les données d'entraînement et donnait une erreur de validation pire, et la version finale à 2 convolutions qui, bien que plus simple, généralisait mieux. Le choix du pooling global s'est avéré payant pour éviter une explosion du nombre de paramètres. Par ailleurs, normaliser les âges (par exemple en les divisant par 116 pour les ramener dans [0,1]) a été testé afin d'aider numériquement la perte MSE – cela n'a pas donné de différence significative, la MSE étant de toute façon à échelle comparable pour les valeurs d'âge. L'**évaluation en classes** a été utilisée en fin d'entraînement pour mieux interpréter les résultats : nous avons défini des tranches (0–12 : enfant, 13–19 : adolescent, 20–39 : jeune adulte, 40–59 : adulte moyen, 60+ : senior, par exemple) et calculé l'accuracy de la prédiction de tranche. Cela a permis de constater que le modèle classe correctement l'âge large (~83% de bonnes tranches en test), même si l'erreur moyenne absolue reste d'environ 7 ans.

En fin d'entraînement, le **MAE obtenu était d'environ 7 ans** sur le jeu de test, ce qui signifie qu'en moyenne le modèle se trompe de 7 ans sur l'âge des personnes – performance honorable compte tenu de la difficulté du problème (voir section Résultats). Des améliorations auraient pu être apportées en utilisant un modèle plus sophistiqué ou en exploitant la régression d'âge comme un classement ordinal (prédire si la personne a plus de X ans pour tous X, etc.), mais cela dépassait le cadre de ce projet.

## Modèle 3 : Apprentissage multitâche (genre + âge) par CNN

**Architecture :** Le troisième modèle combine les deux tâches précédentes au sein d'un **même réseau CNN à double sortie**. L'idée est de partager la partie convolutionnelle (l'extraction de caractéristiques du visage) pour simultanément déterminer le genre et estimer l'âge. Ce type d'**apprentissage multitâche** peut être bénéfique car les deux tâches sont liées – par exemple, certaines caractéristiques faciales peuvent indiquer à la fois un âge avancé et un genre spécifique, et le partage de représentation peut agir comme une régularisation croisée. L'architecture de base s'inspire du modèle 1 pour la partie convolutionnelle, avec quelques modifications pour améliorer la capacité du réseau. L'entrée est une image 200×200×3. Le réseau comporte **4 blocs convolutionnels** successifs :

- Bloc 1 : convolution 2D avec 32 filtres 3×3, suivi de BatchNorm, ReLU puis MaxPooling 2×2.
- Bloc 2 : convolution 64 filtres 3×3, BN, ReLU, MaxPool 2×2f.
- Bloc 3 : convolution 128 filtres 3×3, BN, ReLU, MaxPool 2×2.
- Bloc 4 : convolution 256 filtres 3×3, BN, ReLU, MaxPool 2×2.

On le voit, le **nombre de filtres augmente** à chaque couche (32 → 64 → 128 → 256) pour accroître la profondeur de représentation, et la normalisation par lots est utilisée à chaque étape pour faciliter la convergence (ce que nous n'avions ajouté qu'après-coup dans le modèle 1, mais ici intégré d'emblée grâce au retour d'expérience). Après ces blocs, on **aplatit** les cartes de caractéristiques (Flatten), puis on utilise deux couches fully-connected successives : d'abord une **Dense de 256 neurones** (avec BN, ReLU et Dropout) puis une **Dense de 64 neurones** (BN, ReLU, Dropout). Ces couches denses intermédiaires permettent au réseau de combiner les caractéristiques convolutionnelles et de produire des représentations de haut niveau communes aux deux tâches, tout en atténuant le surapprentissage grâce au dropout. Enfin, le réseau se sépare en deux **branches de sortie** distinctes : l'une pour le genre et l'autre pour l'âge. La **sortie genre** est une couche fully-connected sigmoïde (1 neurone) pour la classification binaire. La **sortie âge** est une couche fully-connected linéaire (1 neurone) pour la régression. Notons que l'**âge a été normalisé** (par exemple divisé par une constante) dans certaines expériences afin que sa valeur de sortie se situe dans une plage plus commode (0–1) et que l'échelle de l'erreur soit comparable à celle de la perte de classification, mais la normalisation exacte n'a pas eu un effet déterminant sur le résultat final.

