REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA



Faculté de **T**echnologie

Mémoire de fin d'études de Master

Filière : Electronique
Spécialité : Electronique des Systèmes Embarqué

Etude et réalisation d'un contrôleur d'accès biométrique

Présenté par :

DJEFFAL Farouk

Proposé et Dirigé par :

Dr.MAAZOUZ Mohammed

Dr. TOUBAL Abdelmoughni

Promotion 2020

DEDICACE

Je dédie cette mémoire à :

Mes parents, Qu'ils puissent recevoir à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.

Remerciements

je remercier avecmes profondes gratitudes
mon promoteur Monsieur Mazzouz Mohamed
pour son encadrement, son suivi
et ses conseils tout au long de cette période.
Nos remerciements et notre gratitude vont aux
professeurs et enseignants de département
d'électronique.

Table des matières

I.1 Introduction 3
I.2 Préambule et cadre générale 3
I.2.1 La biométrie dans le domaine de sécurité 3
I.2.2 Le système biométrique 10
I.3 L'architecture d'un système biométrique 10
I.3.1 La mesure de la performance du système biométrique 11
I.3.2 Les applications des systèmes biométriques 13
I.4 La biométrie du visage 14
I.4.1 Motivation : (pourquoi la reconnaissance de visages ?) 17
I.4.2 Les principales difficultés de la reconnaissance faciale 17
I.5 Multi-modalité 18
I.5.1 Principe 18
I.5.2 Fusion de données biométriques 19
I.6 Conclusion 21
II.1 Introduction : 22
II.2 Vue sur l'intelligence artificielle 22
II.3 Qu'est-ce que le machine Learning 23
II.3.1 performance et sur-apprentissage 25
II.3.2 Les différents types de machine learning 27
II.3.3 Classification , Régression , Association de règle et Clustering 28
II.3.4 Méthodes de Classifications 34
II.3.4.a Les k plus proches voisins 34
II.3.4.b Le classifieur naïf de Bayes 35
II.3.4.c K-moyennes et K-plus proches voisins 36
II.3.4.c Les arbres de décision 38
II.3.4.d Les forets aléatoires 40
II.3.4.e Les machines à vecteur de support (SVM - Support Vector Machine) 41
II.3.4.f Le perceptron multicouches 44
II.4. Conclusion 46
II.5. Deep Learning 46
II.5.1 Artificial Neural Network – ANN 46
II.5.2 Réseaux de neurones profonds (Deep Neural Network - DNN) 47
II.5.3 Réseaux neuronaux de convolutions (CNN) 49
II.5.3.1 L'opération de convolution 50
II.5.3.2 Couche convolutif 51
II.5.3.3 Couche de pooling 52
II.5.3.4 Perceptron 53
II.5.3.5 Quelques réseaux convolutifs célèbres 54
II.6. Optimisation pour l'apprentissage en Deep Learning 55
II.6.1. Les variantes de la descente de gradient 56
II.6.1.1. Batch gradient descent 57
II.6.1.2. Descente de gradient stochastique 57
II.6.1.3. Mini-batch gradient descent 58
II.7. Les fonctions d'activation 59
II.7.1 Caractéristique des fonctions d'activation 59
II.7.2 Quelque fonctions d'activation 59
II.7.2.1 Rectified Linear Units et leur généralisations 59
II.7.2.2 Sigmoïde et tangente hyperbolique 61

II.8. Transfert Learning..... 61
II.9. Evaluation des méthodes..... 63
II.9.1. Métrique d'évaluation 63
II.9. Conclusion 66
III .1 Introduction..... 67
III.2 Logiciels et outils..... 67
III.2.1 Anaconda (Python distribution)..... 67
III.2.2 Présentation des outils de programmation : 67
III.3 Description de la base de données utilisé..... 70
III.3.1. Séparation des bases de données..... 71
III.4 Implémentation et réalisation : 71
4.1 module de détection :..... 71
4.2 module de reconnaissance : 71
III.5. Reconnaissance des visages et résultats 86
III.5.1. Résultats expérimentaux et discussions 86
III.6. Conclusion 87

Table des figures

Figure I.1 : Exemples de modalités biométriques. 03
Figure I.2 : Exemples de traitement pour la reconnaissance de visages. 04
Figure I.3 : Exemples de reconnaissance d'empreintes digitales. 05
Figure I.4 : Exemple d'un réseau vasculaire. 05
Figure I.5 : Exemple de rétine et d'iris. 06
Figure I.6 : Exemple d'une signature. 06
Figure I.7 : Illustration d'un signal vocale. 07
Figure I.8 : exemple d'un empreinte ADN. 07
Figure I.9 : Exemple d'un électrocardiogramme de trois personnes. 08
Figure I.10 : L'architecture du système biométrique 11
Figure I.11 : Illustration du FRR et du FAR en fonction de scores 12

Figure I.12 : Exemples de dispositifs biométriques intégrés aux smartphones. 13
Figure I.13 : Caméras intelligentes intérieure et extérieure Netatmo 14
Figure I.14 : Scores de compatibilité pour différentes technologies biométriques dans un système MRTD.. 15
Figure I.15 : Principe de fonctionnement de base d'un système de reconnaissance faciale. 15
Figure I.16 : Sources d'informations dans un système de fusion biométrique. 19
Figure I.17 : Les différents niveaux de fusion biométrique. 21
Figure II.1 : Relations entre les ensembles "intelligence artificielle", "apprentissage automatique" et "apprentissage profond". 23
Figure II.2 : le processus typique du ML. 24
Figure II.3 : un modèle de ML qui essaye d'obtenir la meilleure approximation possible. 24
Figure II.4 : Le sur-apprentissage. 25
Figure II.5 : La ligne verte représente un modèle surentraîné et la ligne noire représente un modèle régularisé. Ce dernier aura une erreur de test moins importante. 26
Figure II.6 : *La validation croisée* 27
Figure.II.7 : Chaîne des étapes de classification d'images communes à la majorité des applications. 30
Figure II.8 : Catégories de méthodes d'extractions de caractéristiques dans l'image. 32
Figure II.9 : Pour k = 3 la classe majoritaire du point central est la classe B, mais si on change la valeur du voisinage k = 6 la classe majoritaire devient la classe A.. 35
Figure II.10: Le classifieur naïf de Bayes est basé sur le théorème de Bayes avec une indépendance (dite naïve) des variables prédictives. 36
Figure II.11 : L'algorithme *k*-means regroupe les données en *k* cluster, ici *k* = 3. Les centres de gravité sont représentés par de petit cercle.. 37
Figure II.12 : L'ensemble d'apprentissage contient 12 observations décrites par 10 variables prédictives et une variable cible. 39
Figure II.13 : L'ensemble d'apprentissage contient 12 observations décrites par 10 variables prédictives et une variable cible 40
Figure II.14 : L'arbre de décision déduit à partir des 12 exemples d'apprentissage 41
Figure II.15 : Création de B boostrap à partir des exemples d'apprentissage, chacun servira à entraîner un arbre de décision. Si c'est un cas de classification 42
Figure II.16 : *prétraitements.* 43
Figure II.17 : L'hyperplan optimal (en vert) avec la marge maximale. Les échantillons remplis sont des vecteurs supports. 43
Figure II.18 : Exemple d'un problème non linéairement séparable. La courbe devient une bande linéaire après avoir appliqué la transformationϕ non linéaire . 44

Figure II.19 : La figure représente le fonctionnement d'un neurone formel. C'est un composant 45

Table des figures

calculatoire qui fait la somme pondérée des signaux reçus en entrée, puis on leur applique une fonction d'activation afin d'obtenir Y.

Figure II.20 : Structure d'un RN composé d'une couche d'entrée, deux couches cachées et une couche de sortie. 47

Figure II.21 : Principe d'un réseau de neurones artificiels. 48

Figure II.22 : Deep Learning Neural Network. 50
.
Figure II.23 : Différence entre CNN et Machine Learning Traditionnel pour l'analyse d'images. 51

Figure II.24 : Exemple d'une convolution 2D. 53

Figure II.25 : Sparse interactions. 53

Figure II.26: Exemple image 1000x1000. (à gauche) Le non paramètre sharing nous oblige à concevoir une couche cachée de 106 neurones. 54

Figure II.27:(à gauche) Average pooling : chaque case correspond à la moyenne du carré d'entrée de la même couleur, ex de la case jaune) Max pooling. 54

Figure II.28 :Un réseau de neurones convolutif qui reçois une image 2D comme entrée et qui est composé d'une couche convolutive, une fonction d'activation non linéaire, une couche MAX pooling et enfin un perceptron multi couche. 56

Figure II.29 :Le taux d'erreur dans ImageNet Visual recognition Challenge. Le Deep Learning dépasse la performance humaine. 56
.
Figure II.30 : Optimisation par descente de gradient 58

Figure II.31 : Les oscillations causées par SGD. 59

Figure II.32 : Saddle point : Le point noir joue le rôle du minimum par rapport à l'axe A-B et du maximumpar rapport à l'axe C-D. 61
.
Figure II.33 : La fonction d'activation ReLu**.** 62

Figure II.34 :(a) Sigmoïde. (b) tangente hyperbolique 70

Figure I.15 : Principe de fonctionnement de base d'un système de reconnaissance faciale. 70

Figure III.1 : Quelques exemples de variations d'images de la base de données de visages FEI. 72

Figure III.2 : Quelques exemples de variations d'images sur la région d'intérêt. 74

Figure III.3 : l'architecture standard de VGG16. 75

Figure III.4 : exemple des couches Conv2D suivi par les couches de pooling. 75

Figure III.5 : exemple d'une couche Conv2D. 77

Figure III.6 : Diagramme de la première couche de convolution $a^{[1]}$ avec des dimensions (224,224,64). 79

Figure III.7 : Schéma de transition de x à $z^{[1]}$. Une convolution est appliquée à x 6 fois en 82

Table des figures

utilisant 6 filtres différents.

Figure III.8 : la fonction d'activation ReLU 83

Figure III.9 : Principaux modules de la phased'apprentissage. 84

Figure III.10 : le contenu de la fenêtre de sortie pendant la phase d'apprentissage. 86
.

Figure III.11 : Principaux modules de la phasede prédiction. 87

Figure III.12 : Aperçu sur les fenêtres de sortie (reconnaissance de deux visages étique par leurs noms 88

Figure III.13 : Apprentissage et validation de la précision dans VGG16 88

Figure III.14 : Apprentissage et validation de la perte dans VGG16 89

Liste de tableaux

Table I.1 : Comparaison des modalités biométriques les plus connues . 09

TableII.1 : Comparaison d'algorithmes de classification supervisée. 46

Table II.2 : Matrice de confusion des prédictions d'un classifieur à C classes. 64

Table II.3 : Matrice de confusion des prédictions d'un classifieur à deux classes. 65

Table III.1 : Couches , dimensions des filtres et les nombres des paramètres pour notre architecture vgg16 73

Introduction générale

La croissance internationale des communications, tant en volume qu'en diversité (déplacements physiques, transactions financières, accès aux services...), implique le besoin de s'assurer de l'identité des individus. En effet, l'importance des enjeux peut motiver les fraudeurs à mettre en échec les systèmes de sécurité existants. Il existe donc un intérêt grandissant pour les systèmes électroniques d'identification et de reconnaissance. Leur dénominateur commun est le besoin d'un moyen simple, pratique, fiable et peu onéreuse de vérifier l'identité d'une personne sans l'assistance d'un tiers. Le marché du contrôle d'accès s'est ouvert avec la prolifération de systèmes, mais aucun ne se révèle efficace contre la fraude, car tous utilisent un identifiant externe tel que : badge/carte, clé, code. Il est fréquent d'oublier un code d'accès. Il existe d'ailleurs de nombreux bureaux où les mots de passe sont notés dans des listes, ce qui représente une dangereuse faille dans la sécurité informatique de l'entreprise puisque toute confidentialité est alors perdue .de même, un badge ou une clé peuvent être, volés ou copiés par des personnes mal intentionnées. Le défaut commun à tous les systèmes d'authentification est que l'on identifie un objet (code, carte...) et non la personne elle-même. Face à la contrainte de l'authentification par « objets », la biométrie apporte une simplicité et un confort aux utilisateurs.

Cette discipline s'intéresse en effet, à l'analyse du comportement ainsi qu'à l'analyse de la morphologie humaine et étudie, par des méthodes mathématiques (statistiques, probabilités,…), les variations biologiques des personnes. Ce thème se situe dans la problématique générale de la biométrie qui est une science qui propose d'identifier les personnes à partir de la mesure de leurs indices biologiques. La biométrie recouvre deux approches principales : analyse comportementale (vitesse de signature, marche,…) ou analyse de la morphologie humaine (empreintes digitales, iris, rétine, voix, main, visage, …). Un des objectifs de la biométrie est de sécuriser des accès à des locaux ou à des matériels. Ceci peut se faire aujourd'hui par un contrôle de pièce d'identité ou par la saisie d'un mot de passe, mais les deux modes de contrôle sont contraignants et peuvent donner lieu à des falsifications. L'utilisation de techniques biométriques doit permettre d'identifier une personne à travers la consultation d'une base de données, ou de vérifier l'identité affirmée d'un individu. Nous avons retenu la modalité « visage » car c'est un indice biologique très fort contenant de nombreuses indications sur l'identité de la personne et dont l'image peut être acquise de manière non invasive. La reconnaissance de la forme du visage est la technique la plus commune etpopulaire. Elle est la plus acceptable parce qu'on peut l'utiliser à distance sans contact avecl'objet. Utiliser une caméra permet d'acquérir la forme du visage d'un individu et puis retirercertaines caractéristiques. Les caractéristiques essentielles pour la reconnaissance du visagesont: les yeux, la

bouche, le tour du visage, le bout du nez,… etc. Selon le système utilisé,l'individu doit être positionné devant la caméra où peut être en mouvement à une certainedistance. Les données biométriques qui sont obtenues sont comparées au fichier référence. Lelogiciel doit être capable d'identifier un individu malgré différents artifices physiques(moustache, barbe, lunettes, etc..).

Le succès actuel des réseaux de neurones convolutif (CNN) dans la classification d'images s'est étendu au problème de la reconnaissance faciale. Contrairement aux méthodes traditionnelles d'apprentissage par la machine et de vision par ordinateur où les caractéristiques sont définies par des méthodes traditionnels , CNN apprend à les extraire directement de la base de données d'apprentissage. le réseau CNN est généralement associé à un classifieur qui permet de former le modèle de bout en bout sur l'ensemble de données. Dans la plupart des cas, il nécessite beaucoup de données d'entraînement pour bien généraliser. La disponibilité des bases de données et la puissance de calcul des machines actuelles présentent des valeurs à ajouter pour les CNNs. Cependant, trouver une architecture idéale pour optimiser les performances des système FER (*facial-expression-recognition*) n'est pas une tâche facile, la plupart des CNN fournissent des résultats moins précis quand on diminue le volume du réseaux et les paramètres à traiter . L'objectif de ce projet de fin d'étude est de concevoir et de réaliser une application qui permet de reconnaitre les visages en temps réel en utilisant la méthode d'apprentissage profond. Dans ce mémoire, nous utilisons l'architecture VGG16qui est élaboré pour la classifications des images.

Ce mémoire est constitué en quatre chapitres, et organisé comme suit :
Le premier chapitre est consacré à la présentation générale de la biométrie. Il décrit le principe de fonctionnement des systèmes biométriques puis définit les outils utilisés pour évaluer leurs performances. Ensuite, la place de la reconnaissance faciale parmi les autres techniques biométriques est analysée. A travers ce chapitre, nous voulons positionner le problème de la reconnaissance faciale et présenter ses enjeux et intérêts par rapport aux autres techniques. Enfin, nous mettons en lumière les difficultés rencontrées par les systèmes de reconnaissance de visage.. Ensuite, dans le second chapitre, nous le consacrons à la présentation de l'apprentissage profond, où nous détaillons plus les réseaux de neurone convolutifs. La conception de notre approche de reconnaissance des expressions faciales basée sur les CNNs est présentée dans le troisième chapitre avec la description des différents outils utilisés dans le développement de notre application, ainsi que les différents résultats obtenus.

I.1 Introduction

La biométrie représente la mesure biologique ou les caractéristiques physiques qui peuvent être utilisées pour identifier les individus. Le mappage des empreintes digitales, la reconnaissance faciale et les empreintes rétiniennes sont tous des formes de technologie biométrique. Puisque les caractéristiques physiques sont relativement fixes et personnalisées, elles sont utilisées pour remplacer ou au moins accroître la sécurité des systèmes de mots de passe pour les ordinateurs, les téléphones, ainsi que pour les salles et les immeubles à accès restreint, à travers cette introduction, exposer le cadre de notre thèse. Nous en ferons une présentation plus détaillée dans le but de préciser les enjeux et les motivations de notre travail.

I.2 Préambule et cadre générale

I.2.1 La biométrie dans le domaine de sécurité

La biométrie consiste à définir l'identité d'une personne à l'aide de ses caractéristiques biométriques, que l'on appelle modalités, et que l'on classe parmi trois catégories : biologiques, morphologiques et comportementales. Inconsciemment, nous avons recours à la biométrie via nos capteurs sensoriels pour nous distinguer les uns des autres en reconnaissant le visage, l'écriture, le rythme de frappe sur un clavier,la démarche, l'odeur ou encore la voix de nos proches (figure 1).



Figure I.1. Exemples de modalités biométriques

Désormais, grâce aux avancées technologiques, un échantillon d'une modalité biométrique peutêtre comparé automatiquement et rapidement à plusieurs autres échantillons de cette même

modalité,c'est ce que l'on appelle la "reconnaissance biométrique" (ou identification : 1 pour N). Onla distingue de la "vérification biométrique" (ou authentification : 1 pour 1) qui consiste à comparerun échantillon à un autre et de confirmer ou non la correspondance. Dans le domaine desécurité, la biométrie permet de protéger physiquement l'accès à un lieu, ou bien virtuellementl'accès à des données. Au travers d'une authentification ou une identification, cette technologiegarantit un niveau de sécurité supérieur tout en s'affranchissant des risques d'oublis et de pertesliés à l'utilisation de mots de passe ou de clés physiques. Afin d'illustrer ces termes, nous pouvonsprendre comme exemple le passeport biométrique, garant de l'identité d'une personne accédant àun pays grâce au visage et aux empreintes digitales de chaque doigt. Désormais implantée dans lesecteur du grand public, la biométrie se retrouve également dans nos smartphones pour protégerl'accès aux données personnelles en utilisant les empreintes digitales comme protection, le visageou même l'iris.

I.2.1.a Modalités morphologique

- Visages [1,2] :

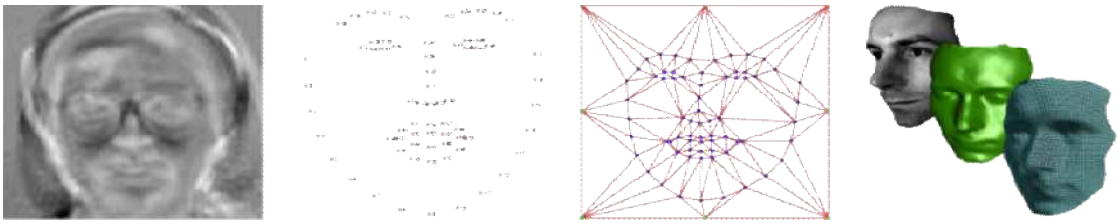


Figure I.2. : Exemples de traitement pour la reconnaissance de visages

La mesure de cette modalité est non intrusive (pas de contact physique) et bien acceptée dugrand public du fait de sa similarité avec le processus humain de reconnaissance des personnes.Cette technique, considérée comme mature à l'heure actuelle, présente l'avantagede pouvoir utiliser les systèmes de vidéosurveillance existant si l'algorithme ne nécessitepas de mesures particulières. En effet, pour la plupart des traitements, une caméra standardsuffit. Cependant, le système est alors soumis à certaines limitations telles que lesvariations de luminosité, les variations du visage (vieillissement, barbe, lunettes, expressionsfaciales, etc.), l'angle de mesure ou contourné dans certains cas par l'usage d'unsimple masque ou d'une photo. L'utilisation d'information 3D (stéréovision, lumièrestructurée, etc.) permet d'outrepasser la majorité de ces problèmes mais nécessite alors uneimplantation particulière et donc un coût supérieur de déploiement.Afin de réduire la complexité des calculs et donc d'accélérer le traitement, une majoritédes méthodes travaille avec des images en niveau de gris (1 canal). Les approches lesplus populaires sont basées soit sur la localisation des points caractéristiques du visage(yeux, sourcils,

nez, bouche, menton, etc.) et leur relation spatiale, soit sur l'analyse del'aspect général d'une image représentant une combinaison linéaire d'un certain nombrede visages normés. D'autres informations complémentaires peuvent être estimées, commele genre du sujet, son âge et la teinte de la peau, ou, en exploitant l'information thermiquedans des longueurs d'onde plus lointaines.

- Empreintes digitale [3] :

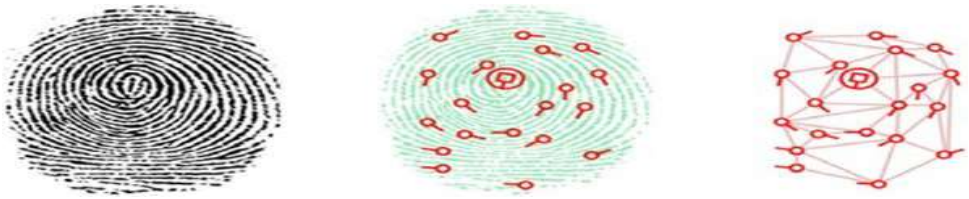


Figure I.3. : Exemples de reconnaissance d'empreintes digitales.

Très largement utilisée, cette modalité est une des plus anciennes utilisées et une des plus connues du grand public. La quantité d'informations disponibles est importante, mais seuls quelques points caractéristiques suffisent à discriminer une empreinte avec une grande fiabilité. La taille du modèle est donc réduite, nécessitant peu d'espace de stockage et permettant une reconnaissance rapide. Les empreintes digitales ont l'avantage d'être présentes sur chaque doigt de la main, ce qui en fait d'autant de motifs différents pour caractériser une personne. Les empreintes digitales ne changent pas de manière naturelle, ce qui garantit la pérennité des mesures dans le temps et réduit les risques de faux négatifs entre deux acquisitions. En revanche, des impuretés externes, des brûlures, des cicatrices, d'autres dommages ou des empreintes inexistantes peuvent être un obstacle à la reconnaissance. Les empreintes digitales peuvent également être usées ou altérées avec le temps selon le métier des sujets. La nécessité d'un contact physique avec le système d'acquisition pose problème sur la question de l'hygiène et peut-être un frein à l'usage pour certaines personnes.

- Réseau vasculaire [4] :

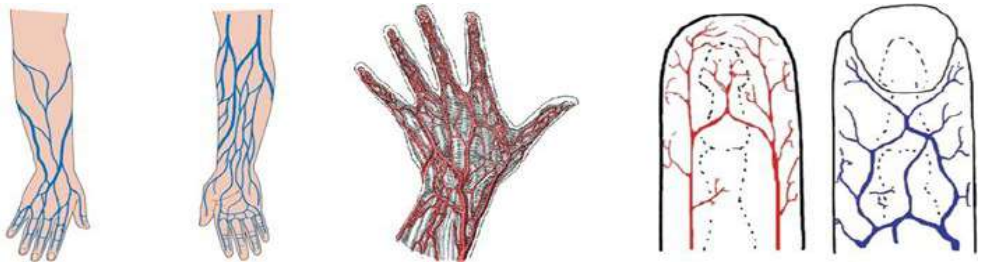


Figure I.4. : Exemple d'un réseau vasculaire.

La mesure du réseau vasculaire s'effectue dans des longueurs d'ondes particulières du domaineproche infrarouge ([750 nm - 1000 nm]), pour lesquelles la lumière pénètre les tissusà faible profondeur pour atteindre les veines où le sang l'absorbe. Ces longueurs d'ondes,proches de

la plage du visible, peuvent être captées par certains capteurs CMOS de qualitédisposant d'un rendement quantique important, dépassant le domaine visible.

- Rétine / Iris [5] :

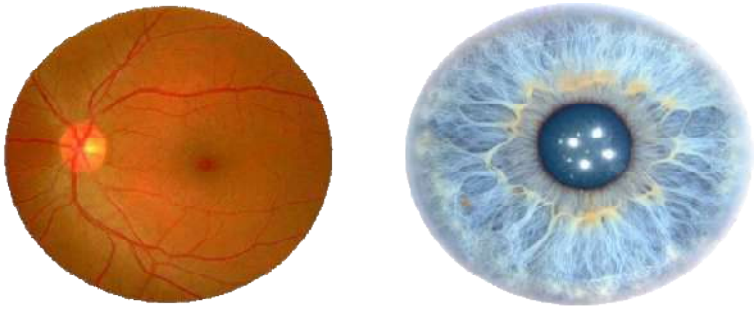


Figure I.5. : Exemple de rétine et d'iris.

La mesure du réseau vasculaire s'effectue dans des longueurs d'ondes particulières du domaine proche infrarouge ([750 nm - 1000 nm]) [5], pour lesquelles la lumière pénètre les tissus à faibleprofondeur pour atteindre les veines où le sang l'absorbe. Ces longueurs d'ondes, proches de la plage du visible, peuvent être captées par certains capteurs CMOS de qualité disposant d'un rendement quantique important, dépassant le domaine visible

Cette acquisition peut s'effectuer sur l'intérieur du bras, le dos de la main, ou l'intérieur d'un doigt. Le motif représenté par les veines est unique et permet de d'identifier une personne en détectant les bifurcations, les jonctions et la forme générale. Mais cette technique est encore peu connue du grand public. La mesure peut principalement être perturbée par la chaleur environnante et la température du corps.

- Signature [6] :



Figure I.6. : Exemple d'une signature

Il est question de mesurer la manière dont une personne signe à l'aide d'un stylet et d'une tablette graphique. En plus de l'aspect général de la signature, les caractéristiques mesurées incluent les pauses, les variations de rythme, la pression, la direction des traits et la vitesse. Cette signature change dans le temps et est influencée par la condition physique du sujet et le système d'écriture utilisé.

