

Rapport de PTA

Filière : Ingénierie en Systèmes numériques

Niveau : 3^{ème} année

Sujet :

Détection du genre et de l'âge

Encadré par : Mme Maya KALLAS, Mr Steven LE CAM

Elaboré par : Meriam JARDAK

Année Universitaire : 2023-2024

Remerciements

Je tiens à exprimer mes plus sincères remerciements à mes deux encadrants, Madame Maya KALLAS et Monsieur Steven LE CAM, pour leur précieuse aide et leur soutien inestimable tout au long de ces mois consacrés à l'élaboration de ce travail. Leur expertise, leurs conseils éclairés et leur dévouement ont grandement contribué à la réussite de ce projet. Je suis reconnaissante de la chance d'avoir pu bénéficier de leur guidance. Merci infiniment pour votre engagement et votre professionnalisme, qui ont rendu cette expérience enrichissante et formatrice.

Résumé : Ce rapport explore les applications de l'Intelligence Artificielle (IA) dans la détection de l'âge et du genre, en utilisant des Réseaux Neuraux Convolutifs (CNN). En abordant les défis de la détection, l'étude analyse le jeu de données Adience, comprenant 12 000 images faciales diverses, mettant en avant le rôle crucial des Réseaux Neuraux pour assurer l'uniformité. Le prétraitement du jeu de données construit un dataframe informé par CNN avec les identifiants faciaux, les références des photos, les âges et les genres, formant la base des analyses ultérieures. L'extraction de caractéristiques, guidée par les CNN, fournit des références essentielles pour l'apprentissage approfondi. L'architecture, cruciale pour l'apprentissage, implique deux approches : des CNN séparés pour l'âge et le genre, et un CNN ensembliste unifié pour une détection simultanée. Les deux subissent des phases distinctes d'entraînement et de test pour une évaluation robuste. Les résultats montrent un seuil de performance de 92% pour la première solution et 77% pour la deuxième solution d'apprentissage informées par CNN, mettant en évidence l'efficacité des réseaux neuronaux dans la détection de l'âge et du genre. Des recherches ultérieures devraient prioriser l'optimisation des CNN et des paramètres pour atteindre une performance de 100%, assurant des avancées continues dans les solutions d'IA pour la détection de l'âge et du genre.

Mots-clés : Détection de l'âge et du genre, CNN, Jeu de données Adience, Phases d'entraînement et de test, Seuil de performance.

Abstract: This report explores Artificial Intelligence (AI) applications in Age and Gender Detection, using Convolutional Neural Networks (CNNs). Addressing detection challenges, the study analyzes the Adience dataset, housing 12,000 diverse facial images, underscoring the Neural Networks' crucial role for uniformity. Preprocessing the dataset constructs a CNN-informed dataframe with face IDs, photo references, ages, and genders, forming the foundation for subsequent analyses. Feature extraction, guided by CNNs, provides essential references for deep learning. The architecture, pivotal for learning, involves two approaches: separate CNNs for age and gender and a unified ensemble CNN for concurrent detection. Both undergo unique training and testing phases for robust evaluation. Results show a 92% and 77% performance threshold for both CNN-informed learning solutions, highlighting Neural Networks' effectiveness in age and gender detection. Further research should prioritize optimizing CNNs and parameters for 100% performance, ensuring continual advancements in AI solutions for age and gender detection.

Keywords: Age and Gender Detection, CNN, Adience Dataset, Training and Testing Phases, Performance Threshold.

Table des matières

Chapitre 1 : Introduction

- 1.1 Contexte du projet
- 1.2 Objectif
- 1.3 Plan d'étude

Chapitre 2 : Méthodologie de Travail

- 2.1 Organisation des réunions et gestion du projet
- 2.2 Environnement et outils de Travail
- 2.3 Base de données Utilisée
- 2.4 Bibliothèques Utilisées

Chapitre 3 : Étude de la base de données

- 3.1 Préparation des données
- 3.2 Analyse des données

Chapitre 4 : Réseaux de neurones convolutifs

- 4.1 Présentation des CNNs
- 4.2 Extraction des caractéristiques

Chapitre 5 : Première solution - Deux réseaux de neurones séparés

- 5.1 Description de l'architecture pour la détection de l'âge et son entraînement
- 5.2 Description de l'architecture pour la détection du genre et son entraînement
- 5.3 Phase de Test
- 5.4 Présentation des résultats obtenus

Chapitre 6 : Deuxième solution - Un seul réseau de neurones ensembliste

- 6.1 Description de l'architecture pour la détection de l'âge et de genre et son entraînement
- 6.2 Présentation des résultats obtenus

6.3 Phase de Test

Chapitre 7 : Évaluation de la solution et Problèmes rencontrés

7.1 Discussion des deux solutions

7.2 Problèmes rencontrés

Chapitre 8 : Conclusion et Perspectives

8.1 Conclusion

8.2 Perspectives

Chapitre 9 : Bibliographie et Annexes

Liste des Figures

Figure 1 : Environnement de Travail “Kaggle”

Figure 2 : Base de données “Adience”

Figure 3 : Bibliothèques Utilisées

Figure 4 : Format fichier age_train.txt

Figure 5 : Dataframe pour l’âge “df_age”

Figure 6 : Dataframe global “df_total”

Figure 7 : Dataframe global “df_total_val”

Figure 8 : Distribution de l’âge

Figure 9 : Distribution du genre

Figure 10 : Analyse visuelle de la base des données

Figure 11 : Conversion des étiquettes aux classes d’âge

Figure 12 : l’architecture du modèle construit pour l’âge

Figure 13 : les epochs pour l'entraînement du modèle de l’âge

Figure 14 : l’architecture du modèle construit pour le genre

Figure 15 : les epochs pour l'entraînement du modèle genre

Figure 16 : Enregistrement des modèles après entraînement

Figure 17 : La précision pour les données d'entraînement et de validation

Figure 18 : La perte pour les données d'entraînement et de validation

Figure 19 : Distribution de l’âge pour les données de Test

Figure 20 : Distribution de genre pour les données de Test

Figure 21 : Architecture du modèle ensembliste

Figure 22 : Les epochs d'entraînement

Figure 23 : La précision pour les données d'entraînement et de validation

Figure 24 : La perte pour les données d'entraînement et de validation

Chapitre 1 : Introduction du projet

1.1 Contexte du projet

L'analyse faciale à partir d'images a gagné en popularité en raison de ses applications étendues, telles que le ciblage publicitaire, les systèmes de recommandation, la surveillance et la sécurité.

La détection du genre et de l'âge constitue une composante cruciale de cette analyse, permettant une meilleure personnalisation des expériences des utilisateurs.

1.2 Objectif

L'objectif de ce projet est d'explorer les possibilités offertes par les réseaux de neurones convolutifs (CNN) pour la détection du genre et de l'âge à partir d'images faciales. Nous visons à développer des solutions efficaces en utilisant deux approches distinctes et à évaluer leurs performances.

1.3 Plan d'étude

Notre plan d'étude est structuré de manière séquentielle, guidant le lecteur à travers les différentes étapes de notre projet. Nous débutons avec une exploration approfondie de l'environnement de travail, mettant en lumière les outils employés et notre choix de plateforme principale, ainsi que la base de données utilisée.

Cette immersion préliminaire offre un contexte technologique essentiel à la compréhension du projet. Ensuite, nous plongeons dans la préparation minutieuse des données, soulignant les étapes cruciales pour garantir la qualité et l'uniformité du jeu de données.

Nous introduisons également les réseaux de neurones convolutifs (CNN), soulignant leur rôle central dans notre approche. Ces fondations sont ensuite mises en œuvre dans deux solutions distinctes, chacune détaillant l'architecture, les phases d'entraînement et de test, ainsi que les résultats obtenus. L'évaluation comparative des solutions et la discussion des problèmes rencontrés sont ensuite traitées.

Enfin, la conclusion récapitule les points essentiels de notre étude et propose des perspectives pour des recherches futures, permettant une transition fluide entre chaque étape du rapport.