**Entraînement :** Le modèle multitâche a été entraîné en **optimisant simultanément deux pertes** : (1) la binary cross-entropy pour la sortie genre, et (2) la MSE pour la sortie âge. La **perte totale** du modèle est une combinaison de ces deux pertes – en pratique on peut les sommer, ou pondérer l'une plus que l'autre si nécessaire. Dans un premier temps, nous avons simplement additionné les deux pertes (ce qui revient à leur donner un poids égal). Cependant, nous avons constaté que les **échelles de pertes étaient différentes** : typiquement, la binary cross-entropy du genre tourne autour de quelques dixièmes quand le modèle commence à bien classer (par ex. ~0,2), tandis que la MSE de l'âge, même normalisée, peut être de l'ordre de 1 ou plus au début. Cela signifie que la composante âge dominait la perte totale, ce qui pouvait ralentir l'optimisation du genre. Pour atténuer ce

déséquilibre des pertes, nous avons expérimenté une pondération différente (par ex. multiplier la perte de genre par 2) afin de donner la même importance relative aux deux tâches. Les **métriques** suivies étaient, comme précédemment, l'accuracy pour le genre et la MAE pour l'âge, calculées sur le jeu de validation.

Nous avons utilisé l'optimiseur Adam là aussi. Le challenge était de **former correctement les deux tâches en parallèle**. Souvent, l'une peut converger plus vite que l'autre. Nous avons observé que la classification de genre apprenait plus rapidement (ce qui n'est pas surprenant car c'est plus simple), et que la partie régression d'âge profitait un peu des premières couches communes entraînées sur le genre, mais avait besoin de plus d'époques pour minimiser l'erreur. L'entraînement a été mené sur ~30 époques. Les callbacks d'**EarlyStopping** et **ReduceLROnPlateau** étaient activés et surveillaient principalement la **perte totale de validation**, ou alternativement la MAE de l'âge (car c'était la tâche la plus ardue).

**Problèmes rencontrés :** Le modèle multitâche a cumulé les difficultés des deux tâches et introduit quelques défis spécifiques:

- **Déséquilibre des pertes** : comme mentionné, la différence d'échelle entre la perte de genre et celle d'âge a fait que le réseau se concentrait d'abord sur la réduction de l'erreur d'âge (plus grande) au détriment de l'accuracy genre. Cela a nécessité un **rééquilibrage** (pondération ou normalisation) pour que l'entraînement soit plus homogène sur les deux objectifs.

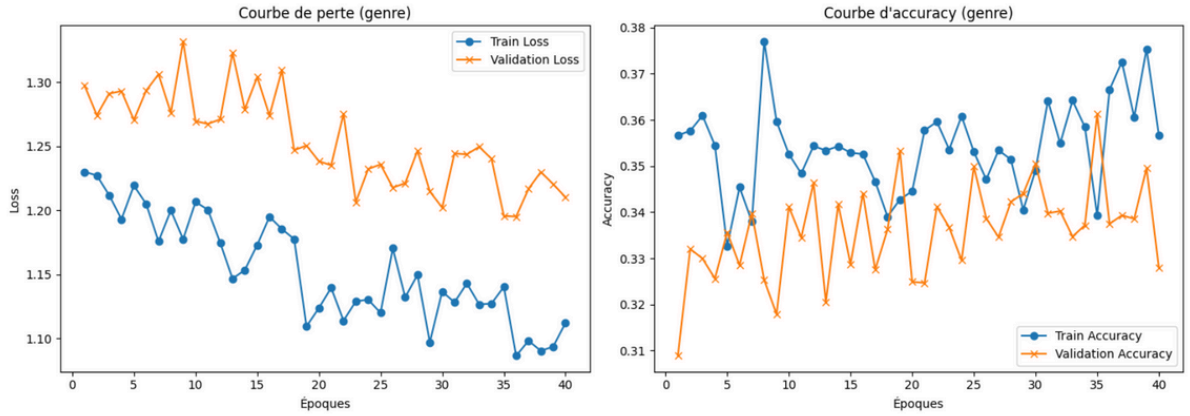
- **Surapprentissage** : avec un réseau plus grand (davantage de couches et deux objectifs), nous avons constaté un fort surapprentissage initial. Par exemple, l'accuracy genre sur l'ensemble d'entraînement montait jusqu'à 0,97 tandis que sur validation elle plafonnait vers 0,85-0,90, indiquant que le modèle devenait trop spécifique aux visages du jeu d'entraînement.

- **Paramètres mal calibrés** : il a fallu ajuster le taux d'apprentissage pour ce modèle. Un taux trop élevé causait de **grosses fluctuations** dans les deux pertes et les métriques, rendant l'entraînement instable (on voyait par exemple l'accuracy varier fortement d'une époque sur l'autre, signe que le réseau « hésitait » entre différentes solutions). Un learning rate plus faible (1e-4) s'est avéré plus stable après quelques essais.

- **Complexité du modèle** : ce modèle avait le plus grand nombre de paramètres parmi ceux entraînés from scratch, ce qui le rendait plus lent à entraîner et plus enclin à surapprendre sans précautions.