- Voix [8] :

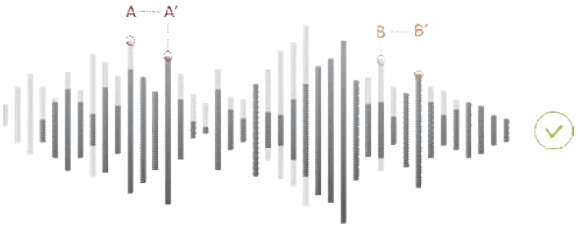


Figure I.7. : Illustration d'un signal vocale.

Le son de notre voix est directement influencé par des facteurs physiques comme le nez, la bouche, les cordes vocales, ou par l'état émotionnel, la langue natale, les conditions médicales, et d'autres paramètres. La qualité de l'enregistrement dépend du capteur ou de l'environnement (écho, bruit, etc.). Il est possible d'isoler ces perturbations en utilisant plusieurs capteurs tout autour du sujet.

I.2.1.b Modalités biologique

- ADN [9]:



Figure I.8. : exemple d'un empreinte ADN.

Présent dans toutes les cellules, l'Acide Désoxyribonucléique (ADN) est une macromolécule biologique contenant toute l'information génétique permettant le développement, le fonctionnement et la reproduction des êtres vivants. L'ADN extrait d'un quelconque échantillon biologique issu d'une personne (sang, salive, fragment de peau ou de poil, etc.) peut être utilisé pour déterminer son empreinte génétique. Le plus souvent utilisée en médecine légale, cette méthode isole et compare les séquences de segments d'ADN de différents individus, avec un risque de similitude entre deux personnes de moins d'un pour cent milliards. Très coûteuse en temps de traitement et en matériel, cette méthode n'est pas temps réel et peu commode dans le prélèvement des échantillons.

- Électrocardiogramme [10] :

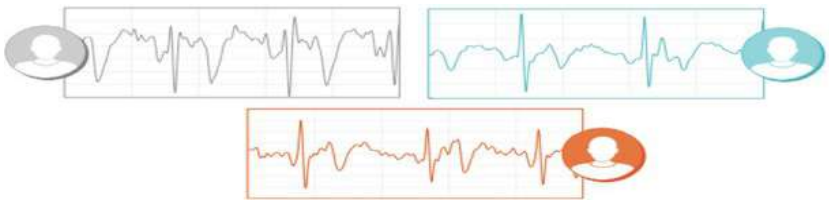


Figure I.9. : Exemple d'un électrocardiogramme de trois personnes.

Les mesures biométriques conventionnelles souffrent d'un risque de falsification via une reproduction de la modalité (masque pour le visage, empreintes digitales en silicone). Les mesures des signaux bioélectriques du corps comptent parmi des plus difficiles à contourner. L'électrocardiogramme (ECG) en fait partie, traduisant un changement du potentiel électrique des cellules cardiaques et possédant des caractéristiques uniques. Le motif de l'acquisition biométrique de l'ECG correspond à la dépolarisation et la repolarisation des battements de cœur. Les caractéristiques mesurées sur ces signaux représentent l'intervalle entre les extremums, leur amplitude, la pente et la largeur des pics.

On distingue sept particularités [11] pour chaque attribut biométrique permettant de lescomparer les unes aux autres et ainsi déterminer la plus appropriées à une situation donnée :

1. L'universalité : détermine si la modalité existe et si elle est présente quel que soit l'individu.
2. L'unicité : définit la probabilité de ne pas trouver de similitudes entre les mesures d'unemême modalité sur des personnes différentes.
3. La pérennité : indique si la caractéristique reste inchangée sur une période donnée.
4. La collectabilité : détermine le degré de facilité de l'acquisition, de la mesure et de l'exploitationde la modalité,
5. La performance : caractérise la robustesse, la fiabilité et la vitesse de la mesure.
6. L'acceptabilité : indique si la modalité est acceptée sans objection par l'utilisateur.
7. Le contournement : représente la difficulté de contourner le système, par usurpationd'identité ou d'autres techniques de fraude.

Modalité	univ	unic	Péren	Collec	Perf	Accept	Cont
Visage	**	*	**	***	*	***	*
Empreinte digitale	*	***	***	**	***	**	***
Réseau vasculaire	*	**	**	**	**	**	***
Rétine	**	***	**	*	***	*	***
Iris	**	***	***	**	***	*	***
Signature	*	*	*	***	*	***	*
Voix	**	*	*	**	*	***	*
ADN	***	***	***	*	***	*	*

TableI.1:Comparaisondesmodalitésbiométriqueslesplusconnues[12](***r eprésenteles meilleuresperformances)

Les points précédents nous démontrent que certaines modalités, considérées seules, peuvent être peu performantes, que ce soit par une faible unicité, une falsification aisée ou des contraintes trop importantes. Une première approche permettant de réduire le nombre de faux positifs se base sur des capteurs complémentaires afin de contrôler les bonnes conditions d'acquisition (pas de falsification), comme par exemple en mesurant sans contact le rythme cardiaque sur le visage du sujet avant de l'authentifier. Une seconde approche pouvant améliorer les performances consiste en l'utilisation d'un système "multi-biométrique", pouvant prendre les formes suivantes (ou une combinaison de celles-ci) :

- Multi-capteurs : Une seule modalité est utilisée ici, mais mesurée par le biais de différents capteurs. Par exemple, pour acquérir la modalité sous différents angles simultanément, ou à différentes échelles, etc.
- Multi-algorithmes : La même mesure est traitée via différents algorithmes, permettant, via une simple acquisition, d'extraire des informations différentes. Cette approche est "économique" en coût matériel mais implique plus de complexité dans les calculs.

- Multi-exemples : Pour les modalités présentes à différents endroits du corps, plusieurs instances sont mesurées sur chacun d'eux. Comme par exemple l'acquisition de différentes empreintes digitales au lieu d'une, ou l'utilisation de l'iris ou de la rétine de chaque œil.
- Multi-échantillons : Plusieurs échantillons d'une même modalité sont mesurés par un seul capteur, en changeant la position autour du sujet, ou faisant varier l'éclairage par exemple.
- Multi-modalités : Différentes modalités sont combinées, via une fusion de décision sur chacune ou de par une fusion des caractéristiques extraites. Cette technique requiert des ressources algorithmiques et matérielles supplémentaires.

I.2.2 Le système biométrique

Un système biométrique est essentiellement un système de reconnaissance de formes quifonctionne en acquérant des données biométriques à partir d'un individu, extrayant unensemble de caractéristiques à partir des données acquises, et comparant ces caractéristiquescontre la signature dans la base de données. Il sert à vérifier l'identité d'une personne à l'aided'une ou plusieurs modalités qui lui sont propres (voix, iris, empreintes digitales, visage …).On peut dire qu'un système de contrôle biométrique est un système automatique de mesurebasé sur la reconnaissance de caractéristiques propres à l'individu. Aussi, on peut adapter ladéfinition suivante : Un système de reconnaissance d'individus est un système biométrique quipermet l'identification d'une personne sur la base de caractères physiologiques ou de traitscomportementaux automatiquement reconnaissables et vérifiables. On peut définir deuxmodes de fonctionnement d'un système biométrique:

- Identification : Procédé permettant de déterminer l'identité d'une personne, il necomprend qu'une étape : L'utilisateur fournit un échantillon biométrique qui va êtrecomparé à tous les échantillons biométriques contenus dans la base de donnéesbiométrique du système. Si l'échantillon correspond à celui d'une personne de la base,on renvoie son numéro d'utilisateur.
- Authentification : Procédé permettant de vérifier l'identité d'une personne. Ilcomprend deux étapes :L'utilisateur fournit un identifiant « Id » au système de reconnaissance (Numéro) ;L'utilisateur fournit ensuite un échantillon biométrique qui va être comparé àl'échantillon biométrique correspondant à l'utilisateur « Id » contenu dans la base dedonnées biométrique du système. Si la comparaison correspond, l'utilisateur est authentifié.

I.3 L'architecture d'un système biométrique

Bien que sous formes diverses et mesurant des caractéristiques différentes, lessystèmes biométriques partagent tous la même architecture. Ce sont tous des systèmes dereconnaissance de

formes. Ils sont composés d'un ou plusieurs systèmes d'acquisition qui vontmesurer le ou les traits physiques ou comportementaux de l'individu. Lorsque le systèmeutilise plusieurs traits qu'il lie à un individu, on parle de systèmes multimodaux. A l'inverse, si une seule caractéristique est utilisée, on utilise le terme de systèmes unimodal.Un système d'information extrait, encode, stocke et compare ces données (figure 1.10).

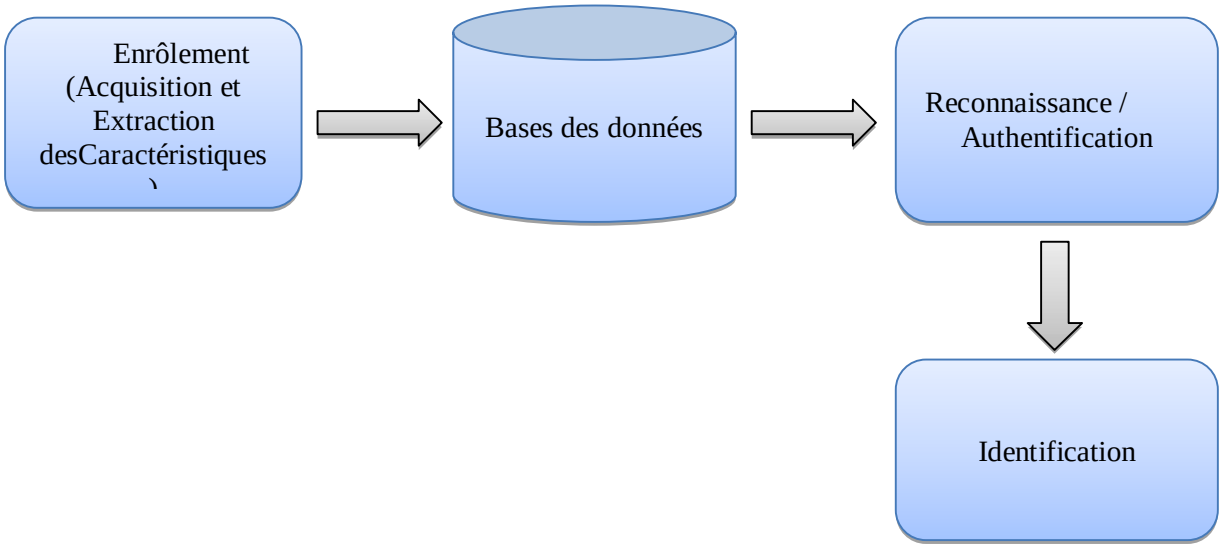


Figure I.10*:* L'architecture du système biométrique.

Les systèmes biométriques sont tous automatisés permettant un traitement rapide del'information. Que le système soit utilisé en mode authentification ou identification, deuxétapes sont nécessaires avant que celui-ci soit pleinement opérationnel [13]:

- Apprentissage :

C'est la phase initiale qui s'effectue une seule fois. Au cours de l'apprentissage, on fait une capture (acquisition) de la caractéristique biométrique. Engénéral cette capture n'est pas directement stockée, des transformations lui sontappliquées, le modèle étant une représentation compacte du signal permettant defaciliter la phase de reconnaissance et de diminuer la quantité des données à stocker.Les tâches essentielles de cette phase sont : l'extraction de paramètres et laconstruction d'un modèle de représentation (appelé gabarit).

- Reconnaissance :

Opération se déroulant à chaque fois qu'une personne se présente devant le système, elle consiste en l'extraction d'un ensemble de paramètres commepour l'étape d'apprentissage suivie d'une autre étape de comparaison et de prise dedécision selon le mode opératoire du système : identification (partir de l'échantillonbiométrique recherche du gabarit correspondant) ou vérification (échantillonbiométrique correspond bien au gabarit).

I.3.1 La mesure de la performance du système biométrique

Tout d'abord, afin de comprendre comment déterminer la performance d'un systèmebiométrique, il nous faut définir clairement trois critères principaux. [14]

1. Le premier critère s'appelle le taux de faux rejet ("False Reject Rate" ou FRR). Ce tauxreprésente le pourcentage de personnes censées être reconnues mais qui sont rejetées parle système.

$$FRR = \frac{nombre\ des\ clients\ rejetés\ (FR)}{Nombre\ total\ d'accés\ de\ clients}$$

Tel que FR Le faux rejet correspond au cas où le système rejette un client légitime.

2. Le deuxième critère est le taux de fausse acceptation ("False Accept Rate" ou FAR).Cetaux représente le pourcentage de personnes censées ne pas être reconnues mais qui sonttout de même acceptées par le système.

$$FAR = \frac{nombre\ d'imposteurs\ accepté\ (FA)}{Nombre\ total\ d'accés\ de\ imposteurs}$$

Tel que FA correspond au cas où le système accepte un individu qui a proclamé uneidentité qui n'est pas la sienne.

3. Le troisième critère est connu sous le nom de taux d'égale erreur ("Equal Error Rate" ou EER). Ce taux est calculé à partir des deux premiers critères et constitue un point de mesure de performance courant. Ce point correspond à l'endroit où FRR = FAR, c'est-à-direle meilleur compromis entre les faux rejets et les fausses acceptations [15].

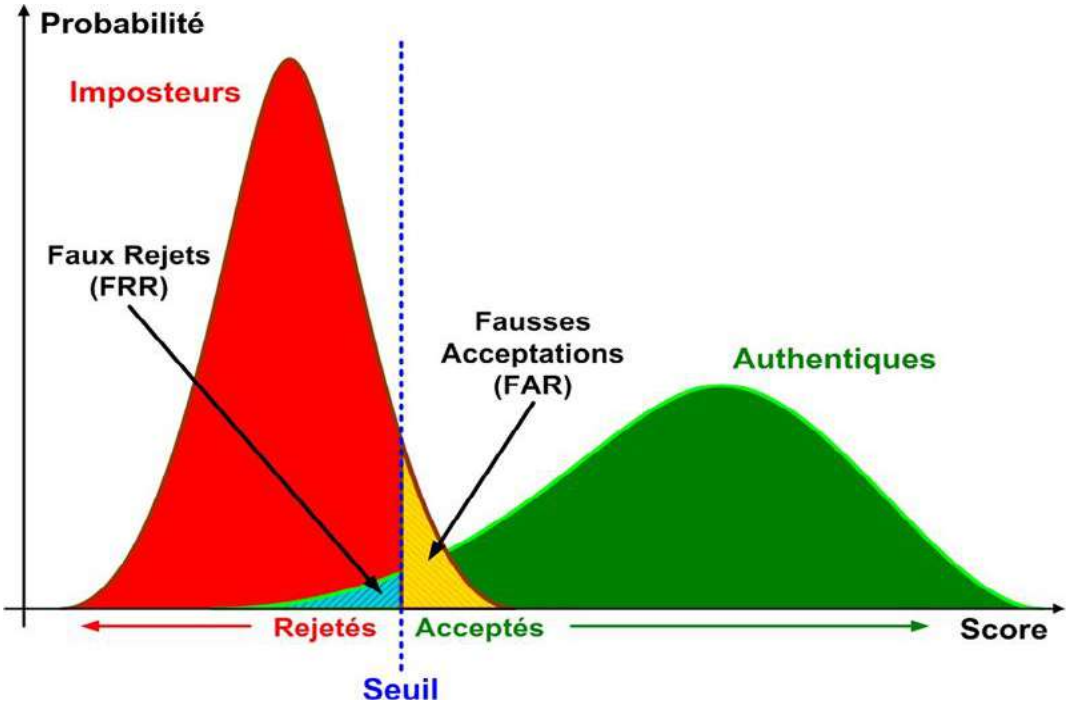


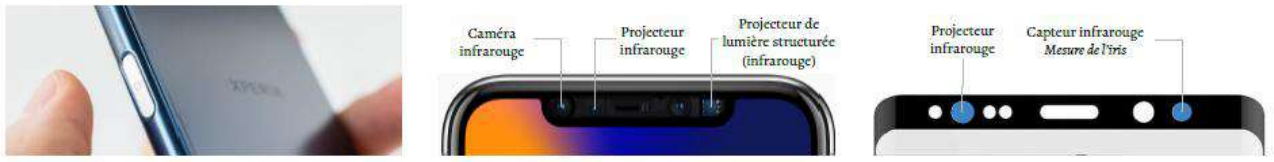
Figure I.11 : Illustration du FRR et du FAR en fonction de scores

I.3.2 Les applications des systèmes biométriques

Les applications de la biométrie peuvent être divisées en trois groupes principaux [16] :

- Applications commerciales (grand publique) : telles que l'ouverture de réseau informatique, la sécuritéde données électroniques, l'e-commerce, l'accès Internet, la carte de crédit, le contrôled'accès physique, le téléphone cellulaire, la gestion des registres médicaux, l'étude àdistance, etc.

La biométrie est déjà bien acceptée du grand public de par son intégration dans les smartphoneset les ordinateurs portables, pour sécuriser l'accès "physique" aux données personnelles, avec la démocratisation du déverrouillage par empreintes digitales, par le visage, par l'iris ou encore la voix. Les implémentations biométriques sur téléphones portables, illustrées par les exemples suivants, sont des méthodes hors ligne, dont les calculs sont strictement limités à la plateforme cible .



(a) Sony Empreint (b) Apple – Face ID (c) Samsung – Scanner d'iris

Figure I.12 – Exemples de dispositifs biométriques intégrés aux smartphones

Les systèmes biométriques pour le grand public ne sont pas limités aux cas très répandu des smartphones,et des systèmes pour la maison leurs sont également proposés. Nous pouvons prendre pourexemple deux dispositifs de l'entreprise française Netatmo présentés figure 1.13. Conçus en tantqu'objets connectés pour la surveillance de la maison, ils disposent de fonctions de communicationsavec une gestion d'évènements afin d'avertir les propriétaires sur leur smartphone. La caméradestinée à un usage intérieur, figure 1.13a, effectue une reconnaissance de visages et notifie de laprésence des membres de la famille ou de personnes inconnues. La caméra à usage extérieur, figure1.13b, n'effectue pas de reconnaissance biométrique mais est capable de capter la présenced'une voiture, de personnes ou d'animaux.



(a) "Welcome" - Caméra intérieure(b) "Presence" - Caméra extérieure

Figure I.13 – Caméras intelligentes intérieure et extérieure Netatmo

- Applications gouvernementales: telles que la carte d'identité nationale, le permis deconduire, la sécurité sociale, le contrôle des frontières, le contrôle des passeports, etc.
- Applications légales : telles que l'identification de corps, la recherche criminelle, l'identification de terroriste, etc.

I.4 La biométrie du visage

La reconnaissance faciale est une tâche que les humains effectuent naturellement et sans effort dans leurs vies quotidiennes. La grande disponibilité d'ordinateurs puissants et peu onéreux ainsi que des systèmes informatiques embarqués ont suscité un énorme intérêt dans le traitement automatique des images et des vidéos numériques au sein de nombreuses applications, incluant l'identification biométrique, la surveillance, l'interaction homme-machine et la gestion de données multimédia. La reconnaissance faciale, en tant qu'une des technologies biométriques de base, a pris une part de plus en plus importante dans le domaine de la recherche, ceci étant dû aux **avances rapides** dans des technologies telles que les appareils photo numériques, Internet et les dispositifs mobiles, le tout associé à des **besoins en sécurité** sans cesse en augmentation.

La reconnaissance faciale possède plusieurs avantages sur les autres technologies biométriques : elle est **naturelle**, **non intrusive** et **facile à utiliser**. Parmi les six attributs biométriques considérés par Hietmeyer[17], les caractéristiques faciales marquent un score de compatibilité le plus élevé dans un systèmeMRTD("*Machine Readable Travel Documents*") [18], ce score étant basé sur **plusieursfacteurs d'évaluation** tels que l'enrôlement, le renouvellement des données, les requis matériels et la perception des utilisateurs (Fig. 1.14). [19]

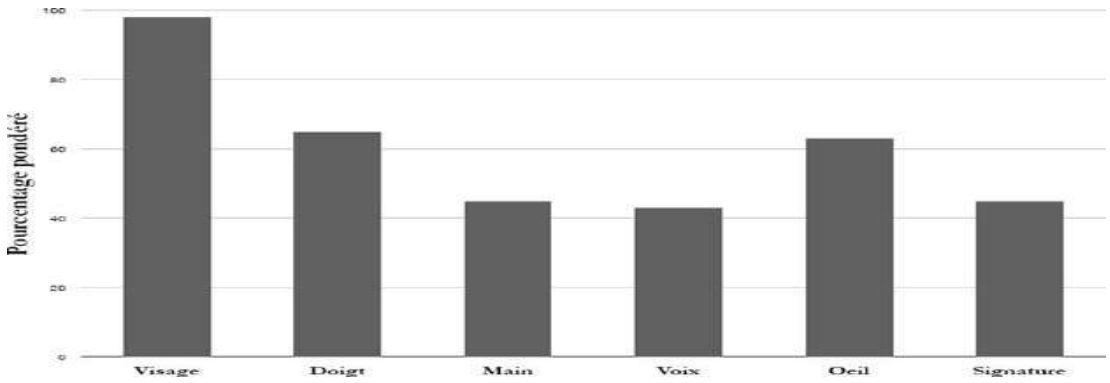


Figure I.14 – Scores de compatibilité pour différentes technologies biométriques dans un système MRTD.

Idéalement, un système de reconnaissance faciale doit pouvoir **identifier des visages** présents dans une image ou une vidéo **de manière automatique**. Le système peut opérer dans les deux modes suivants : authentification ou identification ; on peut également noter qu'il existe un autre type de scénario de reconnaissance faciale mettant en jeu une vérification sur une liste de surveillance ("*watch-list*"), où un individu est comparé à une liste restreinte de suspects. Le principe de fonctionnement de base d'un système de reconnaissance faciale (Figure 1.15.) peut être résumé comme :

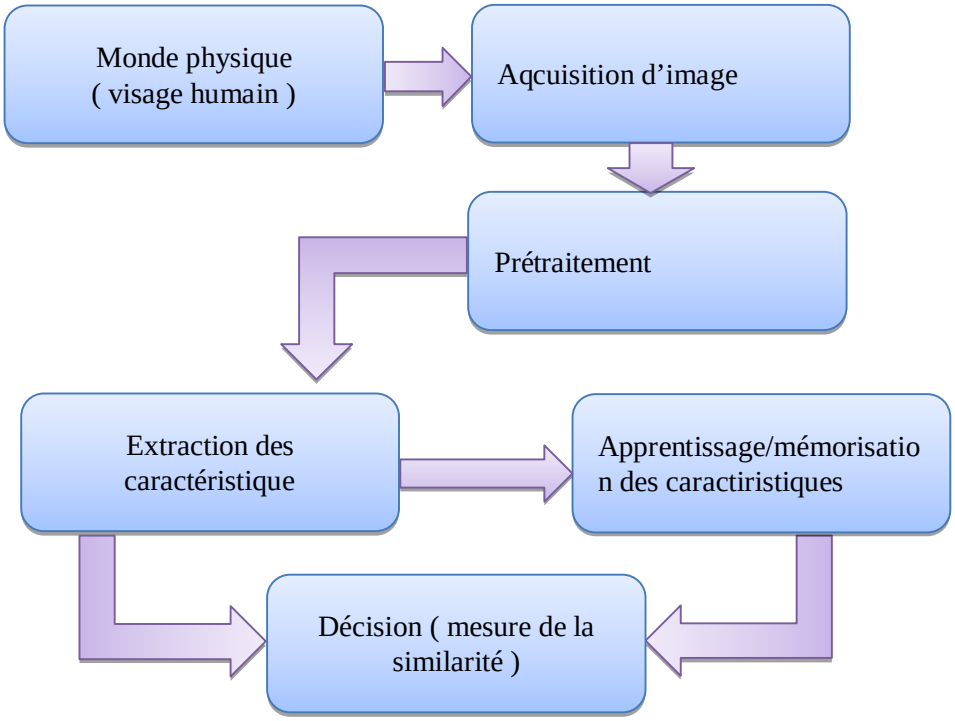


Figure I.15. – Principe de fonctionnement de base d'un système de reconnaissance faciale.

Donc pour être identifié, l'image d'une personne dans un système de reconnaissancefaciale suit les étapes suivantes :

- Détection et acquisition de l'image :

Cette étape consiste à capturer l'image de l'utilisateur du monde extérieur dans un état statique à l'aide d'un appareil photo ou dynamique à l'aide d'une caméra. Après, l'image extraite sera digitalisée ce qui donne lieu à une représentation bidimensionnelle au visage. L'image dans cette étape est dans un état brut ce qui engendre un risque de bruit qui peut dégrader les performances du système.

- Prétraitement:

Le rôle de cette étape est d'éliminer les parasites causés par laqualité des dispositifs optiques ou électroniques lors de l'acquisition de l'image enentrée, dans le but de ne conserver que les informations essentielles et donc préparerl'image à l'étape suivante. Elle est indispensable car on ne peut jamais avoir uneimage sans bruit à cause du background et de la lumière qui est généralementinconnue. Il existe plusieurs types de traitement et d'amélioration de la qualité del'image, telle que : la normalisation, l'égalisation et le filtre médian. Cette étape peutégalement contenir la détection et la localisation du visage dans une image, surtout làoù le décor est très complexe.

- L'extraction descaractéristiques :

L'étape de l'extraction des caractéristique représente le cœur du système de reconnaissance, elle consiste à effectuer le traitement de l'image dansun autre espace de travail plus simple et qui assure une meilleure exploitation dedonnées, et donc permettre l'utilisation, seulement, des informations utiles,discriminantes et non redondantes.

- La classification (Modélisation) :

Cette étape consiste à modéliser les paramètres extraits d'un visage ou d'un ensemble de visages d'un individu en se basant sur leurs caractéristiques communes. Un modèle est un ensemble d'informations utiles, discriminantes et non redondantes qui caractérise un ou plusieurs individus ayant des similarités.

- L'apprentissage :

C'est l'étape où on fait apprendre les individus au système, elle consiste à mémoriser les paramètres, après extraction et classification, dans une base de données bien ordonnée pour faciliter la phase de reconnaissance et la prise d'une décision, elle est en quelque sorte la mémoire du système.

