

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Kasdi Merbah Ouargla

Faculté des Nouvelles Technologies de l'Information et de Communication

Département d'Informatique et de la Technologie de l'Information



MEMOIRE DE FIN D'ETUDES

Pour l'obtention du diplôme de Master en Informatique

OPTION : Informatique Industrielle

Classification d'âge par Deep Learning

Réalisé par :

IATARIEN Abderraouf

Présenté le :

Membres du jury :

Encadreur : Ch.BENKHROUROU MA (A) UKM Ouargla

Examineur : UKM Ouargla

Examineur : UKM Ouargla

Promotion : 2019/2020

Remerciements

C'est pour moi un plaisir autant qu'un devoir de remercier toutes les personnes qui ont pu contribuer de près ou de loin à l'établissement de ce projet, qui m'ont aidé, m'ont soutenu et ont fait sorte que ce travail ait eu lieu.

Ainsi, j'exprime ma gratitude et je tiens à remercier Mme Chafika BENKHROUROU qui m'a encadré, et qui n'a épargné aucun effort pour m'orienter afin que je puisse mener à bien ce projet.

Je tiens aussi à remercier les membres du jury pour leur précieux temps accordé à l'étude de mon mémoire.

Enfin, je remercie affectueusement mes parents et ma sœur, qui m'ont toujours fait confiance et qui m'ont soutenu dans mes études.

Dédicaces

Je dédie ce mémoire :

A mes très chers parents pour leur soutien et encouragement durant toutes mes années d'études et sans lesquels je n'aurais jamais réussi.

A ma chère sœur Soumia

J'espère toujours être à la hauteur de vos espérances.

A toute personne ayant contribué à ce travail de près ou de loin.

A tous mes professeurs et enseignants que j'ai eu durant tout mon cursus scolaire et qui m'ont permis de réussir dans mes études.

Liste des abréviations

IA : Intelligence Artificielle

Chatbot : Chat robot (Rebot de chat)

RNA : Réseau de Neurone Artificiel

ConvNet : Convolutional Neural Network (Réseau de neurones convolutionnels)

CNN : Convolutional Neural Network

RELU : Rectified Linear Unit

ACP : Analyse des Composants principales

GUI : Graphical User Interface (Interface Utilisateur Graphique)

Liste des figures

Figure 1.1 : Paradigmes d'apprentissage automatique

Figure 1.2 : Exemple de régression linéaire

Figure 1.3 : Exemple d'un arbre de décision

Figure 1.4 : Exemple d'un SVM

Figure 1.5 : Exemple de KNN

Figure 1.6 : Exemple de K-Means

Figure 1.7 : Exemple de Random Forest

Figure 1.8 : Neurone artificielle et neurone biologique

Figure 1.9 : Architecture d'un RNA

Figure 1.10 : Feedforward dans un neurone

Figure 1.11 : Feedforward dans un RNA

Figure 2.1 : Les ConvNet et le cerveau humain

Figure 2.2 : Couche convolutionnelle

Figure 2.3 : Couche Pooling

Figure 2.4 : Max Pooling

Figure 2.5 : Fonction RELU

Figure 2.6 : Couche entièrement connectée

Figure 3.1 : Les ratios utilisés

Figure 3.2 : Le modèle Ke Zhang Na Liu, Xingfang Yuan, Xinyao Guo, Ce Gao, Zhenbing Zhao et Zhanyu Ma.

Figure 4.1 : L'architecture de notre modèle

Figure 4.2 : Logo du langage python

Figure 4.3 : Logo de Tensorflow

Figure 4.4 : Logo de Keras

Figure 4.5 : Logo de PyQt

Figure 4.6 : Logo d'Open CV

Figure 4.7 : La base d'images IMDB-WIKI

Figure 4.8 : Exemple de recadrement d'une image

Figure 4.9 : Exemple de conversion en grayscale d'une image

Figure 4.10 : Graphique linéaire de la précision du modèle

Figure 4.11 : Page d'accueil

Figure 4.12 : Test à travers la caméra de l'ordinateur

Figure 4.13: Test à travers une image importée

Table des matières

| | |
|--|-----------|
| Introduction générale | 1 |
| Chapitre 01 : Généralités sur l'apprentissage automatique | 2 |
| 1. Introduction | 3 |
| 2. Définition | 3 |
| 3. Application de l'apprentissage automatique | 3 |
| 4. Avantages de l'apprentissage automatique | 6 |
| 5. Paradigmes l'apprentissage automatique | 7 |
| 5.1. Apprentissage supervisé | 8 |
| 5.2. Apprentissage non supervisé..... | 9 |
| 5.3. Apprentissage semi supervisé | 9 |
| 5.4. Apprentissage par renforcement..... | 10 |
| 6. Algorithmes d'apprentissage automatique | 10 |
| 7. Conclusion | 17 |
| Chapitre 02 : Les réseaux de Neurones Convolutionnels | 18 |
| 1. Introduction | 19 |
| 2. Définition | 19 |
| 3. Fonctionnement des réseaux de Neurones Convolutionnels | 20 |
| 3.1. Couche Convolutionnelle | 20 |
| 3.2. Couche Pooling | 21 |
| 3.3. Fonction d'activation..... | 22 |
| 3.4. Couche entièrement connectée..... | 23 |
| 4. Les ConvNet les plus connus | 23 |
| 4.1. LeNet | 24 |
| 4.2. AlexNet..... | 24 |
| 4.3. GoogleNet..... | 24 |
| 5. Conclusion | 25 |

| | |
|--|-----------|
| Chapitre 03 : Etat de l'art des approches utilisées dans la détection d'âge | 26 |
| 1. Introduction..... | 27 |
| 2. Les approches utilisées dans le domaine de détection d'âge | 27 |
| 2.1 L'approche basée sur la théorie du développement cranio-facial..... | 27 |
| 2.1.1 Détection des caractéristiques faciales | 28 |
| 2.1.2. Evaluation de différents ratios de caractéristiques faciales | 28 |
| 2.1.3. Analyse des rides | 29 |
| 2.1.4. Combinaison des ratios et des rides | 29 |
| 2.2. L'approche basée sur un modèle de vieillissement | 29 |
| 2.2.1. Sous espace du modèle de vieillissement | 30 |
| 2.2.2. Estimation d'âge | 31 |
| 2.3. L'approche basée sur l'apprentissage automatique | 32 |
| 3. Conclusion | 33 |
| Chapitre 04 : Conception et implémentation | 34 |
| 1. Introduction..... | 35 |
| 2. Conception..... | 35 |
| 2.1. Architecture de notre modèle | 36 |
| 3. Implémentation | 38 |
| 3.1. Outils de programmation utilisés | 38 |
| 3.2 La base d'images | 41 |
| 3.3. Configuration utilisée dans l'implémentation | 41 |
| 3.4. Le Prétraitement | 42 |
| 3.5. L'entraînement (Training) | 44 |
| 4. Résultats | 45 |
| 5. Discussions..... | 46 |
| 6. Présentation de l'application | 47 |
| 7. Conclusion | 48 |
| Conclusion générale..... | 49 |
| Bibliographie..... | 50 |

Introduction générale

Dans un monde connecté et de plus en plus informatisé, la technologie est devenue indispensable dans la vie quotidienne des gens. Non seulement les adultes en ont besoin, mais aussi les adolescents et les enfants qui deviennent de plus en plus attachés à la technologie à travers les jeux vidéo et les sites web. Parmi lesquels il existe ceux dont le contenu ne convient qu'aux personnes ayant un certain âge le plus souvent plus de 18 ans.

Cependant, les créateurs de ces sites web et ces jeux vidéo ne font pas assez d'efforts pour les rendre accessibles qu'aux adultes. D'abord, les enfants se trouvent très souvent confrontés à un contenu trop violent pour leur âge ce qui peut engendrer un comportement trop violent chez eux. En plus, l'accès à des sites web et des plateformes ayant des contenus trop graphiques et inappropriés à un âge très jeune peut avoir des conséquences néfastes sur les personnalités de ces jeunes individus qui peuvent devenir des harceleurs ou des personnes qui ne peuvent pas contrôler leurs émotions.

Pour cela, un outil efficace doit être mis en place pour empêcher les mineurs d'accéder aux plateformes et aux programmes destinés qu'aux adultes.

La démarche que nous allons suivre pour réaliser cet objectif consiste à concevoir et réaliser un programme qui peut classer les êtres humains en deux catégories (adulte et mineur) à travers les traits de leurs visages en appliquant l'apprentissage automatique. D'abord, nous allons effectuer un prétraitement sur la base d'images IMDB-WIKI comportant des milliers d'images de célébrités et de personnes connues. Ensuite, nous allons créer un modèle en se basant sur les Réseaux de Neurones Convolutionnels et l'entraîner sur une partie de la base de données prétraitées. Enfin, nous allons tester notre modèle sur des images qui n'ont pas été utilisées lors du traitement pour valider le modèle et calculer sa précision.

Ce rapport se compose de quatre grandes sections : la première est une vue d'ensemble de l'apprentissage automatique. La deuxième est consacrée aux Réseaux de Neurones Convolutionnels. Alors que la troisième section est une présentation de l'état de l'art du domaine de détection d'âge. La quatrième présente la conception et le développement de la partie pratique de notre projet. Nous terminerons notre rapport par une conclusion et des perspectives.

Chapitre 01: Généralités sur L'Apprentissage Automatique

1. Introduction :

Aujourd'hui, l'intelligence artificielle est devenue le domaine le plus important de l'informatique, et de nombreux domaines tels que l'économie, l'industrie et la médecine ont énormément bénéficié de la montée de cette science qui se développe et évolue constamment jour après jour.

L'évolution de l'Intelligence Artificielle a dépassé les attentes. Il y a quelques décennies, disposer d'appareils ou de logiciels proche de l'intelligence humaine semblait impossible et irréaliste. Mais aujourd'hui, il existe des logiciels qui dépassent les humains dans de nombreux domaines tels que les jeux vidéo, la prise de décision, diagnostics médicaux ...etc

Cette incroyable évolution de l'IA est fortement liée à l'énorme montée de l'apprentissage automatique (Machine Learning), qui est devenu le domaine de l'IA le plus populaire à partir des années 2010 et le plus utilisé par la majorité des chercheurs. Dans ce chapitre nous allons parler de l'apprentissage automatique et ses algorithmes.

2. Définition de l'apprentissage automatique :

L'apprentissage automatique est une application de l'intelligence artificielle (IA) qui offre aux systèmes la capacité d'apprendre et de s'améliorer automatiquement à partir des données sans être explicitement programmé. Des techniques basées sur l'apprentissage automatique ont été appliquées avec succès dans divers domaines allant de la reconnaissance de formes, la vision par ordinateur, l'ingénierie de vaisseaux spatiaux, la finance, le divertissement et la biologie computationnelle aux applications biomédicales et médicales. [1]

3. Applications de l'apprentissage automatique :

Les applications de l'apprentissage automatique sont partout, et beaucoup de personnes les utilisent chaque jour. Nous citons quelques exemples de ces applications.

3.1. Assistants personnels virtuels : Ces assistants comme leur nom l'indique, ils aident à trouver des informations lorsqu'ils sont interrogés à voix haute. Il suffit de les activer et de demander une question. Pour répondre, l'assistant personnel recherche les informations, rappelle la requêtes associées ou envoie une commande à d'autres ressources (comme les applications téléphoniques) pour collecter des informations. Les assistant virtuels reçoivent même des instructions pour effectuer certaines tâches notamment réglage de l'alarme pour le lendemain, rappel des rendez-vous et des dates limites ...etc

L'apprentissage automatique est une partie importante de ces assistants personnels car ils collectent les informations à partir des interactions quotidiennes avec l'utilisateur.

3.2. Vidéosurveillance : De nos jours, beaucoup de systèmes de vidéosurveillance sont alimentés par l'IA qui permet de détecter des crimes avant qu'ils ne se produisent. Ils suivent le comportement inhabituel de personnes telles que rester immobile pendant longtemps, trébucher ou faire une sieste sur des bancs, etc. Le système peut ainsi donner une alerte aux assistants humains, ce qui peut finalement aider à éviter les accidents. Lorsque de telles activités sont signalées et considérées comme vraies, elles contribuent à améliorer les services de surveillance. Cela se produit lorsque l'apprentissage automatique fait son travail au niveau du backend (arrière plan).

3.3. Filtrage des emails indésirables et des programmes malveillants : Il existe un certain nombre d'approches de filtrage anti-spam que les services de messagerie utilisent. Pour vérifier que ces filtres anti-spam sont mis à jour en permanence, ils sont alimentés par l'apprentissage automatique. Lorsque le filtrage du spam est effectué à partir d'une base de règles, il ne parvient pas à traquer les dernières techniques adoptées par les spammeurs. Par contre, les systèmes de filtrage alimentés par l'apprentissage automatique comprennent le modèle de codage. Par conséquent, ils détectent facilement les courriers indésirables ainsi que les nouveaux logiciels malveillants et offrent une protection contre eux.

3.4 Détection de fraude en ligne:

L'apprentissage automatique prouve son potentiel pour faire du cyberspace un endroit sûr et sécurisé à travers le suivi des fraudes monétaires en ligne. Par exemple: Paypal utilise l'apprentissage automatique pour se protéger contre le blanchiment d'argent. L'entreprise utilise un ensemble d'outils qui l'aide à comparer des millions de transactions en cours et à distinguer les transactions légitimes ou illégitimes entre les acheteurs et les vendeurs.

3.5 Moteurs de recherche : Google et d'autres moteurs de recherche utilisent l'apprentissage automatique pour améliorer les résultats de recherche. Chaque fois qu'un client exécute une recherche, les algorithmes du backend surveillent la façon dont il réagit aux résultats. S'il ouvre les premiers résultats et reste longtemps sur la page Web, le moteur de recherche suppose que les résultats affichés sont conformes à la requête. De même, si il atteint la deuxième ou la troisième page des résultats de la recherche mais n'ouvre aucun des résultats, le moteur de recherche estime que les résultats affichés ne correspondent pas aux exigences. De cette façon, les algorithmes appliqués au niveau du backend améliorent les résultats de la recherche.

3.6 Service client en ligne : Un certain nombre de sites Web offrent de nos jours la possibilité de discuter avec le représentant du support client pendant que les clients naviguent sur le site. Cependant, la majorité des sites Web n'utilisent pas des êtres humains pour répondre aux questions de ces clients. Dans la plupart des cas, ils utilisent un chatbot. Ces bots ont tendance à extraire des informations du site Web et à les présenter aux clients. Pendant ce temps, les chatbots avancent avec le temps. Ils ont tendance à mieux comprendre les requêtes des utilisateurs et à leur fournir de meilleures réponses, ce qui est possible grâce aux algorithmes d'apprentissage automatique.

3.7 Reconnaissance des objets : La reconnaissance d'objets est l'un des exemples les plus importants d'apprentissage automatique et d'intelligence artificielle. C'est une approche pour identifier et détecter une caractéristique ou un objet dans l'image numérique. De plus, cette technique peut être utilisée pour une analyse plus approfondie, telle que la reconnaissance de formes, la détection de visages, la reconnaissance de visages, la reconnaissance optique de caractères et bien d'autres.