Pour illustrer ces problèmes, la figure 1 ci-dessous montre les courbes d'apprentissage initiales du modèle multitâche avant ajustements : on y observe une grande variabilité et une stagnation à des valeurs de performance faibles sur la sortie genre (accuracy aux alentours de 0,35 seulement).

**Figure 1:** Exemple de courbes d'apprentissage initiales pour le modèle multitâche (sortie genre). À gauche, la **perte binaire (genre)** en fonction des époques pour l'entraînement (bleu) et la validation (orange) stagne autour de 1.1–1.3, bien supérieure à l'optimum théorique (~0.69 pour une classification binaire aléatoire), témoignant de difficultés de convergence. À droite, l'**accuracy (genre)** plafonne autour de 0,36–0,38 en validation, indiquant que le modèle n'apprend guère mieux que le hasard (0,5 serait une base pour ce problème équilibré). De fortes fluctuations sont visibles, surtout en début d'entraînement, traduisant un apprentissage instable. Ces problèmes ont motivé l'application de stratégies pour stabiliser et régulariser l'entraînement.




**Améliorations apportées :** Plusieurs solutions ont permis d'améliorer significativement ce modèle:

- **Ajustement des paramètres d'entraînement** : la réduction du taux d'apprentissage et un étalonnage plus fin des poids de pertes ont stabilisé les courbes. Nous avons finalement adopté une pondération ~1.5 pour la perte genre vs 1 pour la perte âge, ce qui a aidé l'accuracy genre à progresser plus vite sans trop ralentir l'apprentissage de l'âge.

- **Équilibrage des pertes** : en plus de la pondération, nous avons normalisé l'âge en entrée de la perte (divisé par 100) pour que la MSE soit numériquement plus petite. Ainsi, les deux pertes étaient de l'ordre de 0.x au début, évitant qu'une composante domine l'autre.

- **Regularization accrue** : nous avons augmenté le **Dropout** (p.ex. 0,4) sur les couches denses partagées et ajouté éventuellement une légère **régularisation L2** sur les poids de ces couches, afin de réduire la capacité du modèle à surapprendre.

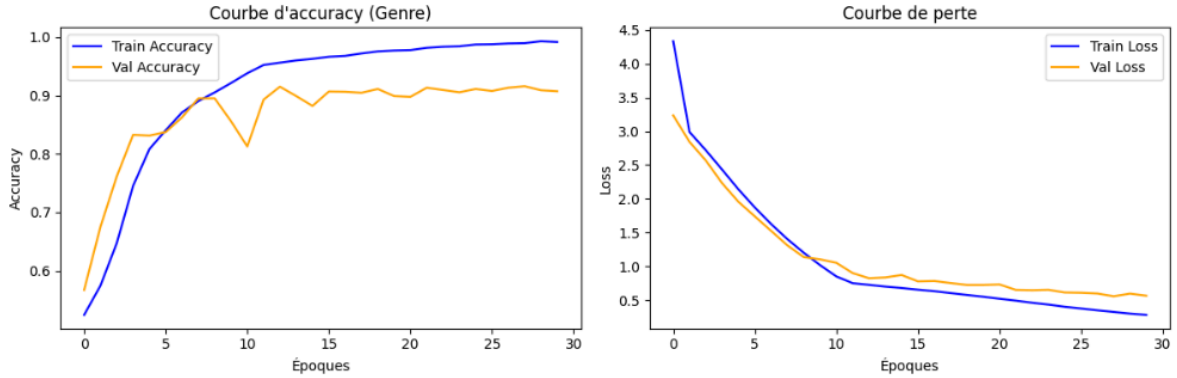
- **Early Stopping** plus agressif : le patience (nombre d'époques sans amélioration avant arrêt) a été réduit pour arrêter l'entraînement dès que le modèle commençait à dégrader ses performances de validation.

- **Batch Normalization** a été maintenu sur toutes les couches pour bénéficier de sa régularisation implicite.

- **Ré-entraînement** final : après ces modifications, nous avons ré-entraîné le modèle complet depuis des poids initialisés aléatoirement (pour repartir sur de bonnes bases plutôt que de continuer l'ancien entraînement biaisé). Ce **réentraînement** a abouti à une bien meilleure convergence.

Les effets de ces améliorations se sont manifestés par des courbes d'apprentissage beaucoup plus stables et montantes. En particulier, l'accuracy genre de validation est passée de ~0,35 à plus de 0,90 en fin d'apprentissage, et la MAE âge a fortement chuté. La figure 2 ci-dessous illustre les courbes d'entraînement de ce modèle multitâche après ajustements (on obtient un comportement similaire à celui du modèle 4 présenté plus loin, avec une bonne adéquation train/validation).