Chapitre 2 : Méthodologie de Travail

2.1 Organisation des réunions et gestion du projet

Dans le cadre de ce projet de détection d'âge et de genre, un processus de supervision et de suivi régulier a été mis en place avec mes deux encadrants, Mme Maya Kallas et M. Steven Le Cam. Les réunions ont lieu environ toutes les deux semaines à l'ENSEM.

Ces réunions étaient cruciales pour valider l'avancement du travail accompli et pour discuter des problèmes rencontrés. Chaque séance durait en moyenne 45 minutes, offrant un temps suffisant pour examiner en détail les différentes étapes du projet, les résultats obtenus et les défis rencontrés.

Lors de ces réunions, nous avons passé en revue les tâches réalisées depuis la dernière rencontre et discuté des prochaines étapes à suivre. Les retours d'expérience de mes encadrants ont été particulièrement précieux pour résoudre les problèmes techniques et méthodologiques rencontrés tout au long du projet.

En outre, ces réunions ont également permis d'ajuster et de réorienter, si nécessaire, la direction du projet en fonction des nouveaux développements ou des recommandations des encadrants.

2.2 Environnement et outils de Travail

Notre projet repose sur **Kaggle** en tant que plateforme principale, un choix motivé par ses avantages en termes de ressources, de collaboration et de bibliothèques préinstallées essentielles pour le développement de solutions en apprentissage profond. Tout au long du projet, nous avons opté pour le langage de programmation **Python** en raison de sa flexibilité, de sa richesse en bibliothèques dédiées à l'apprentissage profond.

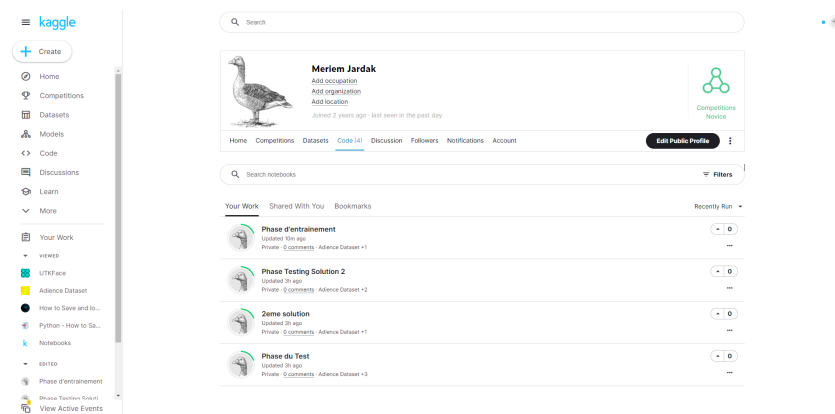


Figure 1: Environnement de Travail “Kaggle”

2.3 Base de données Utilisée

La base de données fondamentale sous-jacente à notre étude est la **base Adience**, et plus spécifiquement, le **fold numéro 3** a été sélectionné pour notre analyse. Ce fold contient **11,175 données** relatives à l'âge et **12,854 données** concernant le genre, capturées sous différents angles et représentant diverses personnes. Ce choix stratégique sert à garantir une diversité suffisante dans les données, essentielle pour entraîner des modèles de manière robuste.

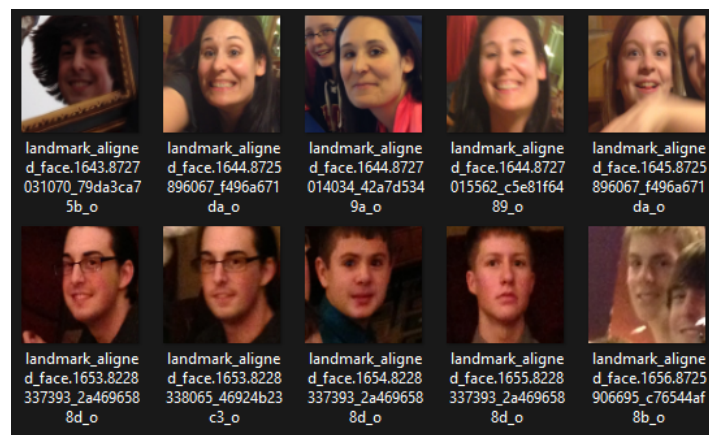


Figure 2: Base de données “Adience”

2.4 Bibliothèques Utilisées

Pour la mise en œuvre de notre première solution, nous avons utilisé un ensemble de bibliothèques essentielles pour le traitement des données, la création de modèles de réseaux de neurones, et l'évaluation des résultats. Notamment, nous avons fait appel à **numpy** et **pandas** pour la manipulation des données, **cv2** pour le traitement des images, et **matplotlib** ainsi que **seaborn** pour la visualisation des données.

En ce qui concerne la création du modèle, nous avons exploité les fonctionnalités de **tensorflow** et **keras**, en utilisant des couches telles que **Dense**, **Conv2D**, et **Flatten**. L'optimisation du modèle a été réalisée avec l'optimiseur **Adam**. De plus, nous avons employé **LabelEncoder** et **to_categorical** de **scikit-learn** pour la gestion des labels.

```
import numpy as np
import pandas as pd
import cv2
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
from tqdm.notebook import tqdm
warnings.filterwarnings('ignore')
%matplotlib inline
from keras.preprocessing.image import load_img
from keras.models import Sequential, Model
from keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPooling2D, Input
from keras.optimizers import Adam
from sklearn.preprocessing import LabelEncoder
from keras.utils import to_categorical
from PIL import Image
```

Figure 3 : Bibliothèques Utilisées

Ces bibliothèques ont été sélectionnées pour leur adaptabilité aux tâches spécifiques du projet, garantissant une mise en œuvre robuste et efficiente des solutions proposées.

Chapitre 3 : Étude de la base de données

3.1 Préparation des données

La préparation des données a été entreprise avec rigueur afin de créer des dataframes structurés à partir des informations extraites des fichiers texte disponibles dans la **base de données Adience**. Nous avons commencé par la création de deux dataframes distincts pour gérer les informations relatives à l'âge et au genre des individus représentés dans les images.

Pour chaque image, nous avons extrait **le répertoire, la référence de l'image** et le **face_id**, qui servent d'identifiants uniques pour chaque visage dans la base de données. En parallèle, nous avons récupéré **les classes d'âge** et de **genre** associées à chaque image à partir des fichiers texte correspondants. Ces informations ont été soigneusement structurées dans les dataframes, comprenant des colonnes pour **le répertoire, la référence de l'image, le face_id**, ainsi que **l'âge** ou **le genre** de chaque individu.

```
114841417@N06/landmark_aligned_face.488.12059875396_f5c3a70550_o.jpg 3
28754132@N06/landmark_aligned_face.612.9367256464_38d751ea5c_o.jpg 5
10792106@N03/landmark_aligned_face.532.11080871614_ee39c8d59e_o.jpg 3
30872264@N00/landmark_aligned_face.603.9206683814_2fe1c35b50_o.jpg 5
62501130@N02/landmark_aligned_face.97.11305791453_0e76106203_o.jpg 0
```

Figure 4 : Format fichier age_train.txt

	Répertoire	Reference_image	Face_id	Age
0	114841417@N06	landmark_aligned_face.488.12059875396_f5c3a705...	488	3
1	28754132@N06	landmark_aligned_face.612.9367256464_38d751ea5...	612	5
2	10792106@N03	landmark_aligned_face.532.11080871614_ee39c8d5...	532	3
3	30872264@N00	landmark_aligned_face.603.9206683814_2fe1c35b5...	603	5
4	62501130@N02	landmark_aligned_face.97.11305791453_0e7610620...	97	0

Figure 5 : Dataframe pour l'âge "df_age"

Il convient de noter que la base de données Adience utilise un système de classement des âges en **8 catégories, numérotées de 0 à 7**, en fonction des tranches d'âge des individus représentés. De même, les classes de genre sont codées en tant que **0 pour les hommes et 1 pour les femmes**.