3.8 Analyse de sentiments : L'analyse des sentiments est une autre application d'apprentissage automatique en temps réel. Il se réfère également à l'exploration d'opinion, à la classification des sentiments, etc. C'est un processus qui détermine l'attitude ou l'opinion du locuteur ou de l'écrivain. En d'autres termes, c'est le processus de recherche de l'émotion à partir du texte. Par exemple si quelqu'un écrit "le film n'est pas si bon". Découvrir la pensée ou l'opinion réelle du texte (est-elle bonne ou mauvaise) est la tâche de l'analyse des sentiments.

3.9 Services médicaux : Les méthodes d'apprentissage automatique sont largement utilisées dans le domaine médical. Notamment pour détecter une maladie, planification de la thérapie, recherche médicale, prédiction de la situation de la maladie etc L'utilisation de logiciels basés sur l'apprentissage automatique dans le domaine de la santé apporte une percée dans notre science médicale.

3.10 Identification âge / sexe : C'est un sujet qui attire beaucoup de chercheurs, l'identification de l'âge ou du sexe est une tâche importante dans de nombreux cas. L'identification peut être effectuée à l'aide des algorithmes d'apprentissage automatique.

4. Les avantages de l'apprentissage automatique :

L'intelligence Artificielle est sûrement le domaine dont on parle le plus au milieu informatique, Et à partir des années 2010 l'apprentissage automatique est devenu le domaine d'IA qui attire le plus de chercheurs au monde. Plusieurs domaines tels que la médecine et l'industrie ont bénéficié énormément de sa croissance. Ce qui pose la question pourquoi l'apprentissage automatique est si important ? Voici quelques avantages de l'apprentissage automatique :

4.1. Identification des modèles et l'extraction des informations utiles : L'apprentissage automatique peut examiner de grands volumes de données et découvrir des tendances et des modèles spécifiques qui ne seraient pas apparents aux humains. Par exemple, pour un site Web de commerce électronique, il sert à comprendre les comportements de navigation et les historiques d'achat des utilisateurs pour les aider à trouver les bons produits. Il utilise les résultats pour leur révéler des publicités pertinentes.

4.2. Aucune intervention humaine nécessaire (automatisation) Avec l'apprentissage automatique, le projet ne doit pas être suivi et surveillé à chaque étape. Comme l'apprentissage se fait par les machines elles mêmes, cela leur permet de faire des prédictions et également d'améliorer les algorithmes par eux-mêmes. Un exemple courant de cela est les logiciels antivirus; ils apprennent à filtrer les nouvelles menaces au fur et à mesure qu'elles sont reconnues.

4.3. Amélioration continue : Lorsque les algorithmes d'apprentissage automatique continuent à acquérir de l'expérience, ils continuent de s'améliorer en précision et en efficacité. Cela leur permet de prendre de meilleures décisions. Si on prend l'exemple d'un modèle de prévisions météorologiques, au fur et à mesure que la quantité de données augmente, les algorithmes apprennent à faire des prédictions plus précises plus rapidement.

4.4. Traitement des données hétérogènes et multidimensionnelles : Les algorithmes d'apprentissage automatique sont bons pour gérer des données multidimensionnelles et variées.

5. Les paradigmes de l'apprentissage automatique :

Les paradigmes d'apprentissage automatique diffèrent dans leur approche. Chaque paradigme est caractérisé par le type de données qu'il entre et sort, et le type de tâche ou de problème qu'il est censé résoudre.

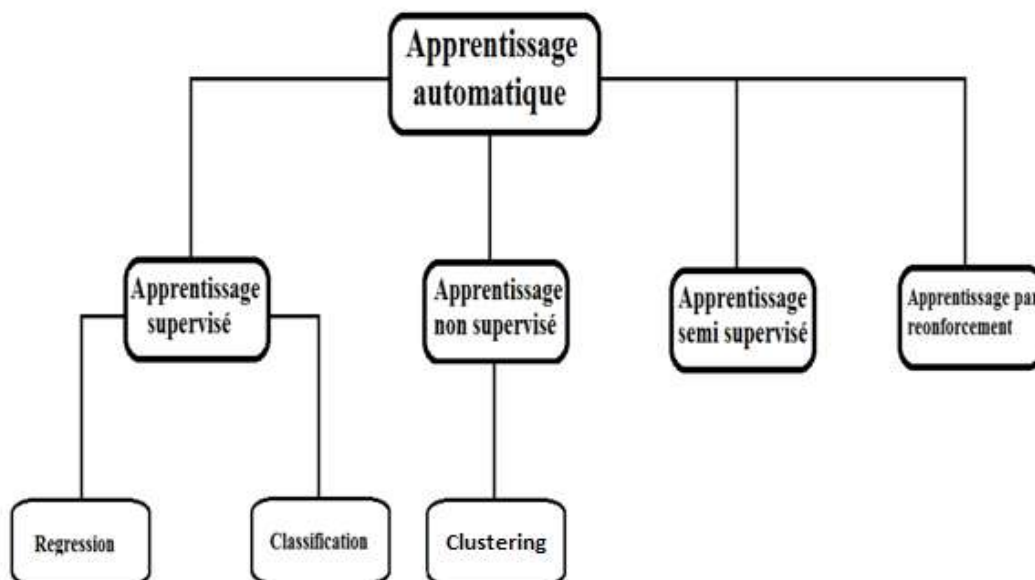


Figure 1.1 : Paradigmes d'apprentissage automatique

5.1 Apprentissage supervisé (Supervised learning) : Les algorithmes d'apprentissage supervisé construisent un modèle mathématique d'un ensemble de données qui contient à la fois les entrées et les sorties souhaitées [2]. Les données sont appelées données d'entraînement (training data) et se composent d'un ensemble d'exemples. Chaque exemple de l'entraînement a une ou plusieurs entrées et la sortie souhaitée. Dans le modèle mathématique, chaque exemple d'apprentissage est représenté par un tableau ou un vecteur, parfois appelé vecteur d'entités, et les données d'apprentissage sont représentées par une matrice. Grâce à l'optimisation itérative d'une fonction objective, les algorithmes d'apprentissage supervisé apprennent une fonction qui peut être utilisée pour prédire la sortie associée à de nouvelles entrées [3]. Une fonction optimale permettra à l'algorithme de déterminer correctement la sortie pour les entrées qui ne faisaient pas partie des données d'apprentissage. Les algorithmes d'apprentissage supervisé incluent la classification et la régression.

- **5.1.1 Classification :** Les algorithmes de classification sont utilisés lorsque les sorties sont limitées à un ensemble limité de valeurs. le programme apprend des données d'entrée qui lui sont données dans la phase d'entraînement, puis utilise cet apprentissage pour classer de nouvelles observations, où le processus consiste à trouver les caractéristiques qui aideront à séparer les données en différentes classes. Les algorithmes de classifications les plus connus sont SVM, KNN et les réseaux de neurones [20].
- **5.1.2 Régression :** La régression est une technique d'apprentissage supervisé prédisant des réponses continues (des valeurs numériques), les modèles de régression sont utilisé pour prédire une valeur continue comme le salaire, les prix et les poids, il est l'un des plus importants outils d'apprentissage automatique et de statistiques largement utilisés, il permet de faire des prédictions en apprenant la relation entre les caractéristiques des données et les résultats observés à valeur continue. Plusieurs différents modèles peuvent être utilisés, tels que la régression polynomiale, SVR (Support Vector Regression) et les arbres de régression [21].

5.2 Apprentissage non supervisé (*Unsupervised learning*) : Les algorithmes d'apprentissage non supervisés prennent un ensemble de données qui ne contient que des entrées et trouvent une structure dans les données, comme le regroupement ou le regroupement de points de données. Les algorithmes apprennent donc des données de test qui n'ont pas été étiquetées, classées ou catégorisées. Ces algorithmes identifient les points communs dans les données et réagissent en fonction de la présence ou de l'absence de ces points communs dans chaque nouvelle donnée. Une application centrale de l'apprentissage non supervisé se situe dans le domaine de l'estimation de la densité en statistique, bien que l'apprentissage non supervisé englobe d'autres domaines impliquant de résumer et d'expliquer les caractéristiques des données. Lorsqu'on parle d'apprentissage automatique non supervisé on parle essentiellement de clustering [22].

- **5.2.1 Clustering :** Le clustering est l'affectation d'un ensemble d'observations en sous-ensembles (appelés clusters) de sorte que les observations au sein d'un même cluster soient similaires selon un ou plusieurs critères pré désignés, tandis que les observations tirées de clusters différents sont différentes. Différentes techniques de clustering font différentes hypothèses sur la structure des données, souvent définies par une métrique de similitude, tel que la compacité interne, ou la similitude entre les membres d'un même cluster, ou la différence entre les clusters. D'autres méthodes sont basées sur une densité estimée et une connectivité graphique. Les algorithmes de clustering les plus connus sont K-Means et le Mean-Shift Clustering [23].

5.3 Apprentissage semi supervisé (*Semi supervised learning*) : Dans le cas des algorithmes d'apprentissage semi-supervisés, certains exemples manquent d'étiquettes de formation, mais ils peuvent néanmoins être utilisés pour améliorer la qualité d'un modèle. Dans un apprentissage faiblement supervisé, les étiquettes de formation sont bruyantes, limitées ou imprécises; cependant, ces étiquettes sont souvent moins chères à obtenir, ce qui donne des ensembles de formation plus efficaces.

5.4 Apprentissage par renforcement (Reinforcement learning) : L'apprentissage par renforcement est un domaine de l'apprentissage automatique qui concerne la façon dont les agents logiciels doivent prendre des mesures dans un environnement afin de maximiser une certaine notion de récompense cumulative. En raison de sa généralité, le domaine est étudié dans de nombreuses autres disciplines, telles que la théorie des jeux, la théorie du contrôle, la recherche opérationnelle, la théorie de l'information, l'optimisation basée sur la simulation, les systèmes multi-agents, les statistiques et les algorithmes génétiques. Les algorithmes d'apprentissage par renforcement ne supposent pas la connaissance d'un modèle mathématique exact. Ils sont utilisés lorsque les modèles exacts sont irréalisables. Ils sont fréquemment utilisés dans des véhicules autonomes ou pour apprendre à jouer à un jeu contre un adversaire humain.

6. Les algorithmes de l'apprentissage automatique :

Il y a tellement d'algorithmes qu'il est presque impossible de les connaître tous. Mais il existe des algorithmes qui sont plus connus et plus utilisés que d'autres. Nous présentons ici les algorithmes les plus connus dans le domaine de l'apprentissage automatique.

6.1. Régression linéaire : Utilisé pour estimer les valeurs réelles (coût des maisons, nombre d'appels, ventes totales, etc.) sur la base de variables continues. Ici, une relation entre les variables indépendantes et dépendantes est établie en ajustant une meilleure ligne. Cette ligne de meilleur ajustement est connue sous le nom de ligne de régression et représentée par une équation linéaire $Y = a * X + b$. La régression linéaire est principalement de deux types: la régression linéaire simple et la régression linéaire multiple. La régression linéaire simple est caractérisée par une variable indépendante, et la régression linéaire multiple (comme son nom l'indique) est caractérisée par plusieurs (plus de 1) variables indépendantes. Tout en trouvant la ligne la mieux adaptée [27].

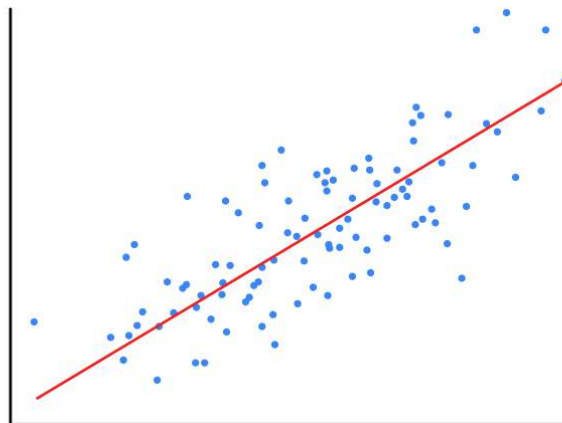


Figure 1.2 : Exemple de régression linéaire

6.2 Arbre de décision : Il s'agit d'un type d'algorithme d'apprentissage supervisé qui est principalement utilisé pour les problèmes de classification. Dans cet algorithme, nous divisons la population en deux ou plusieurs ensembles homogènes. Cela se fait sur la base des attributs les plus significatifs / les plus indépendants pour créer des groupes aussi distincts que possible [24].

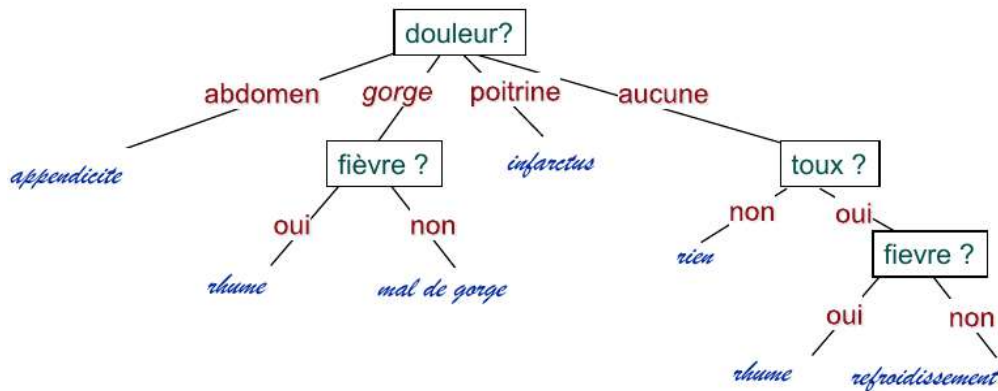


Figure 1.3 : Exemple d'un arbre de décision

6.3 SVM (Séparateurs à Vaste Marge) : Les séparateurs à vaste marge (en anglais support vector machine, SVM) sont des modèles d'apprentissage supervisé avec des algorithmes d'apprentissage associés qui analysent les données utilisées pour la classification et la régression. Étant donné un ensemble d'exemples d'entraînement, chacun marqué comme appartenant à l'une ou l'autre des deux catégories, un algorithme d'entraînement SVM construit un modèle qui attribue de nouveaux exemples à une catégorie ou à l'autre, ce qui en fait un classificateur linéaire binaire non probabiliste (bien que les méthodes tels que la mise à l'échelle de Platt existent pour utiliser SVM dans un cadre de classification probabiliste). Un modèle SVM est une représentation des exemples sous forme de points dans l'espace, cartographiée de manière à ce que les exemples des catégories distinctes soient divisés par un espace clair aussi large que possible. De nouveaux exemples sont ensuite mappés dans ce même espace et devraient appartenir à une catégorie en fonction du côté de l'écart sur lequel ils se trouvent [25].