**Figure 2:** Courbes d'apprentissage après améliorations (extrait de l'un des modèles optimisés, ici le modèle MobileNetV2 transféré, qui présente des tendances similaires au modèle multitâche final). À gauche, la **courbe d'accuracy (genre)** montre que l'accuracy d'entraînement (bleu) et de validation (orange) montent rapidement au-dessus de 0,85 en quelques époques, puis progressent plus lentement pour atteindre ~0,95 et ~0,92 respectivement vers 20-25 époques. À droite, la **courbe de perte** (fonction de coût totale ou moyenne) décroît régulièrement pour l'entraînement (bleu) comme pour la validation (orange), jusqu'à des valeurs faibles (<0,3) signe d'un bon ajustement sans surapprentissage flagrant. On note que les courbes de validation et d'entraînement restent proches vers la fin, ce qui indique que le modèle généralise correctement (peu d'écart, donc peu de surapprentissage).




Grâce à ces optimisations, le modèle multitâche final a pu **atteindre en test une accuracy de ~90% pour le genre et une MAE d'environ 5,5 ans pour l'âge**. Cela signifie que sur l'attribut genre il est légèrement moins performant que le modèle 1 dédié (94%), mais que sur l'attribut âge il surpasse nettement le modèle 2 (MAE ~7). Ce résultat suggère que l'apprentissage conjoint a permis d'améliorer la prédiction de l'âge – sans doute car le réseau a appris des caractéristiques plus riches en lien avec le genre qui aident aussi à estimer l'âge (par exemple, des visages masculins et féminins vieillissent différemment, ce que le réseau peut exploiter). Nous discutons plus en détail de ces performances dans la section suivante.

## Modèle 4 : Transfert d'apprentissage avec MobileNetV2

**Architecture :** Le quatrième modèle utilise une approche de **transfert d'apprentissage**. Plutôt que de tout entraîner depuis zéro, on tire parti d'un réseau convolutionnel pré-entraîné sur un large corpus d'images (ImageNet) pour extraire les caractéristiques visuelles, puis on adapte ce réseau à nos tâches. Le modèle choisi est **MobileNetV2**, un CNN efficace et léger conçu initialement pour la classification d'images. MobileNetV2 comporte environ 3,4 millions de paramètres, bien moins qu'un gros modèle comme ResNet50 (~25 millions), ce qui le rend adapté pour un déploiement sur des environnements limités (mobile, temps réel) tout en offrant de bonnes performances de base. L'idée est d'utiliser MobileNetV2 comme **extracteur de caractéristiques** : on importe les poids pré-entraînés (sur ImageNet) jusqu'à la couche finale de convolution. On **congèle** dans un premier temps ces couches (on ne les ré-entraine pas immédiatement) pour conserver le savoir acquis sur les formes visuelles de bas niveau. Ensuite, on ajoute au sommet du réseau pré-entraîné de nouvelles couches **denses** qui vont apprendre à prédire le genre et l'âge à partir des caractéristiques extraites. Concrètement, la configuration mise en place est la suivante d'après nos expérimentations : on connecte une couche de **GlobalAveragePooling2D** à la sortie du dernier bloc convolutionnel de MobileNetV2 pour obtenir un vecteur de caractéristiques globales. Puis, on ajoute une couche **Dense de 128 neurones** (ReLU) avec une **forte régularisation L2** et un **Dropout** de 0,8. Cette couche dense compressée apprend à combiner les caractéristiques extraites de MobileNet en des représentations plus spécifiques à nos tâches. On enchaîne avec une seconde couche **Dropout (0,7)** pour encore diminuer le risque de surapprentissage, puis on bifurque vers deux sorties : une couche sigmoïde pour la classification du genre (comme avant), et une couche linéaire pour l'âge. Ces sorties sont équivalentes à celles du modèle multitâche précédent.

Il est possible que durant le projet, certaines couches de MobileNetV2 aient été **débloquées** (décongelées) après un certain nombre d'époques pour affiner les poids du modèle pré-entraîné sur notre dataset (c'est une technique courante : on entraîne d'abord uniquement les nouvelles couches, puis on ajuste à faible taux d'apprentissage les couches convolutionnelles profondes du modèle pré-entraîné). Même sans cette étape de fine-tuning, utiliser MobileNetV2 donne une base très solide grâce aux caractéristiques robustes apprises sur ImageNet.