- La décision :

C'est l'étape qui fait la différence entre un système d'identificationd'individus et un autre de vérification. Dans cette étape, un système d'identificationconsiste à trouver le modèle qui correspond le mieux au visage pris en entrée à partirde ceux stockés dans la base de données, il est caractérisé par son taux dereconnaissance. Par contre, dans un système de vérification il s'agit de décider si levisage en entrée est bien celui de l'individu (modèle) proclamé ou il s'agit d'unimposteur, il est caractérisé par son EER (Equal Error Rate).

I.4.1 Motivation : (pourquoi la reconnaissance de visages ?)

Durant les vingt dernières années, la reconnaissance automatique des visages estdevenue un enjeu primordial, notamment dans les domaines de l'indexation de documentsmultimédias et surtout dans la sécurité, ceci est dû aux besoins du monde actuel mais aussi àses caractéristiques avantageuses dont on peut citer :

- La disponibilité des équipements d'acquisition, leur simplicité et leurs coûts faibles.
- Passivité du système : un système de reconnaissance de visages ne nécessite aucunecoopération de l'individu, du genre : mettre le doigt ou la main sur un dispositifspécifique ou parler dans un microphone. En effet, la personne n'a qu'à rester oumarcher devant une caméra pour qu'elle puisse être identifiée par le système.En plus, cette technique est très efficace pour les situations non standards, c'est les cas où onne peut avoir la coopération de l'individu à identifier, par exemple lors d'une arrestation descriminels. Certes que la reconnaissance des visages n'est pas la plus fiable comparée auxautres techniques de biométrie, mais elle peut être ainsi si on utilise des approches pluseffficaces en plus du bon choix des caractéristiques d'identification représentant le visage enquestion.

I.4.2 Les principales difficultés de la reconnaissance faciale

Pour le cerveau humain, le processus de la reconnaissance de visages est une tâchevisuelle de haut niveau. Bien que les êtres humains puissent détecter et identifier des visagesdans une scène sans beaucoup de peine, construire un système automatique qui accomplit detelles tâches représente un sérieux défi. Ce défi est d'autant plus grand lorsque les conditionsd'acquisition des images sont très variables. Il existe deux types de variations associées auximages de visages: inter et intra sujet. La variation inter-sujet est limitée à cause de laressemblance physique entre les individus. Par contre la variation intra-sujet est plus vaste.Elle peut être attribuée à plusieurs facteurs que nous analysons ci-dessous.

- Changement d'illumination :

Les variations d'éclairage rendent la tâche dereconnaissance de visage très difficile. En effet, le changement d'apparence d'unvisage du a l'illumination, se révèle parfois plus critique que la différence physiqueentre les individus, et peut entrainer une mauvaise classification des images d'entrée.

- Variation de pose :

Le taux de reconnaissance de visage baisse considérablementquand des variations de pose sont présentes dans les images. La variation de pose estconsidérée comme un problème majeur pour les systèmes de reconnaissance faciale.Quand le visage est de profil dans le plan image (orientation <30°), il peut êtrenormalisé en détectant au moins deux traits faciaux (passant par les yeux).Cependant, lorsque la rotation est supérieure à 30°, la normalisation géométrique n'estplus possible.

- Présence ou absence des composants structurels :

La présence des composantsstructurels telle que la barbe, la moustache, ou bien les lunettes peut modifierénormément les caractéristiques faciales telles que la forme, la couleur, ou la taille duvisage. De plus, ces composants peuvent cacher les caractéristiques faciales de basecausant ainsi une défaillance du système de reconnaissance.

- Les vrais jumeaux :

Qui ont le même indicatif d'ADN, peuvent tromper lespersonnes qui ne les connaissent pas (les personnes familières avec les jumeaux ontreçu une grande quantité d'information sur ces derniers et sont donc beaucoup plusqualifiés à distinguer les jumeaux.).

- Expressions faciales :

La déformation du visage qui est due aux expressions facialesest localisée principalement sur la partie inférieure du visage. L'information faciale sesituant dans la partie supérieure du visage restequasi invariable. Elle est généralementsuffisante pour effectuer une identification. Toutefois, étant donné que l'expressionfaciale modifie l'aspect du visage, elle entraine forcement une diminution du taux dereconnaissance. L'identification de visage avec expression faciale est un problèmedifficile qui est toujours d'actualité et qui reste non résolu.

I.5 Multi-modalité

I.5.1 Principe

Les systèmes de sécurité biométrique à modalité unique, bien qu'ayant été considérablement amélioréscesdernières années de par des algorithmes plus performants et des capteurs plus précis,soufrent de certains problèmes de fiabilité notamment liés aux critères de contournement et

d'unicité,souvent trop faibles selon les modalités. Par conséquent, les systèmes biométriques multimodauxsont au cœur des solutions les plus performantes en matière de sécurité, permettant decombiner plusieurs modalités (le plus souvent deux) acquises simultanément avec un ou plusieurs dispositifs de mesure [20]. Le système est ainsi capable d'effectuer un contrôle de sécurité sur unnouveau jeu de caractéristiques émanant de la fusion des attributs de ces modalités. Il est égalementpossible de compléter ce contrôle en traitant indépendamment en parallèle les caractéristiques dechacune des modalités mesurées. Un système multimodal est donc en mesure de compenser leslacunes des systèmes uni-modaux, en proposant une solution plus difficilement falsifiable et plusrobuste aux similitudes entre les individus ainsi qu'aux données d'acquisition bruitées. En effet,il est indéniable qu'un usurpateur aura plus de mal à copier plusieurs modalités d'une mêmepersonne. Il est également peu probable que plusieurs modalités soient similaires entre deux personnes. Cette fusion peut s'effectuer à différents niveaux des chaînes de traitement et de décision.

I.5.2 Fusion de données biométriques

La fusion biométrique multimodale combine des mesures de différents traits biométriques pour renforcer les points forts et réduire les points faibles des différents processus biométriques fusionnés. Ainsi, la fusion des informations biométriques peut se faire dans différents niveaux : niveau capteur, niveau caractéristiques, niveau score, niveau décision ou niveau rang.

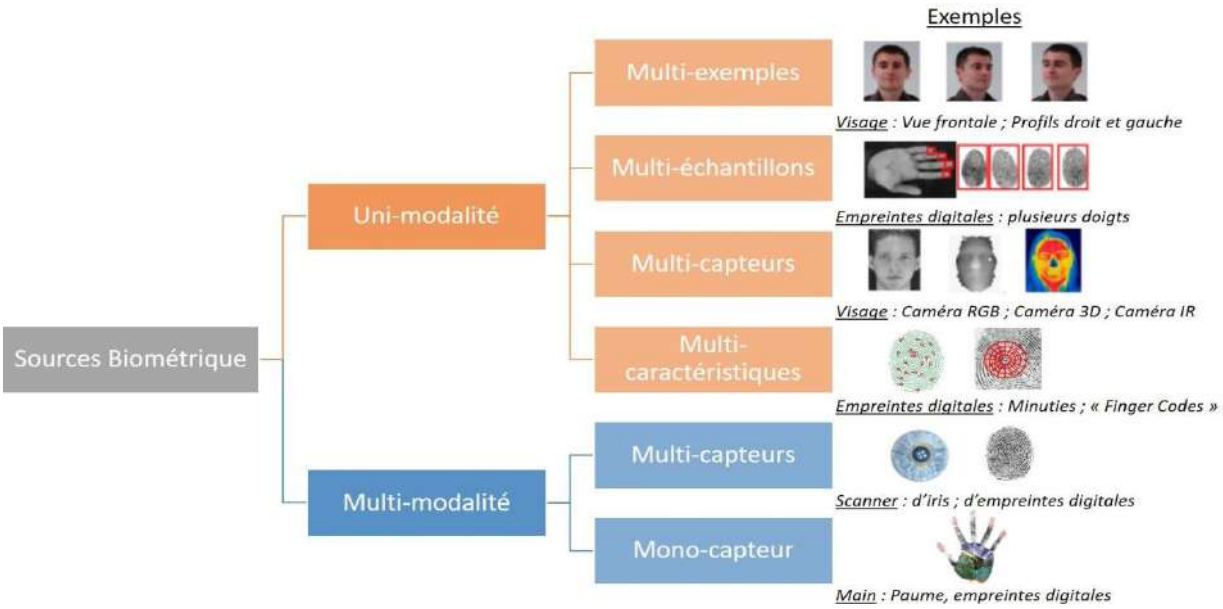


Figure I.16. Sources d'informations dans un système de fusion biométrique.

- Fusion au niveau de capteur : La fusion au niveau de capteur peut se faire uniquement si les diverses captures sont des instances du même trait biométrique obtenu à partir deplusieurs capteurs compatibles entre eux ou plusieurs instances du même traitbiométrique obtenu à

partir d'un seul capteur.De plus, les captures doivent être compatibles entre elles et la correspondance entre lespoints dans les données brutes doit être connue par avance. Par exemple, les images de visageobtenues à partir de plusieurs caméras peuvent être combinées pour former un modèle 3D duvisage. Un autre exemple de fusion au niveau de capteur consiste à mettre en mosaïqueplusieurs images d'empreintes digitales afin de former une image d'empreinte digitale finaleplus complexe. La fusion au niveau de capteur n'est généralement pas possible si les instancesdes données sont incompatibles (par exemple, il est peut être difficile de fusionner des imagesde visages provenant de caméras ayant des résolutions différentes).

- Fusion au niveau de caractéristiques : cette fusion consiste à combiner différentsvecteurs de caractéristiques qui sont obtenus à partir des sources suivantes ; plusieurscapteurs du même trait biométrique, plusieurs instances du même trait biométrique,plusieurs unités du même trait biométrique ou encore plusieurs traits biométriques.Quand les vecteurs de caractéristiques sont homogènes (par exemple, plusieurs imagesd'empreinte digitale du doigt d'un utilisateur), un unique vecteur de caractéristiquesrésultant peut être calculé comme une somme pondérée des vecteurs decaractéristiques individuels. Lorsque les vecteurs de caractéristiques sont hétérogènes(par exemple, des vecteurs de caractéristiques de différentes modalités biométriquescomme l'iris et l'empreinte digitale), nous pouvons les concaténer pour former un seul vecteur de caractéristiques [21].
- Fusion au niveau de scores : Après les vecteurs de caractéristiques, les scores (decorrespondance) donnés en sortie par les matchers contiennent l'information la plusriche à propos du modèle d'entrée. En fait, la fusion au niveau de scores donne lemeilleur compromis entre la richesse d'information et la facilité d'implémentation.Aussi, il est relativement facile d'accéder et de combiner les scores générés par lesdifférents matchers. En conséquence, l'intégration d'information au niveau scores estl'approche la plus courante dans les systèmes biométriques multimodaux [22].
- Fusion au niveau de décision : L'intégration d'information au niveau de décision peutêtre mis en place lorsque chaque matcher biométrique décide individuellement de lameilleure correspondance possible selon l'entrée qui lui est présentée. Les méthodescomme le « *majority voting* », le « *behavior knowledge space* », le « *weighted voting* » basé surla théorie Dempster-Shafer, les règles ET et OU, etc. peuvent être utilisées afind'arriver à la décision finale [23].

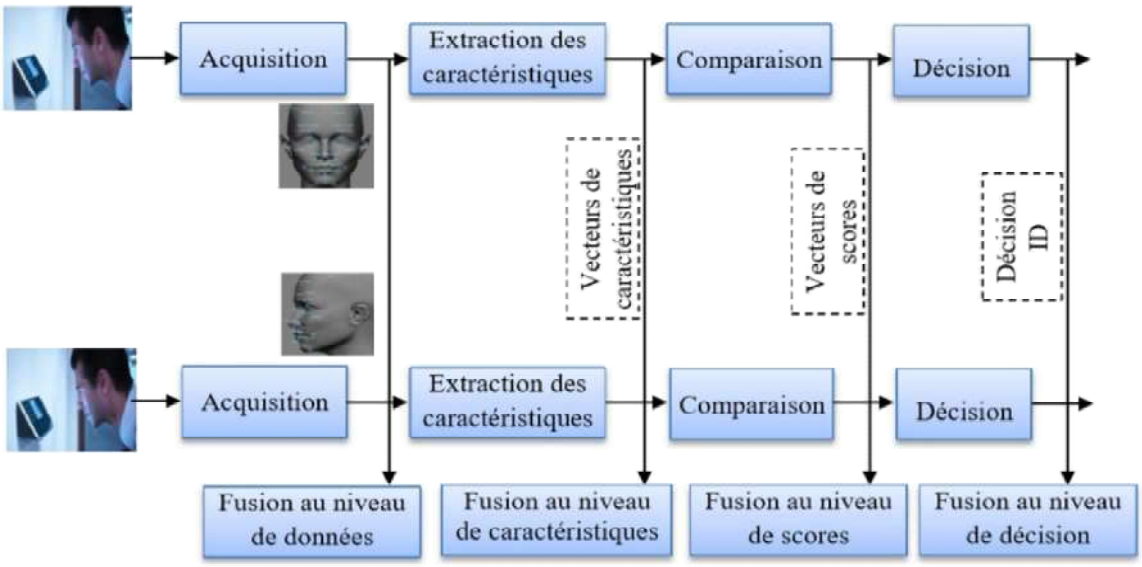


Figure I.17: Les différents niveaux de fusion biométrique.

I.6 Conclusion

Tout ce qui a été présenté jusque-là nous a permis de constater que la biométrie, demanière générale, et que la reconnaissance de visage suscite de plus en plus l'intérêt de la communauté scientifique, car elle présente plusieurs challenges et verrous technologiques.Nous avons mis en évidence les différentes difficultés inhérentes à la reconnaissanceautomatique de visages, ce qui nous a permis de définir les problématiques traitées dans cemémoire.

Divers changements au niveau des images présentent un grand défi face à un systèmede reconnaissance qui doit être robuste en ce qui concerne les grandes variabilités des imagesdu visage comme les expressions faciales, la pose du visage et l'éclairage. Pour faire face à ceproblème, il est important de choisir une représentation appropriée des images du visage.Cette représentation doit être compacte et significative.Nous avons ensuite présenté une façon de réduire les limitations des systèmesbiométriques monomodaux en combinant plusieurs systèmes ou algorithmes conduisant ainsià fiabiliser le système de reconnaissance. Les systèmes multimodaux peuvent être dedifférentes natures, nous avons donc présenté dans ce chapitre, les différents types decombinaisons de modalités possibles, mais aussi les architectures et les niveaux de fusion quipeuvent être utilisés dans un tel système.

II.1Introduction :

Dans le but de faciliter la compréhension du fonctionnement des méthodes employées, il est question ici d'introduire de manière générale l'apprentissage automatique (Machine Learning) pour la classification d'images. l'apprentissage automatique fait partie de l'intelligenceartificielle et permet, avec l'aide des machines, d'effectuer des calculs afin de résoudreun problème complexe. La particularité du Machine Learning, est que les méthodes utilisées permettentà la machine d'apprendre à réaliser une tâche en fonction de nombreuses données d'entrée.Ainsi, l'algorithme ne se contente pas d'appliquer une consigne définie par son concepteur, maiss'adapte aux données qui lui sont transmise afin d'apprendre comment répondre à la problématiquequi lui est fournie. L'apprentissage profond (Deep Learning), que nous traitons égalementdans ce manuscrit, est quant à lui un sous ensemble du Machine Learning. Le Deep Learningmet en œuvre, sous forme de couches, un enchaînement de traitements algorithmiques propres auMachine Learning afin de répondre à un problème complexe découpé en plusieurs tâches, chaquecouche utilisant la sortie de la couche précédente comme données d'entrée.

II.2 Vue sur l'intelligence artificielle

Intelligence Artificielle (IA) est l'un des domaines les plus récents de la science et de l'ingénierie. Les travaux ont sérieusement débute après la seconde guerre mondiale, et le nom lui-même a été inventé en 1956. Régulièrement cité comme "domaine ou j'aimerai bien y être" par les scientifiques dans d'autres disciplines.

Un étudiant en physique peut raisonnablement se dire que toutes les bonnes idées ont été trouvées par Galilée, Newton, Einstein et le reste. De l'autre côté, tout reste ouvert en IA.

Historiquement, quatre approches de l'IA ont été suivies, chacune par des gens différents avec des méthodes différentes. Une approche axée sur l'homme doit être en partie une science empirique, impliquant observation et hypothèse sur le comportement humain. Une approche rationnelle implique une combinaison de mathématique et d'ingénierie . Les différents groupes se sont décriés et se sont aides mutuellement, voici les quatre approches :

- *Acting humanly* : Le test de Turing "L'art de créer des machines qui exécutent des fonctions requérant une intelligence lorsqu'elles sont exécutées par des êtres humains" [24]
- *Thinking humanly* : La modélisation cognitive "L'excitent nouveau défi de construire des ordinateurs qui pensent, des machines avec des consciences, au sens propre comme au sens figuré "[25]

— *Thinking rationally* : Les lois de la pensée "L'étude des calculs qui rendent possibles la perception, le raisonnement et les actes"[26]

— *Acting rationally* : les agents rationnels "L'IA est l'étude de la conception des agents intelligents"[27] L'IA englobe plusieurs sous domaines allant du plus général (apprentissage et perception) au plus spécifique, comme jouer aux échecs, démontrer des théorèmes mathématiques, écrire des poèmes, conduire une voiture ou diagnostiquer des maladies. L'IA se révèle être utile dans toutes les taches intellectuelles. C'est vraiment un domaine universel et pluridisciplinaire.

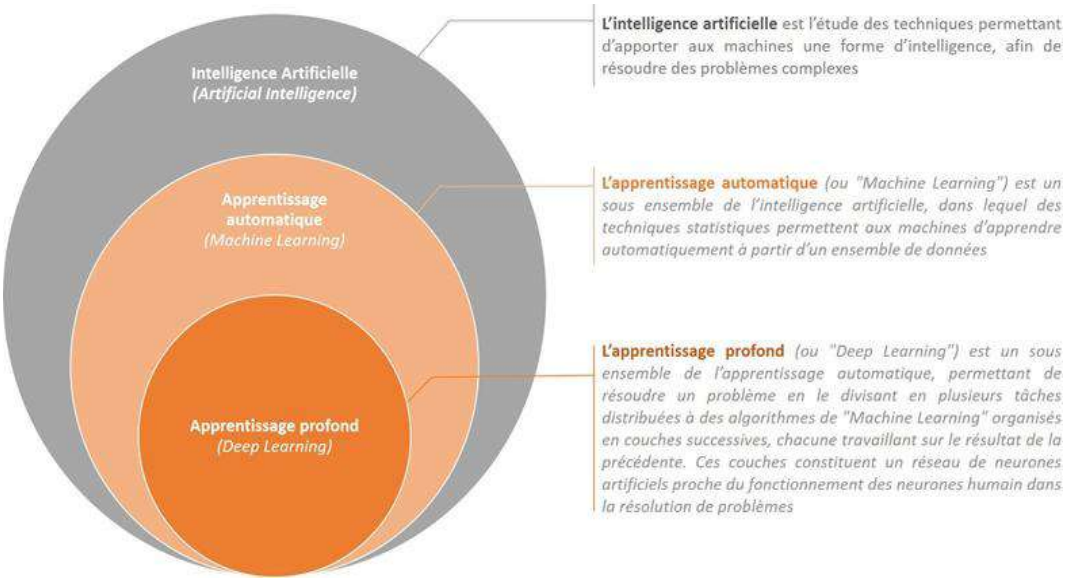


Figure II.1 : Relations entre les ensembles "intelligence artificielle", "apprentissage automatique" et "apprentissage profond" [28]

II.3 Qu'est-ce que le machine Learning

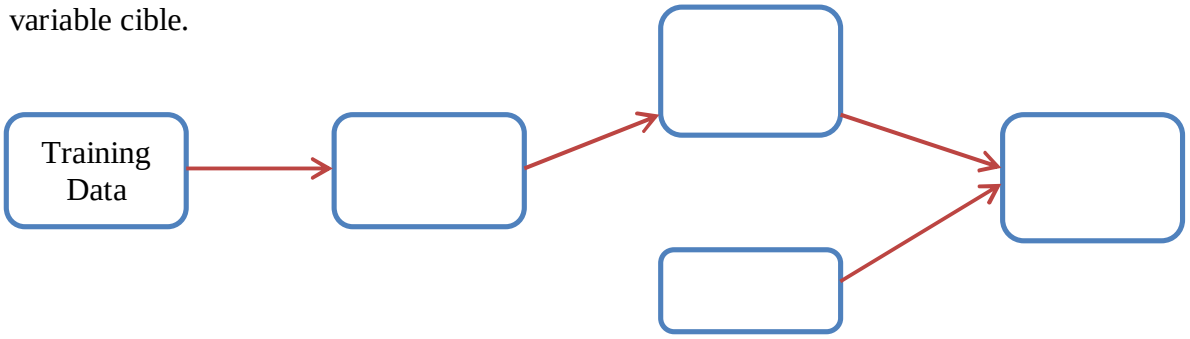
Le ML est une discipline de l'IA qui offre aux ordinateurs la possibilité d'apprendre à partir d'un ensemble d'observations que l'on appelle ensemble d'apprentissage.

Chaque observation, comme par exemple « j'ai mangé tels et tels aliments à tel moment de la journée pendant telle période ce qui a causé telle maladie » est décrite au moyen de deux types de variables :

- Les premières sont appelées les variables prédictives (ou attributs ou caractéristiques), dans notre exemple les caractéristique des visages sont les variables à partir desquelles on espère pouvoir faire des prédictions. les n variables prédictives associées à une observation seront

notées comme un vecteur $x = (x1, \dots, x2)$,à n composantes. Un ensemble de M observations sera constitué de M tels vecteurs $x^{(1)}$, ..., $x^{(M)}$.

- Une variable cible dont on souhaite prédire la valeur pour des événements non encore observés. Dans notre exemple, il s'agirait de la reconnaissance faciale . on notera y cette variable cible.



FigureII.2 : le processus typique du ML

En résume, la valeur de la variable y dépend de :

- Une fonction F(x) déterminée par les variables prédictives.
- Un bruit ε(x) qui est le résultat d'un nombre de paramètres dont on ne peut pas tenir compte.

Aussi bien F que ε ne seront jamais connues mais l'objectif d'un modèle de ML est d'obtenir la meilleure approximation possible de F à partir des observations disponibles. Cette approximation sera notée f, on l'appelle la fonction de prédiction.

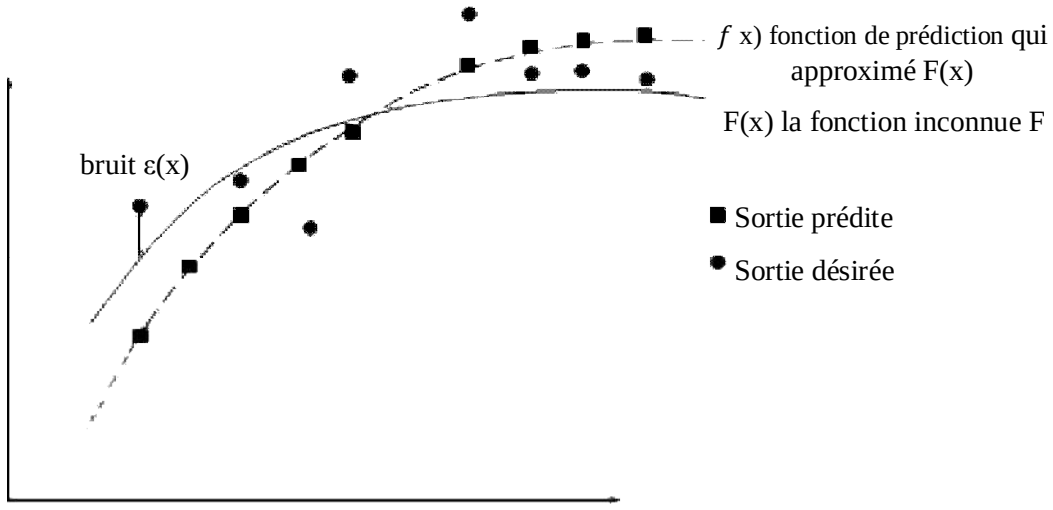


Figure II.3 : un modèle de ML qui essaye d'obtenir la meilleure approximation possible de F.

Voici quelques exemples d'utilisation du ML :

- Vision par ordinateur .
- Classification (images , textes , son , vidéo ..) .
- Détection de fraude .
- Diagnostic médical .
- Les publicités ciblé .

II.3.1 performance et sur-apprentissage

On peut penser que la performance d'un modèle sera en fonction des prédictions correctes faites sur l'ensemble d'observation utilisées pour l'apprentissage, plus elle est élevé, mieux c'est.

Pourtant c'est complètement faux, ce qu'on cherche à obtenir du ML n'est pas de prédire avec exactitude les valeurs des variables cibles connues puisque elles ont été utilisées pour l'apprentissage mais bien de prédire celles qui n'ont pas encore été observées. Par conséquent, la qualité d'un algorithme de ML se juge sur sa capacité à faire les bonnes prédictions sur les nouvelles observations grâce aux caractéristiques apprises lors de la phase d'entraînement. Il faut donc éviter le cas où on a un modèle de ML tellement trop entraîné qu'il arrive à prédire à la perfection les données d'apprentissage mais qui n'arrive pas à généraliser sur les données de test. On l'appelle le sur-apprentissage.

La cause du sur-apprentissage est que le modèle est trop complexe par rapport à la fonction F que l'on souhaite apprendre.

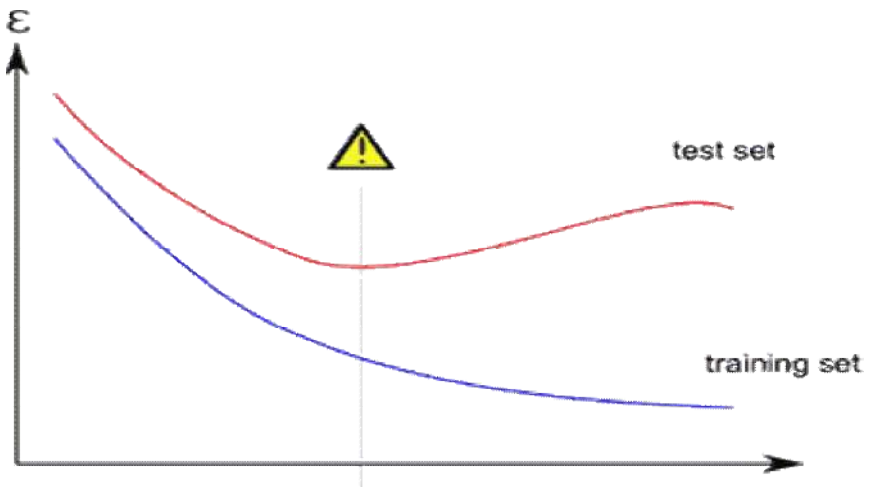


Figure II.4 : Le sur-apprentissage : le graphe montre l'évolution de l'erreur commise sur l'ensemble de test par rapport à celle commise sur l'ensemble d'apprentissage, les deux erreurs diminuent mais dès que l'on

rentre dans une phase de sur-apprentissage, l'erreur d'apprentissage continue de diminuer alors que celle du test augmente.