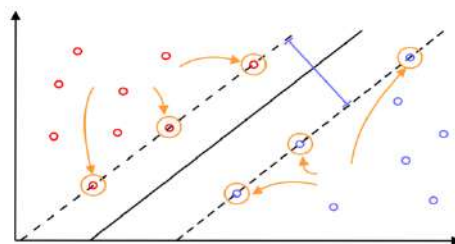


Figure 1.4 : Exemple d'un SVM

6.4 KNN (K-nearest Neighbors) : C'est un algorithme qui peut être utilisé à la fois pour des problèmes de classification et de régression. Cependant, il est plus largement utilisé dans les problèmes de classification dans l'industrie. KNN est un algorithme simple qui stocke tous les cas disponibles et classe les nouveaux cas par un vote majoritaire de ses k voisins. Le cas attribué à la classe est le plus courant parmi ses K voisins les plus proches mesurés par une fonction de distance. Ces fonctions de distance peuvent être la distance euclidienne, Manhattan, Minkowski et Hamming. Les trois premières fonctions sont utilisées pour la fonction continue et la quatrième (Hamming) pour les variables catégorielles. Si $K = 1$, le cas est simplement affecté à la classe de son plus proche voisin. Parfois, choisir K s'avère être un défi lors de la modélisation KNN [26].

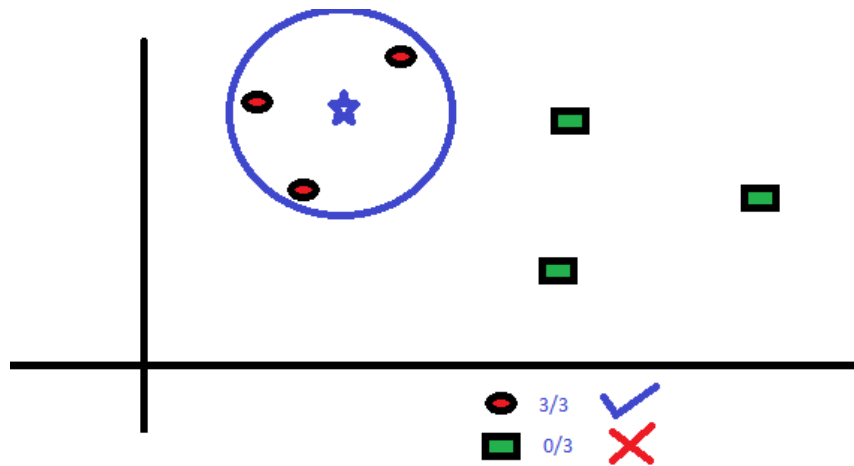


Figure 1.5 : Exemple de KNN

6.5. K-Means : Il s'agit d'un type d'algorithme non supervisé qui résout le problème de clustering. Sa procédure suit une méthode simple et facile de classer un ensemble de données à travers un certain nombre de clusters (supposons k clusters). Les points de données à l'intérieur d'un cluster sont homogènes et hétérogènes aux autres groupes [28].

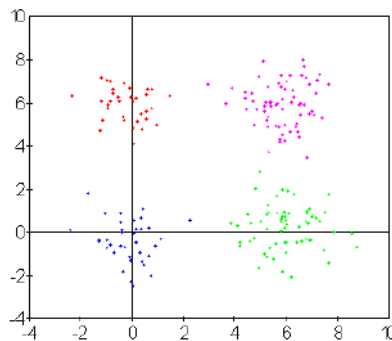


Figure 1.6 : Exemple de K-Means

6.6 Random Forest (Forêt aléatoires) : Dans cet algorithme, une collection d'arbres de décision (appelés «forêt»). Pour classer un nouvel objet en fonction d'attributs, chaque arbre donne une classification et l'arbre « vote » pour cette classe. La forêt choisit la classification ayant le plus de votes (sur tous les arbres de la forêt) [29]. Chaque arbre est planté et cultivé comme suit:

- 1- Si le nombre de cas dans l'ensemble de formation est N , un échantillon de N cas est prélevé au hasard mais avec remplacement. Cet échantillon sera l'ensemble de formation pour la croissance de l'arbre.
- 2- S'il y a M variables d'entrée, un nombre $m < M$ est spécifié de telle sorte qu'à chaque nœud, m variables sont sélectionnées au hasard parmi les M et la meilleure répartition sur ces m est utilisée pour diviser le nœud. La valeur de m est maintenue constante pendant la croissance de la forêt.
- 3- Chaque arbre est cultivé autant que possible. Il n'y a pas d'élagage.

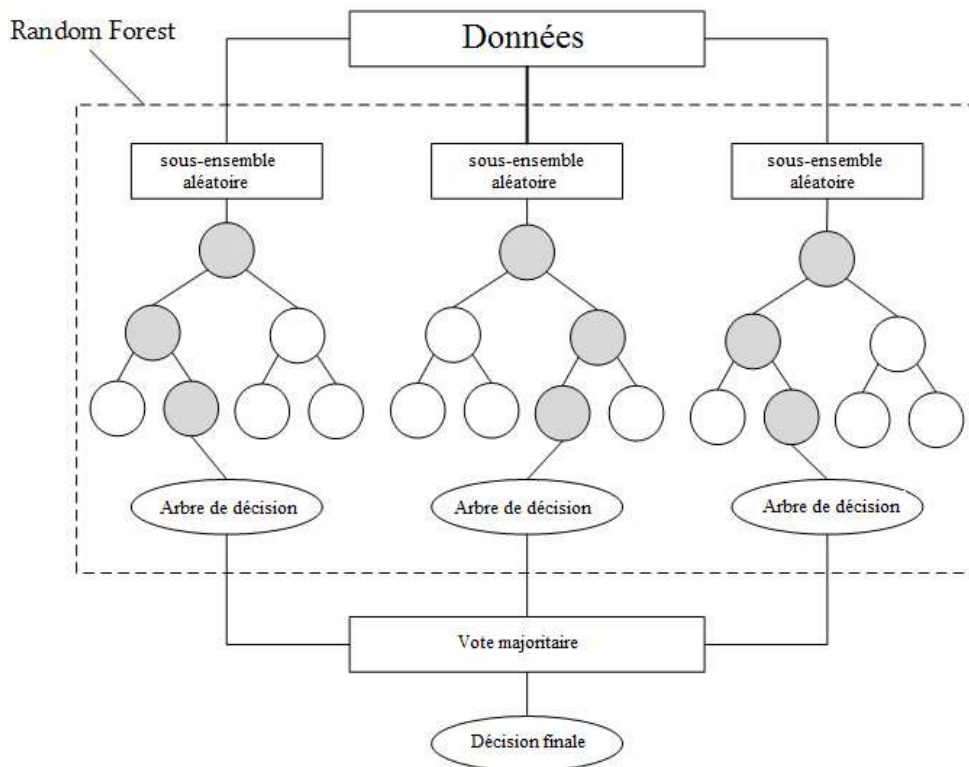


Figure 1.7 : Exemple de Random Forest

6.7 Les réseaux de neurones artificielles :

- **6.7.1 Définition :** Les réseaux de neurones artificiels sont un ensemble de neurones interconnectés où chaque neurone calcule une sortie unique en fonction des informations qu'il reçoit. Leur conception a été à l'origine inspirée par le fonctionnement des neurones biologiques dans le cerveau. [4]

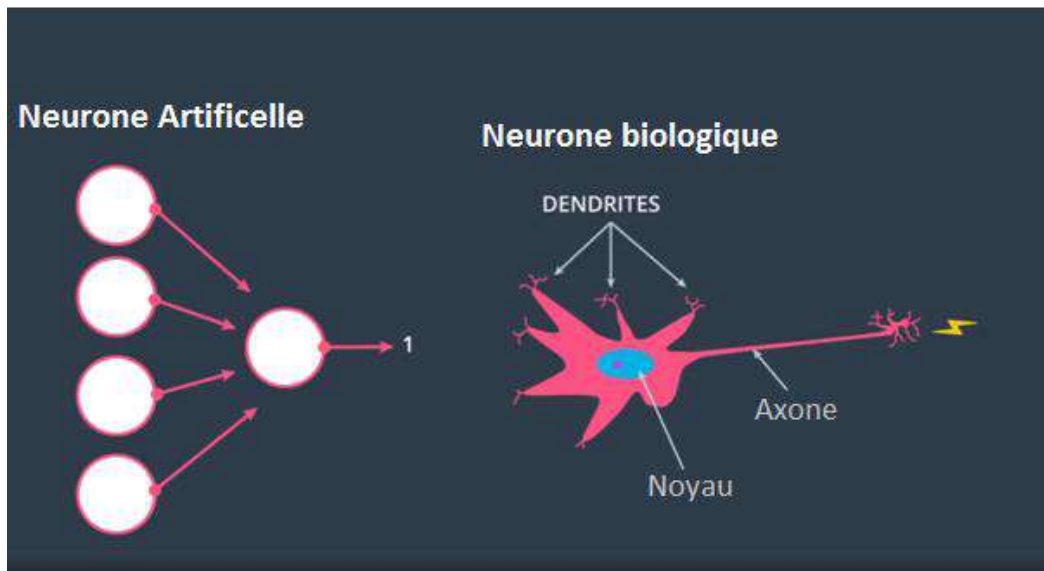


Figure 1.8 : Neurone artificielle et neurone biologique

- **6.7.2 Architecture des RNA :** Les RNA sont constitués de couches de neurones. La première couche est la couche d'entrée qui reçoit les données. La dernière couche est la couche de sortie qui prédit la sortie. Entre ces 2 couches existent les couches cachées qui effectuent la plupart des calculs requis par le réseau.

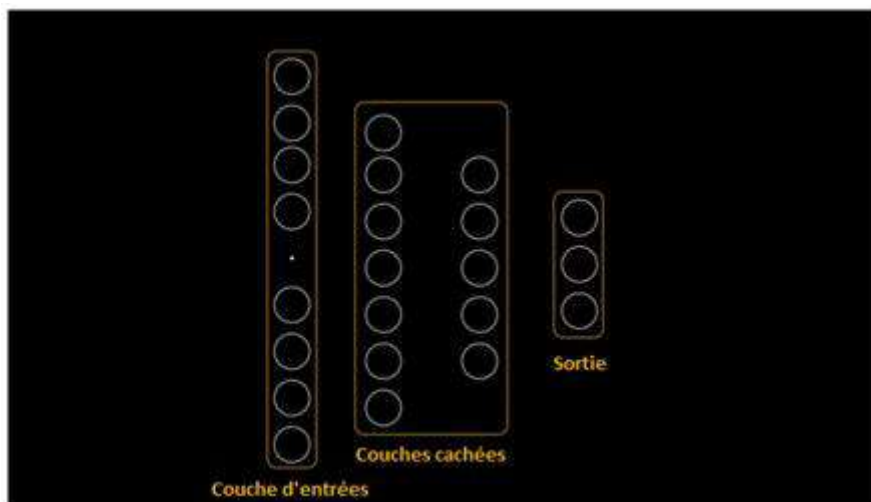


Figure 1.9 : Architecture d'un RNA

- **6.7.3 Fonctionnement des réseaux de neurones artificiels** : Le fonctionnement des RNA est simple. Premièrement, ils prennent des entrées de données. Ils s'entraînent ensuite à reconnaître les traits dans ces données. Enfin, ils produisent une prédiction ou un nouvel ensemble de données.

Pour entraîner les RNA la première chose qui est nécessaire sont les données, et dans la plupart des cas, d'énormes quantités de données sont indispensables pour obtenir de bons résultats. Une fois les données acquises, elles doivent être divisées en un ensemble d'entraînement et un ensemble de test (qui seront utilisés pour calculer la précision des RNA plus tard). Après cela, il y a deux étapes principales dans la phase d'entraînement qui sont le Feedforward et la Backpropagation :

- **6.7.3.1 Feedforward** : D'abord, les données sont transmises au réseau (via la couche d'entrée) où chaque entrée (x_i) est multipliée par un poids aléatoire (W_i). Ensuite, le résultat de ($W * x$) qui est le produit de chaque entrée et son poids est ajoutée à une valeur (b) appelée biais. Enfin, une fonction d'activation est appliquée au résultat.

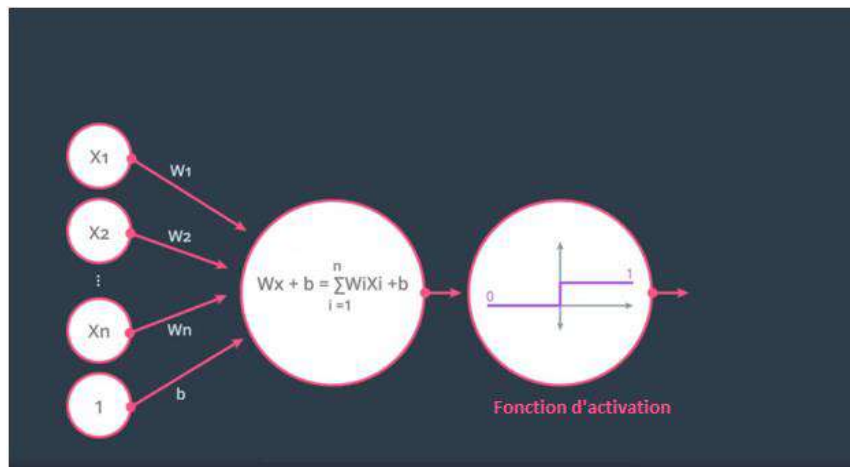


Figure 1.10 : Feedforward dans un neurone

Ensuite, la sortie de chaque neurone devient l'entrée des neurones dans la couche suivante et le même processus est répété jusqu'à la couche finale qui émet une prédiction dénoté \bar{y} .

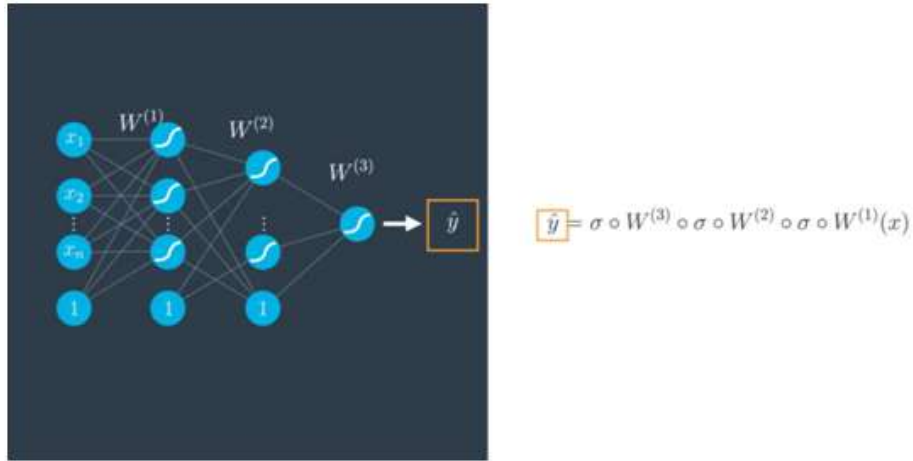


Figure 1.11 : Feedforward dans un RNA

- 6.7.3.2 Backpropagation** : Après le FeedForward vient l'étape de Backpropagation dans laquelle se fait la comparaison entre la prédiction \bar{y} et l'étiquette réelle y . Après cela, les poids (W) sont misent à jour mais cette fois pas au hasard. Tout d'abord, pour chaque neurone de la couche de sortie, la fonction d'erreur est calculée par la formule :

$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$

Où m est le nombre d'échantillons (exemples) dans l'ensemble de données (sélectionnée pour l'apprentissage). y_i est l'étiquette (valeur correcte) et \bar{y}_i est la prédiction. Ensuite, le gradient qui est la dérivée partielle de E par rapport à chaque poids (W_{kij}) où k est la couche, i le neurone et j l'entrée est calculé :

$$\nabla E = \left(\frac{\partial E}{\partial W_{11}^{(1)}}, \dots, \frac{\partial E}{\partial W_{31}^{(2)}}, \frac{\partial E}{\partial W_{ij}^{(k)}} \right)$$

Après cela, la mise à jour des poids en soustrayant la dérivée partielle par rapport à ce poids multipliée par un taux d'apprentissage α (learning rate) :

$$W_{ij}'^{(k)} \leftarrow W_{ij}^{(k)} - \alpha \frac{\partial E}{\partial W_{ij}^{(k)}}$$

Un processus Feedforward et une BackPropagation sont appelés une époque (epoch en Anglais). Pour obtenir de bons résultats, il est généralement nécessaire d'entraîner les RNA sur plusieurs époques.