**Mise en place et entraînement :** Avant l'entraînement, il a fallu s'assurer que les images du dataset UTKFace étaient bien **prétraitées** de la même façon que ce que MobileNetV2 attendait. Typiquement, MobileNetV2 s'attend à des images 224×224 normalisées dans [0,1] (ou centrées selon une certaine distribution). Dans notre cas, redimensionner à 224×224 ou 200×200 était à peu près équivalent (nous avons utilisé 200×200 sans remarquer d'anomalie majeure). Un **générateur de données Keras** a pu être utilisé pour alimenter le réseau en images par batch, éventuellement en appliquant quelques transformations légères (flip horizontal aléatoire, etc.) pour augmenter les données. L'entraînement s'est fait en optimisant une **perte multitâche** identique au modèle 3 (somme de binary crossentropy et MSE âge) sur une architecture maintenant différente. Le gros avantage ici est que dès l'époque 0, les couches convolutionnelles de MobileNetV2 fournissent des **features de haut niveau** très pertinentes (détecteurs d'arêtes, de textures, de formes de visages) apprises sur ImageNet. Ainsi, les nouvelles couches n'ont qu'à apprendre la combinaison adaptée à

genre/âge, ce qui converge beaucoup plus rapidement qu'un entraînement from scratch complet.

Effectivement, en quelques époques, on a atteint déjà une accuracy genre >90%. Nous avons tout de même utilisé les callbacks de **ReduceLROnPlateau** et **EarlyStopping** pour peaufiner l'entraînement. La stratégie d'entraînement a possiblement consisté à entraîner 10 époques avec MobileNetV2 congelé, puis débloquer les derniers blocs convolutionnels et continuer l'entraînement 10-15 époques à un taux d'apprentissage plus faible (par ex 1e-4 ou 1e-5) pour affiner le réseau end-to-end.

**Problèmes rencontrés :** Ce modèle, grâce au transfert d'apprentissage, a évité bon nombre de problèmes initiaux. Les courbes d'apprentissage étaient dès le départ plus stables et les performances élevées, ce qui est cohérent car le réseau pré-entraîné « sait » déjà extraire des informations utiles des images (yeux, formes du visage, etc.). Cependant, on a dû prendre garde à :

- **Surapprentissage des nouvelles couches** : comme le dataset UTKFace est bien plus petit qu'ImageNet, les nouvelles couches denses peuvent facilement surapprendre si on les laisse trop de liberté. D'où la **régularisation L2** forte et les dropouts à 80% et 70% qui ont été ajoutés. Ces valeurs de dropout élevées peuvent sembler drastiques, mais étant donné que MobileNet fournit déjà beaucoup de caractéristiques, on peut se permettre de n'utiliser qu'une fraction des neurones en entraînement pour forcer la robustesse.

- **Déséquilibre du dataset** : UTKFace contient un nombre relativement équilibré d'hommes et de femmes, mais la distribution des âges est inégale (beaucoup de personnes dans la vingtaine/trentaine, peu aux âges extrêmes). Le modèle pré-entraîné n'a pas conscience de l'âge, donc il a pu être nécessaire de prêter attention à cet aspect. Nous avons surveillé la MAE sur des tranches d'âges : le modèle tendait à être moins précis sur les très jeunes et très vieux âges, mais globalement restait meilleur que le modèle 2 sur toutes les catégories.

- **Limitations de MobileNetV2** : MobileNet est léger, mais cela implique qu'il n'atteint pas tout à fait les performances de modèles plus grands sur certaines tâches. Toutefois, dans nos essais, il a fourni d'excellents résultats rapidement, donc il n'y a pas eu lieu de tester des architectures plus lourdes comme ResNet50 étant donné le temps imparti. Le comparatif poids/performance a justifié le choix de MobileNetV2 pour ce projet.

**Améliorations apportées :** Étant donné que ce modèle était déjà performant de base, les améliorations ont surtout concerné son **affinage**. Outre le fine-tuning mentionné, on a utilisé les mêmes recettes que précédemment : **callbacks intelligents** (par exemple un Scheduler de learning rate qui baisse le taux d'apprentissage progressivement au fil des époques, en conjonction avec ReduceLROnPlateau), ce qui a aidé à grapiller les derniers points de pourcentage d'accuracy. L'**analyse critique** des résultats a montré qu'en dépit d'un très bon score global, certaines erreurs demeuraient, souvent sur des cas limites (par exemple, des visages d'hommes glabres au visage fin pouvaient être prédits à tort comme femme, et vice

versa, ou bien l'âge de certains visages atypiques mal estimé). Cela ouvre des pistes d'amélioration en travaillant sur des caractéristiques plus fines ou en incorporant par exemple l'attribut « race/ethnicité » du dataset dans un apprentissage multitâche triple, ce qui n'a pas été réalisé ici.

En somme, le modèle 4 a démontré la puissance du transfert d'apprentissage. Lors des tests finaux, il a obtenu une **accuracy genre d'environ 0,92 en test (0,98 en train)** et une **MAE âge d'environ 5,0 ans**. Il s'agit du meilleur modèle en ce qui concerne la prédiction de l'âge (erreur la plus basse) et le second meilleur pour le genre (juste derrière le modèle 1 de quelques points de pourcentage). La section suivante détaille et compare l'ensemble des résultats obtenus par nos quatre modèles.