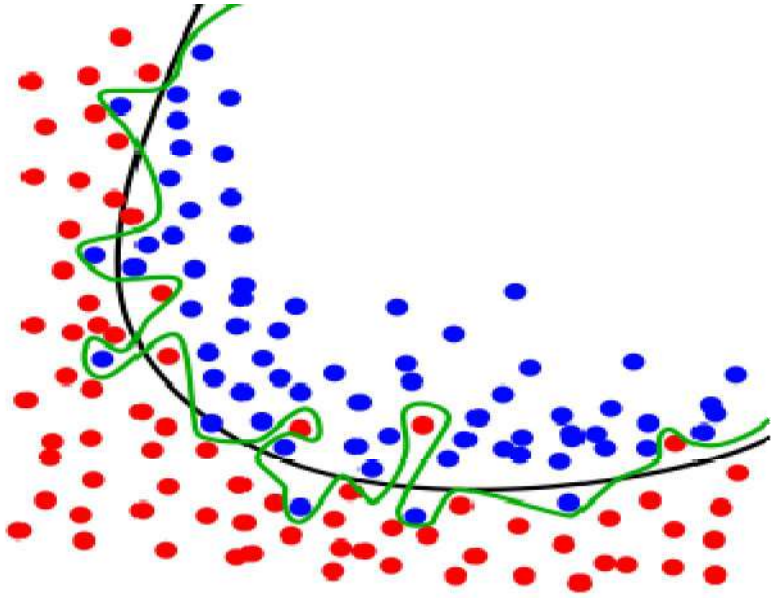


Figure II.5 : La ligne verte représente un modèle surentraîné et la ligne noire représente un modèle régularisé. Ce dernier aura une erreur de test moins importante.

Pour résoudre ce problème, on divise les données disponibles en deux groupes distincts. Le premier sera l'ensemble d'apprentissage, et le deuxième sera l'ensemble de test.

Pour avoir une bonne séparation des données en données d'apprentissage et données de test, on utilise la validation croisée. L'idée c'est de séparer aléatoirement les données dont on dispose en k parties séparées de même taille. Parmi ces k parties, une fera office d'ensemble de test et les autres constitueront l'ensemble d'apprentissage. Après que chaque échantillon ait été utilisé une fois comme ensemble de test. On calcule la moyenne des k erreurs moyennes pour estimer l'erreur de prédiction.

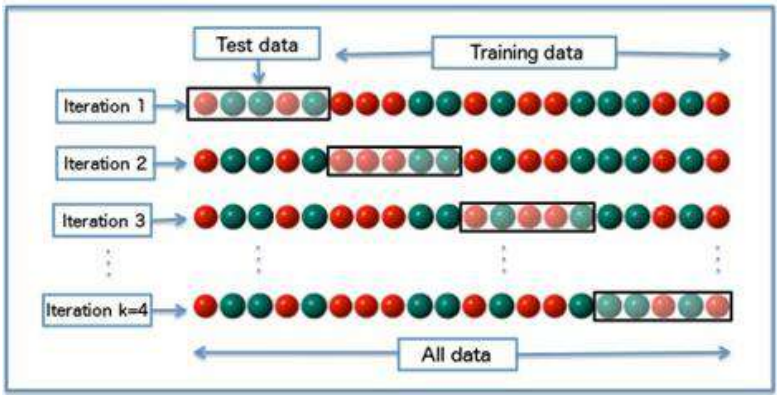


Figure II.6 : La validation croisée.

II.3.2 Les différents types de machine learning

De manière générale, un algorithme d'apprentissage automatique utilise un certain nombre de données d'entrées (pouvant être nécessairement très important pour de Deep Learning), représentant des données liées au problème à résoudre. Cet apprentissage dépend de la manière dont sont transmises ces données à l'algorithme de traitement.

Il existe donc différentes manières de procéder, et on distingue trois grands groupes d'apprentissage : l'apprentissage supervisée (supervised learning), l'apprentissage non supervisée (unsupervised learning), et l'apprentissage par renforcement (reinforcement learning).

— Apprentissage supervisé :

Dans le cas de l'apprentissage supervisé, les données d'apprentissage fournies à l'algorithme contiennent également les solutions associées, que l'on appelle des "étiquettes" ou des "labels". Toutes ces informations sont fournies par un opérateur humain. L'objectif de l'algorithme est alors trouver les relations (f) entre les données d'entrée (X) et la sortie (Y). Ainsi, la relation entre les données d'entrées et la sortie est définie par Y = f (X). L'apprentissage permet donc d'approximer cette fonction afin d'être capable, après l'apprentissage, de prédire une sortie appropriée pour une nouvelle entrée correspondant aux données d'apprentissage (sans label cette fois-ci).

— Apprentissage non-supervisé :

Ici, contrairement à l'apprentissage supervisé, aucune donnée d'entrée ne porte de label. Avec l'apprentissage non-supervisé, on laisse l'algorithme découvrir les relations entre les données d'entrée et leur structure. Le système cherche donc à découvrir et apprendre tout ce qu'il est capable de détecter (relations et différences) à partir de ces informations d'entrée et la sortie dépend donc de ce que l'algorithme a su décrire.

— Apprentissage par renforcement :

L'apprentissage par renforcement tire son nom du fait que l'apprentissage est amélioré à chaque itération du processus par un retour d'expérience. Les performances de l'algorithme sont maximisées à mesure qu'il prend des décisions à partir de ses données d'apprentissage. L'algorithme est plongé au sein d'un environnement, et prend ses décisions en fonction de son état courant. Après chaque étape d'apprentissage, l'environnement retourne une récompense, qui peut être positive ou négative, en fonction du résultat de l'étape précédente. Ainsi, au travers d'expériences itérées, l'algorithme cherche un comportement décisionnel optimal, afin qu'il maximise la somme des récompenses au cours du temps.

On peut également citer l'apprentissage semi-supervisé (semi-supervised learning). Cependant, ce n'est pas un groupe à part entière, du fait qu'il dérive à la fois de l'apprentissage supervisé et non supervisé. Certaines données portent une étiquette mais la majorité demeure non annotées. L'acquisition de données annotées pour l'apprentissage requiert des compétences humaines ainsi que du temps, tandis que le coût associé à l'acquisition de données non annotées est très court en comparaison. Ainsi, les algorithmes d'apprentissages semi-supervisés cumulent les avantages des deux méthodes, permettant, avec la faible quantité de données annotées, d'orienter l'apprentissage, puis, avec les données non annotées, de découvrir une plus large quantité d'informations relationnelles entre les données.

II.3.3 Classification , Régression , Association de règle et Clustering

Dans le cadre de l'apprentissage automatique, on peut établir plusieurs groupes de problématiques, auxquels les types d'apprentissages peuvent répondre. On retrouve principalement les problématiques de "classification", de "régression", d'"associations de règles" et de "clustering", Dans certains cas, afin d'utiliser un algorithme supervisé pour la classification de données non annotées, il est possible d'utiliser en complément un algorithme non supervisé dans le but de séparer les données de manière automatique. Le choix de l'algorithme dépend alors de nombreux facteurs tels que la quantité de données, leur qualité et leur nature, la puissance de calcul disponible ainsi que les temps de calculs acceptables selon l'application, etc.

La classification a pour objectif d'assigner des données à des catégories définies au préalable. Lors de l'apprentissage, les relations entre les données et leurs catégories (ou classes) sont déterminées et apprises. Ainsi, la classification permet d'obtenir, pour un nouvel échantillon, une prédiction d'appartenance à une classe connue, telle que "malade"/"sain" et "rouge"/"bleu"/.../"vert". Dans le

cas d'un problème à deux classes on parle de classification binaire ou binomiale, et dans le cas à plus de deux classes, on parle de classification à classes multiples ou multi classes.

La régression a pour but de prédire la valeur d'une certaine quantité à l'évolution continue, dépendant de certaines caractéristiques. Ce problème est établi lorsque la donnée de sortie est une valeur réelle, telle qu'une somme d'argent ou un poids. On peut citer l'estimation de valeurs telles que l'évolution des prix de locations d'appartement selon le secteur et les caractéristiques du bien.

Un problème d'apprentissage *d'association de règles* se présente comme un problème où l'on souhaite associer des règles décrivant majoritairement des grandes parties des données (ex. les personnes qui achètent un objet X ont également tendance à acheter un objet Y). Une règle d'association comporte deux parties : un antécédent ("si") et un résultant ("alors"). Un antécédent est un élément compris dans les données, et un résultant est une combinaison d'antécédent suivant une relation précise. Ainsi, l'algorithme recherche au travers des données la fréquence de modèles "si-alors" afin d'identifier les relations les plus importantes entre les données.

Le clustering se présente comme un problème où l'on souhaite établir des groupes (ou des "clusters") selon des caractéristiques particulières propres aux données (ex. regrouper des clients selon leurs habitudes et leur comportement d'achat). Comme exemple particulier au clustering on peut citer la détection d'anomalies dont l'objectif est d'identifier les points de données qui sont inhabituels. Dans certains cas, les variations possibles sont si nombreuses et les exemples réels d'apprentissage si rares, qu'il n'est pas possible de savoir à quoi ressemble une activité frauduleuse. L'approche de la détection des anomalies consiste donc simplement à apprendre à quoi ressemble l'activité normale et d'identifier tout ce qui est très différent.

Pour la suite de cette thèse, nous allons nous intéresser à l'apprentissage supervisé dans le cadre de la classification d'images. Du fait de notre problématique, à savoir déterminer si une personne est autorisée ou non autorisée, nous nous focaliserons sur une étude de classification binaire. Avant de considérer une quelconque application à la biométrie, nous allons établir les généralités propres au traitement d'image et communes à toutes les applications de ce domaine.

— *Chaine de traitement pour la classification d'image :*

la chaine de traitement générale (commune aux algorithmes de classification d'images) comporte plusieurs étapes clés telles que la détection et l'isolation de zone d'intérêt (Region of Interest - ROI), des prétraitements afin d'améliorer l'image dans le sens voulu suivi d'une potentielle normalisation, l'extraction de caractéristiques ainsi que leur sélection, puis la classification

effectuant soit un apprentissage d'un modèle, soit une prédiction à partir du modèle appris. Cette prédiction donne lieu à une décision qui, à partir de multiples résultats moyennés sur une base de données d'images, fourni un certain nombre d'informations servant à évaluer les performances du modèle.

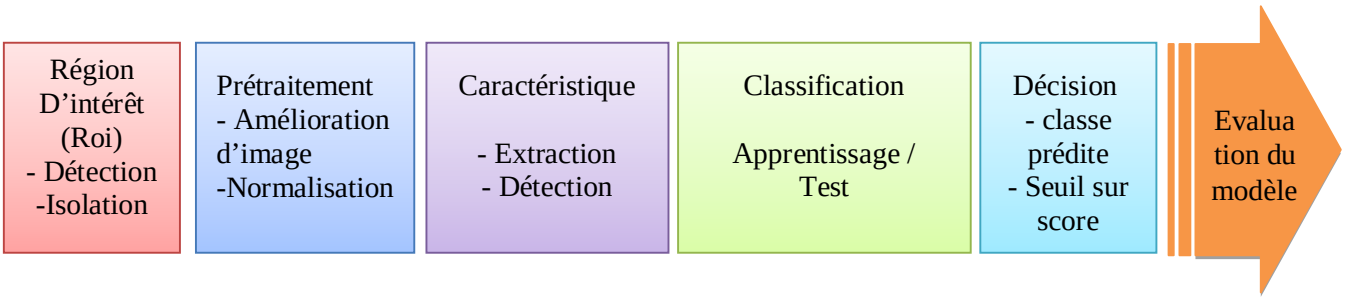


Figure II.7 : Chaîne des étapes de classification d'images communes à la majorité des applications

La première étape consiste à extraire la zone d'intérêt dans laquelle se trouve les éléments que l'on veut classifier. Cette étape, bien que facultative, permet de réduire l'influence des données peu pertinentes dans l'image. Par exemple, dans le cas de la reconnaissance de visages, il est important d'isoler le visage du fond de l'image. En revanche, dans le cas d'une détection multiple (piétons, voiture, panneaux, etc.), les informations sont réparties dans toute l'image.

Les prétraitements constituent une étape non négligeable de cette chaîne. La manipulation de l'image (filtres, segmentation binaire, superpixels, changement d'espaces couleurs, etc.) permet de renforcer ou faire apparaître les éléments importants tout en réduisant les contributions non utiles. La phase de normalisation permet de conserver une homogénéité entre les images utilisées pour l'apprentissage. Les images sont redimensionnées à une même dimension et une rotation de l'image peut être envisagée pour conserver une orientation similaire du sujet à observer.

L'extraction de caractéristiques est une étape cruciale, permettant de transcrire l'information visuelle en une information plus stable et compréhensible par l'algorithme d'apprentissage. Certains algorithmes, basés sur la répartition statistique des données, permettent de trouver les caractéristiques communes aux images d'une simple classe ou bien de détecter les caractéristiques discriminant au mieux une classe par rapport à une autre tout en cherchant à réduire l'impact des caractéristiques communes à plusieurs classes. Ces caractéristiques ne représentent pas toutes la même contribution dans l'image et leur nombre peut rapidement être important. C'est pourquoi une phase complémentaire de sélection de caractéristiques permet de limiter leur nombre tout en ne conservant que les plus pertinentes.

La classification est divisée en deux phases : l'apprentissage et le test (ou l'utilisation). Lors de l'apprentissage, les caractéristiques des images d'apprentissages sont utilisées par le classifieur afin de construire un modèle à partir de règles séparant au mieux les classes. Selon l'application et les images utilisées, il est nécessaire d'ajuster les paramètres de classification, et l'apprentissage peut être perfectionné en utilisant une technique de validation croisée *(ou "cross validation")*, permettant de scinder les images en plusieurs groupes et d'effectuer un apprentissage sur chacun de ces groupes en faisant varier les paramètres.

La phase d'utilisation permet, à partir des caractéristiques d'une image inconnue, de déterminer l'appartenance de l'échantillon à une certaine classe en utilisant le modèle de classification. Ainsi ce modèle fourni l'étiquette de la classe prédite et le score associé attestant de la probabilité d'une bonne prédiction. L'étape de décision consiste à définir si la prédiction du modèle est juste ou non. Soit la classe prédite est utilisée directement (prédiction juste), soit, un seuil peut être défini et ajusté sur le score de la prédiction afin de considérer une prédiction comme juste seulement si le score est supérieur au seuil fixé. Pour finir, l'évaluation permet de quantifier les performances du modèle avec des métriques calculées à partir d'informations issues des résultats de prédiction sur des images de test. Ces métriques permettent de comparer des modèles entre eux et ainsi sélectionner le modèle optimal dans un cas donné.

Lors de l'apprentissage, une mauvaise répartition des données d'entrées ainsi qu'un mauvais para- métrage de la classification peuvent mener à un "sur-apprentissage" ou à un "sous-apprentissage". Le sur-apprentissage correspond à un modèle considérant les détails et le bruit comme des données importantes, menant à d'excellentes performances lors de l'apprentissage, mais de très mauvaises sur de nouvelles images inconnues. Le sous-apprentissage fait référence à un modèle incapable de fournir de bonnes performances en apprentissage ni à partir de nouvelles données.[29]

— *Extraction et traitement des caractéristiques :*

L'extraction de caractéristiques est une étape cruciale de la chaîne de traitement de la classification d'images, les méthodes peuvent être regroupées selon deux catégories principales [30], comme le montre la figure II.8 , à savoir les méthodes globales et les méthodes locales. Les méthodes globales sont indépendantes de l'application (du type d'élément présent dans l'image) et se focalisent sur des caractéristiques telles que les textures, la couleur ou encore les formes. On peut considérer trois niveaux d'abstraction pour ces caractéristiques : le niveau pixel (couleur, coordonnées dans l'image, etc.), le niveau local (caractéristiques issues de *subdivision* de l'image, de segmentation, ou autre) et le niveau global (caractéristiques issues de l'image entière ou d'une

seule région de l'image). En comparaison, les méthodes locales sont dépendantes de l'application et représentent souvent une synthèse de caractéristiques bas-niveau. Chacune de ces deux catégories peuvent être sous divisées selon le type de techniques [31].

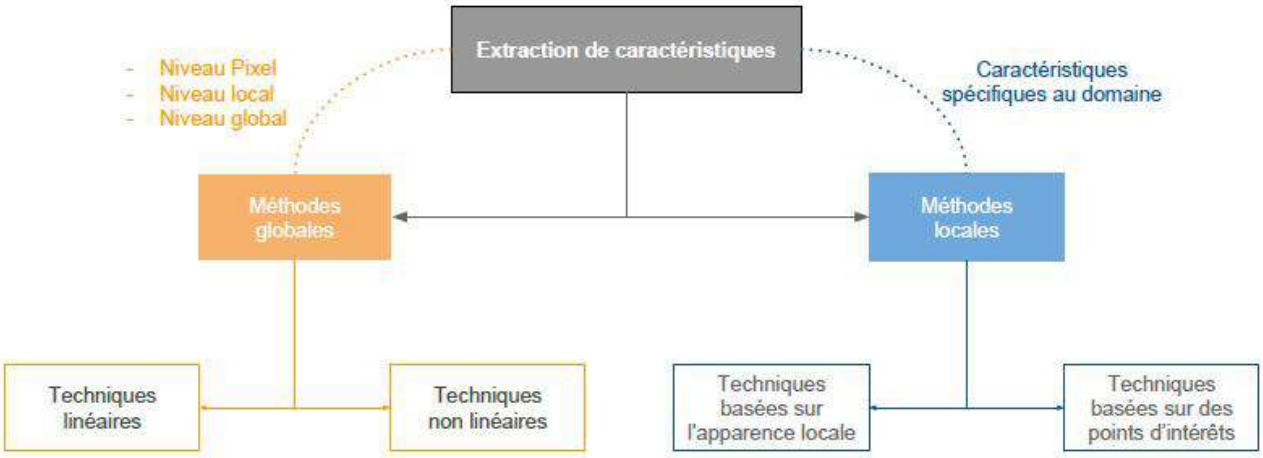


Figure II.8 : Catégories de méthodes d'extractions de caractéristiques dans l'image.

— Méthodes globales :

Les approches globales, également appelées méthodes basées sur l'apparence (Appearance-based methods), utilisent l'image complète. De manière générale, l'image est représentée selon une matrice de pixels, qui est transformée en un vecteur de pixels pour faciliter les manipulations. Ces approches sont sensibles aux variations d'acquisition (orientation, éclairage, etc.) du fait que le moindre changement induit une variation de la valeur des pixels. Ces méthodes utilisent un sous espace (de faible dimension) du sujet à considérer, afin d'y projeter de nouvelles images et ainsi réduire l'influence d'éléments non pertinents dans l'image. Considérant les techniques utilisées pour générer cette projection, cette catégorie regroupe deux types de techniques : les linéaires et les non linéaires.

Les techniques linéaires effectuent une projection linéaire des données d'entrées, représentées dans un espace de très grande dimension (dépendant du nombre de pixels et de canaux de l'image), dans un nouvel espace relativement réduit et dédié au "sujet". Cet espace est composé de caractéristiques propres aux types d'éléments observés.

De nombreuses techniques peuvent être classées comme linéaires : l'Analyse en Composantes Principales "ACP" (ou "PCA" pour "Principle Component Analysis"), l'Analyse en Composantes

In- dépendantes "ACI" (ou "ICA" pour "Independant Component Analysis"), l'Analyse Discriminante Linéaire "ADL" (ou "LDA" pour "Linear Discriminent Analysis"), les ondelettes de Gabor, et beaucoup d'autres. L'ACP est une approche très populaire, sur laquelle nous nous attarderons plus particulièrement, de fait de son importante utilisation par de nombreux algorithmes dans le domaine de la biométrie.

L'ACI [32] est une méthode issue d'applications statistiques et de traitement du signal, spécialement utilisée pour effectuer une séparation "aveugle" de sources. l'ACI consiste à exprimer un ensemble de N variables aléatoires x_1, ..., x_n comme une combinaison linéaire de N variables aléatoires s j statistiquement indépendantes.

L'ADL [33], contrairement à l'ACP qui construit un sous espace pour décrire de manière optimale un type "d'objet", construit un sous espace discriminant pour les distinguer, de manière optimale, les variations entre différentes catégories d'un même type d'objet. En d'autres termes, l'ADL permet de trouver des caractéristiques séparant au mieux les classes. Lorsque la structure des données d'entrée n'est pas linéaire, une solution consiste à utiliser une fonction nommée fonction noyau (ou kernel). Dans ce cas, un espace de grande dimension est créé dans lequel la représentation du problème devient linéaire. Les méthodes linéaires peuvent être adaptées afin de répondre à ce problème, de par l'ajout d'un noyau [34]. Ainsi, on retrouve l'Analyse en Composantes Principales avec Noyau (Kernel PCA - KPCA) , l'Analyse en Composantes Indépendantes avec Noyau (Kernel ICA - KICA) ,Analyse Discriminante Exponentielle "ADE" (ou EDA - "Exponential Discriminant Analysis") [35], et nombreux autres.

— Méthodes locales :

Les méthodes locales, ne traitant que des caractéristiques très spécifiques au type d'informations observées dans l'image, peuvent être divisées en deux catégories : les techniques basées sur l'apparence locale et les techniques basées sur des points d'intérêts. Les techniques basées sur l'apparence locale divisent l'image en petites régions ou "patchs" desquelles des caractéristiques locales sont directement extraites. Une fois ces régions définies, il advient de choisir la meilleure façon de représenter l'information de chaque région. Les caractéristiques les plus généralement utilisées dans la littérature sont les coefficients de Gabor [36], les ondelettes de Haar [], la transformation de caractéristiques visuelles invariante à l'échelle (Scale-Invariant Feature Transform - SIFT) [37], les caractéristiques basées sur les motifs binaires locaux (Local Binary Pattern - LBP) et leurs extensions tels que les motifs ternaires locaux (Local Ternary Pattern LTP) [38], et d'autres.

Les techniques basées sur des points d'intérêts détectent d'abord ces points spécifiques, permettant par la suite d'extraire les caractéristiques représentant les diverses relations entre ces points tels que leur distance les uns avec les autres, leur angle, etc. Ce type d'extraction est réalisé avec des approches telles que l'Architecture de Liens Dynamiques (Dynamic Link Architecture - DLA) [39], l'extraction par filtre de Gabor [40], etc.

Une troisième catégorie pourrait être définie, composée de méthodes hybrides combinant des techniques des deux catégories précédentes et de nouvelles méthodes basées sur des modèles statistiques. L'objectif de cette méthode, dans un contexte d'analyse d'images, consiste à exprimer un ensemble de M images selon une base de vecteurs orthogonaux particuliers (des vecteurs propres), contenant des informations indépendantes d'un vecteur à l'autre. Ainsi, ces vecteurs sont une représentation simplifiée et optimale de ces images dans le sens où la variance est maximisée. Chaque vecteur décrit alors des caractéristiques indépendantes. Une projection d'une nouvelle image dans cette base permet de représenter l'information de manière normalisée, selon une combinaison linéaire de ces vecteurs propres, et donc de faciliter la comparaison de ces nouvelles données.

— *Classification*

L'extraction de caractéristiques fournit donc un vecteur composé d'éléments représentant ces caractéristiques.

L'étape suivante de la chaîne est la classification. Son but est de calculer le degré

similitude entre 2 vecteurs (caractéristiques cible et caractéristique mesurée) ou entre 1 vecteur (caractéristique mesurée) et un ensemble de vecteurs (formant une classe). Cette comparaison peut être effectuée de différente façons, plus ou moins efficaces selon la complexité des données (dimensions des vecteurs, variance interclasse, séparation des classes, etc.).

II.3.4 Méthodes de Classifications

II.3.4.a Les k plus proches voisins

L'algorithme des K-Nearest Neighbors (KNN) (K plus proches voisins) est un algorithme de classification supervisé. Chaque observation de l'ensemble d'apprentissage est représentée par un point dans un espace à n dimensions ou n est le nombre de variables prédictives. Pour prédire la classe d'une observation, on cherche les k points les plus proches de cet exemple. La classe de la variable cible, est celle qui est la plus représentée parmi les k plus proches voisins. Il existe des variantes de l'algorithme ou on pondère les k observations en fonction de leur distance à l'exemple

dont on veut classer [41], les observations les plus éloignées de notre exemple seront considérées comme moins importantes. Une variante de l'algorithme est utilisée par NetFlix [48] pour prédire les scores qu'un utilisateur attribuera à un film en fonction des scores qu'il a attribués à des films similaires .

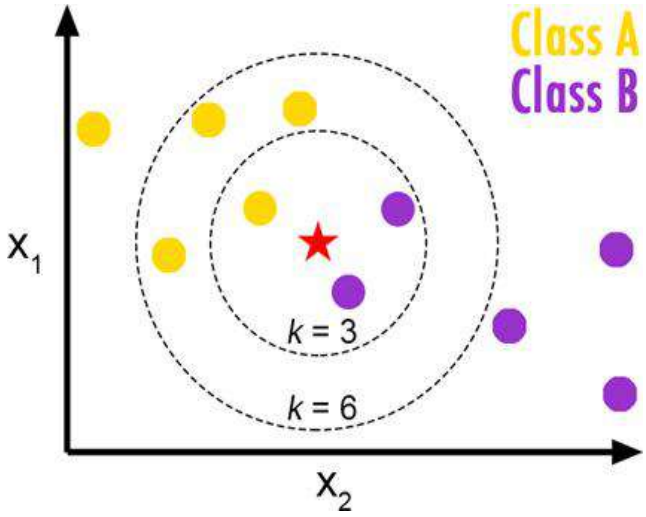


Figure II.9 : Pour k = 3 la classe majoritaire du point central est la classe B, mais si on change la valeur du voisinage k = 6 la classe majoritaire devient la classe A.