La préparation des données a également impliqué la création d'un dataframe global consolidant les informations sur l'âge et le genre de chaque individu représenté dans les images. À cette fin, nous avons combiné les données extraites des dataframes individuels relatifs à l'âge et au genre, en utilisant le **face_id** comme clé de fusion. Ce processus a permis de créer un dataframe global, appelé **df_total**, contenant toutes les informations pertinentes pour notre étude.

	Répertoire	Reference_image	Face_id	Age	Gender
0	10280355@N07	landmark_aligned_face.1842.11367072686_307323b...	1842	6	1
1	64504106@N06	landmark_aligned_face.912.11813883246_a6cd49ff...	912	5	0
2	8147776@N04	landmark_aligned_face.2263.9237730868_93f71f02...	2263	3	1
3	40410287@N02	landmark_aligned_face.1259.11554934376_5562949...	1259	4	1
4	20632896@N03	landmark_aligned_face.555.9826157586_43d0ca0d5...	555	3	1
5	115021413@N07	landmark_aligned_face.952.12059491173_ab67a08c...	952	2	1
6	8187011@N06	landmark_aligned_face.989.8728378639_60aa6a744...	989	1	1
7	113525713@N07	landmark_aligned_face.1048.11784469696_63969f5...	1048	3	0
8	16886060@N03	landmark_aligned_face.1911.10393595085_ff05342...	1911	4	1
9	115126086@N07	landmark_aligned_face.1144.12076432206_5e32465...	1144	7	1

Figure 6 : Dataframe global “df_total”

De plus, nous avons fait le même processus de préparation des données pour les données de validation et de test . Nous avons veillé à maintenir un équilibre entre les deux ensembles pour garantir une représentation adéquate des différentes classes d'âge et de genre.

	Répertoire	Reference_image	Face_id	Age	Gender
0	10280355@N07	landmark_aligned_face.1848.9496769730_46a8129a...	1848	6	1
1	31183835@N08	landmark_aligned_face.2095.8754328439_3011d832...	2095	4	0
2	115321157@N03	landmark_aligned_face.1747.12112823904_5b6678b...	1747	4	0
3	115321157@N03	landmark_aligned_face.1744.12113088635_400af32...	1744	1	0
4	113528649@N08	landmark_aligned_face.335.11777308434_e08b6f01...	335	1	0

Figure 7 : Dataframe global “df_total_val”

Pour garantir la cohérence des données, nous avons effectué des étapes de nettoyage et de normalisation. Cela inclut le redimensionnement des images pour une taille uniforme. La taille initiale des images est **(816, 816,3)** qui est devenue **(128,128,3)**.

Cette phase de préparation des données était cruciale pour créer un jeu de données de haute qualité, prêt à être utilisé dans la construction et l'évaluation des modèles de détection du genre et de l'âge à partir des images faciales.

3.2 Analyse des données

La préparation des données constitue une étape fondamentale pour assurer la qualité et l'homogénéité des données utilisées dans notre projet. Nous avons fait ensuite une exploration approfondie de la base de données Adience, en mettant l'accent sur **le fold numéro 3**, qui contient **11,175 données** pour l'âge et **12,854 données pour le genre** de différentes personnes, prises sous différents angles.

Pour garantir la représentativité des données, nous avons pris la décision de retirer **aléatoirement 50%** des lignes associées à la classe d'âge 4, tout en préservant l'équilibre global de la distribution des classes d'âge dans le jeu de données.

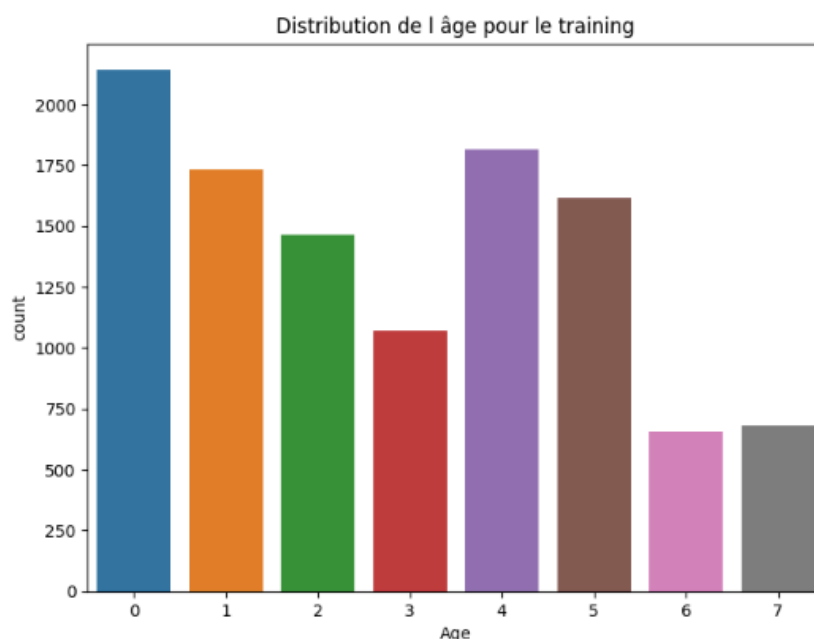


Figure 8 : Distribution de l'âge

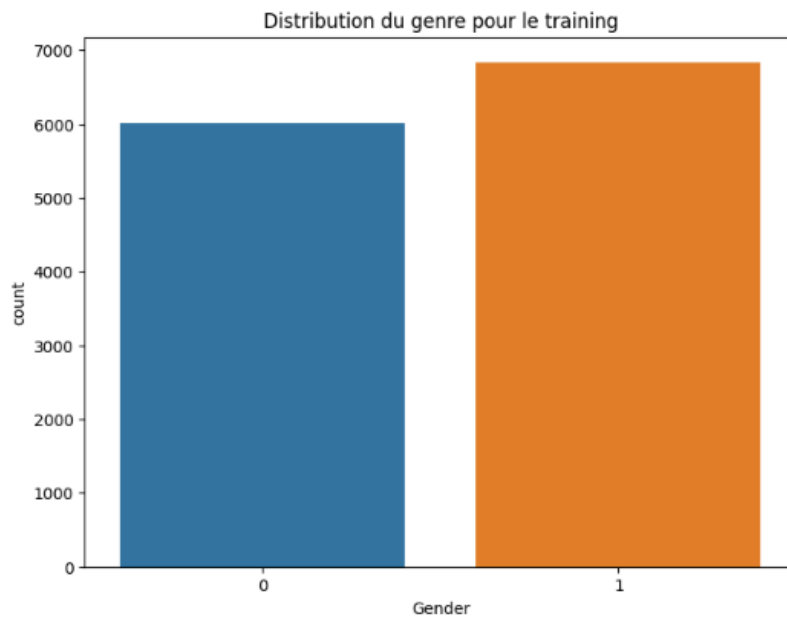


Figure 9 : Distribution du genre

Pour mieux comprendre la distribution des classes d'âge et de genre dans notre jeu de données, nous avons procédé à une analyse visuelle en affichant un échantillon d'images avec leurs étiquettes d'âge et de genre correspondantes. L'utilisation de cette représentation graphique nous a permis d'avoir un aperçu intuitif de la diversité des visages présents dans notre ensemble de données.