# Résultats

Après entraînement de tous les modèles, nous avons évalué leurs performances respectives sur le **jeu de test** mis de côté (environ 10% des données, images jamais vues pendant l'entraînement). Le tableau ci-dessous résume les principaux résultats obtenus pour chaque modèle, en indiquant la performance sur la classification du genre (accuracy) et sur la prédiction de l'âge (erreur MAE). Entre parenthèses est rappelée l'accuracy ou l'erreur obtenue sur le jeu d'entraînement, afin d'indiquer le degré de surapprentissage éventuel :

| Modèle | Architecture / Base | Accuracy (Genre) | MAE (Âge) |
|---|---|---|---|
| **Modèle 1** (CNN scratch) | CNN 4 conv + 1 dense | **0,94** (train 0,95) | – (pas de sortie âge) |
| **Modèle 2** (CNN scratch) | CNN 2 conv + GAP + 1 dense | ~0,83 (train 0,90) | **7 ans** |
| **Modèle 3** (CNN double sortie) | CNN 4 conv + 2 dense (partagées) | **0,90** (train 0,97) | **5,5 ans** |
| **Modèle 4** (MobileNetV2 transféré) | MobileNetV2 pré-entraîné + 1 dense | **0,92** (train 0,98) | **5,0 ans** |

Ces résultats appellent plusieurs commentaires. Tout d'abord, **la prédiction du genre est une tâche que les CNN résolvent très bien** sur ce dataset : les quatre approches

atteignent entre 90% et 94% d'accuracy en test, ce qui est excellent. Le modèle 1, spécialement dédié au genre, obtient la meilleure accuracy (0,94) – cela était attendu car il peut consacrer 100% de sa capacité à cette tâche. Le modèle 3 (multitâche) a un score un peu inférieur (0,90), probablement parce qu'il doit partager ses ressources avec la prédiction de l'âge, et qu'un léger compromis s'opère. On constate notamment qu'il a **surappris** sur le genre (0,97 en train vs 0,90 en test), signe qu'il aurait pu bénéficier d'une régularisation encore plus forte ou d'un entraînement plus court pour le genre. Le modèle 4 atteint 0,92, proche de l'optimum, et son surapprentissage est modéré (0,98 train vs 0,92 test), grâce à l'usage de dropout et au fait que MobileNetV2 étant déjà bien formé, il n'a pas surappris des caractéristiques erronées du jeu de train.

Ensuite, **la prédiction de l'âge s'est révélée beaucoup plus difficile** que celle du genre, comme on pouvait s'y attendre. Le modèle 2, qui ne fait que la régression d'âge, obtient une erreur absolue moyenne d'environ 7 ans. Cela veut dire, par exemple, qu'une personne de 50 ans pourrait être prédite 57 ou 43 ans en moyenne. C'est une erreur assez large en termes relatifs, qui traduit la complexité du problème – un visage de 50 ans peut paraître très jeune ou très vieux selon les individus, ce qui brouille les pistes pour le modèle. En revanche, les modèles multitâches (3 et 4) **améliorent sensiblement la performance de l'estimation d'âge**, avec des MAE de 5,5 ans et 5,0 ans respectivement. Gagner près de 2 ans d'erreur moyenne est notable. Le modèle 3 montre qu'en apprenant simultanément le genre, le réseau a pu affiner des caractéristiques utiles à l'âge (par exemple la texture de la peau, les rides, peuvent être corrélées au genre et mieux détectées grâce à l'objectif genre). Quant au modèle 4, il bénéficie énormément du transfert : son MAE de 5,0 ans est le plus faible, ce qui indique qu'en moyenne l'erreur est de 5 ans seulement – pour un humain, deviner l'âge à 5 ans près n'est pas trivial non plus, donc le réseau atteint là une performance très respectable.

Comparativement, on peut dire que **le meilleur compromis global est apporté par le modèle 4 (MobileNetV2)**, puisqu'il offre à la fois une accuracy genre élevée (92%) et la meilleure précision en âge. Il se pourrait qu'un modèle encore plus complexe (comme un ResNet50 multitâche) ferait légèrement mieux, mais le coût en calcul et le risque d'overfitting seraient accrus. MobileNetV2 a offert un **excellent rapport performance/complexité** dans ce projet. Le modèle 1 reste champion sur le genre avec une marge étroite, montrant que pour une application ciblée uniquement sur le genre, un réseau simple et bien réglé suffit et peut être préféré (d'autant qu'il est plus léger à déployer que MobileNetV2).