- Avantages :

  — simple à concevoir .

- Inconvénients :

  — Sensible aux bruits
  — Pour un nombre de variable prédictives très grands, le calcul de la distance devient très coûteux.

II.3.4.b Le classifieur naïf de Bayes

Le classifieur naïf de Bayes est un algorithme supervisé probabiliste qui suppose que l'existence d'une caractéristique pour une classe, est indépendante de l'existence d'autres caractéristiques, raison pour laquelle on utilise l'adjectif «naif» . Une personne peut être considérée comme un homme si il pèse un certain poids et mesure une certaine taille. Même si ces caractéristiques sont

liées dans la réalité, un classifieur bayésien naïf déterminera que la personne est un homme en considérant indépendamment ces caractéristiques de taille et de poids.

Malgré des hypothèses de base extrêmement simplistes, ce classifieur conduit à de très bons résultats dans beaucoup de situations réelles complexes. En 2004, un article a montré qu'il existe des raisons théoriques derrière cette efficacité inattendue [42]. Toutefois, une autre étude de 2006 montre que des approches plus récentes (arbres renforcés, forêts aléatoires) permettent d'obtenir de meilleurs résultats[43].

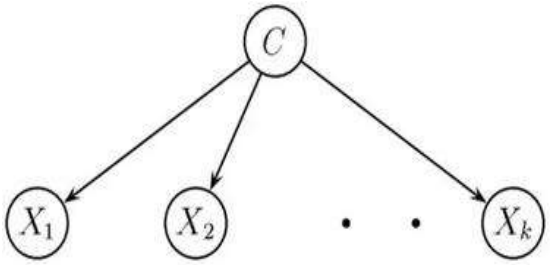


Figure II.10 : Le classifieur naïf de Bayes est basé sur le théorème de Bayes avec une indépendance (dite naïve) des variables prédictives.

- Avantages :

  — L'algorithme offre de bonne performance

- Inconvénients :

  — La prédiction devient erronée si L'Hypothèse indépendance conditionnelle est invalide.

II.3.4.c K-moyennes et K-plus proches voisins

La première étape du regroupement par les K-moyennes [44] consiste à déterminer le nombre K de groupes et de considérer le centre de chacun d'entre eux. Ce centre est défini à priori de manière aléatoire au sein du groupe (apprentissage non supervisé) mais peut être choisi selon certaines données annotées (apprentissage semi-supervisé). Par la suite, l'algorithme effectue en boucle trois étapes jusqu'à la convergence : recalculer les coordonnées du barycentre de chaque ensemble, déterminer la distance entre chaque donnée et ce centre, puis compléter les ensembles en

regroupant les échantillons les plus proches. Les K-plus proches voisins (k-Nearest Neighbors - KNN) [45] font, quant à eux, partie de l'apprentissage supervisés. Les K-NN n'ont pas besoin de construire un modèle prédictif, il n'existe pas de phase d'apprentissage proprement dite (ils sont donc souvent associés au terme "Lazy Learning"). Pour effectuer une prédiction, l'algorithme se base sur le jeu de données complet et, pour une nouvelle observation, va rechercher les K instances du jeu de données les plus proches de l'observation. Ensuite pour ses K voisins, l'algorithme se base sur leurs variables de sortie pour assigner le nouvel échantillon à une classe.

Les étapes de l'algorithme sont :

- Choisir k points qui représentent la position moyenne des cluster.
- répéter jusqu'à stabilisation des points centraux :
    - affecter chacun des M points au plus proche des *k* points centraux .
    - mettre à jour les points centraux en calculant les centres de gravité des *k* cluster .
- Avantages :

— implémentable pour des grands volumes de données .

- Inconvénients :

— Le choix du paramètre *k* n'est pas découvert mais choisi par l'utilisateur .

— La solution dépend des *k* centre de gravité choisi lors de l'initialisation.

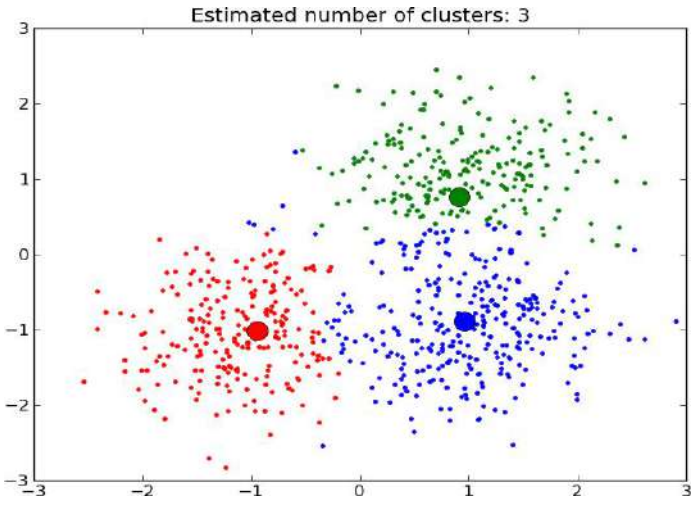


Figure II.11 : L'algorithme *k*-means regroupe les données en *k* cluster, ici *k* = 3. Les centres de gravité sont représentés par de petit cercle.

II.3.4.c Les arbres de décision

Les arbres de décision sont des modèles de ML supervisés, pouvant être utilisés pour la classification que pour la régression.

Un arbre de décision représente une fonction qui prend comme entrée un vecteur d'attributs et retourne une décision qui est une valeur unique. Les entrées et les sorties peuvent être discrètes ou continues.

Une arbre de décision prend ses décisions en exécutant une séquence de test, chaque nœud interne de l'arbre correspond à un test de la valeur d'un attribut et les branches qui sortent du nœud sont les valeurs possibles de l'attribut. La classe de la variable cible est alors déterminée par la feuille dans laquelle parvient l'observation à l'issue de la séquence de test.

La phase d'apprentissage consiste à trouver la bonne séquence de test. Pour cela, on doit décider des bons attributs à garder. Un bon attribut divise les exemples en ensembles homogènes c.à.d. qu'ils ne contiennent que des observations appartenant à la même classe, alors qu'un attribut inutile laissera les exemples avec presque la même proportion de valeur pour la variable cible .
Ce dont on a besoin c'est d'une mesure formelle de "bon" et "inutile". Pour cela, il existe des métriques standards homogénéisées avec lesquels on peut mesurer l'homogénéité d'un ensemble. Les plus connus sont l'indice de diversité de Gini et l'entropie [46] .

En général l'entropie d'une variable aléatoire V avec des valeurs v_k chacune avec une probabilité $P(v_k)$ est définie comme :

$$\text{Entropie : } H(V) = -\sum P(v_k)\, log_2\, P(v_k)$$

- Avantages :
  - C'est un modèle boîte blanche, simple à comprendre et à interpréter .
  - Peu de préparation des données.
  - Les variables prédictives en entrée peuvent être aussi bien qualitatives que quantitatives.
  - performant sur de grands jeux de données.

Example	Input Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 = Yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 = No$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 = Yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 = Yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = No$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 = Yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 = No$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 = Yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = No$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$y_{10} = No$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$y_{11} = No$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$y_{12} = Yes$

Figure II.12 : L'ensemble d'apprentissage contient 12 observations décrites par 10 variables prédictives et une variable cible. [47]

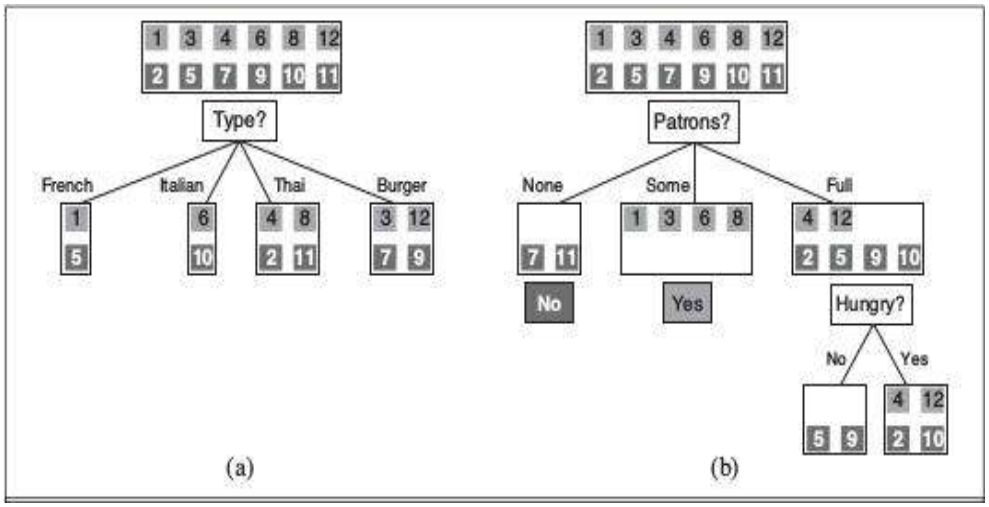


Figure II.13 : Les exemples positifs sont représentés par des cases claires alors que les exemples négatifs sont représentés par des cases sombres. (a) montre que la division par l'attribut *Type* n'aide pas à avoir une distinction entre les positifs et les négatifs exemples. (b) montre qu' avec la division par l'attribut *Patrons* on obtient une bonne séparation entre les deux classes. Après division par *Patrons*, *Hungry* est un bon second choix. [48]

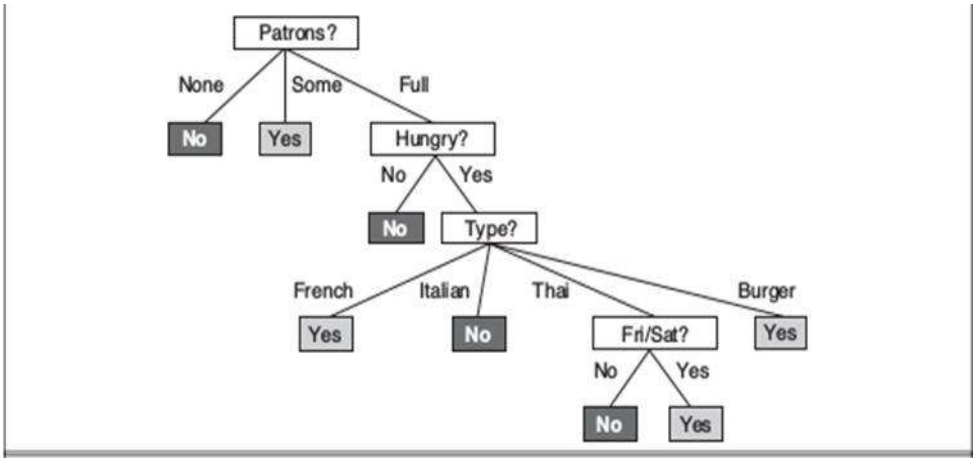


Figure II.14 : L'arbre de décision déduit à partir des 12 exemples d'apprentissage. [49]

- Inconvénients :
  - — L'existence d'un risque de sur-apprentissage si l'arbre devient très complexe. On utilise des procédures d'élagage pour contourner ce problème.

II.3.4.d Les forets aléatoires

Combiner un ensemble de classifieurs individuels faibles pour former un unique système de classification appelé Ensemble de Classifieurs a suscité un intérêt grandissant de la communauté scientifique. L'efficacité des combinaisons de classifieurs repose principalement sur leur capacité à tirer parti des complémentarités des classifieurs individuels, dans le but d'améliorer autant que possible les performances en généralisation de l'ensemble.

Parmi les différentes approches de construction d'ensembles de classifieurs, l'algorithme des forets aléatoires [50], composé d'un ensemble de classifieurs élémentaires de type arbres de décision, le but de l'algorithme est de conserver les avantages des arbres de décision tout en éliminant leurs inconvénients et particulièrement leur vulnérabilité au sur-apprentissage. C'est un algorithme qui peut être utilisé aussi bien pour la classification que pour la régression.

L'algorithme repose sur trois idées principales :

— Pour M observations de l'ensemble d'apprentissage, chacune décrite par n variables prédictives, on crée B nouveau échantillon de même taille M par tirage avec remise. Cette technique s'appelle le boostrap. Chacun des B échantillon servira à l'apprentissage d'un arbre de décision.

— Pour n caractéristiques, un nombre k < n (généralement $\sqrt{n}$)est tiré aléatoirement de sorte qu'à chaque nœud de l'arbre, un sous-ensemble de k caractéristiques soit tiré aléatoirement, parmi lesquelles la meilleure est ensuite sélectionnée pour le partitionnement.

— Pour classer une nouvelle observation, on procède par vote majoritaire. On fait passer cette observation par les B arbres et sa classe c'est la classe majoritaire parmi les B prédictions

Quelques travaux de recherche se sont intéressés au nombre d'arbres de décision à construire au sein d'une forêt. Quand Breiman a introduit les forets aléatoires, il démontra également qu'au-delà d'un certain nombre d'arbres, en ajouter d'autres ne permettait pas forcement d'améliorer les performances de l'ensemble. Ce résultat indique que le nombre d'arbres d'une foret aléatoire ne doit pas nécessairement être le plus grand possible pour produire un classifieur performant.

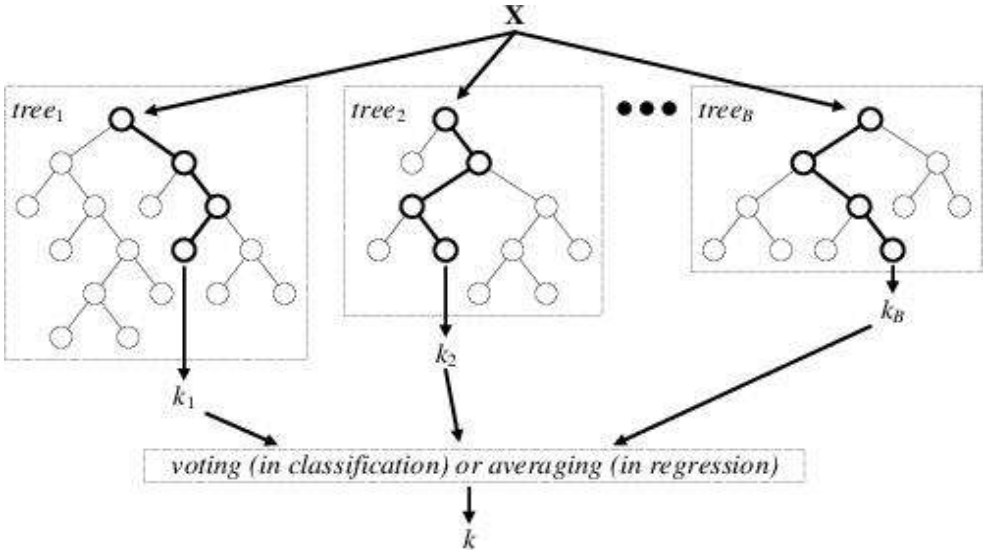


Figure II.15 : Création de B boostrap à partir des exemples d'apprentissage, chacun servira à entraîner un arbre de décision. Si c'est un cas de classification, la classe finale est attribuer par vote majoritaire. Si c'est un cas de régression, on fait la moyenne des prédictions .

- Avantages :

— C'est un des meilleur algorithme pour ce qui est de la précision.
— Incorporation de la validation croisée.
— Il conserve une bonne puissance de prédiction même si il y' a des données manquantes .
— Il ne souffre pas du sur-apprentissage .

- Inconvénients

— Une implémentation difficile .

II.3.4.e Les machines à vecteur de support (SVM - Support Vector Machine)

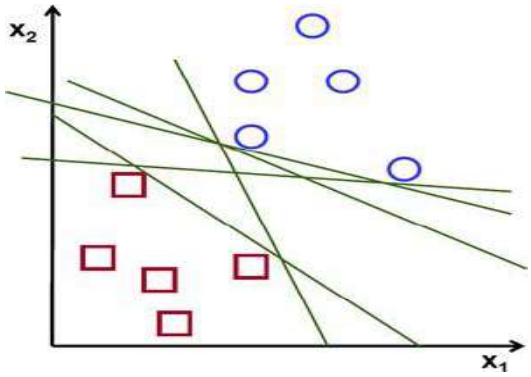
Les Support Vector Machine (SVM) (machines à vecteur de support) [51] sont des algorithmes de classification binaire non linaire très puissant. Le principe des SVM consiste à construire une bande séparatrice non linéaire de largeur maximale qui sépare deux ensembles ou classes d'observations et à l'utiliser pour faire des prédictions. l'astuce des SVM pour y parvenir consiste à utiliser une

transformation ϕnon linéaire qui envoie les points $x^{(1)},\dots , x^{(M)}$ de l'espace original à n dimensions (n est le nombre de variable prédicatives) vers des nouveaux points $\phi(\mathbf{x}^{(1)}) , \dots, \phi(\mathbf{x}^{(M)})$ d

Les SVM sont des classificateurs qui reposent sur deux idées clés :

La première idée consiste à trouver un séparateur linéaire de largeur maximale, c'est la notion de marge maximale. La marge est la distance entre la frontière de séparation et les échantillons les plus proches. Ces derniers sont appelés vecteurs supports. Le problème est de trouver cette frontière séparatrice optimale.

Dans le cas où le problème est linéairement séparable, le choix de l'hyperplan séparateur n'est pas évident. Il existe en effet une infinité d'hyperplans séparateurs, dont les performances en phase d'apprentissage sont identiques, mais dont les performances en phase de test peuvent être très différentes. Pour résoudre ce problème, il a été montré [52], qu'il existe un unique hyperplan



optimal, défini comme l'hyperplan qui maximise la marge entre les échantillons et l'hyperplan séparateur. Il existe des raisons théoriques à ce choix. Vapnik a montré [53] que la capacité des classes d'hyperplans séparateurs diminue lorsque leur marge augmente.

Figure II.16 : On cherche un hyperplan qui divise les observations en deux catégories. Considérons un exemple x qu'on veut classer, si f(x) > 0, il appartient à la classe des cercles, sinon il appartient à la classe des carrés. Dans cette figure on peut voir qu'il existe une infinité d'hyperplans séparateurs possibles.

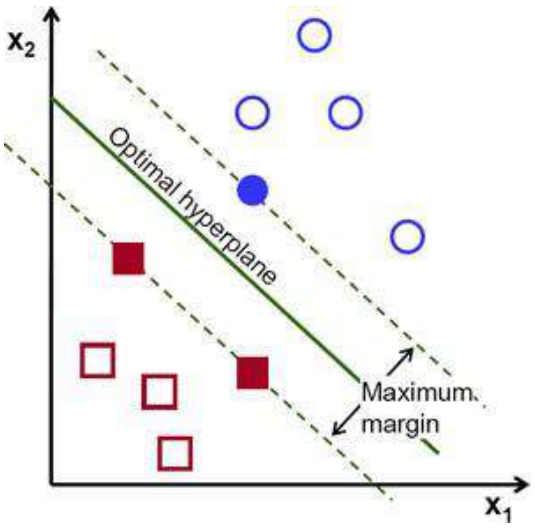


Figure II.17 : L'hyperplan optimal (en vert) avec la marge maximale. Les échantillons remplis sont des vecteurs supports.

Afin de pouvoir traiter des cas où les données ne sont pas linéairement séparables, la deuxième idée clé des SVM est de transformer l'espace de représentation des données d'entrées en un espace de plus grande dimension, dans lequel il est probable qu'il existe une séparation linéaire. Ceci est réalisé grâce à une fonction noyau, qui doit respecter les conditions du théorème de Mercer [54], et qui a l'avantage de ne pas nécessiter la connaissance explicite de la transformation à appliquer pour le changement d'espace. Les fonctions noyaux permettent de transformer un produit scalaire dans un espace de grande dimension, ce qui est coûteux, en une simple évaluation ponctuelle d'une fonction. Cette technique est connue sous le nom de kernel trick.

Les deux fonctions noyaux les plus utilisées sont le noyau polynomial et le noyau gaussien.

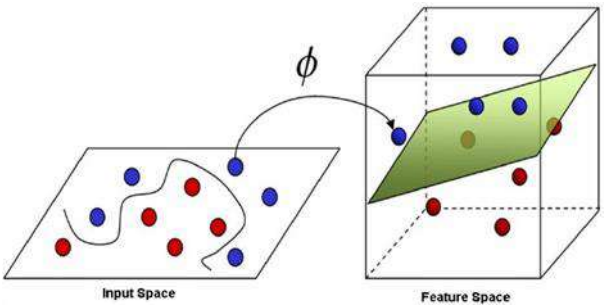


Figure II.18 : Exemple d'un problème non linéairement séparable. La courbe devient une bande linéaire après avoir appliqué la transformationϕ non linéaire .

- Avantages :

— Il permet de traiter des problèmes de classification non linéaire complexe.

— Les SVM constituent une alternative aux réseaux de neurones car plus faciles à entraîner.

- Inconvénients :

— Les SVM sont souvent moins performants que les forets aléatoires.

II.3.4.f Le perceptron multicouches

Multi Layer Perceptron (MLP) (perceptron multicouches)[55] est un classifieur de type réseaux de neurones qui est organisé en plusieurs couches, chaque couche étant formée d' un ou plusieurs neurones formels. Utilisé dans un cas d'apprentissage supervisé, il utilise l'algorithme de retro propagation de gradient [56].

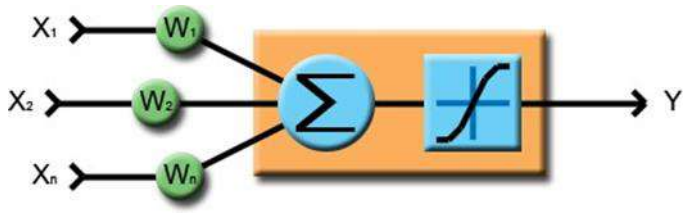


Figure II.19 : La figure représente le fonctionnement d'un neurone formel. C'est un composant calculatoire qui fait la somme pondérée des signaux reçus en entrée, puis on leur applique une fonction d'activation afin d'obtenir Y.

Voici les caractéristiques du modèle :

- Il comporte une seule couche d'entrée et une seule couche de sortie.
- il peut comporter une ou plusieurs couches cachées.
- Chaque neurone est relié uniquement à tous les neurones de la couche suivante.
- Chaque lien de la couche i vers la couche suivante j sert à propager l'activation a_i de i jusqu'à j et qui possède un poids $wi\ j$ qui détermine l'intensité du signal de la connexion. Chaque unité de la couche j calcul la somme pondéré de ses entrées :

$$in\ j = \sum_{i=0}^{n} wi\ j\ ai$$

puis leur applique une fonction d'activation :

$$aj = g(inj) = g(\sum_{i=0}^{n} wijai)$$

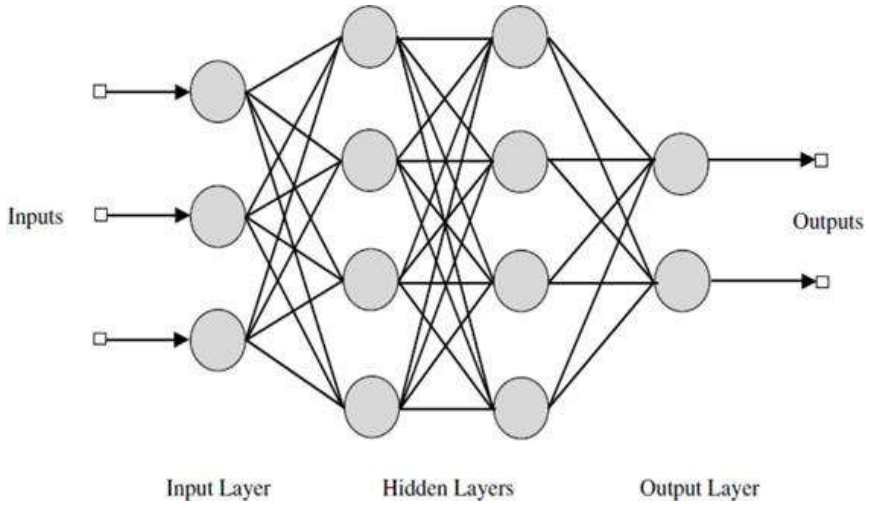


Figure II.20 : Structure d'un RN composé d'une couche d'entrée, deux couches cachées et une couche de sortie.

L'algorithme d'apprentissage consiste à présenter au réseau des entrées et on lui demande de modifier sa pondération de façon à retrouver la sortie correspondante. D'abord on propage vers l'avant les entrées jusqu'à obtenir une sortie calculée par le réseau puis on compare cette sortie avec la sortie désirée, enfin on modifie les poids de telle sorte qu'à la prochaine itération l'erreur commise entre les sortie réelles et les sortie désirées soit minimisée. On répète ce processus jusqu'à ce qu'on obtienne une erreur de sortie négligeable.

- Avantages :

— Capacité à découvrir les dépendances par lui-même.

— Résistance aux bruits

- Inconvénients :

— C'est un modèle boite noire qui n'explique pas ses décisions.

Le tableau II.1 établit une comparaison rapide des certains algorithmes présentés précédemment, à partir d'une évaluation sur certaines caractéristiques de ces classifieurs. Les performances sur ces évaluations sont représentées par des étoiles "*". Le maximum de quatre étoiles fait référence aux meilleures capacités, tandis que le minimum d'une étoile représente les plus faibles performances. On observe alors que les SVM, les arbres de décision et les réseaux bayésiens sont globalement les plus performants.

AD RN KNN SVM

Performances de classification	**	***	**	****
Vitesse d'apprentissage selon le nombre d'éléments	***	*	****	*
Tolérance aux données incomplètes	****	****	*	****
Tolérance aux données erronées	***	*	**	****
Tolérance aux données redondantes	**	**	**	***
Tolérance au bruit** **	*	**		

Table II.1 – Comparaison d'algorithmes de classification supervisée [57] (**** représente les meilleures performances ; AD : Arbre de Décision, RN : Réseaux de Neurones peu profond ,KNN : Plus proches voisins, SVM : Séparateurs à Vastes Marges)

II.4. Conclusion

Le ML est un sujet vaste en évolution permanente. Les algorithmes qu'il met en œuvre ont des sources d'inspiration variées qui vont de la théorie des probabilités aux intuitions géométriques en passant par des approches heuristiques.