Une fois que le processus d'entraînement se termine, le RNA est testé sur des échantillons de données qui n'ont pas été utilisés dans le processus d'entraînement pour évaluer son efficacité.

7. Conclusion :

L'apprentissage automatique est le présent et l'avenir de l'informatique, et la plupart des percées technologiques récentes ont été réalisées avec cet outil qui prouve sa puissance et son efficacité jour après jour.

Dans ce chapitre nous avons défini l'apprentissage automatique ainsi que ses paradigmes. Nous avons aussi mentionnés un ensemble de ses algorithmes tel que les réseaux de neurones artificiels. Dans le prochain chapitre nous allons parler d'un type de réseaux de neurones qui sont largement utilisées dans le domaine de vision par ordinateur et plusieurs autres domaines qui sont les réseaux de neurones convolutionnels.

Chapitre 02:

Les réseaux de

Neurones

Convolutionnels

1. Introduction :

A partir de l'année 2012, une révolution se produit grâce à un algorithme de Deep Learning (apprentissage profond) qui a explosé tous les records. Il s'agit d'un réseau de neurones convolutif appelé AlexNet [6], et depuis, les réseaux de neurones convolutionnels sont devenus dominants dans diverses tâches de vision par ordinateur. Ils suscitent un intérêt dans une variété de domaines, y compris la classification d'images, les moteurs de recherche, les systèmes de recommandation, les réseaux sociaux, la reconnaissance faciale et d'autres.

Les réseaux de neurones convolutionnels sont conçus pour apprendre automatiquement et de manière adaptative les caractéristiques et les hiérarchies spatiales d'entités en utilisant plusieurs blocs de construction, tels que des couches de convolution et des couches entièrement connectées.

Dans ce chapitre nous donnons une perspective sur les concepts de base des réseaux de neurones convolutionnels et leurs fonctionnements.

2. Définition :

Un réseau de neurone convolutionnel (ConvNet / CNN) est un algorithme d'apprentissage profond (deep learning) qui peut prendre une image comme entrée, attribuer une importance (poids et biais apprenants) à divers aspects / objets de l'image et être capable de différencier les uns des autres. Le prétraitement requis dans un ConvNet est beaucoup plus faible par rapport à d'autres algorithmes de classification. Alors que dans les méthodes primitives, les filtres sont conçus à la main, avec un entraînement suffisant, les réseaux de neurones convolutionnels ont la capacité d'apprendre ces filtres / caractéristiques eux-mêmes. L'architecture d'un ConvNet est analogue à celle du modèle de connectivité des neurones dans le cerveau humain et a été inspirée par l'organisation du Visual Cortex. Les neurones individuels ne répondent aux stimuli que dans une région restreinte du champ visuel. Une collection de ces champs se chevauche pour couvrir toute la zone visuelle [30].

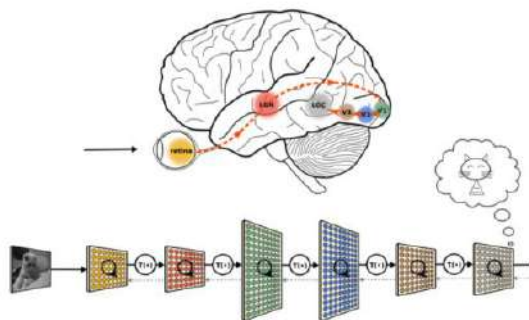


Figure 2.1 : Les ConvNet et le cerveau humain

3. Fonctionnement des réseaux de Neurones Convolutionnels :

Un réseau neuronal convolutionnel se compose d'une couche d'entrée et d'une sortie, ainsi que de plusieurs couches cachées. Les couches cachées d'un RNC consistent généralement en une série de couches convolutionnelles qui sont convoluées avec une multiplication ou un autre produit scalaire. Les couches convolutionnelles sont généralement suivies d'une fonction d'activation (généralement la fonction RELU) et ensuite suivie de convolutions supplémentaires telles que des couches de mise en commun, des couches entièrement connectées et des couches de normalisation, appelées couches cachées car leurs entrées et sorties sont masquées par la fonction d'activation et la convolution finale.

3.1 Les couches convolutionnelles : Une couche convolutionnelle au sein d'un réseau de neurones est caractérisée par des filtres (ou noyaux/kernels). Plusieurs filtres de convolutions parcourent l'image et calculent un produit scalaire. Chaque filtre extrait différentes caractéristiques de l'image. Une multiplication est effectuée entre les valeurs des pixels de l'image qui correspondent à la taille du noyau et le noyau lui-même. Ce qui donne à la fin une matrice ayant la même taille du filtre.

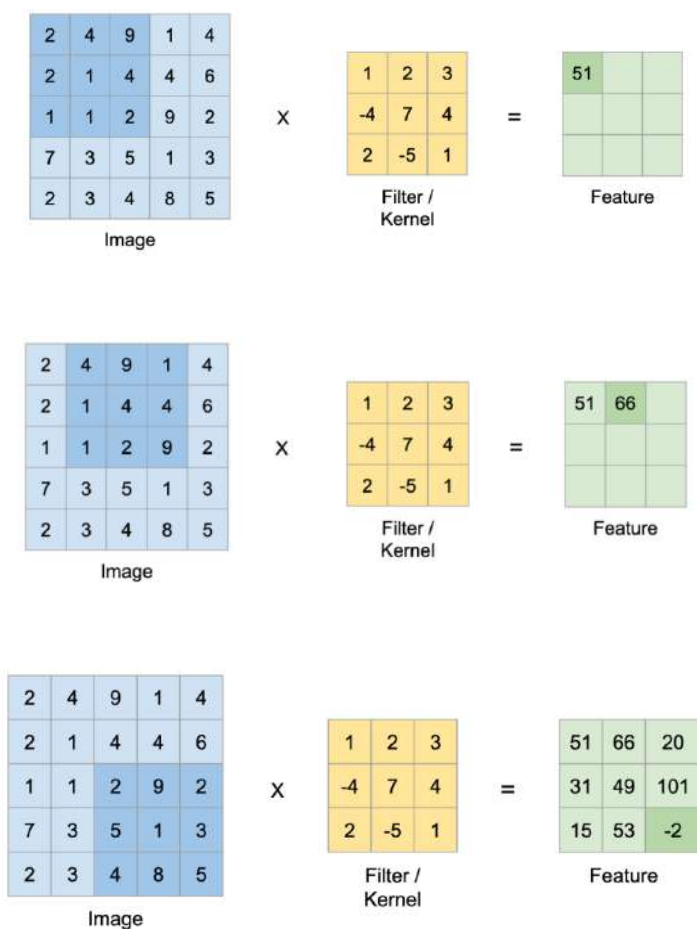


Figure 2.2 : Couche convolutionnelle

3.2 La couche pooling : Semblable à la couche convolutionnelle, la couche pooling est responsable de la réduction de la taille spatiale de l'entité convolutionnée. Il s'agit de diminuer la puissance de calcul requise pour traiter les données grâce à la réduction de dimensionnalité. En outre, elle est utile pour extraire des caractéristiques dominantes qui sont invariantes en rotation et en position, gardant ainsi l'efficacité du processus d'apprentissage du modèle.

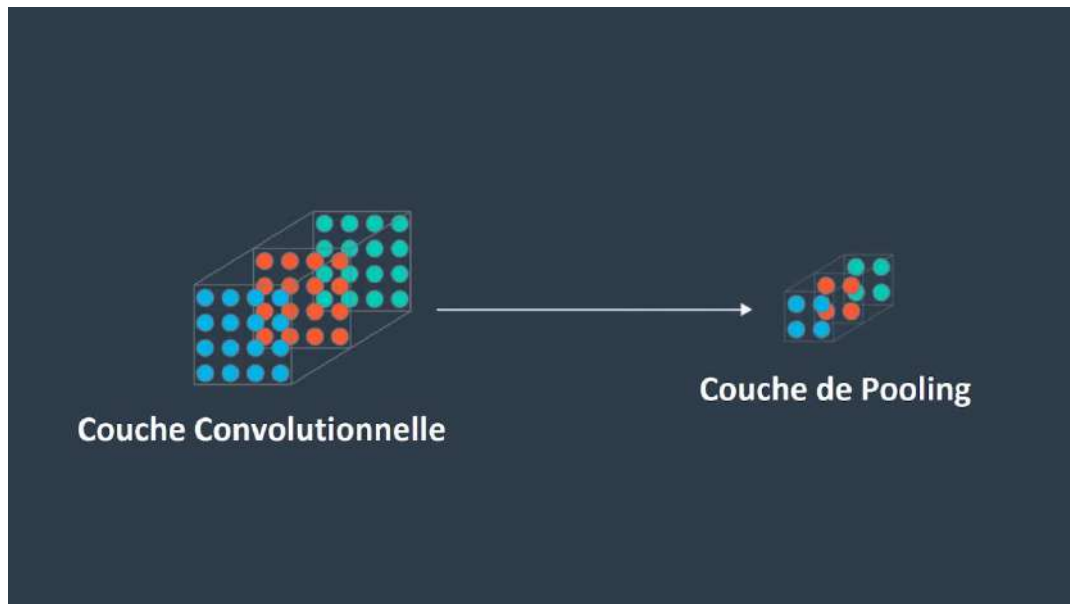


Figure 2.3 : Couche Pooling

Il existe deux types de couche de pooling : pooling maximal (Max Pooling) et le pooling moyen (Average Pooling). Max Pooling renvoie la valeur maximale de la partie de l'image couverte par le filtre.

| | | | | |
|-----|----|----|-----|-----|
| 21 | 59 | 37 | -19 | 2 |
| 30 | 51 | 66 | 20 | 43 |
| -14 | 31 | 49 | 101 | -19 |
| 59 | 15 | 53 | -2 | 21 |
| 49 | 57 | 64 | 76 | 10 |

Matrice convoluée

| | | | |
|----|----|-----|-----|
| 59 | 66 | 66 | 43 |
| 51 | 66 | 101 | 101 |
| 59 | 53 | 101 | 101 |
| 59 | 64 | 76 | 76 |

Matrice après la max pooling

Figure 2.4 : Max Pooling

D'un autre côté, le regroupement moyen renvoie la moyenne de toutes les valeurs de la partie de l'image couverte par le noyau. Max Pooling fonctionne également comme un supprimeur de bruit. Il supprime complètement les activations bruyantes et effectue également le débruitage avec réduction de dimensionnalité. D'un autre côté, l'Average Pooling effectue simplement une réduction de dimensionnalité comme mécanisme de suppression du bruit. Par conséquent, le Max Pooling est considéré plus efficace que l'Average Pooling.

3.3 Fonction d'activation : Les fonctions d'activation introduisent une non-linéarité au modèle qui lui permet d'apprendre des mappages fonctionnels complexes entre les entrées et les variables de réponse. Il existe plusieurs fonctions d'activation différentes comme sigmoïde, softmax, ReLU, etc. La fonction la plus utilisée avec les Réseaux de neurones convolutionnels est ReLU qui est une fonction linéaire par morceaux, elle sort directement l'entrée si elle est positive, c'est-à-dire > 0 , sinon, elle sortira zéro [31].

$$\text{ReLU}(x) = \max(0, x)$$

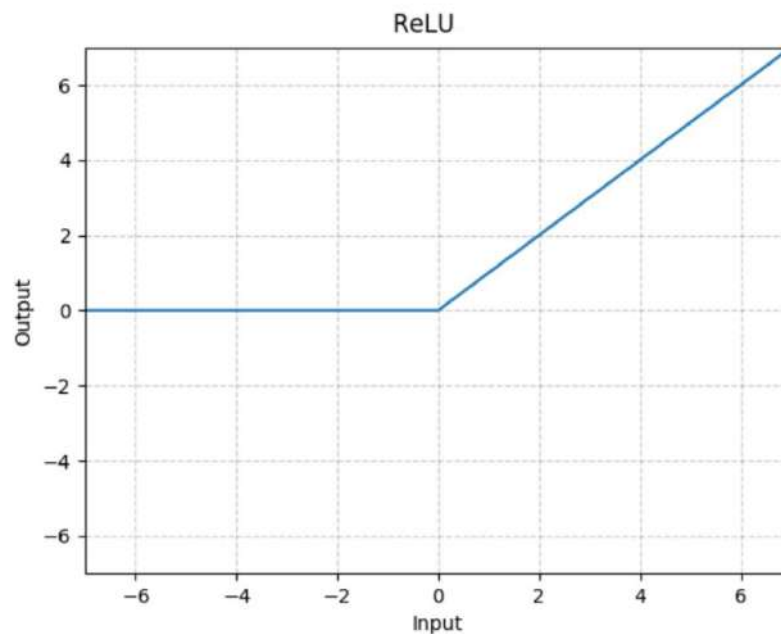


Figure 2.5 : Fonction RELU

La fonction Relu peut être utilisée avant la couche pooling ou après la couche pooling. Ils satisfont donc la propriété communicative et peuvent être utilisés dans les deux sens. En pratique, la fonction d'activation ReLU est appliquée juste après une couche de convolution, puis cette sortie est regroupée au maximum.

3.4. Couche entièrement connecté : Dans une couche entièrement connectée, les nœuds de la couche d'entrée sont connectés à chaque nœud de la deuxième couche. Une ou plusieurs couches entièrement connectées sont utilisées à la fin d'un ConvNet. L'ajout d'une couche entièrement connectée permet d'apprendre des combinaisons non linéaires des fonctionnalités de haut niveau générées par les couches convolutionnelles.