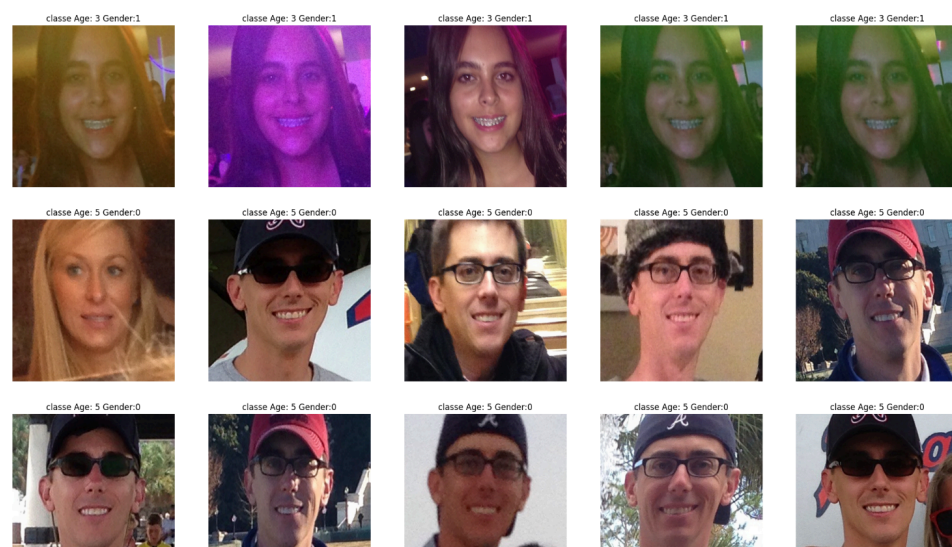


Figure 10 : Analyse visuelle de la base des données

Chapitre 4 : Réseaux de neurones convolutifs

4.1 Présentation des CNN

Les réseaux de neurones convolutifs (CNN) constituent une architecture fondamentale dans le domaine de la vision par ordinateur, reconnue pour son efficacité dans la reconnaissance et l'analyse d'images. Inspirés par le fonctionnement du cortex visuel humain, les CNNs sont caractérisés par l'utilisation de **couches de convolution** pour extraire des caractéristiques pertinentes des images en entrée.

Ces couches de convolution sont suivies de **couches de pooling** qui réduisent la dimensionnalité des caractéristiques extraites tout en préservant leur information discriminante. Enfin, **des couches entièrement connectées** sont utilisées pour classer les caractéristiques extraites en différentes catégories.

Dans le chapitre suivant, nous explorerons en détail l'architecture et le fonctionnement des CNNs utilisés pour les deux solutions proposées, mettant en évidence leur utilisation dans notre projet pour la détection du genre et de l'âge à partir d'images faciales.

Nous décrirons également les différentes couches qui composent les CNNs et expliquerons comment elles contribuent à la capacité des réseaux à apprendre des caractéristiques visuelles importantes à partir des données d'entrée.

4.2 Extraction des caractéristiques

Dans le processus de préparation des données pour l'entraînement de notre modèle de détection du genre et de l'âge à partir d'images faciales, une étape cruciale consiste à extraire les caractéristiques des images.

Cette extraction de caractéristiques est réalisée en redimensionnant chaque image à une taille de **128x128 pixels** et en les convertissant en tableaux numpy, afin de les rendre compatibles avec notre modèle de réseau de neurones convolutifs (CNN).

La fonction **extract_features** a été développée pour effectuer cette tâche, Cette fonction parcourt chaque image et charge son chemin d'accès spécifié. Ensuite, chaque image est redimensionnée à une taille de **128x128 pixels** à l'aide de la méthode **resize** de la bibliothèque **PIL**. Après cela, chaque image est convertie en un tableau numpy et ajoutée à une liste de caractéristiques. Une fois toutes les images traitées, la liste des caractéristiques est convertie en un tableau numpy bidimensionnel de dimensions **(nombre_d'images, 128, 128, 3)**, où **3** représente les trois canaux de couleur (**rouge, vert, bleu**).

Les données d'entraînement, de validation et de test ont ensuite été préparées en utilisant cette fonction. Une fois les caractéristiques extraites, elles ont été normalisées en divisant les valeurs de **chaque pixel par 255**, Cette normalisation garantit que les valeurs des pixels des images sont mises à l'échelle dans une plage de **[0, 1]**, ce qui facilite l'entraînement du modèle en réduisant la variance des données.

Chapitre 5 : Première solution - Deux réseaux de neurones séparés

5.1 Description de l'architecture pour la détection de l'âge et son entraînement

Avant de concevoir l'architecture pour la détection de l'âge, nous avons procédé à une conversion des étiquettes aux classes d'âge. Cette conversion est cruciale étant donné que notre modèle aura une sortie à huit neurones, correspondant aux huit classes d'âge. Tout d'abord, nous avons utilisé l'outil **"LabelEncoder"** pour mapper chaque classe d'âge à un entier unique. Cette étape permet de convertir les étiquettes d'âge en un format numérique, requis pour l'entraînement du modèle.

Ensuite, afin de réaliser une classification multiclasse à l'aide d'une sortie **softmax**, les étiquettes d'âge ont été converties en format catégoriel à l'aide de la fonction **"to_categorical"**. Cette opération transforme chaque étiquette en un vecteur binaire où une seule dimension contient **une valeur 1**, indiquant la classe correspondante, tandis que toutes les autres dimensions contiennent des zéros.

Ces étapes de conversion assurent que nos étiquettes d'âge sont préparées et prêtes à être utilisées comme cibles d'entraînement pour notre modèle de détection de l'âge à partir d'images faciales.

```
label_encoder = LabelEncoder()
y_train_age_encoded = label_encoder.fit_transform(y_age_train)
y_val_age_encoded = label_encoder.fit_transform(y_age_val)

y_train_age_categorical = to_categorical(y_train_age_encoded, num_classes=8)
y_val_age_categorical = to_categorical(y_val_age_encoded, num_classes=8)
```

Figure 11 : Conversion des étiquettes aux classes d'âge

Passons maintenant à l'architecture de Réseau de neurones. Le modèle a été construit en utilisant l'**API Sequential de Keras**, qui permet de définir les couches du réseau de manière séquentielle.

Ce modèle est composé de plusieurs couches :

- Trois **couches de convolution** avec des filtres de **taille 3x3** et des fonctions d'activation **ReLU**, suivies de couches de max pooling pour réduire la dimensionnalité des données.
- Une couche de **flatten** pour convertir les données en un vecteur unidimensionnel.
- Une couche **dense** de **256 neurones** avec une fonction d'activation **ReLU**, suivie d'une couche de **dropout** avec un taux de régularisation de **30%** pour réduire le surapprentissage.
- Enfin, une couche **dense** de sortie avec **8 neurones** et une activation **softmax** pour la **classification multiclasse** des classes d'âge.

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 126, 126, 16)	448
max_pooling2d (MaxPooling2D)	(None, 63, 63, 16)	0
conv2d_1 (Conv2D)	(None, 61, 61, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 32)	0
conv2d_2 (Conv2D)	(None, 28, 28, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 64)	0
flatten (Flatten)	(None, 12544)	0
dense (Dense)	(None, 256)	3211520
dropout (Dropout)	(None, 256)	0
age_out (Dense)	(None, 8)	2056

```
=====  
Total params: 3237160 (12.35 MB)  
Trainable params: 3237160 (12.35 MB)  
Non-trainable params: 0 (0.00 Byte)
```

Figure 12 : l'architecture du modèle construit pour l'âge

Le modèle a été compilé avec l'optimiseur **Adam** et la fonction de perte **categorical_crossentropy**, adaptée à une tâche de classification multiclasse.

La métrique d'évaluation de la performance du modèle est l'**accuracy** (précision).

Le modèle a été entraîné sur les données d'entraînement pendant **10 epochs** avec un batch size de **32**, et sa performance a été évaluée sur les données de validation.

```

Epoch 1/10
318/318 [=====] - 74s 229ms/step - loss: 1.8180 - accuracy: 0.2792 - val_loss: 1.4995 - val_accuracy: 0.4165
Epoch 2/10
318/318 [=====] - 68s 215ms/step - loss: 1.3808 - accuracy: 0.4604 - val_loss: 1.2886 - val_accuracy: 0.5255
Epoch 3/10
318/318 [=====] - 69s 216ms/step - loss: 1.1070 - accuracy: 0.5795 - val_loss: 1.1978 - val_accuracy: 0.5536
Epoch 4/10
318/318 [=====] - 70s 221ms/step - loss: 0.8679 - accuracy: 0.6754 - val_loss: 1.1290 - val_accuracy: 0.6081
Epoch 5/10
318/318 [=====] - 71s 222ms/step - loss: 0.6647 - accuracy: 0.7556 - val_loss: 1.1379 - val_accuracy: 0.6028
Epoch 6/10
318/318 [=====] - 71s 225ms/step - loss: 0.4621 - accuracy: 0.8308 - val_loss: 1.2885 - val_accuracy: 0.6274
Epoch 7/10
318/318 [=====] - 71s 223ms/step - loss: 0.3470 - accuracy: 0.8762 - val_loss: 1.4089 - val_accuracy: 0.6178
Epoch 8/10
318/318 [=====] - 70s 222ms/step - loss: 0.2493 - accuracy: 0.9122 - val_loss: 1.5638 - val_accuracy: 0.6318
Epoch 9/10
318/318 [=====] - 71s 224ms/step - loss: 0.2111 - accuracy: 0.9239 - val_loss: 1.6562 - val_accuracy: 0.6151
Epoch 10/10
318/318 [=====] - 71s 223ms/step - loss: 0.1705 - accuracy: 0.9412 - val_loss: 1.7371 - val_accuracy: 0.6213

```

Figure 13 : les epochs pour l'entraînement du modèle de l'âge

5.2 Description de l'architecture pour la détection du genre et son entraînement

Pour la détection du genre à partir d'images faciales, nous avons développé un modèle de réseau de neurones convolutifs (CNN) avec une architecture qui utilise aussi des couches convolutionnelles et des couches entièrement connectées pour extraire les caractéristiques des images et prédire le genre de chaque personne.