II.5. Deep Learning

Comme précisé au début de ce chapitre, le Deep Learning, ou apprentissage profond, (également connu sous le nom de deep structured learning, hierarchical learning ou deep Machine Learning) est une branche du Machine Learning. Ce sous-ensemble du deep Machine Learning repose sur le principe des réseaux de neurones artificiels (Artificial Neural Network - ANN), utilisés cependant à une échelle beaucoup plus importante, reposant sur le fait qu'une augmentation du nombre de couches et de neurones d'un ANN augmente les performances de classification. Théorisé à l'origine en 1986 [65,66], le concept du Deep Learning n'est devenu populaire que récemment [67], exigeant un très grand nombre de données annotées et une grande puissance de calculs.

II.5.1 Artificial Neural Network – ANN

Un réseau de neurones artificiels est un réseau composé de neurones artificiels interconnectés, où chaque neurone représente une unité de traitement d'information, comme le montre la figure II.21. Ces neurones traitent et propagent l'information aux autres d'une manière similaire aux neurones du cerveau humain. Chacun reçoit un ensemble d'informations auxquelles il applique une certaine opération avant de transmettre le résultat aux neurones suivant. Les neurones sont rassemblés en couches formant des niveaux hiérarchiques. Les neurones recevant l'information "brute" ou

originale font partie de la couche appelée couche d'entrée et, de la même façon, les neurones fournissant les données de sortie du réseau font partie de la couche appelée couche de sortie. Tous les autres neurones, aux travers desquels l'information se propage de l'entrée à la sortie, font partie des couches cachées.

Sur chaque lien entre les neurones, l'information transmise est pondérée. Ainsi, un neurone reçoit un certain nombre de données de la couche précédente et utilise une fonction de transfert, sommant ces données pondérées, avant de traiter l'information. Le résultat est ensuite transmis à une fonction non linéaire, appelée "fonction d'activation", laquelle utilise une constante de seuil (aussi appelée "biais") afin d'éviter de transmettre une information nulle aux neurones de la couche suivante. Lors de l'apprentissage, les poids sont ajustés en fonction des performances du réseau.

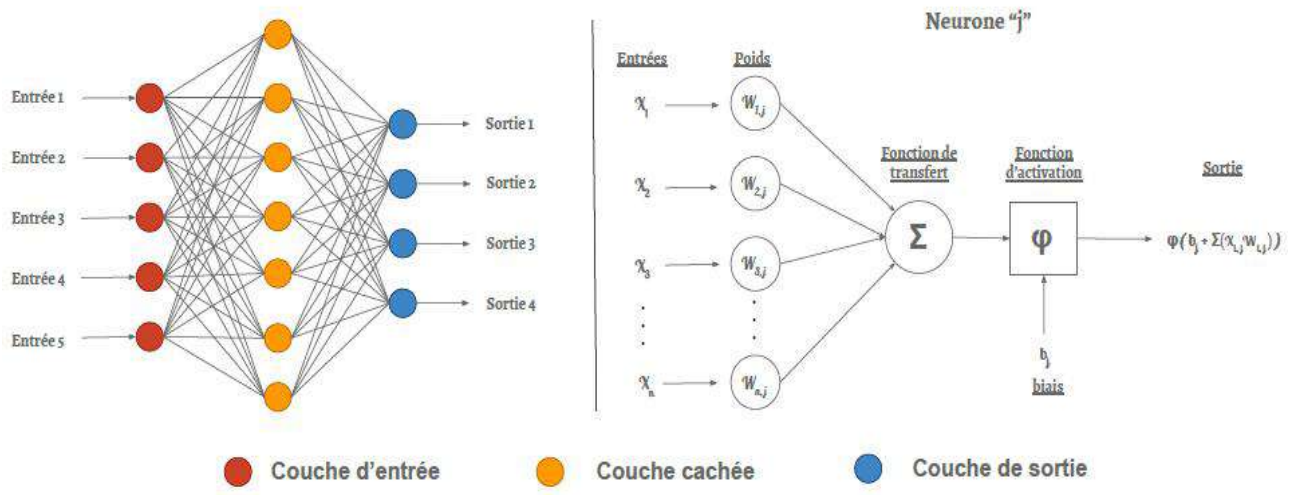


Figure II.21 : Principe d'un réseau de neurones artificiels.

On peut catégoriser les réseaux de neurones les plus simples selon deux familles : réseaux "Feedforward" et réseaux "Feedback". Pour le type "Feedforward", l'entrée est directement propagée vers la sortie (dans une seule direction). Pour le type "Feedforward" (ou récurrents), la sortie de certains neurones est retournée vers l'entrée de neurones précédents ("boucle") permettant ainsi d'augmenter les performances.

II.5.2 Réseaux de neurones profonds (Deep Neural Network - DNN)

Les réseaux de neurones, comme présenté précédemment, comportent un nombre limité de neurones et disposent par conséquent de capacités limitées face aux problèmes de la littérature de plus en plus complexes. Afin de répondre à ces problèmes, les structures du Machine Learning doivent évoluer afin de proposer des modèles d'apprentissage plus complexes, capables de traiter

des informations fournies par des milliers voir des millions de données. Une solution consiste à concevoir des réseaux de neurones disposant d'un plus grand nombre de couches cachées (ainsi qu'un plus grand nombre de neurones par couche), comme le montre la figure II.22, que l'on appelle alors des réseaux de neurones profond [58]. Cette augmentation des capacités et de la complexité du réseau augmente par conséquent la quantité de poids des neurones et la quantité d'informations propagées, conduisant nécessairement à une augmentation du nombre de calculs et donc des besoins en ressources informatiques.

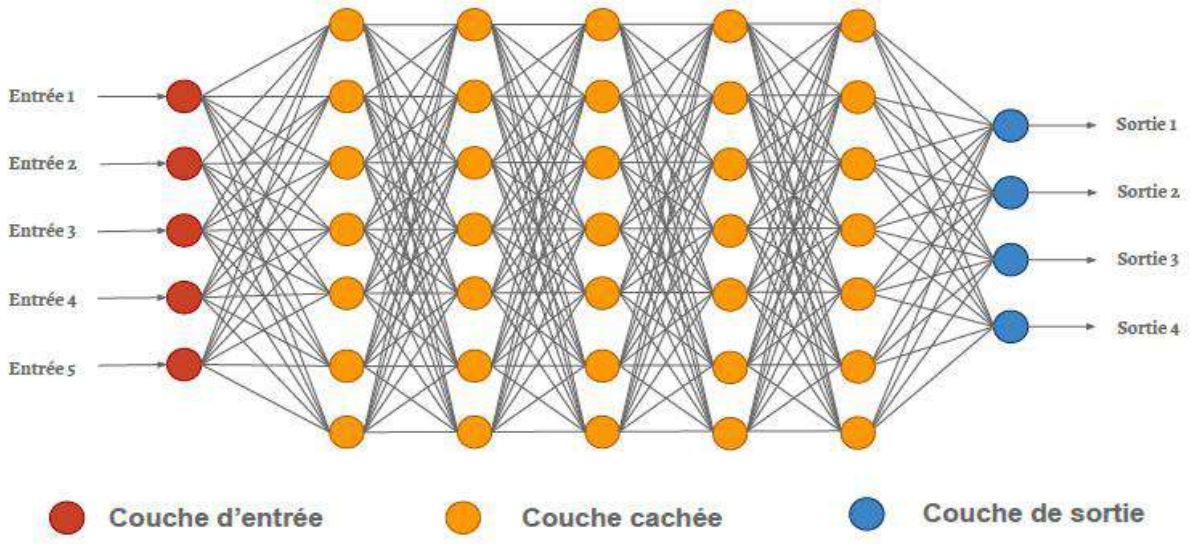


Figure II.22 : Deep Learning Neural Network.

Il existe de nombreux types de réseaux de neurones et réseaux profonds suivant leur architecture. Les plus populaires dans la conception des réseaux de neurones profond étant :

— *Perceptrons multicouches :*

Ce sont les réseaux les plus basiques parmi les réseaux "feedforward". Ils utilisent généralement des fonctions d'activations non linéaires. Ils peuvent être utilisés comme une partie d'un plus grand réseau de neurones profond.

— *Réseaux de neurones récurrents (Recurrent Neural Networks - RNN) :*

Les RNN sont aussi catégorisés comme des réseaux "feedforward", cependant ils possèdent des boucles de connexions récurrentes, propageant le résultat d'un neurone au précédent ou à lui-même. Ainsi, le réseau conserve en "mémoire" tout ou partie des informations précédentes et est donc capable de les utiliser afin d'affiner les résultats suivants. Ces réseaux sont surtout utilisés à des fins prédictives comme reconnaissance de texte ou en traduction.

— Réseaux de neurones convolutifs (Convoluted Neural Network - CNN) :

Les CNN sont similaires aux réseaux de neurones profonds ordinaires mais leur architecture est spécifique au traitement d'images. En effet, l'information de l'image est traitée en différents points par une convolution avec plusieurs filtres pour chaque couche. La récente popularité du Deep Learning est due à ces réseaux, permettant entre autres les dernières avancées en matière de conduite autonome, d'analyse complexe d'images, etc.

II.5.3 Réseaux neuronaux de convolutions (CNN)

Convolutional Neural Network (CNN) (réseaux de neurones convolutifs) sont un type de réseau de neurones spécialisés pour le traitement de données ayant une topologie semblable à une grille. Les exemples comprennent des données de type série temporelle, qui peuvent être considérées comme une grille 1D en prenant des échantillons à des intervalles de temps réguliers et des données de type image, qui peuvent être considérées comme une grille 2D de pixels. Les réseaux convolutifs ont connu un succès considérable dans les applications pratiques. Le nom « réseau de neurones convolutif » indique que le réseau emploie une opération mathématique appelée convolution. La convolution est une opération linéaire spéciale. Les réseaux convolutifs sont simplement des réseaux de neurones qui utilisent la convolution à la place de la multiplication matricielle dans au moins une de leurs couches. Ils ont de larges applications dans la reconnaissance de l'image et de la vidéo, les systèmes de recommandations et le traitement du langage naturel [59].

Dans le ML, un réseau convolutif est un type de réseau de neurones feed-forward, il a été inspiré par des processus biologiques [60]. Le modèle de connectivité entre les neurones d'un CNN s'inspire de l'organisation du cortex visuel animal. Les neurones corticaux individuels répondent aux stimuli dans une région restreinte de l'espace connu sous le nom de champ réceptif. Les champs réceptifs des différents neurones se chevauchent partiellement de sorte qu'ils couvrent le champ visuel. La réponse d'un neurone individuel aux stimuli dans son champ réceptif peut être approchée mathématiquement par une opération de convolution.

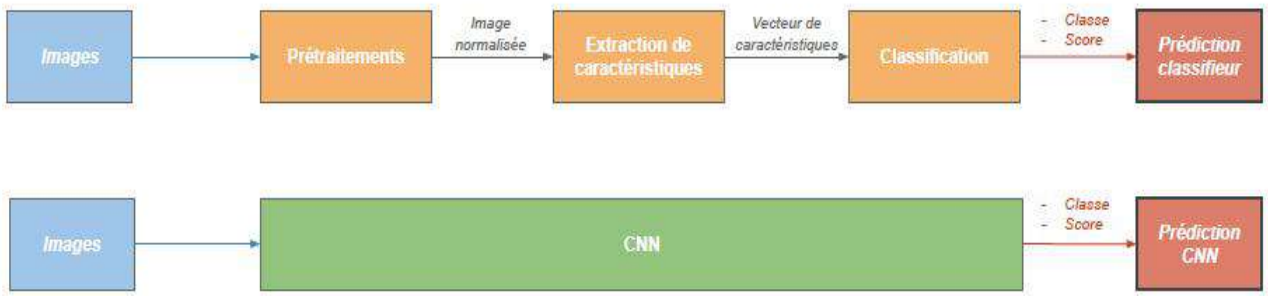


Figure II.23 : Différence entre CNN et Machine Learning Traditionnel pour l'analyse d'images.

Dédiés à l'analyse d'images, les réseaux neuronaux de convolutions (CNN) embarquent toute la chaîne de traitements décrite par la figure II.7 . Contrairement au Machine Learning classique et comme présenté par la figure II.23, ces CNN peuvent être considérés comme une boite noire utilisant un ensemble d'images d'apprentissage de même dimensions pour ajuster les nombreux paramètres du réseau et ainsi se spécialiser à une certaine tâche. Nous nous proposons ici de présenter rapidement les éléments constituant un CNN. Afin de faciliter la compréhension du fonctionnement de ces CNN .

II.5.3.1 L'opération de convolution

Dans sa forme la plus générale, la convolution est une opération sur deux fonctions d' argument réel. Pour comprendre la motivation derrière la convolution, nous commençons par des exemples de deux fonctions qu'on pourrait utiliser.

Supposons que nous suivons l'emplacement d'un vaisseau spatial avec un capteur laser. Notre capteur laser fournit une sortie x(t) qui est la position du vaisseau spatial au moment t. x et t sont des réelles, c'est-à-dire que nous pouvons obtenir une lecture différente du capteur laser à tout moment.

Supposons maintenant que notre capteur laser soit quelque peu bruité. Pour obtenir une estimation moins bruitée de la position du vaisseau spatial, nous aimerions combiner plusieurs mesures. Bien sûr, les mesures plus récentes sont plus pertinentes, donc nous voulons que ces mesures soient une moyenne pondérée et donner plus de poids aux mesures récentes. Nous pouvons le faire avec une fonction de pondération w(a), où a est l'âge de la mesure.

Si nous appliquons une telle opération de moyenne pondérée à chaque instant, nous obtenons une nouvelle fonction qui fournit une estimation lisse de la position du vaisseau spatial :

$$s(t) = \int x(a)w(t - a)da$$

Cette opération s'appelle convolution. L'opération de convolution est généralement désignée par un astérisque :

$$s(t) = (x * w)t$$

Dans la terminologie du réseau convolutif, le premier argument (dans cet exemple, la fonction x) de la convolution est souvent appelé l'entrée (input)et le second argument (dans cet exemple, la fonction w) comme noyau (kernel). La sortie est parfois appelée *feature map.*

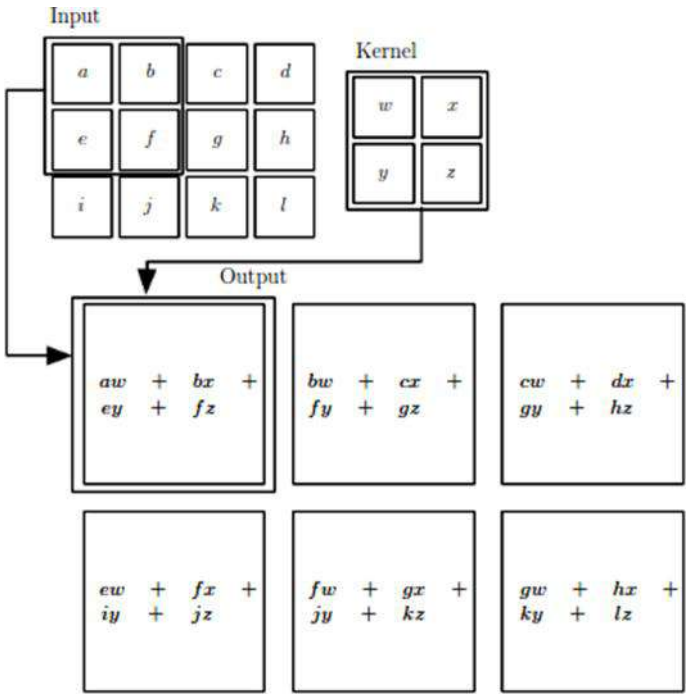


Figure II.24 : Exemple d'une convolution 2D. [61]

II.5.3.2 Couche convolutif

La convolution s'appuie sur trois idées importantes qui peuvent aider à améliorer un système de ML *: sparse interactions, parameter sharing* et *equivariant representations.*

- Sparse interactions : Les réseaux de neurones traditionnelles utilisent la multiplication matricielle par une matrice de paramètres avec des paramètres séparés décrivant l'interaction entre chaque unité d'entrée et chaque unité de sortie. Cela signifie que chaque unité de sortie interagit avec chaque unité d'entrée ce qui n'est pas le cas des réseaux de neurones convolutifs. Ceci est accompli en rendant le noyau plus petit que l'entrée. Par exemple, lors du traitement d'une image, l'image d'entrée peut avoir des milliers ou des millions de pixels, mais nous pouvons détecter de petites caractéristiques significatives telles que les bords avec des noyaux qui n'occupent que des dizaines ou des centaines de pixels.

Cela signifie que nous pouvons stocker moins de paramètres, ce qui réduit les besoins en matière de mémoire du modèle et améliore son efficacité. Cela signifie également que le calcul des sortie nécessite moins d'opérations. Ces améliorations en matière d'efficacité sont généralement assez importantes. Figure II.25.

- parameter sharing : (le partage des paramètres) Se réfère à l'utilisation du même paramètre pour plus d'une fonction dans un modèle. Dans un réseau de neurones convolutif, chaque élément du noyau est utilisé à chaque position de l'entrée (sauf peut-être quelques-uns des pixels des bords, selon le choix de conception concernant la frontière). Le parameter sharing utilisé par l'opération de convolution signifie que, plutôt que d'apprendre un ensemble de paramètres distincts pour chaque emplacement, nous n'apprenons qu'un ensemble ce qui réduit encore davantage les exigences de stockage du modèle. Figure II.26.
- equivariant representations : Dans le cas de la convolution, la particularité du parameter sharing fait que la couche possède une propriété appelée equivariant representations. Pour dire qu'une fonction est équivariante signifie que si l'entrée change, la sortie change de la même manière. Spécifiquement, une fonction f(x) est équivariante à la fonction g si $f(g(x)) = g(f(x))$. Dans le cas du traitement d'image, la convolution crée une carte 2-D de certaines caractéristiques apparaissant dans l'entrée. Si nous déplaçons l'objet dans l'entrée, sa représentation va se déplacer de la même façon dans la sortie. Par exemple, dans le cas d'une image, Il est utile de détecter les arêtes dans la première couche d'un réseau convolutif. Les mêmes arêtes apparaissent plus ou moins partout dans l'image, donc il est pratique de partager des paramètres sur l'image entière.

II.5.3.3 Couche de pooling

Une architecture atypique d'un réseau convolutif se compose de trois types de couches différentes. D'abord une couche convolutive pour générer un ensemble d' activations linéaires ensuite, on les fait passer à travers une couche d'activation non linéaire telle que Rectified Linear Unit (ReLu), enfin on utilise la fonction pooling.

— Il permet de réduire progressivement la taille des représentations afin de réduire la quantité de paramètres et de calcul dans le réseau et, par conséquent, de contrôler également le sur-apprentissage.
— Il permet l'invariance aux petites translations .
— Utile lorsque on préfère savoir si une caractéristique est présente plutôt que la région de sa présence.
— Plusieurs type de pooling diffèrent (MAX pooling (très populaire), AVG pooling, ...).

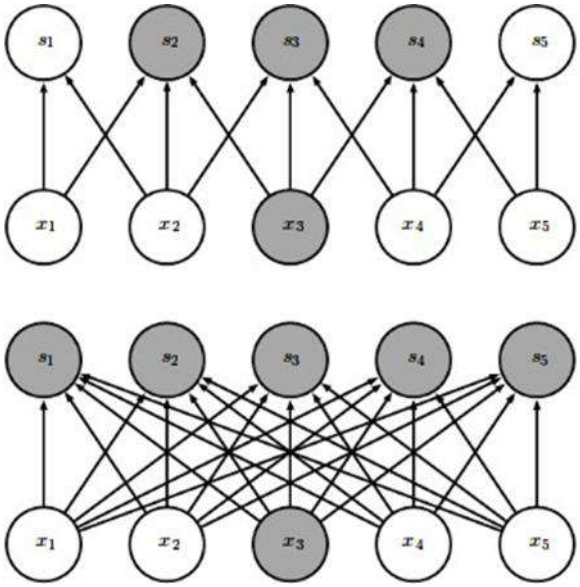


Figure II.25 : Sparse interactions : on met en évidence l' unité d'entrée, x3, et les unités de sortie qui sont affectées par cette unité. (en haut) Lorsque s'est formé par convolution avec un noyau de largeur 3, seules trois sorties sont affectées par x. (en bas) Lorsque s'est formé par une multiplication matricielle, toutes les sorties sont atteintes par x3. [62]

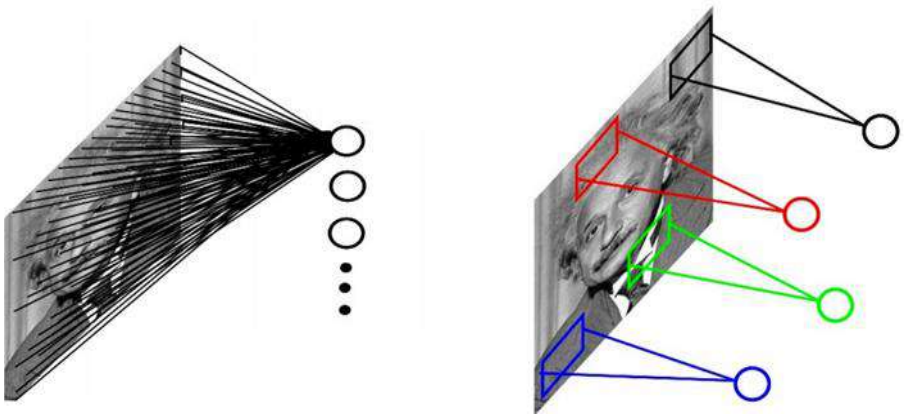


Figure II.26: Exemple image 1000x1000. (à gauche) Le non paramètre sharing nous oblige à concevoir une couche cachée de 106 neurones, chaque neurone est connecté à 106 pixels, en tout ça fait 1012 paramètres. (à droite) Avec un noyau 10x10 et une couche cachée de 106 neurones, le nombre de paramètres est de 108 .

II.5.3.4 Perceptron

Après avoir extrait les caractéristiques des entrées, on attache à la fin du réseau un perceptron ou bien un MLP (Multi layer Perceptron) . Le perceptron prend comme entrée les caractéristiques extraites et produit un vecteur de N dimensions ou N est le nombre de classe ou chaque élément est

la probabilité d'appartenance à une classe. chaque probabilité est calculée à l'aide de la fonction *softmax* dans le cas où les classes sont exclusivement mutuelles.

$$softmax(z)i = \frac{e^{zi}}{\sum j\ ezj}$$

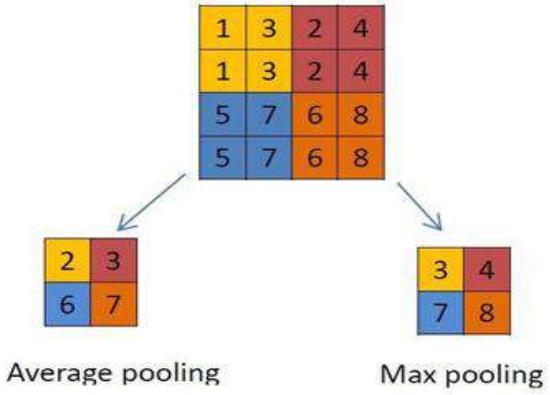


Figure II.27:(à gauche) Average pooling : chaque case correspond à la moyenne du carré d'entrée de la même couleur, ex de la case jaune : (1+3+1+3)=4 = 2. (à droite) Max pooling : chaque case correspond à la valeur maximum du carré d'entrée de la même couleur, ex de la case bleu : max(5; 7; 5; 7) = 7

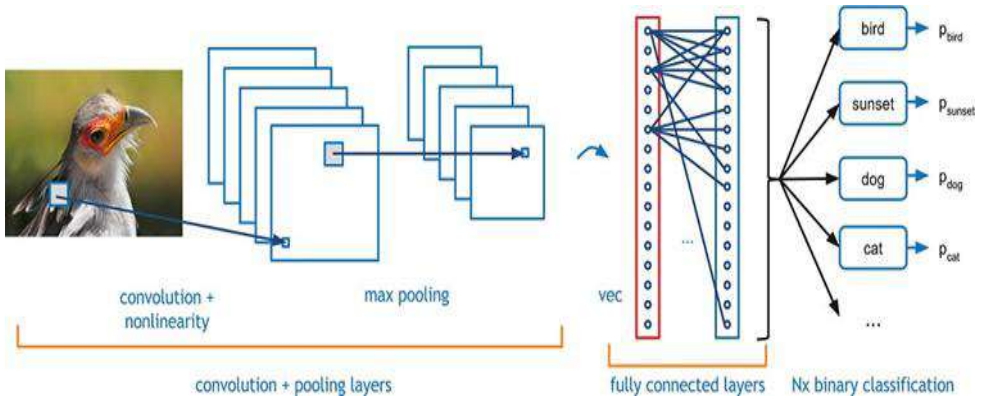


Figure II.28 :Un réseau de neurones convolutif qui reçois une image 2D comme entrée et qui est composé d'une couche convolutive, une fonction d'activation non linéaire, une couche MAX pooling et enfin un perceptron multi couche.

II.5.3.5 Quelques réseaux convolutifs célèbres

- LeNet [63] : Les premières applications réussies des réseaux convolutifs ont été développées par Yann LeCun dans les années 1990. Parmi ceux-ci, le plus connu est l'architecture LeNet utilisée pour lire les codes postaux, les chiffres, etc.

- AlexNet[64] : Le premier travail qui a popularisé les réseaux convolutifs dans la vision par ordinateur était AlexNet, développé par Alex Krizhevsky, Ilya Sutskever et Geoff Hinton. AlexNet a été soumis au défi ImageNet ILSVRC [76] en 2012 et a nettement surpassé ses concurrents. Le réseau avait une architecture très similaire à LeNet, mais était plus profond, plus grand et comportait des couches convolutives empilées les unes sur les autres (auparavant, il était commun de ne disposer que d'une seule couche convolutifs toujours immédiatement suivie d'une couche de pooling).
- ZFnet[65] : Le vainqueur de ILSVRC challenge 2013 était un réseau convolutif de Matthew Zeiler et Rob Fergus. Il est devenu ZFNet (abréviation de Zeiler et Fergus Net). C'était une amélioration de AlexNet en ajustant les hyper-paramètres de l'architecture, en particulier en élargissant la taille des couches convolutifs et en réduisant la taille du noyau sur la première couche.
- GoogLeNet[66] : Le vainqueur de ILSVRC challenge 2014 était un réseau convolutif de Szegedy et al. De Google. Sa principale contribution a été le développement d'un module inception qui a considérablement réduit le nombre de paramètres dans le réseau (4M, par rapport à AlexNet avec 60M). En outre, ce module utilise le global AVG pooling au lieu du PMC à la fin du réseaux, ce qui élimine une grande quantité de paramètres. Il existe également plusieurs versions de GoogLeNet, parmi elles, Inception-v4.
- ResNet[67] : Residual network développé par Kaiming He et al. A été le vainqueur de ILSVRC 2015. Il présente des sauts de connexion et une forte utilisation de la batch normalisation. Il utilise aussi le global AVG pooling au lieu du PMC à la fin.