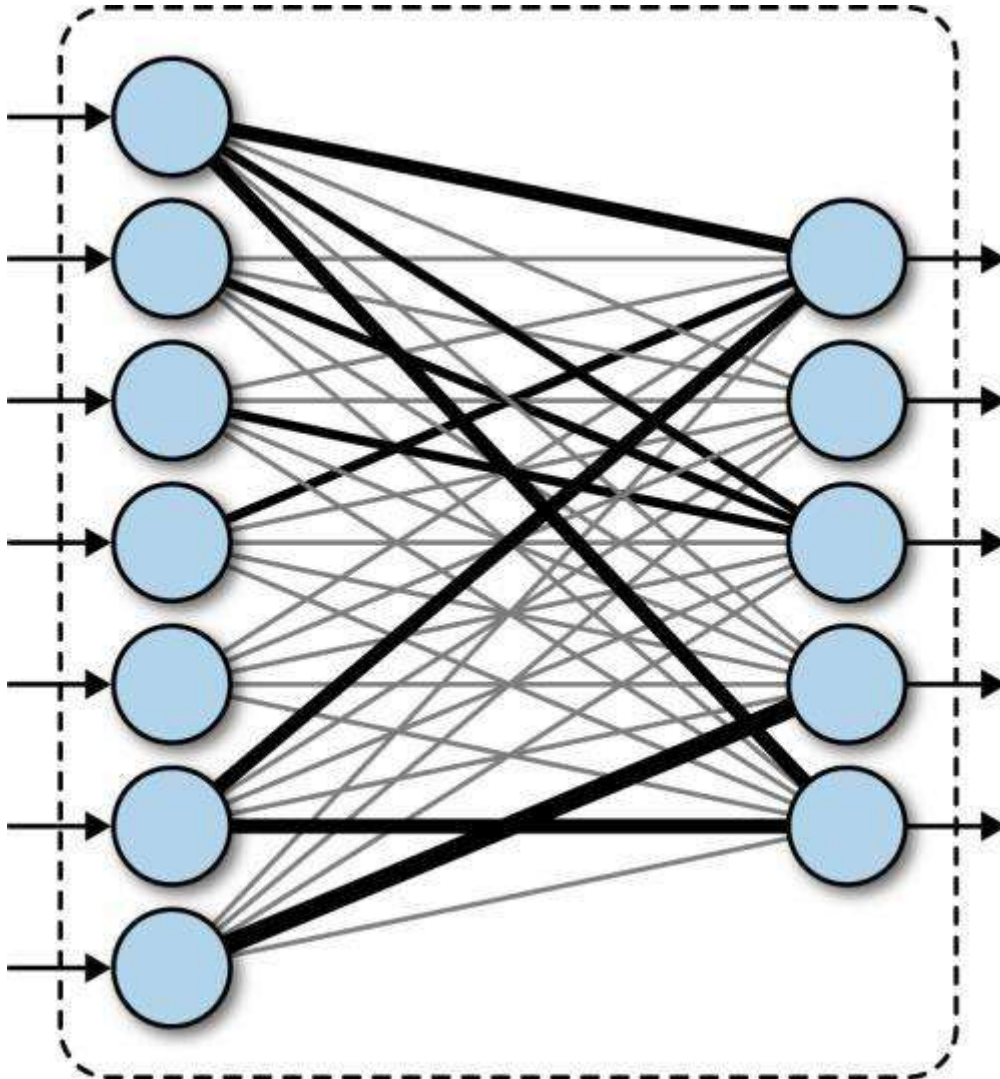


Figure 2.6 : Couche entièrement connectée

À la dernière couche entièrement connectée, la taille de sortie est choisie en fonction de l'application. Pour une classification binaire de classes la dernière couche sera de taille 2. Pour classer les chiffres manuscrits, la dernière couche sera de taille 10, c'est-à-dire un nœud pour chaque chiffre. Ensuite un softmax de la sortie est pris qui donne un vecteur à 10 dimensions contenant des probabilités (un nombre allant de 0 à 1) pour chacun des chiffres [32].

4. Les ConvNet les plus connus :

Il existe différentes architectures de Réseaux de Neurones Convolutionnels disponibles qui ont été essentielles dans la construction d'algorithmes qui alimentent et alimenteront l'IA dans un avenir prévisible. Notamment :

4.1 LeNet: il s'agit d'une structure de réseau de neurones convolutionnel proposée par Yann LeCun et Al. en 1989. L'application la plus classique de LeNet est la reconnaissance d'images de chiffres simples. L'article *Backpropagation Applied to Handwritten Zip Code Recognition* [5] montre comment de telles contraintes peuvent être intégrées dans un réseau de backpropagation à travers l'architecture du réseau, et il avait été appliqué avec succès à la reconnaissance des chiffres du code postal manuscrits fournis par le service postal américain.

4.2 AlexNet : Un réseau de neurones convolutionnel conçu par Alex Krizhevsky avec Ilya Sutskever et le conseiller doctoral de Krizhevsky Geoffrey Hinton. AlexNet a participé au défi de reconnaissance visuelle à grande échelle d'ImageNet le 30 septembre 2012. Le réseau a réalisé une erreur dans le top 5 de 15,3%, soit plus de 10,8 points de pourcentage de moins que celle du finaliste. Le premier résultat du document original était que la profondeur du modèle était essentielle pour ses hautes performances, qui étaient coûteuses en termes de calcul, mais rendues possibles en raison de l'utilisation d'unités de traitement graphique (GPU) pendant l'entraînement. [6]

AlexNet est considéré comme l'un des articles les plus influents publiés en vision par ordinateur, ayant stimulé de nombreux autres articles publiés utilisant des ConvNet et des GPU pour accélérer l'apprentissage en profondeur.

4.3 GoogleNet : GoogLeNet est un réseau de neurones convolutionnel de 22 couches de profondeur. Le réseau formé sur la base d'images ImageNet classe les images en 1000 catégories d'objets, comme le clavier, la souris, le crayon et de nombreux animaux. Ce réseau a appris différentes représentations de caractéristiques pour un large éventail d'images. [7]

Il est possible de recycler un réseau GoogLeNet pour effectuer une nouvelle tâche à l'aide de l'apprentissage par transfert. Lors de l'apprentissage par transfert, l'approche la plus courante consiste à utiliser des réseaux pré-formés sur l'ensemble de données ImageNet.

5. Conclusion :

Dans ce chapitre, nous avons présenté les réseaux de neurones convolutionnels. Ces réseaux sont capables d'extraire des caractéristiques d'images présentées en entrée et de classifier ces caractéristiques.

Malgré leur puissance, les réseaux de neurones convolutionnels présentent un certain nombre de limitations. Premièrement, il nécessite un grand nombre de données pour donner de bons résultats. Deuxièmement, les hyper paramètres du réseau sont difficiles à évaluer a priori. En effet, le nombre de couches, les nombre de neurones par couche ou encore les différentes connexions entre couches sont des éléments cruciaux et essentiellement déterminés par une bonne intuition ou par une succession de tests/calcul d'erreurs ce qui est peut être coûteux. Ils existent cependant des solutions pour améliorer le nombre d'images dans la base d'images (data augmentation) qui consiste à créer des répliques des images déjà existantes en changeant l'angle de certaines images.

Dans le chapitre suivant nous allons présenter un état de l'art sur les approches utilisées dans le domaine de détection d'âge.

Chapitre 03:
Etat de l'art des
approches
utilisées pour la
détection d'âge.

1. Introduction :

La détection d'âge est l'un des sujets brûlants dans le monde de la recherche. De nombreux chercheurs travaillent à apporter un système efficace et efficient pour développer un système enrichi qui permet de détecter automatiquement l'âge d'une personne avec une efficacité respectable.

Dans ce chapitre nous allons mentionner les approches utilisées dans le domaine de détection d'âge.

2. Les approches utilisées dans le domaine de détection d'âge :

Différentes approches ont été utilisées par les chercheurs à travers les années pour obtenir de bons résultats dans le domaine de détection d'âge. Nous présentons ici les approches les plus connus dans ce domaine.

2.1. L'approche basée sur la théorie du développement cranio-facial :

Présenté par Young H. Kwon et Niels da Vitoria Lobo en 1996 [8]. Cette approche présente une théorie et des calculs pratiques pour la classification visuelle de l'âge à partir d'images faciales. La théorie n'a été mise en œuvre que pour classer les images d'entrée dans l'un des trois groupes d'âge: les bébés, les jeunes adultes et les personnes âgées. Les calculs sont basés sur la théorie du développement cranio-facial et l'analyse des rides cutanées. Dans l'implémentation, les caractéristiques principales du visage sont trouvées en premier, suivies d'une analyse des caractéristiques secondaires. Les principales caractéristiques sont les yeux, le nez, la bouche, le menton, le sommet virtuel de la tête et les côtés du visage. À partir de ces caractéristiques, les ratios qui distinguent les bébés des jeunes adultes et des personnes âgées sont calculés. Dans l'analyse d'entités secondaires, une carte de géographie des rides est utilisée pour guider la détection et la mesure des rides. L'indice de rides calculé est suffisant pour distinguer les personnes âgées des jeunes adultes et des bébés. Une règle de combinaison pour les ratios et l'indice de rides permet ainsi de catégoriser un visage en l'une des trois classes. Des résultats utilisant des images réelles sont présentés. Il s'agit du premier travail impliquant la classification par âge et le premier travail qui réussit à extraire et à utiliser les rides naturelles.

2.1.1 Aperçu de l'approche :

Il s'agit d'effectuer 4 tâches successives. Premièrement, trouver les traits du visage. Deuxièmement, calculer les ratios des traits du visage. Troisièmement, l'analyse des rides. Enfin, combiner les calculs effectués précédemment pour conclure la catégorie d'âge. Nous parlons brièvement de chacune des tâches.

2.1.1.1 Détection de caractéristiques faciales : La localisation des traits du visage est effectuée par étapes. À chaque étape, un paramètre de caractéristique faciale particulier est trouvé. La position centrale de la tête dans l'image est initialisée manuellement, avec une marge d'erreur importante. La phase initiale de recherche d'ovale trouve un ovale qui correspond le mieux au visage / à la tête, et par conséquent la position centrale de la tête est automatiquement mise à jour. La phase de recherche du menton trouve le meilleur menton dans la zone rectangulaire spécifiée par les paramètres ovales. La phase de recherche des côtés du visage trouve les côtés gauche et droit du visage dans la zone spécifiée par les paramètres menton et ovale. Le sommet virtuel de la tête est alors acquis à partir de l'ovale généré par le menton et les deux côtés du visage. Le paramètre menton, s'il est trouvé de manière robuste, est ensuite utilisé pour affiner l'ovale initial. Sinon, l'ovale initial est utilisé pour les étapes restantes. L'étape d'attraction d'iris place les centres d'iris gauche et droit du modèle près des centres d'iris respectifs dans l'image spécifiée par le paramètre ovale. L'étape d'ajustement de l'iris tente d'ajuster plus précisément le contour de l'iris en estimant le rayon de l'iris et en améliorant simultanément les estimations des deux positions centrales de l'iris. La phase de recherche de bouche trouve la position du centre de la bouche et enfin la phase de recherche de nez trouve la position du bas du nez.

2.1.1.2 Evaluation de différents ratios de caractéristiques faciales : Une fois les principales caractéristiques faciales localisées, elles peuvent être utilisées pour calculer les ratios de classification par âge. Six ratios sont évalués et leur utilité est discutée. Une base de données faciales de 47 visages, comprenant des bébés, des personnes âgées et des adultes jeunes / d'âge moyen, est utilisée pour cette étude. La figure 3.1 explique graphiquement ces ratios. Les cinq premiers ratios sont recalculés après la suppression des données jugées défavorables en raison de l'expression faciale ou de la rotation de la tête. Le seuil bimodal pour chaque rapport est calculé selon la méthode d'Otsu [9].

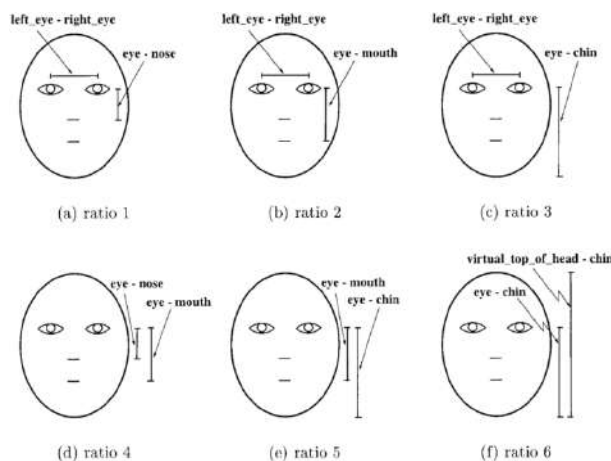


Figure 3.1 : Les ratios utilisés

2.1.1.3 L'analyse des rides : Une fois que les principales caractéristiques ont été trouvées pour le visage, la carte géographique des rides est utilisée pour déterminer où les snakelets doivent être déposés pour rechercher les rides. Étant donné que la résolution d'une image de 256 x 256 ne capture aucune information sur les rides, il est nécessaire de zoomer sur les zones représentées par la géographie des rides pour capturer plus de détails. Avec un zoom à contrôle actif, la tâche de zoom avant peut être rendue automatique. Une autre possibilité consiste à prendre des images de plus haute résolution au départ et à rechercher des rides dans les zones représentées par la géographie des rides. Pour détecter des caractéristiques individuelles et en effectuer une analyse géométrique. Il est nécessaire d'effectuer une correspondance entre un modèle déformable d'une ride et l'image. Le mode de réalisation élémentaire utilisé ici consiste à confirmer simplement que les candidats pour les morceaux de rides ne reposent pas tous sur une seule courbe sous-jacente. Si tous les candidats pour les plis se situent sur une même courbe (avec très peu de valeurs aberrantes), les candidats ne sont pas étiquetés comme des rides. De cette manière, les courbes résultant de marques d'ombrage ou de taches de peau bruyantes ne sont pas étiquetées par erreur comme des rides.

2.1.1.4 Combinaison des ratios et les rides : Avec seulement trois groupes d'âge à classer, la règle de combinaison est simple. Si les proportions sont semblables à celles d'un bébé et qu'aucune ride n'est détectée, le visage est étiqueté bébé. Si des rides sont détectées et que les proportions sont de type adulte, le visage est étiqueté adulte. Pour tous les autres cas, le visage est étiqueté comme un adulte d'âge moyen. Pour décider si les ratios sont semblables à ceux d'un bébé, les ratios 1 et 2 sont utilisés, car ils ont donné les pourcentages corrects les plus élevés. Ce sont également les deux ratios avec l'écart le plus large entre leurs moyennes pour les groupes.

2.2. L'approche basé sur un modèle de vieillissement :

Présenté pour la première fois en 2007 par Xin Geng, Zhi-Hua Zhou et Kate Smith-Miles [10]. L'idée de base est de modéliser le modèle de vieillissement, qui est défini comme la séquence d'images de visage d'un individu particulier triées par ordre temporel, en construisant un sous-espace représentatif. Le modèle de vieillissement approprié pour une image de visage est déterminé par la projection dans le sous-espace qui peut reconstruire l'image de visage avec une erreur de reconstruction minimale, tandis que la position de l'image de visage dans ce modèle de vieillissement indiquera alors son âge.

2.2.1 Aperçu de l'approche :

Cette approche basée sur la fonction de vieillissement considère l'estimation de l'âge comme un problème de classification classique: les données sont les images de visage, la cible est leur étiquette d'âge.

2.2.1.1 Sous-espace du modèle du vieillissement : Un modèle représentatif des schémas de vieillissement peut être construit par l'approche de la théorie de l'information du codage et du décodage. Une technologie largement adoptée utilise ACP [10] pour construire un sous-espace qui capture la principale variation de l'ensemble de données. La projection dans le sous-espace est calculée par :

$$y = W^T(x - u)$$

Où u est le vecteur moyen de x et $W^T = W^{-1}$ est la transposée de W , qui est composé par les vecteurs propres orthogonaux de la matrice de covariance de x .