Ce modèle comprend plusieurs couches :

- Trois couches convolutionnelles avec **des filtres de taille 3x3** et des fonctions **d'activation ReLU**, suivies de couches de **max pooling** pour réduire la dimensionnalité des données.
- Une couche de **flatten** pour convertir les données en un vecteur unidimensionnel.
- Une couche dense avec **75 neurones** et une **fonction d'activation ReLU**.
- Une couche de **dropout avec un taux de régularisation de 30%** pour réduire le surapprentissage.
- Enfin, une couche **dense de sortie avec un seul neurone** et une activation **sigmoid** pour la classification binaire du genre.

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 128, 128, 3)]	0
conv2d_3 (Conv2D)	(None, 126, 126, 16)	448
max_pooling2d_3 (MaxPooling2D)	(None, 63, 63, 16)	0
conv2d_4 (Conv2D)	(None, 61, 61, 32)	4640
max_pooling2d_4 (MaxPooling2D)	(None, 30, 30, 32)	0
conv2d_5 (Conv2D)	(None, 28, 28, 64)	18496
max_pooling2d_5 (MaxPooling2D)	(None, 14, 14, 64)	0
flatten_1 (Flatten)	(None, 12544)	0
dense_1 (Dense)	(None, 75)	940875
dropout_1 (Dropout)	(None, 75)	0
gender_out (Dense)	(None, 1)	76

=====
Total params: 964535 (3.68 MB)
Trainable params: 964535 (3.68 MB)
Non-trainable params: 0 (0.00 Byte)

Figure 14 : l'architecture du modèle construit pour le genre

Le modèle a été compilé avec la fonction de perte '**binary_crossentropy**' et l'optimiseur '**adam**', adaptés à une tâche de **classification binaire du genre**. La métrique d'évaluation de la performance du modèle est l'**accuracy (précision)**.

Le modèle a ensuite été entraîné sur les données d'entraînement pendant **8 epochs** avec un **batch size de 32**, et sa performance a été évaluée sur les données de validation.

```
Epoch 1/8
318/318 [=====] - 72s 221ms/step - loss: 0.5800 - accuracy: 0.6908 - val_loss: 0.4931 - val_accuracy: 0.7531
Epoch 2/8
318/318 [=====] - 66s 208ms/step - loss: 0.4395 - accuracy: 0.7935 - val_loss: 0.4215 - val_accuracy: 0.8128
Epoch 3/8
318/318 [=====] - 66s 208ms/step - loss: 0.3735 - accuracy: 0.8305 - val_loss: 0.3574 - val_accuracy: 0.8436
Epoch 4/8
318/318 [=====] - 66s 209ms/step - loss: 0.3099 - accuracy: 0.8630 - val_loss: 0.3401 - val_accuracy: 0.8445
Epoch 5/8
318/318 [=====] - 66s 208ms/step - loss: 0.2617 - accuracy: 0.8869 - val_loss: 0.3915 - val_accuracy: 0.8366
Epoch 6/8
318/318 [=====] - 66s 207ms/step - loss: 0.2250 - accuracy: 0.9038 - val_loss: 0.3145 - val_accuracy: 0.8664
Epoch 7/8
318/318 [=====] - 66s 207ms/step - loss: 0.1799 - accuracy: 0.9275 - val_loss: 0.3331 - val_accuracy: 0.8770
Epoch 8/8
318/318 [=====] - 66s 209ms/step - loss: 0.1469 - accuracy: 0.9407 - val_loss: 0.3365 - val_accuracy: 0.8656
```

Figure 15 : les epochs pour l'entraînement du modèle genre

```
model.save('modele_age.h5')
model_gender.save('modele_gender.h5')
```

Figure 16 : Enregistrement des modèles après entraînement

5.3 Présentation des résultats obtenus

Après avoir entraîné nos modèles de détection de l'âge et du genre, nous avons procédé à une analyse des résultats et à la visualisation des performances des modèles.

Nous avons évalué la performance de notre modèle de détection de l'âge en termes de précision et de perte sur les données de validation. Les résultats obtenus sont présentés dans le graphique suivant :

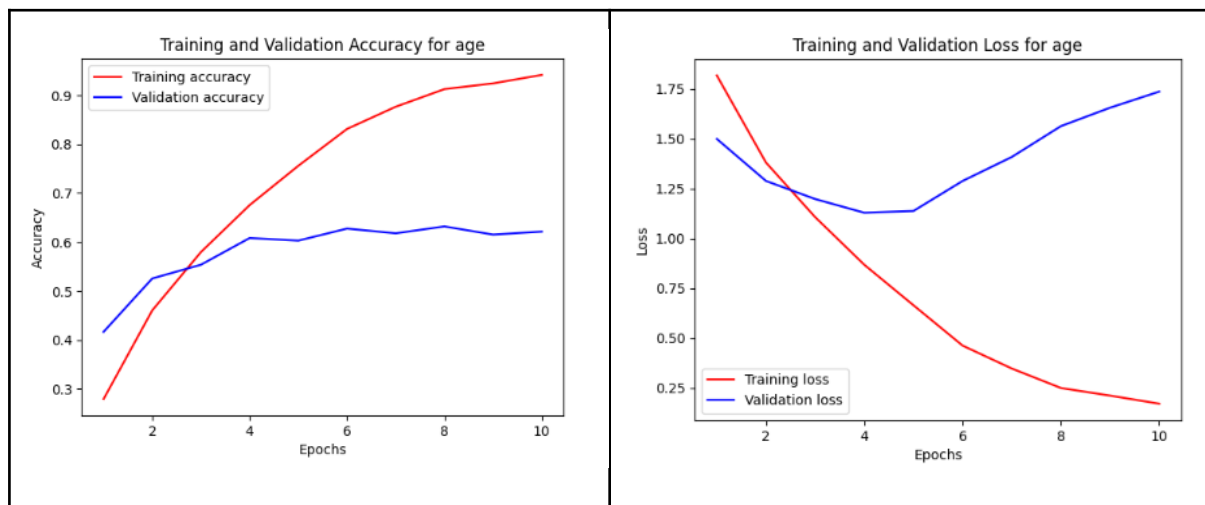


Figure 17: La précision pour les données d'entraînement et de validation

Nous constatons une amélioration de la précision au fil des epochs, tandis que la perte diminue progressivement, indiquant que notre modèle apprend efficacement les caractéristiques des images pour prédire l'âge.