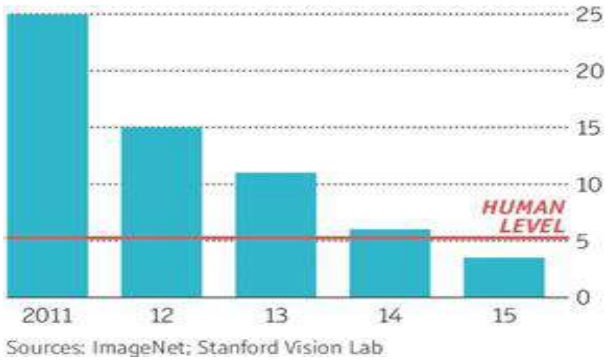
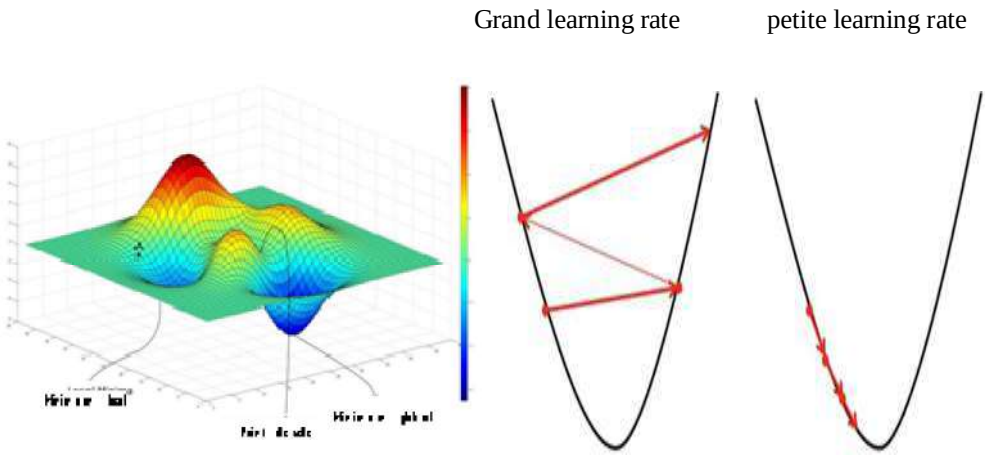


Figure II.29 :Le taux d'erreur dans ImageNet Visual recognition Challenge. Le Deep Learning dépasse la performance humaine.

II.6. Optimisation pour l'apprentissage en Deep Learning

Lors de l'apprentissage, les paramètres du réseau sont initialisés aléatoirement et les images d'apprentissage sont propagées dans le réseau. Avec les étiquettes de données associées, les

performancesde classification sont évaluées par le calcul d'une fonction de perte (ou fonction de coût).Cette fonction mesure l'erreur de classification et donc l'écart entre la probabilité de la prédictionet la vérité de terrain. Les poids et les biais du réseau sont ensuite ajustés par un algorithme d'optimisationitératif, appelé descente de gradient, de telle sorte qu'une nouvelle évaluation produiseun déplacement de l'erreur vers un minimum local ou global du gradient de cette fonction de coût(figure II.30 a). Le pas de ce déplacement à chaque itération est appelé "learning rate". Ce pas estun hyperparamètre pouvant être réglé afin d'influencer la convergence de l'apprentissage. Tel quele décrit la figure II.30 b, un grand pas permet de couvrir une plus grande région du gradient maisdes pentes vers un minimum, pouvant être très brèves, peuvent être manquées. Un petit learningrate est plus précis et permet de capter plus facilement les variations de pente, mais induit un coûtimportant en temps de calcul, fortement lié à la quantité de paramètres à ajuster. Un compromisest donc indispensable.



(a) Illustration 3D de la descente de gradient (b) Effet du *Learning Rate*

Figure II.30 : Optimisation par descente de gradient

II.6.1. Les variantes de la descente de gradient

La descente de gradient est la méthode la plus célèbre pour optimiser un réseau de neurones où on doit minimiser une fonction objective $J(\theta)$ caractérisée par les paramètresθ en les mettant à jour dans la direction opposée de celle de gradient de la fonction objective. Le taux d'apprentissage η détermine l'importance de l'étape qu'on va prendre pour atteindre le minimum local.

Il existe trois variantes de cette méthode. En fonction de la quantité de données, nous ferons un compromis entre la précision de la mise à jour des paramètres et le temps d'exécution de cette mise à jour.

II.6.1.1. Batch gradient descent

C'est la descente de gradient classique, on calcule le gradient de la fonction coût aux paramètres θ pour tout l'ensemble d'apprentissage .

$$\theta = \theta - \eta.\nabla_{\theta}.J(\theta)$$

Comme on doit calculer le gradient pour tout l'ensemble de données pour exécuter juste une seule mise à jour, cette méthode peut être très lente et irréalisable si les données ne peuvent pas être stockées en mémoire.

Avec cette méthode on est sur de converger vers l'optimum global si la fonction cout est convexe et vers un optimum local si elle est non convexe.

II.6.1.2. Descente de gradient stochastique

SGD (la descente de gradient stochastique) met à jour les paramètres pour chaque exemple de l'ensemble de données $x^{(i)}$ et label $y^{(i)}$.

$$\theta = \theta - \eta.\nabla\theta.J(\theta; x^{(i)}; y^{(i)})$$

Cette méthode est plus rapide mais les mises à jours des paramètres trop fréquentes causent à la fonction objectif des oscillations, ces oscillations d'une part permettent d'atterrir dans des minimums locaux potentiellement meilleurs mais d'autre part rendent la convergence plus difficile. Cependant il a été montré qu'en baissant le taux d'apprentissage η, SGD montre la même convergence que la batch gradient descent.

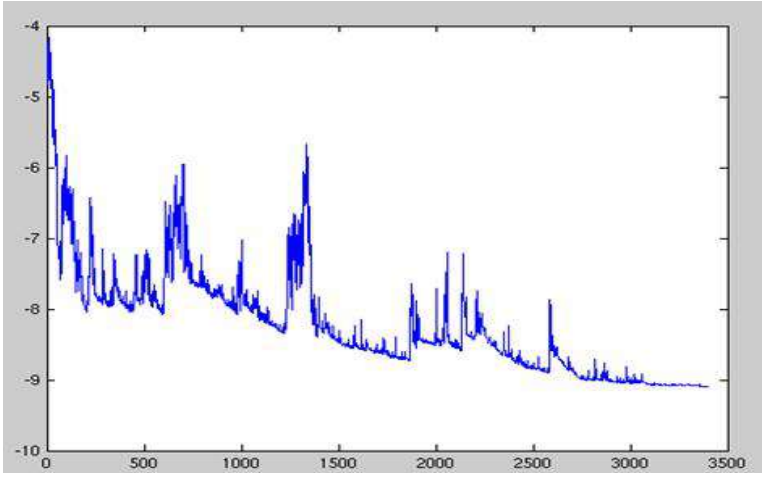


Figure II.31 : Les oscillations causées par SGD.

II.6.1.3. Mini-batch gradient descent

Cette méthode prends le meilleur des deux méthodes et met à jour les paramètres pour chaque mini groupes de n exemples :

$$\theta = \theta - \eta.\nabla\theta.J(\theta; x^{(i:i+n)}; y^{(i:i+n)})$$

Cette méthode réduit la variance des mises à jour des paramètres ce qui conduit à une convergence plus stable. Généralement les mini groupes contiennent de 50 à 256 exemples. C'est une méthode de choix pour entraîner un réseau de neurones. La descente de gradient classique n'offre pas toujours une bonne convergence et pose quelques challenges qui ont besoin d'être résolus :

- Le choix du taux d'apprentissage est difficile, si il est trop petit cela peut emmener à une convergence trop lente, si il est trop grand il peut causer des oscillations à la fonction coût voir même ne pas converger du tout.
- Le même taux d'apprentissage est appliqué à tous les paramètres. Si les caractéristiques des observations dont on dispose ont une fréquence différentes, on peut vouloir appliquer une plus grande mise à jour aux paramètres pour les caractéristiques qui reviennent plus rarement.
- Un autre challenge de la minimisation des fonctions coût non convexe très courantes dans les réseaux de neurones c'est éviter d'être piéger dans des optimums locaux. Dauphin et al. [68] soutiennent que la difficulté ne survient pas des minimum locaux mais des points col (saddle point).

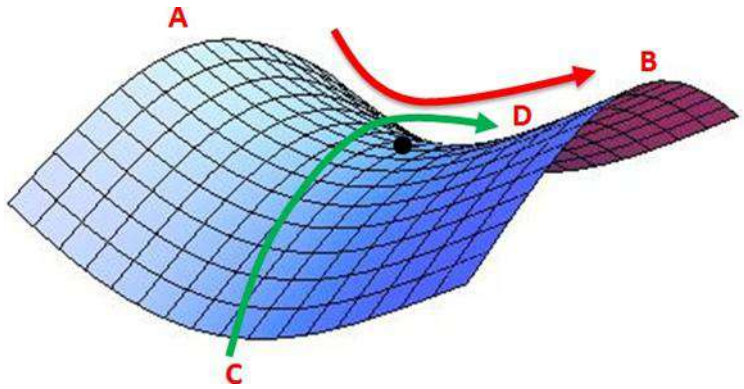


Figure II.32 : Saddle point : Le point noir joue le rôle du minimum par rapport à l'axe A-B et du maximum par rapport à l'axe C-D.

II.7. Les fonctions d'activation

la fonction d'activation est une fonction mathématique appliquée à un signal en sortie d'un neurone artificiel. Le terme de "fonction d'activation" vient de l'équivalent biologique "potentiel d'activation", seuil de stimulation qui, une fois atteint entraîne une réponse du neurone. La fonction d'activation est souvent une fonction non-linéaire. Leur but est de permettre aux réseaux de neurones d'apprendre des fonctions plus complexes qu'une simple régression linéaire car le fait de multiplier les poids d'une couche cachée est juste une transformation linéaire.

II.7.1 Caractéristique des fonctions d'activation

- Non-linéarité [68] : Quand une fonction est non linéaire, un réseau neuronal à deux couches peut être considéré comme un approximateur de fonction universel.
- Partout différentiable : Cette propriété permet de créer des optimisations basées sur les gradients.
- Étendue [69] : Quand la plage d'activation est finie, les méthodes d'apprentissage basées sur les gradients sont plus stables (impact sur un nombre de poids limités). Quand la plage est infinie, l'apprentissage est généralement plus efficace (impact sur davantage de poids).
- Monotone : Lorsque la fonction est monotone, la surface d'erreur associée avec un modèle monocouche est certifié convexe.
- dérivée monotone : Les fonctions à dérivée monotone ont été montrées comme ayant une meilleure capacité à généraliser dans certains cas. Ces fonctions permettent d'appliquer des principes comme le rasoir d'Ockham.
- Identité en 0 : Ces fonctions permettent de faire un apprentissage rapide en initialisant les poids de manière aléatoire. Si la fonction ne converge pas vers l'identité en 0, alors un soin spécial doit être apporté au lors de l'initialisation des poids.

II.7.2 Quelque fonctions d'activation

II.7.2.1 Rectified Linear Units et leur généralisations

Les ReLu utilisent la fonction d'activation :

$$g(z) = max\{0, z\}$$

ReLu sont faciles à optimiser car ils sont tellement similaires aux unités linéaires. La seule différence entre une unité linéaire et ReLu est que ReLu délivre zéro sur tous R- ceci rend les dérivées via ReLu importantes chaque fois que l'unité est active.

Des recherche montrent que ReLu sont un modèle encore meilleur de neurones biologiques [84] et produisent une performance égale ou meilleure que la tangente hyperbolique et un de leur avantage majeur est la réduction de la probabilité du vanishing gradient. Cela se produit lorsque z > 0. Dans ce cas, le gradient a une valeur constante puisque la dérivée est égale à 1 partout. En revanche, le gradient des sigmoïdes devient de plus en plus petit car la valeur absolue de z augmente. Le gradient constant de ReLu entraîne un apprentissage plus rapide.

Lors de l'initialisation des paramètres il peut être judicieux d'initialiser les valeurs des biais à une petite valeur positive, telle que 0,1. Il est très probable que ReLu seront initialement actives pour la plupart des entrées de l'ensemble d'apprentissage et permettent aux dérivées de passer à travers.

Il existe plusieurs généralisations de ReLu. La plupart de ces généralisations fonctionnent de manière comparable et de temps à autre, se comportent mieux. Un inconvénient de ReLu est qu'ils ne peuvent pas apprendre par la descente de gradient sur des exemples pour lesquels leur activation est nulle "dying ReLu" problem. Une variété de générations de ReLu garantit qu'elles reçoivent le gradient partout.

Les généralisations de ReLu sont basées sur l'utilisation d'une pente non nulle $\alpha_i$lorsque $z_i<0$: $h_i = g(z, \alpha) = max(z_i, 0) + \alpha_i min(0, z_i)$. *Absolute value rectiftcation*fixe $\alpha_i = 1$ pour obtenir $g(z) = z$. *Leaky ReLu (LReLu)*fix α_ià une petite valeur comme 0.01 alors *que parametric ReLu (PReLu)* [ ]traite $\alpha_i$comme un paramètre apprenable.

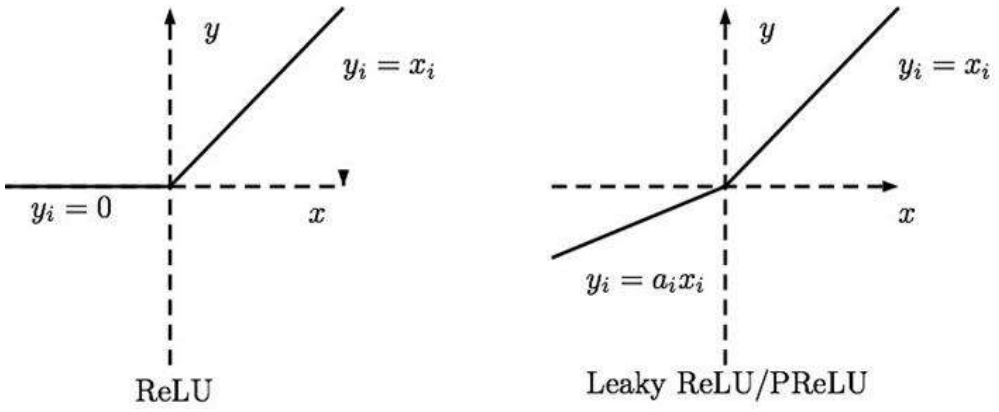


Figure II.33 : (à gauche) La fonction d'activation ReLu. Cette fonction d'activation est la fonction d'activation par défaut recommandée pour l'utilisation des réseaux de neurones feedforward. L'application de cette fonction à la sortie d'une transformation linéaire donne une transformation non linéaire. Ils préservent plusieurs des propriétés qui rendent les modèles linéaires faciles à optimiser avec la descente de gradient. (à droite) Leaky ReLu et Parametric ReLu qui corrige l' inconvénient de ReLu et qui sont différentiables à 0.

II.7.2.2 Sigmoïde et tangente hyperbolique

Avant l'introduction de ReLu, la plupart des réseaux neuronaux utilisent la Sigmoïde ou tanh :

la Sigmoïde :

$$g(z) = \frac{1}{1 + e^{-\lambda z}}$$

tanh :

$$g(z) = \frac{1 - e^{-2z}}{1 + e^{-2z}}$$

Ce qu'on reproche aux unités sigmoïdales c'est qu' elles saturent dans la plupart de leur domaine :

elles saturent à une valeur élevée lorsque z est très positif et à une valeur faible lorsque z est très négatif et ne sont que très sensibles à leur entrée lorsque z est proche de 0. La saturation généralisée des unités sigmoïdales peut rendre l'apprentissage par la méthode de la descente de gradient très difficile. Pour cette raison, leur utilisation dans les couches cachées est maintenant déconseillée.

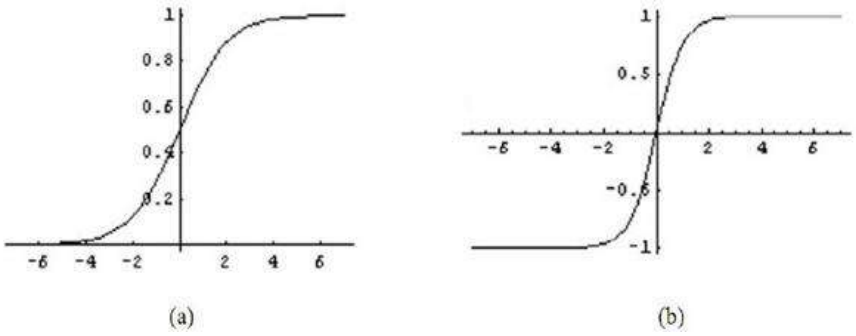


Figure II.34 :(a) Sigmoïde. (b) tangente hyperbolique

II.8. Transfert Learning

En général, peu de personnes entraînent un CNN entièrement (apprentissage à partir d'un réseau vierge, c'est à dire un réseau dont les poids n'ont pas été définis) du fait de la difficulté d'obtenir le grand nombre d'images annotées nécessaire à ce type de réseaux. Il est plus commun de trouver une exploitation des CNN à partir du Transfer Learning. Le Transfer Learning est une technique d'apprentissage, surtout utilisée en Deep Learning, permettant d'entraîner un réseau à une certaine tâche à partir d'un modèle déjà entraîné sur une tâche similaire (e.g réseau original, m classes : classification d'animaux ; nouveau réseau n classes : classification de races de chats ; avec n pouvant être différent de m). Les connaissances de ce réseau pré-entraîné sont transférées au nouveau problème afin d'assister l'apprentissage à partir de nouvelles images, et certaines étapes du

réseau sont alors affinées ("Fine-Tuning"). Cette forme d'apprentissage présente l'avantage de nécessiter moins de données (milliers au lieu de millions) et permet un apprentissage beaucoup plus rapide, passant de centaines ou milliers d'heures de calculs à quelques heures voire quelques minutes. Comme l'apprentissage d'un CNN peut prendre plusieurs semaines, même avec la puissance de calcul de plusieurs GPU (Graphics Processing Unit), il est courant que les grandes structures de recherche rendent disponibles certains de leurs réseaux sur le net. Ces réseaux sont majoritairement entraînés sur les images du challenge ImageNet avec 1000 classes en reconnaissance d'objets. Un CNN pré-entraîné peut être exploité de diffiérentes façons par le Transfer Learning en fonction de la taille du nouveau jeu de données d'entrée et de la similarité des nouvelles images avec celles utilisées lors de l'apprentissage original. Les trois principales formes d'apprentissage par transfert de connaissances sont les suivantes :

— *Fine-tuning total* : Dans ce cas de figure, la dernière couche entièrement connectée (couche de classification) est remplacée par un classifieur adapté au nouveau problème. Toutes les couches du réseau sont ensuite ré-entraînées sur les nouvelles images. Cette stratégie est utilisée lorsque la nouvelle collection d'images est grande. Comme les poids sont initialisés avec les valeurs du réseau pré-entrainé puis affinés, ce nouvel apprentissage est plus rapide.

— *Fine-tuning partiel* : De la même manière que précédemment, la dernière couche entièrement connectée est remplacée par un nouveau classifieur mais seules certaines couches de l'extraction de caractéristiques sont ré-entraînées, les poids des premières couches étant maintenues à leurs valeurs. Les premières couches d'un réseau étant en effet capables d'extraire des caractéristiques abstraites, ou génériques (contours, couleurs, etc.) de l'ancien jeu d'images d'apprentissage, et les dernières couches produisant progressivement des caractéristiques plus spécifiques aux classes des données originales, ce sont les poids de ces dernières qui sont ajustés. Cette stratégie est utilisée lorsque la nouvelle collection d'images est plus petite et différente.

— *Extraction de caractéristiques* : Le dernier cas de figure consiste à se servir des couches extrayant les caractéristiques du réseau pré-entraîné afin de représenter les nouvelles images d'un nouveau problème. Ainsi, la dernière couche entièrement connectée est réinitialisée ou retirée pour être remplacée par un nouveau classifieur, et les paramètres des autres couches sont fixés. Le nouveau classifieur va alors être entraîné à partir des vecteurs de caractéristiques extraites par les autres couches du réseau pré-entraîné. Cette stratégie est utilisée lorsque la nouvelle collection d'images est petite et présente des similitudes aux images originales. [71]

II.9. Evaluation des méthodes

II.9.1. Métrique d'évaluation

Les sections précédentes ont montré qu'il existe un grand nombre de solutions aux problèmes de classification d'images dans le domaine de la biométrie. Les méthodes présentées peuvent être utilisées pour des problèmes de classification à deux ou plusieurs classes, et le comportement des classifieurs dépend du nombre d'échantillons par classe ainsi que de leur composition. Par conséquent, la sélection de la méthode la plus appropriée est dépendante des contraintes liées à l'application ciblée. Une solution consiste à effectuer une première sélection de méthodes, à les tester puis à réaliser une série d'évaluations. Les techniques d'analyse des données utilisent habituellement comme point de départ une représentation sous forme de tableau, que l'on appelle matrice de confusion ou tableau de contingence, mettant en relation le nombre de prédictions "Xi; j"

(Xclasse;prediction) des échantillons de la classe "i" attribués à une classe "j" (parmi C classes). Le nombre d'échantillons formant la classe "i" est noté "Ki", et le nombre total de prédictions attribuées à cette classe est nommé "Mi". La somme des "Ki" ainsi que la somme des "Mi" correspondant au nombre total d'échantillons ($\sum$).

		Prédiction			Total / classes
		classes 1	classes i	classes c	
Classe réelle	classes 1	X1;1	X1;i	X1;C	K1
	classes i	Xi;1	Xi;i	Xi;C	Ki
	classes c	Xc;1	Xc;i	Xc;C	Kc
Total prédictions		M1	Mi	Mc	$\sum$

Table II.2 – Matrice de confusion des prédictions d'un classifieur à C classes .

En considérant le problème précédent, pour chacune des classes "i", comme étant binaire (Classe"i" : positif ; Toutes les autres classes "j , i" : négatif), ou directement dans le cas d'un problème àdeux classes, les prédictions sont résumées sous la forme de quatre informations principales :

— Vrais Positifs - VP ("True Positive" - TP) :

Échantillons de la classe positive ("i") correctement classés (Xi;i)

— Faux Négatifs - FN ("False Negative" - FN) :
Échantillons de la classe positive ("i") incorrectement classés (Xi; j; 8 j , i)

— Vrais Négatifs - VN ("True Negative" - TN) :
Échantillons des classes négatives ("j") correctement classés (Xj;t; 8(j; t) 2 [1;C] , i)

— Faux Positifs - FP ("False Positive" - FP) :
Échantillons des classes négatives ("j") incorrectement classés (Xj;i; 8 j , i)

Pour un problème à N classes, considéré comme binaire pour le test de chacune des classes, ily a autant de matrices de confusion qu'il y a de classes, chacune représentant les performancesde la classification vis à vis de la classe "i". La matrice de confusion pour un problème à deuxclasses établit ainsi le lien entre le nombre d'échantillons total (P) de la classe positive, le nombred'échantillons total (N) de la classe négative, les quatre informations précédentes et par conséquentle nombre total d'échantillons classés positivement (Ppos) et négativement (Pneg). Cette matricede confusion, tableau 3.4, donne une indication sur les résultats de classification. Cependant, du faitque la quantité d'échantillons par classe peut varier entre les différentes classes et des différentesproblématiques, son exploitation n'est pas toujours facile.

		Prédiction		Total / classes
		Classe positive	Classe négative	
Classe réelle	Classe positive	VP	FN	P
	Classe négative	FP	VN	N
Total prédictions		Ppos	Pneg	∑

Table II.3 – Matrice de confusion des prédictions d'un classifieur à deux classes.

Des critères de performances dérivent de cette matrice de confusion et permettent d'évaluer rapidement les performances de classification. Ces nouvelles métriques représentent des indications normalisées de ces performances sous forme d'un pourcentage. Il est ainsi possible d'analyser et de comparer les performances de di_érents classifieurs. Ces mesures sont les suivantes :

— Sensibilité ou Rappel (Sensitivity ou Recall) :
Proportion des solutions pertinentes correctement identifiées (taux de vrais positifs). Mesure la capacité du système à donner toutes les solutions pertinentes.