Pendant le processus d'entraînement les faces manquantes dans les modèles de vieillissement de la formation peuvent être apprises simultanément en reconstruisant l'ensemble des vecteurs de modèle de vieillissement via :

$$\bar{X}_k = u \cdot W_{y_k}$$

Où \bar{X}_k est la reconstruction de X_k . Le but ici est de trouver un W qui minimise l'erreur de reconstruction moyenne des caractéristiques disponibles. L'algorithme est appliqué à la base de données de vieillissement FG-NET [8]. Étant donné que des variations remarquables autres que les effets de vieillissement existent dans la base de données de vieillissement FG-NET et que l'extracteur de fonctionnalités ne les traite pas séparément, certaines faces générées présentent des variations notables d'expression, de pose ou d'éclairage. Ces variations peuvent être traitées, en appliquant LDA aux paramètres du modèle d'apparence. Ainsi, cet algorithme d'apprentissage peut également être utilisé pour simuler les effets du vieillissement sur les visages humains.

2.2.1.2 Estimation d'âge :

La tâche d'estimation de l'âge est basée sur une entrée d'un visage unique et attend une sortie d'âge unique. Étant donné une image de visage I jamais vue auparavant, son vecteur de caractéristiques b est d'abord extrait par l'extracteur de caractéristiques en deux étapes. La première consiste à trouver un modèle de vieillissement approprié pour l'image I . Chaque point dans le sous-espace correspond à un modèle de vieillissement. Ainsi, le modèle de vieillissement approprié pour I peut être sélectionné en trouvant un point dans le sous-espace qui peut le mieux reconstruire b , c'est-à-dire en minimisant l'erreur de reconstruction. Cependant, sans connaître la position de I dans le schéma de vieillissement, qui doit être déterminée dans la deuxième étape, l'erreur de reconstruction ne peut pas être réellement calculée. Ainsi, I est placé à chaque position possible dans le modèle de vieillissement, obtenant p vecteurs de modèle de vieillissement $z_j (j = 1 \dots p)$ en plaçant b à la position j dans z_j , b est la seule fonction disponible dans z_j . Ensuite, la projection y_r qui peut reconstruire b avec une erreur de reconstruction minimale sur toutes les p positions possibles est déterminée par :

$$\varepsilon^a(j) = (b - \mu_{(j)} - \mathbf{W}_{(j)}y_j)^T (b - \mu_{(j)} - \mathbf{W}_{(j)}y_j),$$

où $u(j)$ est la partie dans u et $W(j)$ est la partie dans W qui correspond à la position j . Ensuite, la projection y_r qui peut reconstruire b avec une erreur de reconstruction minimale sur toutes les p positions possibles et qui est déterminée par :

$$r = \arg \min(\varepsilon^a(j)).$$

Ainsi, le schéma de vieillissement approprié pour I est z_r . L'étape 2 devient ensuite triviale car r indique également la position de I dans z_r . Enfin, l'âge associé à la position r est retourné comme l'âge estimé de I . En tant que sous-produit de l'estimation de l'âge, le vecteur du modèle de vieillissement entier peut être reconstruit en $W y_r$, qui peut être utilisé pour simuler des visages à différents âges du sujet dans I . Pendant le processus d'estimation de l'âge le modèle de vieillissement approprié pour l'image de test est généré sur la base à la fois du sous-espace du modèle de vieillissement et de la caractéristique de l'image du visage. Le sous-espace définit la tendance générale du vieillissement et la fonction d'image du visage représente les facteurs personnalisés. En plaçant le vecteur d'entité à différentes positions, des modèles de vieillissement candidats spécifiés pour la face de test sont générés. Parmi ces candidats, un seul correspond à la tendance générale au vieillissement, qui peut être détectée via une erreur de reconstruction minimum par le sous-espace du schéma de vieillissement. En même temps, la position de l'image de test dans ce modèle de vieillissement peut être déterminée et le résultat final est l'âge estimé.

2.3.1. L'approche basée sur l'apprentissage automatique :

Le premier travail basé sur l'apprentissage automatique a été publié en 2014 par Dong Yi, Zhen Lei, et Stan Z. Li [11] qui ont appliqué les Réseaux de Neurones Convolutionnels sur la base d'image MORPH. Comparé aux autres approches, les Réseaux de Neurones Convolutionnels ont des paramètres apprenables et il n'y a aucun écart entre l'étape d'extraction des caractéristiques et l'étape de régression.

Le dernier travail publié est celui de Ke Zhang Na Liu, Xingfang Yuan, Xinyao Guo, Ce Gao, Zhenbing Zhao et Zhanyu Ma [12]. La méthode proposée consiste à ajouter des LSTM (Long Short Memory Cells) à un réseau de neurones convolutionnels pour améliorer les capacités d'extraction des caractéristiques.

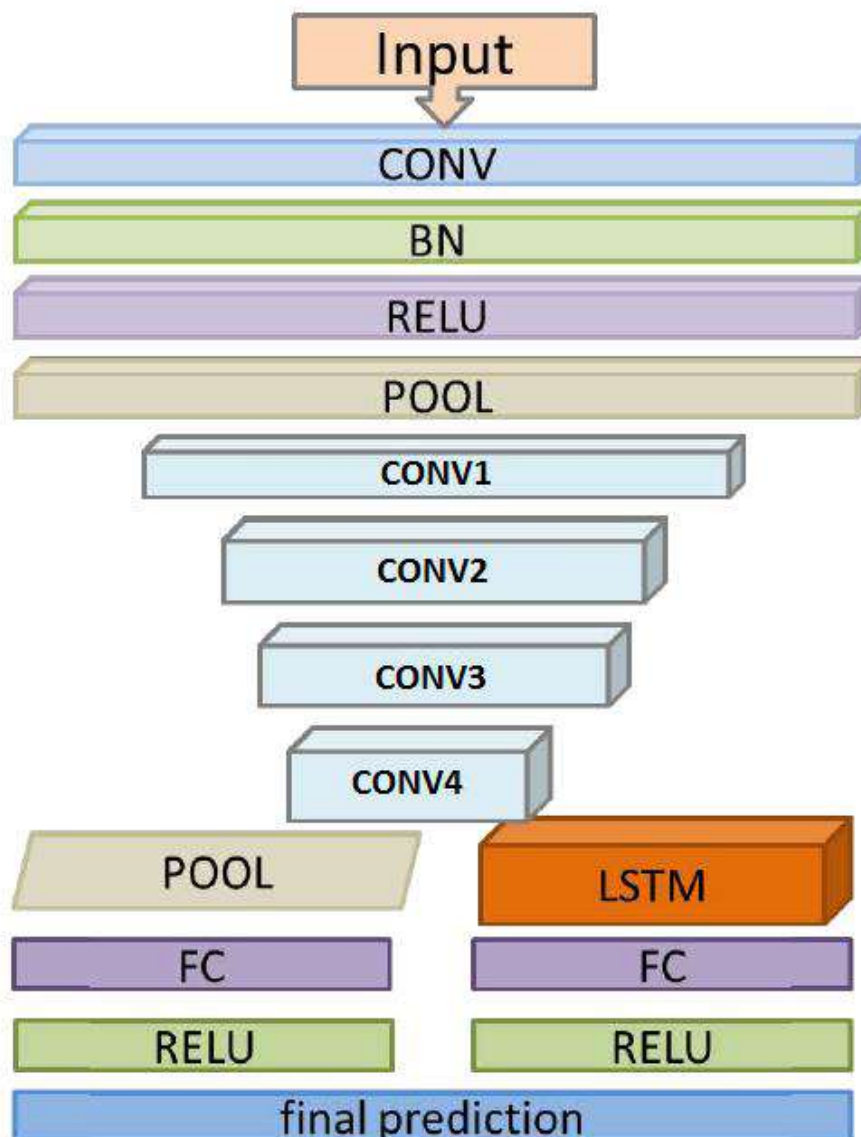


Figure 3.2 : Le modèle Ke Zhang Na Liu, Xingfang Yuan, Xinyao Guo, Ce Gao, Zhenbing Zhao et Zhanyu Ma.

3. Conclusion :

Dans ce chapitre nous avons présenté l'état de l'art du domaine de détection d'âge qui est un domaine qui attire plusieurs chercheurs depuis des années. Plusieurs approches et théories ont été proposées pour atteindre des résultats optimaux dans ce domaine. Après la grande montée de l'apprentissage automatique il est devenu possible d'avoir une estimation d'âge avec une bonne précision. Dans le chapitre suivant, nous allons présenter la partie pratique de notre travail. La conception de l'architecture et l'implémentation de notre application qui va permettre de classer les personnes en deux catégories d'âge à partir des traits de leurs visages seront expliquées.

Chapitre 04 :

Conception et

Implémentation

1. Introduction :

Dans ce chapitre nous allons expliquer la conception et le développement de notre projet. Pour avoir un outil qui peut distinguer les personnes adultes des personnes mineurs nous allons utiliser les notions mentionnées dans le deuxième chapitre pour construire un réseau de neurones convolutionnel.

Nous allons utiliser plusieurs outils de programmation pour organiser la base d'images, implémenter le réseau de neurones et ensuite l'entraîner.

2. Conception :

La conception de notre modèle a été inspiré des travaux « Age Estimation by Multi-scale Convolutional Network » [11] et « Fine-Grained Age Estimation in the Wild with Attention LSTM Networks » [12] qui ont appliqué des CNN pour l'estimation d'âge. Dans le premier travail le CNN donne comme sortie une valeur numérique (l'âge exacte) avec un taux d'erreur un peu élevée car la conception d'un modèle qui donne l'âge exact d'une personne reste très difficile et compliqué. En plus, cette approche multi-scale nécessite un prétraitement qui peut être coûteux en termes de temps. Dans le deuxième travail le CNN effectue une classification avec 8 sorties différentes (0-2, 4-6, 8-13, 15-20, 25-32, 38-43, 48-53, 60-100) et la précision de ce modèle sur la base de données IMDB-WIKI est très élevée (66.03%) par rapport aux autres modèles mais elle reste insuffisante dans le cas d'une application critique. En plus, le modèle utilisée est un peu volumineux en termes de mémoire.

Dans notre projet nous travaillons sur une classification binaire (2 classes) pour répondre à notre problématique en distinguant les adultes et les mineurs et pour obtenir un taux de précision élevé car il est théoriquement plus facile pour un CNN d'apprendre les différentes caractéristiques entre les mineurs et les adultes car ils sont plus distinctifs.

2.1. L'architecture de notre modèle :

A partir des deux modèles mentionnés avant, nous proposons notre propre modèle qui se compose de deux couches convolutionnelles. L'idée est de concevoir un modèle qui peut donner de bons résultats dans un temps plus rapide par rapport autres modèles.

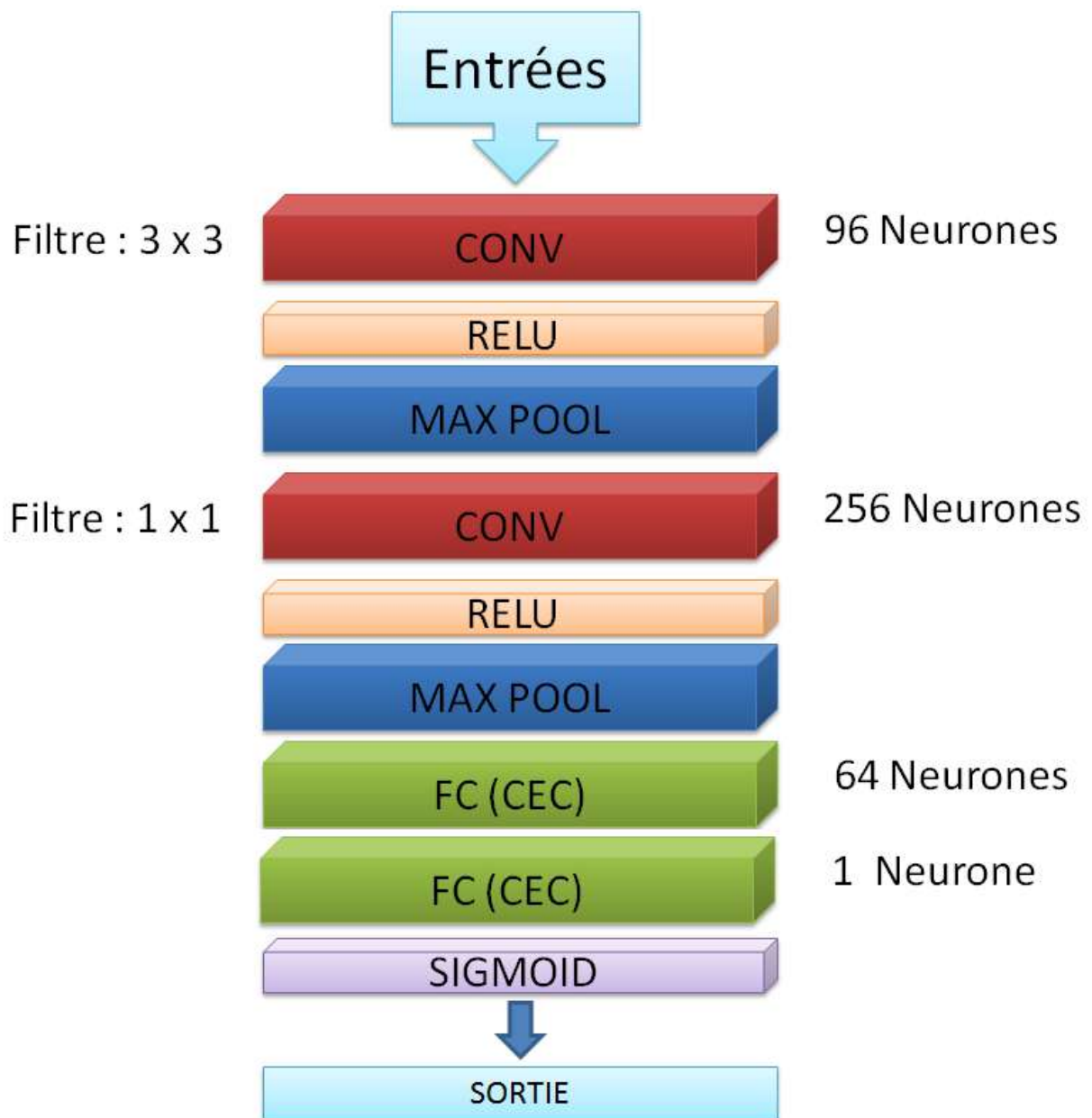


Figure 4.1 : L'architecture de notre modèle

2.1.1 Les entrées : La première couche est la couche d'entrées qui est une matrices de taille 200x200 (chaque case représente un pixel de l'image). On a choisi cette valeur car ça nous permet d'avoir une matrice carré ayant assez d'informations sur le contenu de l'image pour faciliter l'extraction des caractéristiques.

2.1.2 Couche convolutionnel : La deuxième couche est une couche convolutionnelle de 96 neurones avec un filtre 3x3. On a choisi un nombre de neurones pas très élevé avec un filtre un peu vague car cette couche va capter les informations les plus distinctifs dans le visage et éliminer le maximum de bruits.