De même, nous avons évalué la performance de notre modèle de détection du genre sur les données de validation. Les résultats sont représentés dans le graphique ci-dessous :

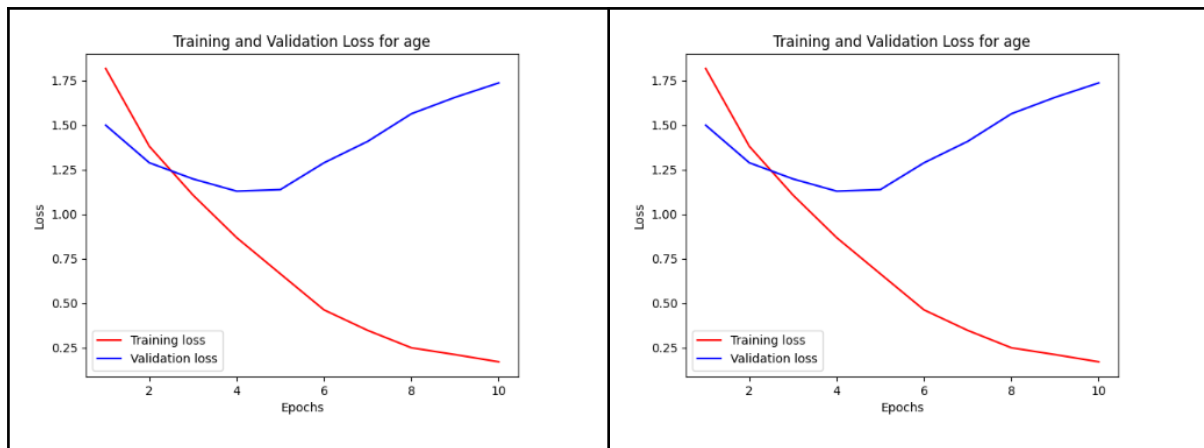


Figure 18 : La perte pour les données d'entraînement et de validation

Nous observons une augmentation de la précision et une diminution de la perte au fur et à mesure de l'entraînement du modèle, ce qui indique une capacité croissante à distinguer les genres à partir des images faciales.

5.4 Phase de Test

Après l'entraînement de nos modèles de détection de l'âge et du genre, nous avons procédé à la phase de test pour évaluer leur performance sur des données de test indépendantes. Cette phase est cruciale pour évaluer la généralisation des modèles sur de nouvelles données. On **2526** données de test pour **l'âge** et **3318** données de test pour **le genre**.

Avant de commencer les prédictions, nous avons analysé **la distribution des âges et des genres** dans notre ensemble de données de test pour garantir qu'elle était **équilibrée**. Cette étape est essentielle pour s'assurer que les résultats de la phase de test sont représentatifs de la population totale.

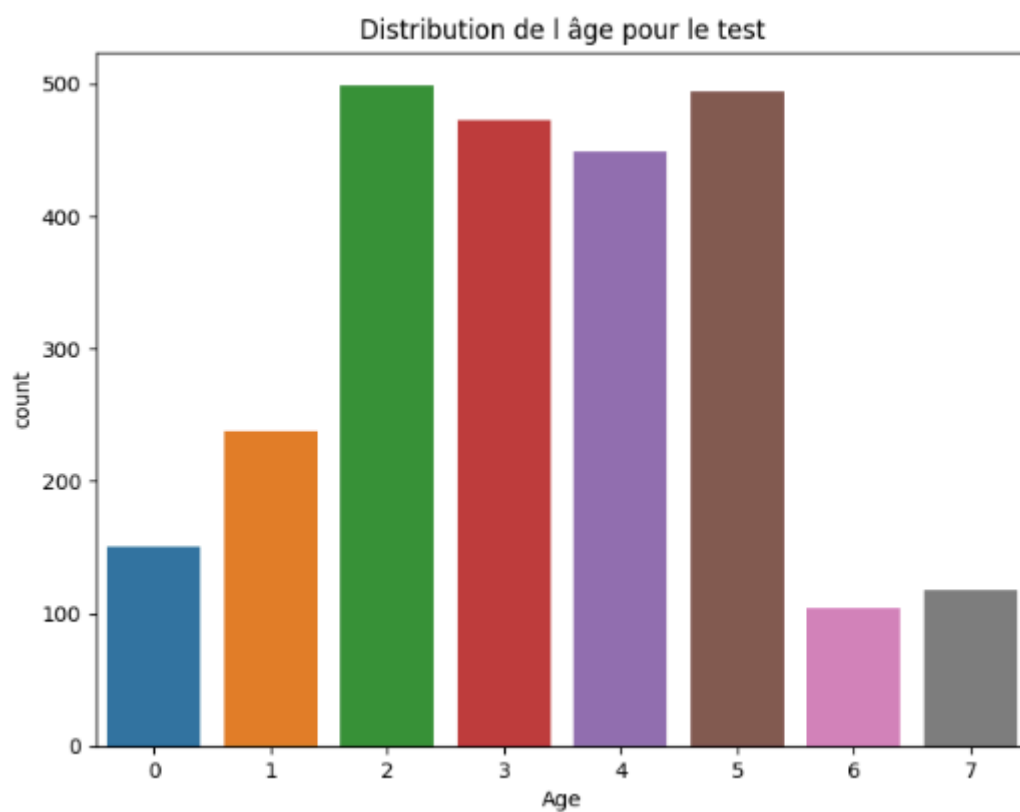


Figure 19 : Distribution de l'âge pour les données de Test



Figure 20 : Distribution de genre pour les données de Test

Nous avons chargé le modèle de détection du genre préalablement entraîné et procédé à la prédiction sur les données de test. En utilisant les prédictions obtenues, nous avons calculé l'accuracy du modèle sur ces données. Pour cela, nous avons comparé les prédictions avec les vraies étiquettes de genre et calculé le pourcentage de prédictions correctes.

→ Le modèle de détection du genre a atteint **une précision de 91.38 %** sur les données de test.

De même, nous avons chargé le modèle de détection de l'âge et effectué des prédictions sur les données de test. Nous avons converti les prédictions en classes d'âge et calculé l'accuracy du modèle en comparant ces prédictions avec les vraies étiquettes d'âge.

→ Le modèle de détection de l'âge a obtenu **une précision de 82.18 %** sur les données de test.

Chapitre 6 : Deuxième solution - Un seul réseau de neurones ensembliste

6.1 Description de l'architecture pour la détection de l'âge et de genre et son entraînement

Dans cette approche, nous avons conçu un seul réseau de neurones pour détecter à la fois l'âge et le genre à partir des images faciales. L'idée était de créer un modèle unique capable de prédire simultanément les deux attributs.

Pour se faire, nous avons adopté une approche novatrice qui consiste à combiner ces deux attributs en une seule neurone de sortie. Cette méthode repose sur un codage spécifique des classes d'âge et du genre, permettant ainsi de représenter chaque combinaison unique de manière efficace.

Chaque **classe d'âge** est représentée par un nombre entier de **0 à 7**, couvrant ainsi toutes les tranches d'âge considérées. De même, le genre est codé avec un chiffre unique, où **0 représente un homme et 1 une femme**.

Pour fusionner ces deux attributs en une seule neurone de sortie, nous avons utilisé une méthode de codage où **l'âge est multiplié par 2** et le résultat est additionné au genre. Ainsi, en combinant l'information sur le genre et l'âge de cette manière, nous obtenons un ensemble de **16 classes uniques**.

Chaque neurone de sortie représente **une classe combinée**, où la valeur active de la neurone indique à la fois le genre et l'âge de la personne détectée dans l'image. Par exemple, si la première neurone est active, cela signifie qu'il s'agit d'un homme de classe d'âge 0, tandis que l'activation de la deuxième neurone indique une femme de classe d'âge 0, et ainsi de suite.