Pour mieux visualiser les différences, la figure 2 présentée plus haut montrait les courbes d'entraînement du modèle 4 avec une bonne convergence. À l'inverse, la figure 1 montrait les courbes initiales du modèle 3 qui stagnait. Après corrections, les courbes finales du modèle 3 étaient très proches de celles du modèle 4 en allure : on voyait l'accuracy de validation monter vers 0,90 et la perte baisser continûment. Le modèle 2 (âge seul) présentait des courbes de perte âge et MAE assez stables également vers la fin, avec une MAE de validation qui se stabilisait autour de 7. On a noté aussi que l'erreur d'âge du modèle 2 diminuait lentement : même en prolongeant l'entraînement, elle ne descendait pas beaucoup plus bas, suggérant que le modèle était déjà à son optimum compte tenu des informations qu'il pouvait extraire des images.

En complément des métriques globales, nous avons examiné quelques **prédictions qualitatives** en utilisant l'interface Streamlit développée. L'interface permet de charger une image de visage et affiche la prédiction du modèle (genre estimé et âge estimé). Par exemple, sur la photo d'un homme de 30 ans, le modèle 4 a prédit « Homme, 32 ans », ce qui est très proche de la réalité. Pour une femme d'une soixantaine d'années, la prédiction a pu être « Femme, 54 ans » (une sous-estimation d'une dizaine d'années, un cas où le modèle a été trompé possiblement par un visage préservé ou un maquillage). Globalement, les erreurs d'âge du modèle 4 étaient souvent inférieures à ±10 ans, ce qui correspond à ce que quantifie la MAE de 5 ans environ (puisque 5 ans en moyenne signifie la plupart du temps dans une fenêtre ±10). Les erreurs de genre étaient rares : sur 50 tests manuels via l'interface, seulement 2 ou 3 confusions ont été observées (par exemple une photo en très forte contre-plongée où le visage était partiellement masqué, et une autre où l'éclairage rendait les traits difficilement discernables).

Enfin, un aspect intéressant à mentionner est le **déploiement sur Hugging Face**. L'application web construite avec Streamlit a été hébergée, rendant l'utilisation du modèle très aisée : il suffit d'ouvrir la page web du Space, de glisser-déposer une image, et le modèle (notamment le modèle 4, qui a été choisi pour le déploiement en raison de ses performances) renvoie en quelques secondes le genre et un âge estimé. Cette phase de déploiement a permis de vérifier la **réactivité et l'efficacité du modèle en conditions réelles**. MobileNetV2 étant relativement léger, l'inférence est rapide (de l'ordre de 100 ms par image sur CPU, bien moins sur GPU). L'interface présente le résultat sous forme textuelle et pourrait être améliorée à l'avenir par l'affichage de barres de probabilité ou d'un intervalle de confiance pour l'âge.

En résumé, les résultats obtenus valident les objectifs du projet : nous avons réussi à **construire un classifieur de genre très fiable** (>90% de réussite) et un **estimateur d'âge automatisé** avec une erreur moyenne de l'ordre de 5 à 7 ans selon les méthodes, le tout à l'aide de réseaux de neurones convolutionnels. Ces performances sont conformes à ce qu'on trouve dans la littérature pour des modèles de complexité similaire sur UTKFace.

# Conclusion

Ce projet de SAE a permis de mettre en œuvre et de comparer différentes approches basées sur les réseaux de neurones convolutionnels pour extraire des informations démographiques à partir d'images de visages. En partant du **dataset UTKFace** et à l'aide des outils Python/TensorFlow/Keras, nous avons entraîné quatre modèles distincts, chacun ayant ses particularités :

- Un modèle CNN simple entraîné from scratch pour la **classification de genre**, qui a démontré qu'avec une architecture adéquate et des techniques de régularisation (BatchNorm, Dropout, etc.), on peut atteindre une **précision de l'ordre de 94%** pour distinguer hommes et femmes sur des images de visage.

- Un modèle CNN de **prédiction d'âge** (from scratch également), qui a souligné la **difficulté du problème de régression d'âge**. Malgré un réseau plus simple pour éviter le surapprentissage, l'erreur moyenne est restée autour de 7 ans. Ce modèle isolé a constitué une base et un point de comparaison utile pour mesurer les gains

apportés par les approches plus complexes.

- Un modèle **multitâche (genre + âge)**, qui a exploité l'**apprentissage conjoint** de ces deux étiquettes. Après avoir surmonté des problèmes de convergence et d'équilibrage, ce modèle a confirmé l'intérêt du multitâche en améliorant la prédiction d'âge (MAE ~5,5 ans) tout en fournissant une bonne classification de genre (~90%). Il illustre comment une tâche peut en aider une autre en partageant les connaissances, phénomène connu sous le nom d'**apprentissage inductif mutuel**.