2.1.3 Couche RELU : Après la couche conventionnelle nous ajoutons une fonction d'activation RELU pour la non-linéarité.

2.1.4 Couche Pooling : La couche MAX Pooling 2x2 réduit la taille spatiale de la représentation et diminue le nombre de paramètres et la charge des calculs dans notre réseau.

2.1.5 Couche convolutionnel : Ensuite nous ajoutons une autre couche convolutionnelle mais cette fois ci avec 256 neurones et un filtre 1x1. On a choisi un nombre de neurones plus élevée avec un filtre strict qui va réduire la dimensionnalité. Un nombre de neurones plus élevée que 256 mène à une surcharge de calcul qui va résulter en un sur surajustement (overfitting).

2.1.6 Couche RELU : Après, les gros calculs dans la couche convolutionnelle il nous faut encore une couche RELU pour augmenter la non-linéarité.

2.1.7 Couche Pooling : Nous ajoutons une autre couche MAX Pooling 2x2 pour la réduction de nombre de paramètres avant de les passer à la couche entièrement connectée.

2.1.8 Couche entièrement connectée : Ensuite nous ajoutons une couche entièrement connectées de 64 neurones à notre réseau qui permet de mélanger des signaux d'informations entre chaque dimension d'entrée et chaque classe de sortie. Nous avons choisi 64 neurones dans cette couche car il n'est pas nécessaire d'avoir un nombre très élevé de neurones vu que ça n'affecte pas le résultat final (Dans le contexte de notre travail seulement).

2.1.9 Couche entièrement connectées : Après, nous ajoutons une couche d'un seul neurone car il s'agit d'une classification binaire donc un seul neurone suffit pour obtenir un résultat finale.

2.1.10 Couche Sigmoid : Enfin nous mettons une fonction d'activation Sigmoid car il s'agit d'une classification binaire et cette fonction retourne une des valeurs entre 0 et 1. Donc pour un résultat inférieur à 0.5 la prédiction est adulte sinon mineur.

2.1.11 Sorties : Enfin la sortie de notre réseau sera la classification de l'âge de la personne dans l'image soit 0 pour adulte soit 1 pour mineur.

3. Implémentation :

Pour implémenter notre projet nous utilisons plusieurs langages et bibliothèques de programmations pour d'abord prétraités les données et ensuite entrainer (training) notre modèle avant de créer l'application.

3.1. Outils de programmation utilisés :

3.1.1 Python : Python est un langage de programmation interprété créée par Guido van Rossum et publiée pour la première fois en 1991, la philosophie de conception de Python met l'accent sur la lisibilité du code avec son utilisation notable d'espaces blancs importants. Ses constructions de langage et son approche orientée objet visent à aider les programmeurs à écrire du code clair et logique pour des projets à petite et à grande échelle. Python est typé dynamiquement et il prend en charge plusieurs paradigmes de programmation, y compris la programmation structurée (en particulier, procédurale), orientée objet et fonctionnelle. Python a été conçu à la fin des années 1980 comme un successeur du langage ABC. Python 2.0, sorti en 2000, a introduit des fonctionnalités telles que la compréhension des listes et un système de récupération de place capable de collecter des cycles de référence. Python 3.0, sorti en 2008, était une révision majeure du langage qui n'est pas complètement rétro compatible, et une grande partie du code Python 2 ne fonctionne pas sans modification sur Python 3. Le langage Python 2 a été officiellement interrompu en 2020. Python 2.7.18 est la dernière version de Python 2.7 et donc la dernière version de Python 2. Plus aucun correctif de sécurité ou autre amélioration ne sera publié pour cela. Avec la fin de vie de Python 2, seuls Python 3.5 et versions ultérieures sont pris en charge. [13]



Figure 4.2 : Logo du langage python

3.1.2 TensorFlow: TensorFlow est une bibliothèque logicielle gratuite et open-source. C'est une bibliothèque mathématique symbolique, qui est également utilisée pour des applications d'apprentissage automatique telles que les réseaux de neurones. Elle est utilisée à la fois pour la recherche et la production chez Google. TensorFlow a été développée par l'équipe de Google Brain pour une utilisation interne de Google. Elle a été publiée sous la licence Apache 2.0 le 9 novembre 2015. [14]



Figure 4.3 : Logo de Tensorflow

3.1.3 Keras: Keras est une bibliothèque de réseau de neurones open source écrite en Python. Conçu pour permettre une expérimentation rapide avec des réseaux de neurones profonds, elle se concentre sur la convivialité, la modularité et l'extensibilité. Elle a été développée dans le cadre de l'effort de recherche du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System) et son auteur principal et mainteneur est François Chollet, un ingénieur de Google. [15]



Figure 4.4 : Logo de Keras

3.1.4 PyQT: PyQt est un module Python de la boîte à outils GUI multiplateforme Qt, implémentée en tant que plug-in Python. PyQt est un logiciel gratuit développé par la firme britannique Riverbank Computing. PyQt prend en charge Microsoft Windows ainsi que diverses versions d'UNIX, y compris Linux et MacOS. PyQt implémente environ 440 classes et plus de 6 000 fonctions et méthodes [16]



Figure 4.5 : Logo de PyQt

3.1.5 Open Cv (Open Source Computer Vision Library) : est une bibliothèque de fonctions de programmation principalement destinées à la vision par ordinateur en temps réel. [17] Développé à l'origine par Intel, il a ensuite été soutenu par Willow Garage puis Itseez (qui a ensuite été acquis par Intel [18]). La bibliothèque est multiplateforme et gratuite.



Figure 4.6 : Logo d'Open CV

2.3. La base d'images :

La base d'images utilisée dans notre projet est IMDB-WIKI qui contient environ 10000 images faciales (avec des étiquettes de sexe et d'âge) de stars de cinéma, principalement des acteurs et actrices hollywoodiens ainsi que plusieurs célébrités de divers domaines, tels que les sports, la politique, les événements sociaux et l'industrie cinématographique [19]. Nous avons choisi cette base de données car elle offre beaucoup de diversités en terme d'ethnicité et de sexe ce qui permet à un notre modèle d'apprendre les caractéristiques d'âge dans les traits de visage de différents personnes quelques soit leur origines ou leur sexe.



Figure 4.7 : La base d'images IMDB-WIKI

2.4. Configuration utilisée dans l'implémentation :

La configuration du matériel utilisé dans notre implémentation est :

- MacBook Pro Retina i7 CPU 2.70 GHZ
- Carte graphique AMD Radeon R9 M370X
- RAM de taille 16 GO
- Disque dur SSD de taille 500 GO
- Système d'exploitation MAC OS High Sierra 10.13

2.5. Le prétraitement :

Pour la phase de prétraitement nous effectuons plusieurs tâches pour organiser les données de la base d'images et les préparer avant de les passer à notre modèle.

2.5.1 La sélection : Vu que la base d'images contient plusieurs images bruyantes qui ne montrent pas clairement les traits des visages des personnes, nous avons éliminés (manuellement) plusieurs images et nous avons gardées 7260 images.

2.5.2 Le recadrement : Nous avons recadré les images en gardant que le visage car c'est la partie sur laquelle nous allons travailler pour la classification d'âge. Pour cela nous utilisons le CascadeClassifier d'OpenCv.

```
def detect_face(image):
    face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

    face = face_cascade.detectMultiScale(gray, 1.1, 4, minSize=(150,50))

    for (x, y, w, h) in face:
        cv2.rectangle(image, (x, y), (x+w, y+h), (255,255,255), 5)
        face = image[y:y + h, x:x + w]

    return face,image
```



Figure 4.8 : Exemple de recadrement d'une image

5.3 Conversion en Grayscale : Ensuite nous effectuons une conversion des images en grayscale car les couleurs ne sont pas des critères qui affectent l'âge. La conversion est effectuée avec la fonction `cvtColor` de la bibliothèque OpenCv.

```
def RGB_to_Gray(image):  
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)  
    return gray
```

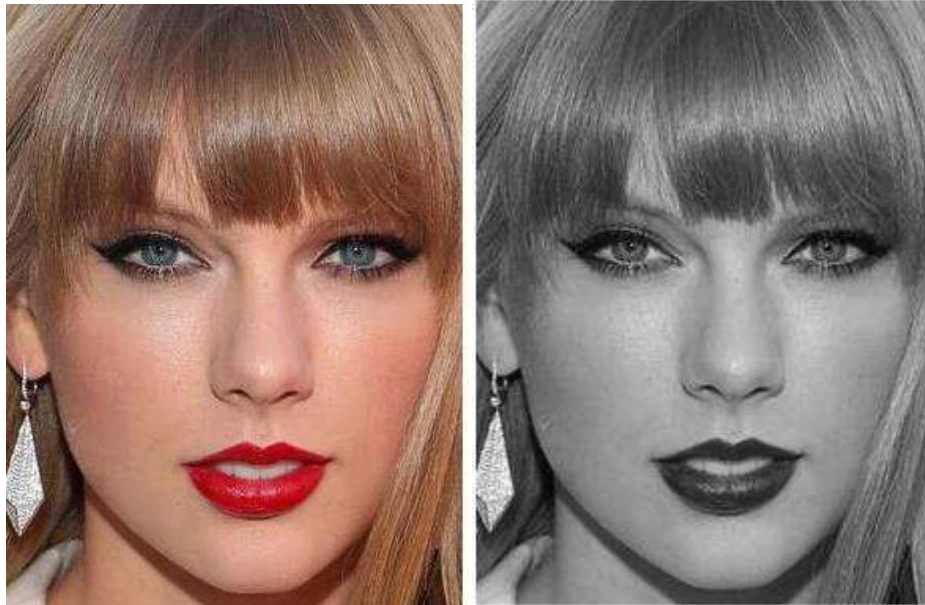


Figure 4.9 : Exemple de conversion en grayscale d'une image

5.4 L'étiquetage : Enfin nous avons effectué un étiquetage en 2 classes. Mineur pour les images des personnes ayant moins de 18 ans et Adultes pour ceux ayant 18 ans ou plus.

```
classes = ["adulte", "mineur"]  
for folder in folders :  
    if int(folder) >= 18 :  
        classe = "adulte"  
    else :  
        classe = "mineur"  
    imgs = [img for img in os.listdir(folder)]  
    for img in imgs :  
        data.append([img, np.argmax(classes)])
```

5.5 Séparation : Avant d'utiliser les données prétraitées pour entraîner notre modèle, nous avons divisés la base de données à deux ensembles. Le premier ensemble qui représente 70% des données est utilisé pour l'entraînement (training). Les 30% restants sont utilisées pour le test afin d'évaluer la précision de notre modèle. La division des images est effectuée aléatoirement.

```
data = load_all_data()
data.shuffle()
training_data = data[:70]
test_data = data[-30:]
```