Notre modèle commence par des couches de convolution suivies de couches de max pooling pour extraire les caractéristiques des images. Ensuite, ces caractéristiques sont aplaties et passées à travers des couches de neurones entièrement connectées pour la classification finale.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 126, 126, 16)	448
max_pooling2d (MaxPooling2D)	(None, 63, 63, 16)	0
conv2d_1 (Conv2D)	(None, 61, 61, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 32)	0
conv2d_2 (Conv2D)	(None, 28, 28, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 64)	0
flatten (Flatten)	(None, 12544)	0
dense (Dense)	(None, 128)	1605760
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 16)	2064

=====
Total params: 1631408 (6.22 MB)
Trainable params: 1631408 (6.22 MB)
Non-trainable params: 0 (0.00 Byte)

Figure 21 : Architecture du modèle ensembliste

Nous avons entraîné le modèle en utilisant les données d'entraînement et validé sa performance sur les données de validation. L'entraînement a été effectué sur **10 epochs** avec une taille de **batch de 32**.

```
Epoch 1/10
255/255 [=====] - 66s 253ms/step - loss: 2.5978 - accuracy: 0.1487 - val_loss: 2.2941 - val_accuracy: 0.2324
Epoch 2/10
255/255 [=====] - 60s 235ms/step - loss: 2.1030 - accuracy: 0.2899 - val_loss: 1.9459 - val_accuracy: 0.3219
Epoch 3/10
255/255 [=====] - 61s 239ms/step - loss: 1.7799 - accuracy: 0.3980 - val_loss: 1.7387 - val_accuracy: 0.4015
Epoch 4/10
255/255 [=====] - 60s 235ms/step - loss: 1.5224 - accuracy: 0.4679 - val_loss: 1.6185 - val_accuracy: 0.4432
Epoch 5/10
255/255 [=====] - 61s 238ms/step - loss: 1.2998 - accuracy: 0.5519 - val_loss: 1.5697 - val_accuracy: 0.4747
Epoch 6/10
255/255 [=====] - 61s 239ms/step - loss: 1.0970 - accuracy: 0.6137 - val_loss: 1.5761 - val_accuracy: 0.4943
Epoch 7/10
255/255 [=====] - 60s 235ms/step - loss: 0.9070 - accuracy: 0.6801 - val_loss: 1.6059 - val_accuracy: 0.4958
Epoch 8/10
255/255 [=====] - 60s 235ms/step - loss: 0.7606 - accuracy: 0.7284 - val_loss: 1.6252 - val_accuracy: 0.5337
Epoch 9/10
255/255 [=====] - 60s 236ms/step - loss: 0.6644 - accuracy: 0.7661 - val_loss: 1.6788 - val_accuracy: 0.5233
Epoch 10/10
255/255 [=====] - 60s 237ms/step - loss: 0.5553 - accuracy: 0.8035 - val_loss: 1.8096 - val_accuracy: 0.5351
```

Figure 22 : Les epochs d'entraînement

Une fois l'entraînement terminé, nous avons sauvegardé le modèle entraîné pour la phase de Test.

6.3 Présentation des résultats obtenus

Nous avons visualisé les métriques d'entraînement, notamment l'accuracy et la perte, pour évaluer la performance du modèle au fil des epochs.

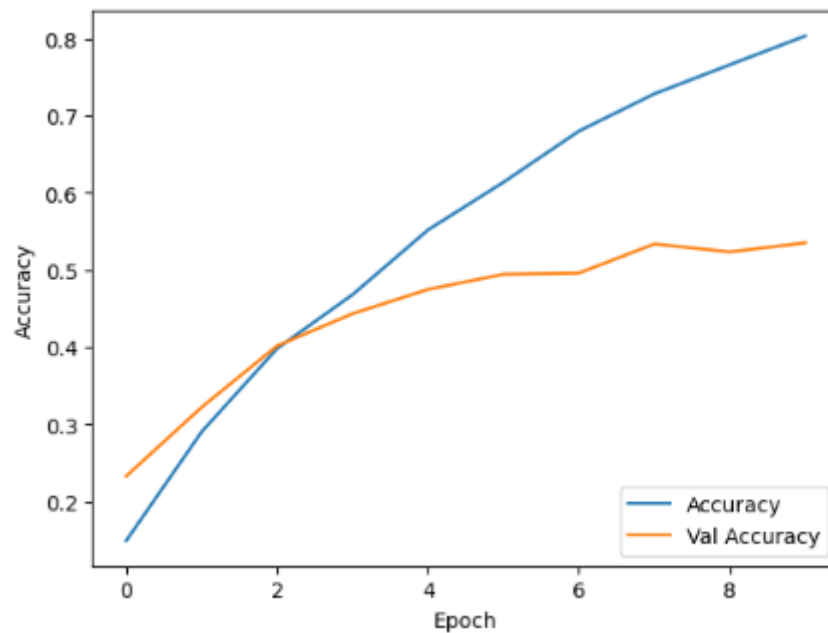


Figure 23 : La précision pour les données d'entraînement et de validation

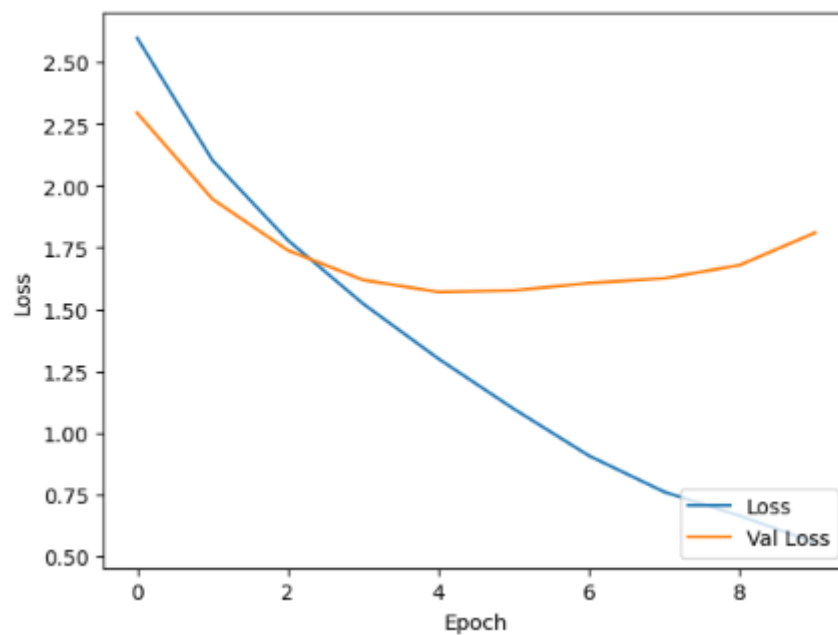


Figure 24 : La perte pour les données d'entraînement et de validation

Les graphiques ci-dessus montrent l'évolution de l'accuracy et de la loss au cours des différentes epochs pendant l'entraînement du modèle. Nous pouvons observer que l'accuracy augmente progressivement au fil des epochs tandis que la loss diminue. Cependant, il est important de noter que la loss sur les données de validation augmente légèrement vers la fin de l'entraînement, ce qui peut indiquer **un léger surapprentissage**.

L'accuracy finale obtenue sur les données de validation est de **80%**, ce qui indique que notre modèle parvient à prédire correctement la classe des données dans une proportion significative des cas. Cependant, la loss sur les données de validation reste relativement élevée, suggérant qu'il existe encore un potentiel d'amélioration du modèle pour réduire l'erreur de prédiction.

6.3 Phase de Test

Dans cette section, nous évaluons les performances de notre modèle sur les données de test après l'entraînement. Nous avons utilisé les mêmes données de test que celles de la première solution pour assurer la cohérence des évaluations.

➤ Performance Globale du Modèle :

Après avoir chargé le modèle entraîné, nous avons procédé à l'extraction des caractéristiques des données de test et calculé l'accuracy globale du modèle sur l'ensemble des prédictions. L'accuracy obtenue pour le modèle total est de **77.00%**, ce qui indique que notre modèle parvient à prédire correctement la classe d'âge et le genre dans une proportion significative des cas.

➤ Performance par Classe :

Nous avons également évalué l'accuracy du modèle en se concentrant sur la prédiction de l'âge seul et du genre seul. Pour ce faire, nous avons construit **deux vecteurs distincts** des classes d'âge et du genre à partir des prédictions du modèle. L'accuracy obtenue pour **l'âge seul** est de **79.46%**, ce qui indique que notre modèle parvient à prédire avec précision la classe d'âge dans la plupart des cas. En ce qui concerne **le genre seul**, l'accuracy atteint

92.37%, ce qui démontre également la capacité de notre modèle à prédire le genre avec une grande précision.