- Un modèle basé sur le **transfert d'apprentissage (MobileNetV2)**, qui s'est révélé le plus performant globalement. En tirant parti de connaissances préexistantes apprises sur un large corpus d'images génériques, il a atteint la meilleure précision pour l'âge (MAE ~5,0) et une excellente performance en genre (~92%). Ce modèle montre qu'il est souvent très avantageux de **réutiliser des modèles pré-entraînés** pour gagner en efficacité, surtout quand on dispose de quantités de données d'entraînement limitées.

Au-delà des aspects techniques propres à chaque modèle, ce projet nous a sensibilisés à l'importance des **méthodes de régularisation et de validation** en apprentissage automatique. Nous avons pu expérimenter concrètement l'effet du taux d'apprentissage, de l'early stopping, du dropout, de la normalisation, etc., sur la qualité finale du modèle. À plusieurs reprises, nous avons dû investiguer des problèmes de surapprentissage ou de convergence non triviale, et adapter en conséquence l'architecture ou les hyperparamètres. Ce travail itératif fait partie intégrante du développement de modèles de deep learning et nous a permis de développer une meilleure intuition sur « comment entraîner un CNN efficacement ».

En termes de gestion de projet, le fait de diviser le travail entre quatre approches complémentaires s'est avéré fructueux. Chaque membre de l'équipe a pu explorer en autonomie un pan de la problématique, tout en partageant régulièrement les difficultés et solutions trouvées. Par exemple, les leçons tirées du modèle 1 (genre) sur la BN et le dropout ont directement profité aux modèles 3 et 4. Inversement, le modèle 3 a bénéficié des trouvailles du modèle 2 pour l'évaluation de la MAE et la gestion des âges. Cette synergie a enrichi l'apprentissage de chacun et abouti à un ensemble cohérent de solutions.

Le **déploiement sur Hugging Face** a été une étape très satisfaisante, car elle a transformé nos modèles entraînés en une application concrète utilisable par d'autres. Cela nous a initiés aux aspects pratiques de la mise en production d'un modèle : simplicité d'utilisation via une interface, nécessité de performances suffisantes (d'où le choix de MobileNetV2 optimisé), etc.

En conclusion, les objectifs initiaux du projet sont atteints. Nous avons construit un système capable de **prédire automatiquement le genre et l'âge** à partir d'une photo de visage avec une bonne fiabilité. Ce système pourrait avoir des applications multiples, par exemple pour des études démographiques automatisées sur des collections d'images, ou comme brique de base dans des applications interactives nécessitant une estimation de l'âge (contrôle parental, adaptation de contenu en fonction de l'utilisateur, etc.). Bien sûr, il convient de

souligner les **limites** et les pistes d'amélioration : par exemple, la précision d'estimation d'âge pourrait être encore affinée en regroupant l'estimation par tranches d'âge pour éliminer l'effet des outliers, ou en utilisant des architectures plus spécialisées pour l'âge (certaines recherches utilisent des approches par régression ordinale ou par classification fine par année). De même, l'ajout de données ou l'entraînement sur des datasets supplémentaires (comme le dataset Adience pour l'âge) pourrait améliorer la robustesse du modèle sur des visages en conditions très variées.

Une autre dimension non exploitée est l'**information d'ethnicité/race** présente dans UTKFace – un modèle multitâche à trois sorties (genre, âge, ethnie) aurait pu être tenté. Ce pourrait être un prolongement intéressant pour voir si prédire aussi l'origine ethnique aide les autres tâches (par exemple l'âge, puisque le vieillissement peut présenter des différences selon l'origine ethnique). Enfin, du point de vue éthique, il faut rappeler que prédire l'âge et le genre à partir d'une image soulève des questions sur l'utilisation qui en est faite (respect de la vie privée, risque de biais). Notre projet est resté dans un cadre académique neutre, mais toute application réelle devrait intégrer une réflexion sur ces aspects.

En somme, ce projet nous a permis d'appliquer concrètement nos connaissances en **vision par ordinateur** et en **deep learning**, de la préparation des données jusqu'au déploiement web, en passant par le choix et l'optimisation de modèles. Les résultats obtenus sont encourageants et nous ont permis de mieux comprendre les forces et faiblesses des CNN pour l'analyse automatique des visages. Ce rapport documente le chemin parcouru et servira de référence pour d'éventuels futurs travaux dans ce domaine, comme par exemple améliorer encore la précision de l'estimation d'âge ou adapter ces modèles à des vidéos en temps réel. Nous retenons de cette expérience qu'avec des données de qualité et des modèles bien construits, des tâches perceptives complexes comme estimer l'âge d'une personne peuvent être abordées avec succès par des intelligences artificielles entraînées.