6. L'entraînement (Training) :

Nous utilisons la bibliothèque Keras pour implémenter notre modèle et l'entraîner sur l'ensemble de données prétraitées.

```
model = Sequential()

model.add(Conv2D(256, (3, 3), input_shape=X.shape[1:]))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(256, (1, 1)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())

model.add(Dense(64))

model.add(Dense(1))
model.add(Activation('sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

7. Résultats :

La métrique sur laquelle nous nous sommes basés pour évaluer notre modèle est la précision qui est calculé par : (Le nombre des images classifiées correctement / nombre totale des images).

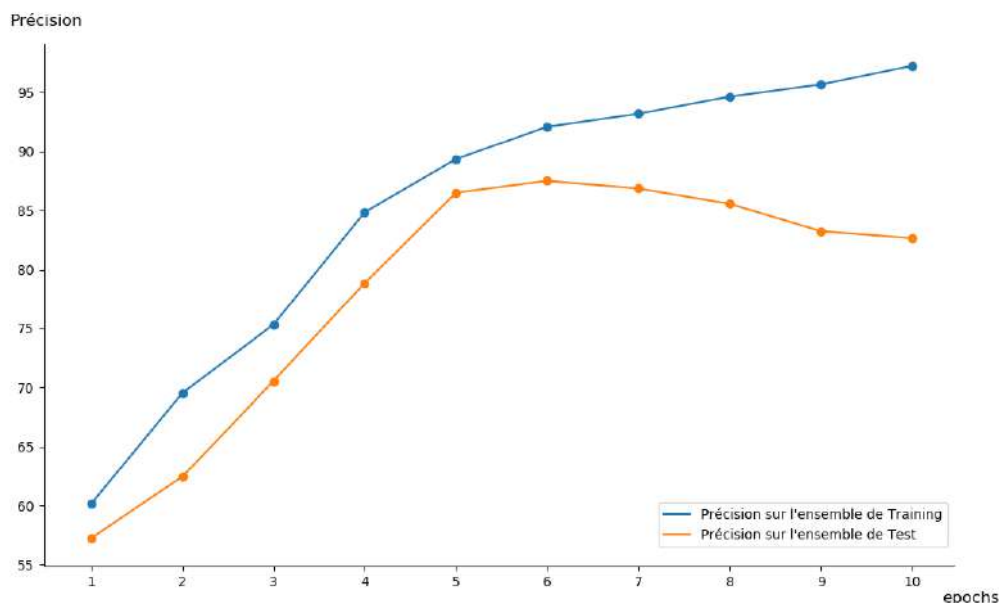


Figure 4.10 : Graphique linéaire de la précision du modèle

Après avoir entraîné notre modèle sur la base d'images IMDB-WIKI sur un CPU i7 pour 10 epochs nous avons remarqué une baisse de précision sur l'ensemble de Test à partir de la septième epoch donc nous avons enregistré les poids du modèle après la 6ème epoch qui ont donné les résultats suivants :

| | Précision sur l'ensemble d'entraînement | Précision sur l'ensemble de Test |
|-----------------------|---|----------------------------------|
| Les images des hommes | 89.1% | 88.2 % |
| Les images des femmes | 95.1 % | 86.8 % |
| Touts les images | 92.1 % | 87.5 % |

Pour mieux évaluer notre modèle nous le testons sur une autre base d'images qui s'appelle UTKFace. Il s'agit d'une base de données de visage à grande échelle avec une longue intervalle d'âge (de 0 à 116 ans). L'ensemble de données comprend plus de 20 000 images de visage avec des annotations d'âge, de sexe et d'origine ethnique. Tous les images sont collectées d'internet [34]. Après le test nous avons obtenus les résultats suivants :

| | Précision |
|-------------------------|---------------|
| Les images des hommes | 93,30 % |
| Les images des femmes | 85.14% |
| Touts les images | 89.22% |

8. Discussion :

- Les résultats que nous avons obtenus après l'entraînement montrent une précision totale de 87.5 % de notre modèle sur l'ensemble de test (les images de la base d'images IMDB-WIKI qui n'ont pas été utilisées dans la phase de l'entraînement), et une précision de 89.22% sur les images d'UTKFace. Ce qui signifie que notre modèle a réussi à extraire les caractéristiques qui distingue les mineurs et les adultes à partir des images utilisées dans la phase d'entraînement.

- La précision de notre modèle sur l'ensemble des images des hommes dans l'ensemble d'entraînement (89.1 %) et dans l'ensemble de test (88.2%) est très proche ce qui signifie que notre modèle a pu généraliser les connaissances qu'il a appris de l'ensemble d'entraînement sur l'ensemble de test.

- La précision du modèle sur l'ensemble des images des femmes dans l'ensemble d'entraînement (95.1 %) est élevée par rapport à la précision sur l'ensemble de test (86.8%) ce qui indique la présence d'un surajustement. Néanmoins le résultat reste favorable.

9. Présentation de l'application :

Pour tester l'efficacité de notre modèle et présenter un prototype d'un moyen de classification d'âge nous avons créé une application qui utilise notre modèle dans le backend et ensuite présente le résultat de la classification à l'écran.

9.1 Page d'accueil : Page d'accueil simple qui offre à l'utilisateur le moyen de tester le modèle en lançant la caméra de l'ordinateur où en important une image du disque.

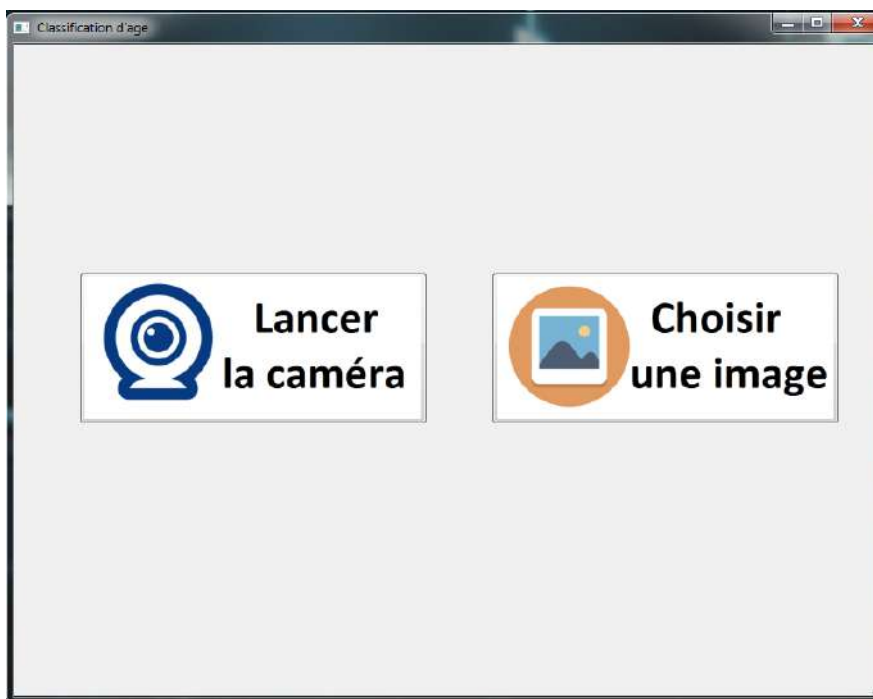


Figure 4.11 : Page d'accueil

9.2 Test à travers la caméra de l'ordinateur : Lors du lancement de la caméra l'application va détecter le visage de l'utilisateur avant de le classier et présenter le résultat.

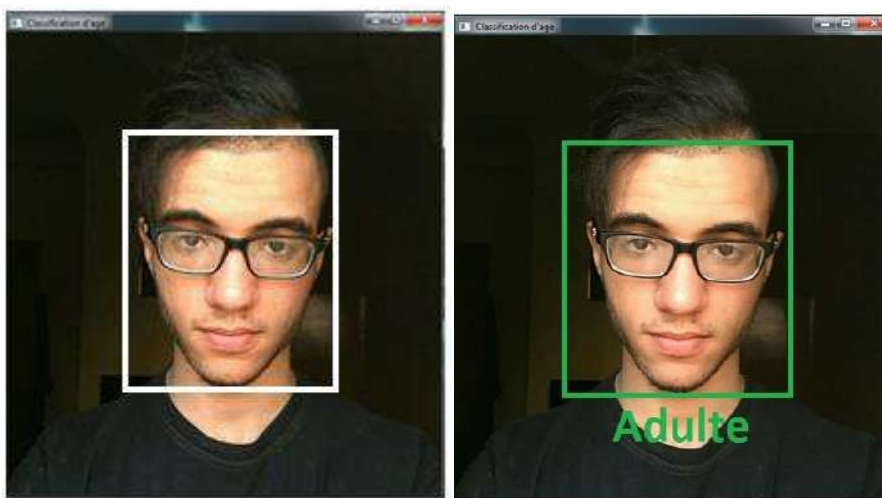


Figure 4.12 : Test à travers la caméra de l'ordinateur

9.3 Test à travers une image importé : Lors de l'importation d'une image du disque l'application détecte le visage de la personne dans l'image avant de le classifier. Ensuite présente le résultat.

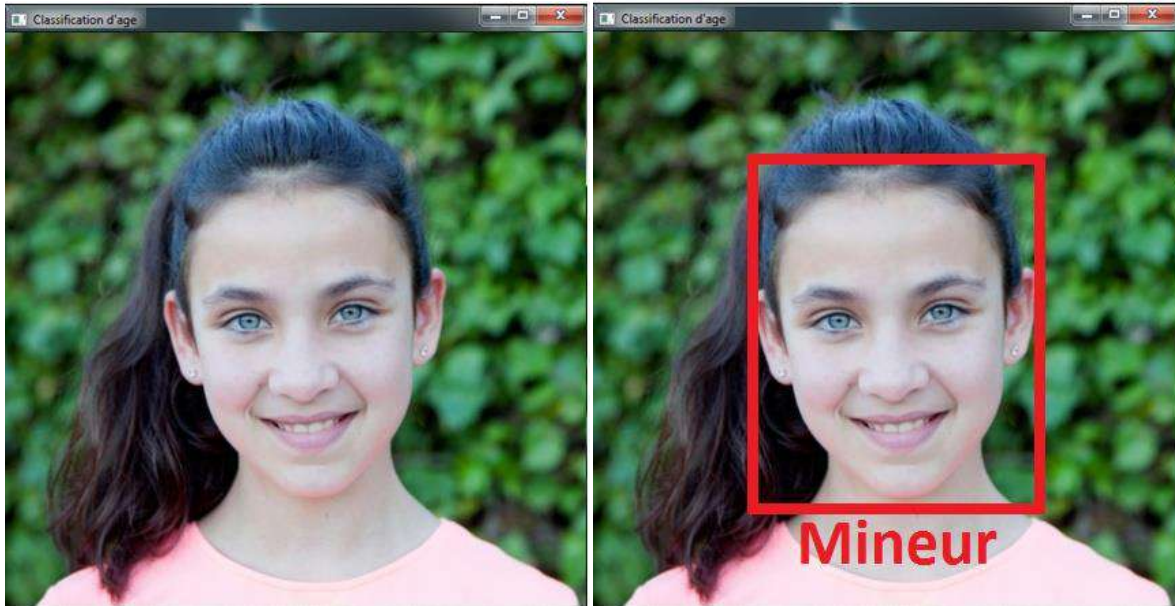


Figure 4.13: Test à travers une image importée

10. Conclusion :

Dans ce chapitre nous avons présenté la partie pratique de notre projet en montrant l'architecture de notre modèle, la base d'images utilisées ainsi que notre application. En plus, nous avons présenté les résultats obtenus de notre modèle et nous avons analysé ces résultats que nous avons trouvé acceptables mais pouvant être améliorés.

Conclusion Générale:

De nos jours il est impossible de surveiller ce que font les enfants et les adolescents sur internet tout le temps, et à tout moment ces mineurs peuvent se retrouver face à un contenu inapproprié pour leur âge ce qui peut affecter leurs personnalités et leurs comportements.

Le travail que nous avons réalisé offre un moyen efficace de classer les humains en deux catégories (mineurs et adultes) avec un bon taux de précision. A travers une application simple qui analyse les traits du visage d'une personne et applique cette classification nous avons pu régler un certain nombre de problèmes qui peuvent mettre les mineurs en danger et offrir les fonctionnalités suivantes :

- Protéger les mineurs d'accéder aux sites web et aux applications montrant un contenu trop graphique et inapproprié pour des audiences trop jeunes.
- Empêcher les mineurs d'accéder aux jeux vidéo trop violents qui peuvent mener à des comportements agressifs chez eux.
- En empêchant les mineurs d'accéder aux applications de rencontre conçus pour adultes. Les pédophiles auront moins de chance de les manipuler à travers ces applications.

L'idée de notre application peut être implémentée dans n'importe quel programme conçu pour les adultes et qui sera plus efficace qu'afficher un message d'alerte qui prévient les mineurs du contenu du programme car la majorité des adolescents sont trop curieux et un message ne va pas les empêcher d'accéder à ces programmes.

Cependant, notre application n'offre pas une solution parfaite et les parents doivent toujours surveiller ce que leurs enfants font avec la technologie et leurs expliquer ses dangers.

Bibliographie :

- [1] « What is Machine Learning ? A definition », <https://expertsystem.com/machine-learning-definition/>, Visité le 14 Mars 2020.
- [2] Russell, Stuart J.; Norvig, Peter (2010). *Artificial Intelligence: A Modern Approach* (Third ed.). Prentice Hall. ISBN 9780136042594.
- [3] Mohri, Mehryar; Rostamizadeh, Afshin; Talwalkar, Ameet (2012). *Foundations of Machine Learning*. The MIT Press. ISBN 9780262018258.
- [4] Claude Touzet. LES RESEAUX DE NEURONES ARTIFICIELS, INTRODUCTION AU CONNEXIONNISME : COURS, EXERCICES ET TRAVAUX PRATIQUES. EC2, 1992, Collection de l'EERIE, N. Giambiasi. hal-01338010, Page 06.
- [5] "LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jackel, L. D. (December 1989). "Backpropagation Applied to Handwritten Zip Code Recognition". *Neural Computation*. 1 (4): 541–551. doi:10.1162/neco.1989.1.4.541. ISSN 0899-7667."
- [6] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E. (2017-05-24). "ImageNet classification with deep convolutional neural networks" (PDF). *Communications of the ACM*. 60 (6): 84–90. doi:10.1145/3065386. ISSN 0001-0782.
- [7] « GoogLeNet convolutional neural network - MATLAB googlenet », <https://www.mathworks.com/help/deeplearning/ref/googlenet.html>, visité le 07 Avril 2020
- [8] Y. H. Kwon and N. da Vitoria Lobo. Age classification from facial images. *Computer Vision and Image Understanding*, 74(1):1–21, 1999.
- [9] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. Systems, Man, Cybernetics SMC-9* (1979), 62–66.
- [10] X. Geng, Z.-H. Zhou, and K. Smith-Miles. Automatic age estimation based on facial aging patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2234–2240, 2007.
- [11] D. Yi, Z. Lei, and S. Z. Li. Age estimation by multi-scale convolutional network. In *Proceedings of Asian Conference on Computer Vision*, 2014.

- [12] H.-F. Yang, L. B-Y, C. K-Y, and C. C-S. Automatic age estimation from face images via deep ranking. In Proceedings of British Machine Vision Conference, 2015.
- [13] Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Section 1.1. Archived from the original (PDF) on 23 June 2012.
- [14] « Why Tensorflow », <https://www.tensorflow.org/about>, Visité le 1er Mai 2020
- [15] « Keras: the Python deep learning API », <https://keras.io/>, Visité le 5 Mai 2020
- [16] « Riverbank Computing | Introduction », <https://riverbankcomputing.com/software/pyqt> , visité le 5 Mais 2020
- [17] Pulli, Kari; Baksheev, Anatoly; Korniyakov, Kirill; Eruhimov, Victor (1 April 2012). "Realtime Computer Vision with OpenCV". Queue: 40:40–40:56. doi:10.1145/2181796.2206309 (inactive 2020-05-21).
- [18] « Intel acquires Itseez », <https://opencv.org/intel-acquires-itseez.html>, visité le 5 Mai 2020
- [19] « IMDB-WIKI dataset | Download », <https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/>, Visité le 20 Décembre 2019.
- [20] « Classification in Machine Learning | edureka », <https://www.edureka.co/blog/classification-in-machine-learning/#:~:text=Artificial%20Neural%20Networks.,Classification%20Algorithms,recognition%2C%20document%20classification%2C%20etc>, Visité le 25 Mai 2020.
- [21] « Regression in Machine Learning | medium », <https://medium.com/datadriveninvestor/regression-in-machine-learning-296caae933ec>, Visité le 25 Mai 2020.
- [22] « Machine Learning paradigms | mc.ai », <https://mc.ai/learning-paradigms-in-machine-learning/>, Visité le 25 Mai 2020.
- [23] « Clustering | analyticsvidhya », <https://www.analyticsvidhya.com/blog/2016/11/anintroduction-to-clustering-and-different-methods-of-clustering/>, Visité le 25 Mai 2020.
- [24] « Decision Trees | Towards data science » <https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052>, Visité le 25 Mai 2020.
- [25] « SVM | Support Vector Machine Algorithm in Machine Learning », <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>, Visité le 26 Mai 2020.

[26] « K-Nearest Neighbor(KNN) Algorithm for Machine Learning – Javatpoint », <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>, Visité le 26 Mai 2020.

[27] « ML | Linear Regression – GeeksforGeeks » <https://www.geeksforgeeks.org/ml-linear-regression/>, Visité le 26 Mai 2020.

[28] « K-Means Clustering Algorithm for Machine Learning | by Madison Schott | Capital One Tech | Medium », <https://medium.com/capital-one-tech/k-means-clustering-algorithm-formachine-learning-d1d7dc5de882>, Visité le 26 Mai 2020.

[29] « Random Forest – Data Analytics », <https://dataanalyticspost.com/Lexique/random-forest/>, Visité le 26 Mai 2020.

[30] « How do CNNs work ? - TechTalks », <https://bdtechtalks.com/2020/01/06/convolutional-neural-networks-cnn-convnets/>, Visité le 26 Mai 2020.

[31] « Rectified Linear Units (ReLU) in Deep Learning | Kaggle », <https://www.kaggle.com/dansbecker/rectified-linear-units-relu-in-deep-learning>, Visité le 26 Mai 2020.

[32] « Fully Connected Layers in Convolutional Neural Networks: The Complete Guide - MissingLink.ai », <https://missinglink.ai/guides/convolutional-neural-networks/fully-connected-layers-convolutional-neural-networks-complete-guide/>, Visité le 26 Mai 2020.

[33] LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W. & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4):541-551.

[34] « UTKFace | Large Scale Face Dataset », <https://susanqq.github.io/UTKFace/>

Résumé : *L'objectif de ce projet est d'offrir un moyen efficace pour empêcher les mineurs d'accéder aux plateformes des adultes afin de les protéger. Car ces plateformes présentent très souvent un contenu trop graphique ou trop violent qui peut affecter le comportement de ces mineurs. Dans notre travail nous présentons une solution sous forme d'un Réseau de Neurones Convolutionnel qui peut distinguer les mineurs et les adultes à partir des traits de leurs visages.*

Mot clés : *Réseaux de Neurones Convolutionnels, Apprentissage automatique, Classification d'âge, mineur, adulte.*

Abstract: *The objective of this project is to provide an effective way to prevent minors from accessing adult platforms in order to protect them. Because these platforms very often present a content that is too graphic or too violent which can affect the behavior of these minors. In our work we present a solution in the form of a Convolutional Neural Network which can distinguish minors and adults from the features of their faces.*

Keywords: *Convolutional Neural Networks, Machine learning, Age classification, minor, Adult.*