On peut dire alors que notre modèle présente des performances solides lors de la phase de test, avec des accuracies satisfaisantes tant pour la prédiction de l'âge que pour celle du genre. Ces résultats confirment l'efficacité de notre approche de modélisation pour la détection simultanée de l'âge et du genre à partir d'images faciales

Chapitre 7 : Évaluation des solutions et Problèmes rencontrés

7.1 Discussion des deux solutions

Les deux solutions mettent en lumière leurs performances respectives, ainsi que d'autres aspects tels que la complexité du modèle, le temps d'entraînement et la facilité de mise en œuvre.

➤ Performance:

- La première solution, avec deux réseaux de neurones séparés pour la détection de l'âge et du genre, a montré de bonnes performances, avec une précision de 91.38% pour le genre et 82.18% pour l'âge.
- La deuxième solution, qui combine la détection de l'âge et du genre en une seule classe, a obtenu une précision globale de 77%. Cependant, en analysant les performances pour chaque attribut individuellement, nous avons constaté des précisions de 79.46% pour l'âge seul et 92.37% pour le genre seul.

➤ Complexité du modèle:

- La première solution implique deux réseaux de neurones séparés, ce qui peut rendre le modèle plus complexe en termes de nombre de paramètres et de structure.
- La deuxième solution utilise un seul réseau de neurones, ce qui peut réduire la complexité du modèle mais nécessite une combinaison des attributs, ce qui pourrait potentiellement introduire des conflits ou une perte d'informations.

➤ Temps d'entraînement:

- Étant donné que la première solution implique deux réseaux de neurones distincts, le temps d'entraînement peut être plus long par rapport à la deuxième solution, qui utilise un seul réseau.
- Cependant, la deuxième solution pourrait également nécessiter des efforts supplémentaires pour trouver le bon équilibre entre la détection de l'âge et du genre dans le même réseau, ce qui peut également affecter le temps d'entraînement.

➤ **Facilité de mise en œuvre:**

- La première solution, avec deux réseaux de neurones distincts, peut être plus facile à mettre en œuvre car chaque réseau est entraîné séparément et peut être optimisé individuellement.
- La deuxième solution, bien qu'elle nécessite moins de composants de modèle, peut nécessiter une attention particulière lors de la conception et de l'entraînement du modèle pour s'assurer que les deux attributs sont correctement représentés et prévus.

La première solution offre une meilleure précision pour chaque attribut individuellement, mais elle peut être plus complexe et prendre plus de temps à entraîner. La deuxième solution simplifie le modèle mais nécessite une bonne gestion de la combinaison des attributs pour obtenir des performances équilibrées.

7.2 Problèmes rencontrés

1. Déséquilibre des données dans la base de données :

Problème : Le déséquilibre des données peut entraîner un biais dans l'apprentissage du modèle, car il peut favoriser les classes majoritaires au détriment des classes minoritaires. Dans ce cas, cela peut affecter la capacité du modèle à prédire avec précision les classes d'âge et de genre moins représentées.

Solution : Pour remédier à cela, il est important de s'assurer que la base de données est équilibrée en termes de distribution des classes d'âge et de genre. Cela peut être réalisé en collectant plus de données pour les classes moins représentées ou en appliquant des techniques de rééchantillonnage telles que le **suréchantillonnage** (oversampling) ou la **sous-échantillonnage** (undersampling).

2. Ajustement des architectures de réseau de neurones :

Problème : L'architecture du réseau de neurones peut avoir un impact significatif sur le **temps d'entraînement** et les **performances du modèle**. Il est souvent nécessaire d'expérimenter avec différentes architectures pour trouver celle qui convient le mieux au problème donné.

Solution : En ajustant les architectures de réseau de neurones, notamment en modifiant **le nombre de couches, les paramètres des filtres et d'autres hyperparamètres**, il est possible d'optimiser le modèle pour obtenir de meilleures performances tout en réduisant le temps d'entraînement.

3. Gestion des ensembles d'entraînement et de validation :

Problème : La division inappropriée des données en ensembles d'entraînement et de validation peut entraîner un mauvais ajustement du modèle et une mauvaise généralisation. De plus, l'utilisation de données spécifiques à la validation pour évaluer les modèles peut conduire à **une sur-estimation** de leurs performances.

Solution : Il est essentiel de diviser correctement les données en ensembles d'entraînement, de validation et de test pour garantir une évaluation impartiale des modèles. De plus, l'utilisation de techniques telles que la validation croisée peut fournir **une estimation plus fiable** des performances du modèle.

4. calcul de l'accuracy pour l'âge et le genre dans la deuxième solution et solution associée

Problème : Lors de l'évaluation de la deuxième solution, j'ai rencontré une difficulté majeure concernant le calcul de l'accuracy pour l'âge seul et le genre seul à partir des prédictions du modèle. Initialement, j'ai supposé que l'accuracy pourrait être directement déduite des prédictions, mais il est rapidement devenu évident que les prédictions indiquent simplement la neurone active dans le modèle final, représentant à la fois l'âge et le genre. Cela a rendu difficile l'évaluation séparée de l'accuracy pour chaque aspect.

Solution : Pour surmonter cette difficulté, j'ai développé une méthode alternative pour évaluer l'accuracy pour l'âge seul et le genre seul à partir des prédictions du modèle. Cette solution impliquait la reconstruction des prédictions pour séparer l'âge et le genre en utilisant les neurones activés. J'ai construit deux vecteurs distincts : un pour l'âge et l'autre pour le genre, en fonction des neurones activés dans les prédictions du modèle. En comparant ensuite ces vecteurs avec les vraies étiquettes d'âge et de genre, j'ai pu calculer l'accuracy pour chaque aspect individuellement. Cette approche a permis d'obtenir une évaluation précise de la performance du modèle pour prédire l'âge et le genre indépendamment l'un de l'autre.

Chapitre 8 : Conclusion et Perspectives

8.1 Conclusion :

La détection de l'âge et du genre à partir d'images faciales est un domaine passionnant de recherche qui présente de nombreux défis et opportunités. Dans ce rapport, nous avons exploré deux approches différentes pour résoudre ce problème complexe : l'utilisation de deux réseaux de neurones distincts et une seule architecture combinée.

Pour la première solution, qui consistait en deux réseaux de neurones séparés pour la détection de l'âge et du genre, nous avons atteint une précision de 91,38 % pour la détection du genre et 82,18 % pour la détection de l'âge sur les données de test. Malgré des résultats encourageants, cette approche présentait des limitations en termes de complexité et de temps d'entraînement.

La deuxième solution, qui impliquait l'utilisation d'un seul réseau de neurones pour détecter à la fois l'âge et le genre, a obtenu une précision globale de 77 %. Bien que cette approche ait été moins précise que la première, elle offrait l'avantage de réduire la complexité du modèle et le temps d'entraînement.

8.2 Perspectives :

Ce projet a été une opportunité précieuse pour enrichir mes connaissances et mes compétences en apprentissage profond et en vision par ordinateur. Cependant, il reste encore beaucoup à explorer et à améliorer dans ce domaine. Voici quelques perspectives pour de futures recherches :

- Augmentation de la taille et de la diversité des données : L'ajout de données provenant de sources diverses et représentatives de la population générale pourrait aider à améliorer la généralisation des modèles et à réduire les biais.

- Intégration de techniques de prétraitement avancées : L'utilisation de techniques de prétraitement telles que l'augmentation de données, la normalisation avancée et la réduction du bruit pourrait contribuer à améliorer la performance des modèles.
- Évaluation sur des données en temps réel : Tester les modèles sur des données en temps réel dans des environnements réels pourrait aider à évaluer leur applicabilité pratique et à identifier les éventuelles limitations.

Chapitre 9 : Bibliographie et Annexes

Site Web :

- <https://talhassner.github.io/home/projects/Adience/Adience-data.html#agegender>
- <https://medium.com/@skillcate/age-detection-model-using-cnn-a-complete-guide-7b10ad717c60>
- [https://www.academia.edu/49636569/Gender and Age Detection using Deep Learning](https://www.academia.edu/49636569/Gender_and_Age_Detection_using_Deep_Learning)
- <https://www.tensorflow.org/guide/keras?hl=fr>
- <https://arxiv.org/ftp/arxiv/papers/2110/2110.12633.pdf>