- Cas général (problème à C classes) :

Sensibilité $_i = \frac{Xi.i}{Ki}$

Sensibilité $_{Système} = \frac{\sum_{i=1}^{C} Sensibilite\ i}{C}$

- <u>Pour un problème à 2 classes :</u>

Sensibilité $= \frac{VP}{VP+FN} = \frac{VP}{P}$

— Robustesse, exactitude ou justesse (Accuracy) :

Proportion des solutions correctes (positives et négatives) parmi toutes les prédictions. Mesure la capacité du système complet à donner toutes les solutions correctes.

$$Justesse = \frac{VP + VN}{VP + VN + FP + FN} = \frac{P}{\sum} \rightarrow \frac{Échantillons\ correctement\ attribués}{Totalité\ des\ échantillons}$$

— F_β-mesure (Fβ-measure) or F-Score :

Combine la précision et le rappel de par leur moyenne harmonique. Fourni une mesure favorisant soit le rappel soit la précision dans le cas d'une classification à partir de donnée non équilibrées.

$$F\beta = \frac{(1 + \beta^2)(Precision.Rappel)}{\beta^2.Precision.Rappel}$$

L'évaluation d'un système complet décrit sa faculté à discriminer une classe par rapport à une autre. Dans le cas d'un système à plus de deux classes, cette évaluation représente une "moyenne" de ces métriques pour chaque classe. Cette moyenne peut être obtenue de deux manières différentes. On peut en effet distinguer la "micro-moyenne" de la "macro-moyenne" (calcul utilisé dans les descriptions précédentes). Chaque donnée (nombre de bonnes réponses et de mauvaises pour tout le système) est prise en compte dans le calcul de la micro-moyenne. Celle-ci calcule d'abord la somme des valeurs les plus fines des termes du numérateur et du dénominateur (présents dans la matrice de confusion) puis établit le critère d'évaluation à partir de ces sommes. Dans le cas du calcul de la macro-moyenne, chaque catégorie est prise en compte (nombre de bonnes réponses et de mauvaises par classe). Comme présentée, elle calcule les critères d'évaluations de chaque classe puis en effectue la moyenne. Cependant, si la quantité d'informations évaluées n'est pas équilibrée entre les classes, ces métriques peuvent conduire à une mauvaise interprétation du comportement du

classifieur. Il est donc nécessaire de comprendre ce que représentent ces métriques, et la manière dont il faut les interpréter, afin de quantifier les performances d'un système.[72]

II.9. Conclusion

Dans cette partie, nous avons vu qu'est-ce que le Deep Learning et comment il se différencie des algorithmes de ML traditionnelles. Nous avons introduit quelques méthodes utilisée par la communauté du Deep Learning et nous avons expliqué le principe de chacun. Nous avons également parlé d'une majeure modèle à savoir le réseaux convolutif ainsi que l'évaluation des méthodes des systèmes sécurité biométriques , ces derniers font encore l'objet de recherches intensives.

III.1 Introduction

Reconnaître automatiquement les visages et identifier les individus est un problème de Grand intérêt et il existe de nombreuses approches pour résoudre ce problème. Récemment, les réseaux de neurones profonds et en particulier les réseaux de neurones convolutifs CNN ont montré des performances de classification impressionnantes dans la reconnaissance faciale. Comme nous le savons déjà, les modèles d'apprentissage profond de grande taille sont gourmands en données. des Une idée fausse courante est que les données de formation et de test doivent provenir de la même source ou être de la même distribution. Grâce à l'apprentissage par « transfert learning », nous sommes en mesure de résoudre une tâche particulière en utilisant tout ou une partie d'un modèle déjà pré-apprenti sur une tâche différente.

III.2 Logiciels et outils

III.2.1 Anaconda (Python distribution)

Anaconda est une distribution libre et open source des langages de programmation Python et R appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique (traitement de données à grande échelle, analyse prédictive, calcul scientifique), qui vise à simplifier la gestion des paquets et de déploiement. Les versions de paquetages sont gérées par le système de gestion de paquets « conda ». La distribution Anaconda est utilisée par plus de 6 millions d'utilisateurs et comprend plus de 250 paquets populaires en science des données adaptés pour Windows, Linux et MacOS.

III.2.2Présentation des outils de programmation :

En utilisant le langage généraliste python, la bibliothèque de vision par ordinateur OpenCV, Tensorflow de google et le Framework Keras pour la mise en oeuvre de nos réseaux de neurones, ainsi que d'autre modules et bibliothèques (numpy, marplotlib, etc...).

2.2 Python :

Le langage de programmation scientifique multi-paradigme Python a été créé en 1989 par Guido van Rossum, aux Pays-Bas. Le nom Python vient d'un hommage à la série télévisée Monty Python's Flying Circus dont G. van Rossum est fan. La première version publique de ce langage a été publiée en 1991. La dernière version de Python est la version 3.7 a été publiée en juin 2018. La version 2 de Python est désormais obsolète et cessera d'être maintenue après le 1er janvier 2020. Ce langage de programmation présente de nombreuses caractéristiques intéressantes :

• Il est gratuit et multiplateforme. : Windows, Mac OS X, Linux, Android, iOS, depuis les mini-ordinateurs Raspberry Pi jusqu'aux supercalculateurs.

• C'est un langage de haut niveau. Il demande relativement peu de connaissance sur le fonctionnement d'un ordinateur pour être utilisé.

• C'est un langage interprété. Un script Python n'a pas besoin d'être compilé pour être exécuté, contrairement à des langages comme le C ou le C++.

• C'est un langage dynamique ,extensible Il favorise la programmation structurée fonctionnelle et orientée objet. C'est-à-dire qu'il est possible de concevoir en Python des entités qui miment celles du monde réel (une cellule, une protéine, un atome, etc.) avec un certain nombre de règles de fonctionnement et d'interactions.

• Il est relativement simple à prendre en main .

• La syntaxe de Python est très simple et, combinée à des types de données évolués (listes, dictionnaires,...), conduit à des programmes à la fois très compacts et très lisibles.

• Il gère ses ressources (mémoire, descripteurs de fichiers...) sans intervention du Programmeur.

• Enfin, il est très utilisé en bio-informatique et l'intelligence artificielle et plus généralement en analyse de données. Toutes ces caractéristiques font que Python est outils idéal pour implémenter notre application .

2.3 OpenCV :

OpenCV (Open Source Computer Vision Library) est une bibliothèque qui aide à la vision par ordinateur Depuis son lancement officiel en 1999 par l'équipe d'Intel, un certain nombre de programmeurs ont contribué aux derniers développements de la bibliothèque. Le dernier changement majeur intervenu en 2009 (OpenCV 2) qui inclut les principales modifications apportées à l'interface C. la version de la bibliothèque est disponible sur le site officiel d'OpenCV. De nos jours, la bibliothèque à 2500 algorithmes optimisés. OpenCv est spécialisé en :

- Manipulation des données d'images et vidéo, les matrices et les vecteurs .
- Différentes structures de données dynamiques (listes, files d'attente, ensembles, arbres, graphiques).
- Analyse du mouvement (flux optique, segmentation du mouvement, suivi), Reconnaissance d'objets.
- Interface graphique de base (image / vidéo à afficher, gestion du clavier et de lasouris, barres de défilement…)

2.4 Tensorflow :

TensorFlow est un framework de programmation pour le calcul numérique de Google initié et développé par l'équipe Google Brain spécialisé dans l'intelligence artificielle, et rendu Open Source en Novembre 2015, et devenir très rapidement l'un des frameworks les plus utilisés pour le Deep Learning, Les caractérestiques principale de tensorflow sont :

- Multiplateformes (Windows , Linux, Mac OS, et même Android et iOS)
- APIs en Python, C++, Java et Go (l'API Python est plus complète cependant, c'est sur celle-ci que nous allons travailler)
- Temps de compilation très courts dû au backend en C/C++
- Supporte les calculs sur CPU, GPU et même le calcul distribué sur cluster
- Une documentation extrêmement bien fournie avec de nombreux exemples et tutoriels

2.5 Keras :

C'est une API de haut niveau permettant de créer et de former des modèles d'apprentissage en profondeur, capable de s'exécuter sur **Tensorflow,** il est utilisé pour le prototypage rapide, la recherche avancée et la production. Les modèles Keras sont fabriqués en reliant des blocs de construction configurables ensemble, donc Keras nous permet de créer de nouvelles couches, des fonctions et développer des modèles à la pointe de la technologie avec peu de restrictions en quelques lignes de code.

2.6 NumPy :

C'est le paquet fondamental du calcul scientifique avec Python. Il contient entre autres des choses:

- Un puissant objet tableau à N dimensions.
- Fonctions sophistiquées (diffusion).
- Des outils pour l'intégration du code C / C ++ et Fortran.
- Capacités utiles d'algèbre linéaire, de transformée de Fourier et de nombres aléatoires Outre ses utilisations scientifiques évidentes, NumPy peut également être utilisé comme conteneur efficace multidimensionnel de données génériques.

III.3 Description de la base de données utilisé

Comme décrit dans l'état de l'art, pour des meilleurs résultats, il est préférable d'entraînés le réseau avec beaucoup d'échantillons d'image. Cela augmenterait la précision et améliorerait la performance des modèles.

Malheureusement, les bases de données de grande taille n'existent pas publiquement, mais nous avons accès à une base des données publique : (*The FEI face database*)[88] c'est une base de données de visages brésiliens qui contient un ensemble d'images de visage prises entre juin 2005 et mars 2006 au laboratoire d'intelligence artificielle de FEI à São Bernardo do Campo, São Paulo, Brésil. Il y a 14 images pour chacun individus, Toutes les images sont colorées et prises sur un fond blanc homogène dans une position frontale verticale avec une rotation de profil jusqu'à environ 180 degrés. L'échelle peut varier d'environ 10% et la taille d'origine de chaque image est de 640x480 pixels. Nous avons choisi de travailler avec 08 classes ou individus à partir de cette base des données avec l'addition d'une classe de moi-même et l'autre de mon collègue pour tester le modèle ça fait un totale de 10classes.



Figure III.1 : Quelques exemples de variations d'images de la base de données de visages FEI.

Il est important de faire l'apprentissage sur une base des données de la région d'intérêt des visages pour que l'application de reconnaissance faciale d'être précise et efficace, pour cela nous avons fait un programme d'extraction de visages.



Figure III.2 : Quelques exemples de variations d'images sur la région d'intérêt.

III.3.1. Séparation des bases de données

Afin de développer une application de reconnaissance de visage, il est nécessaire de disposer de deux bases de données : une base pour effectuer l'apprentissage et une autre pour tester les techniques et déterminer leurs performances, mais Il n'y a pas de règle pour déterminerce partage de manière quantitatif. Il résulte souvent d'un compromis tenant compte du nombrede données dont on dispose et du temps pour effectuer l'apprentissage.Dans les séries de test que nous avons effectué la base a été scindée de la façon suivante :

- *Images apprentissages :*12 images servent pour la phase d'apprentissage.
- *Images Tests***:** Les 2 autre images de chaque individu nous ont servies pour la réalisation des tests.

Le but est d'évaluer le taux de reconnaissance de différents algorithmes présenté, en suivant un protocole de test basé sur la mesure de taux de reconnaissance :

$$\text{Taux de reconnaissance} = \frac{\text{nombre d'image de teste reconnues}}{\text{nombre totale des images de teste}}$$

III.4Implémentation et réalisation :

Notre système de reconnaissance faciale est constitué de deux modèles :

- le premier pour la détection de visage
- et le deuxième pour faire la reconnaissance faciale.

4.1 module de détection :

Pour détecter des visages nous avons utilisé une méthode populaire, proposée par Paul Viola et Michael Jones dans leur article "Détection rapide d'objets utilisant une cascade de fonctions simples" en 2001. [89] Dans notre système nous avons utilisé haarcascade_frontalface.xml de la bibliothèque OpenCV qui fournit la méthode Haar Cascade.

4.2 module de reconnaissance :

Pour reconnaitre les visage détecté par le module précédent, nous utilisons un réseaux de neurones convolutionel. Le premier élément à prendre en compte est le choix d'architecture du CNN. Les architectures de réseaux neuronaux sont des combinaisons de couches qui se transmettent les sorties les unes après les autres. Comme indiqué précédemment nous avons choisi d'utiliser la technique de transfert learning cela nous permettre d'importer divers architectures des modèles disponible pour la classification des images dans notre cas c'est le modèle VGG16.

- L’architecture VGG16 :

VGG16 est une architecture des réseauxneuronaux convolutionel (CNN) qui a été utilisée pour remporter la compétition ILSVR (Imagenet) en 2014. Elle est considérée comme l'une des excellentes architectures de modèles de vision à ce jour. La chose unique à propos de VGG16 est qu'au lieu d'avoir un grand nombre d'hyper-paramètres, ils se sont concentrés sur des couches de convolution de filtre 3x3 avec une foulée 1 et ont toujours utilisé le même rembourrage et une couche maxpool de filtre 2x2 de foulée 2. Il suit cet arrangement de la convolution et les couches maximales de la piscine de manière cohérente dans toute l'architecture. Au final, il a deuxcouches entièrement connectées suivis d'une activation softmax pour la sortie. Le 16 dans VGG16 fait référence à 16 couches qui ont des poids. Ce réseau est un assez grand réseau et il a environ 138 millions de paramètres.

Les figures 3.3 et le tableau 3.1 Présentent l’architectures utilisées :

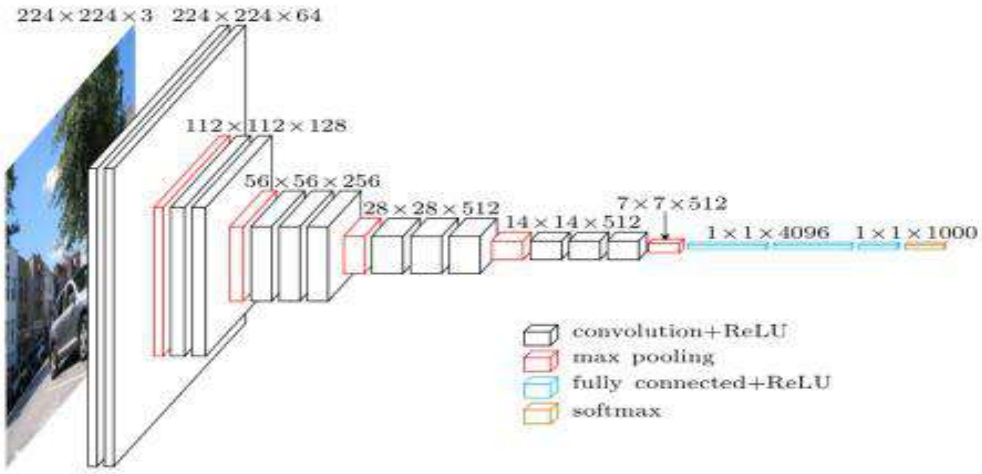


Figure III.3 : l’architecture standard de VGG16

L’architecture de VGG-16 est illustré dans letableau3.1, elle utilise 21 couches convolutives et 3 couches entièrement connectées. Les couches convolutives dans VGG-16 sont toutes la même dimension de 3 × 3, et les couches de pooling sont toutes 2 × 2. La valeur par défaut la taille de l'image d'entrée du VGG-16 est de 224 × 224. Après chaque couche de pooling, la taille de la carte d'entités (features map) est réduite à la moitié. La dernière carte des caractéristiques avant le les couches connectées sont 7 × 7 avec 512 canaux et c'est développé en un vecteur avec 25088 (7 × 7 × 512) canaux.A la fin, on a ajouté une couche de Pooling Global moyen, et on a appliqué la fonction Softmax pour calculer le taux des 10 Classes d’identités , elle renvoie donc un vecteur de taille 10, qui contient les probabilités d'appartenance à chacune des classes.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 1024)	525312
dense_1 (Dense)	(None, 10)	10250

Total params: 21,140,042
Trainable params: 6,425,354
Non-trainable params: 14,714,688

Table III.1 : Couches , dimensions des filtres et les nombres des paramètres pour notre architecture vgg16

La méthode de Transfer Learning appliquant sur ce modèle consiste à améliorer les poids du réseau pré-entrainé. Pour cela, la méthode de backpropagation sera maintenue. En fonction de notre problématique, on peut aussi décider de garder et donc figer une partie des poids des premières

couches du réseau pré-entraîné car les premières couches contiennent des variables très génériques (ex. détection des angles ou des couleurs) et donc pertinentes pour de nombreuses tâches différentes. Plus on avance dans les couches du réseau et plus ces dernières sont spécifiques aux détails des classes contenues dans la base des données.

- Les divers couches de convolution constituons notre réseau de convolution :

La couche d'entrée (input layer) :

Comme appris précédemment, les couches convolutifs sont les principaux éléments constitutifs des modèles. Chaque couche reçoit des informations d'entrée, effectue des calculs et génère enfin les informations transformées. La sortie d'une couche se dirigera vers la couche suivante en tant qu'entrée. Les images donc sont exposé sur la couche d'entrée d'une dimensions 244 et nombre de canaux de 3 c'est à dire les trois couleurs principaux RGB .

Les couches de convolution 2D :

Une couche de convolution 2D signifie que l'entrée de l'opération de convolution est tridimensionnelle. le «2D» en «convolution 2D» fait référence au mouvement du filtre, qui parcourt l'image en deux dimensions. Par exemple, une image couleur qui a une valeur pour chaque pixel sur trois calques: rouge, bleu et vert. Le filtre est ensuite exécuté sur l'image trois fois, une fois pour chaque couche. La même structure convolutive est utilisée successivement, dans un premier temps pour identifier des caractéristiques dans l'image originale, puis pour identifier des sous-caractéristiques dans des parties plus petites de l'image, après sous-échantillonnage ou «pooling» du résultat des convolutions précédentes. Finalement, ce processus vise à identifier les caractéristiques essentielles qui peuvent aider à classer l'image.

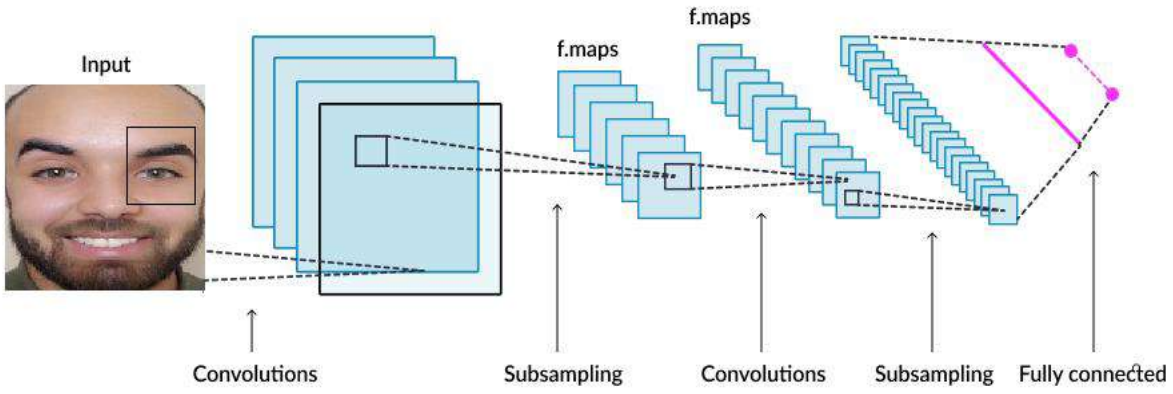


Figure III.4. : exemple des couches Conv2D suivi par les couches de pooling

La transition de la première couche $a^{[0]}$ à la deuxième couche $a^{[1]}$ est une multiplication des matrices contient des poids, l'addition d'un biais et l'application élément par élément de la fonction ReLU.

$$a^{[1]}=g(W^{[1]}a^{[0]}+b^{[1]})$$

Pour VGG-16, la seule différence que nous allons faire est de remplacer la multiplication matricielle générale par une convolution, et au lieu d'une matrice 2-D du poids, nous utiliserons un tenseur de filtre 3-D.

Nous avons discuté plus tôt que notre entrée est une image, avec des dimensions 224 par 224 par 3. Nous avons étiqueté ce tenseur « x »:

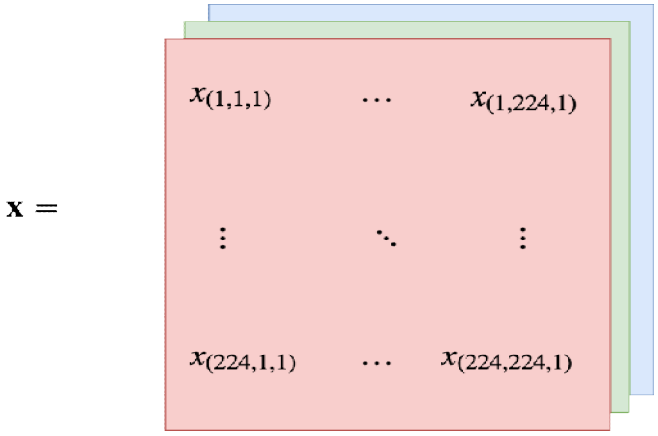


Figure III.5. : exemple d'une couche Conv2D

Nous pouvons considérer cette couche comme la couche $a^{[0]}$. On Appelons la première couche de convolution $a^{[1]}$. Cette couche a des dimensions 224 sur 224 sur 64:

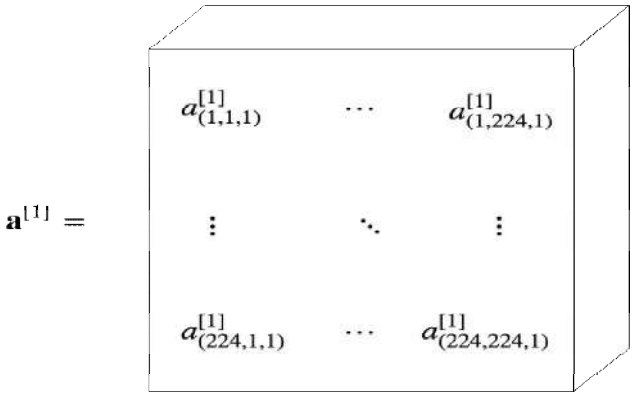


Figure III.6. : Diagramme de la première couche de convolution $a^{[1]}$ avec des dimensions (224,224,64)

Alors, pour passer de x à $a^{[1]}$ Nous commençons par appliquer la fonction de convolution à x 6 fois en utilisant 6 filtres différents puis ajoutons un biais $b^{[1]}$ pour obtenir notre produit intermédiaire, $z^{[1]}$. Nous appellerons la collection de filtres $W^{[1]}c$, de dimensions (16,16,3,6). Les trois premières dimensions représentent la hauteur, la largeur et le nombre des canaux, et la dernière dimension représente le filtre. Nous pouvons représenter chaque filtre par $W^{[1]}c(i)$, où i est le numéro de filtre dans $W^{[1]}c$.

On ajoute le même biais pour chaque filtre, donc $b^{[1]}$ a des dimensions (6,1).

Nous représentons cela visuellement:

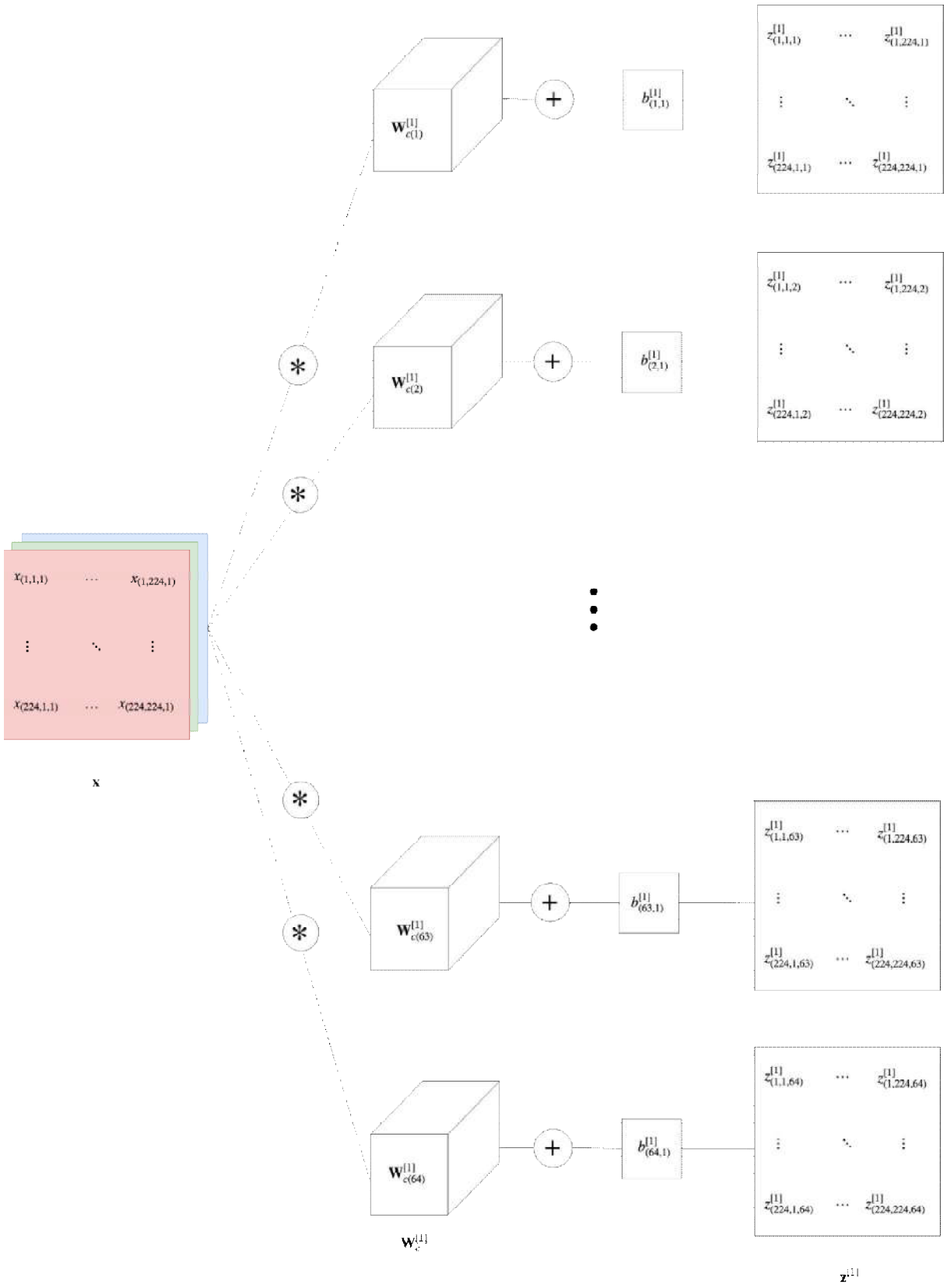


Figure III.7. : Schéma de transition de x à $z^{[1]}$. Une convolution est appliquée à x 6 fois en utilisant 6 filtres différents.

Alors, afin d'obtenir une valeur pour $z^{[1]}_{(i,j,k)}$ pour une ligne i, une colonne j et une canal k donnés dans $z^{[1}$, nous utilisons la fonction de convolution :

$$z^{[1]}_{(i,j,k)} = (x * W^{[1]}_c)(i,j,k) + b^{[1]}_{(k,1)}$$

Ce qui devient:

$$z^{[1]}_{(i,j,k)} = \sum_{l=1}^{3}\sum_{m=1}^{3}\sum_{n=1}^{3} x_{(i+l-1,j+m-1,n)} W^{[1]}_{c(l,m,n,k)} + b^{[1]}_{(k,1)}$$

Les couches convolutif de pooling (MaxPooling2D) :

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs feature maps, et applique à chacune d'entre elles l'opération de pooling. L'opération de pooling consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes.

Pour cela, on découpe l'image en cellules régulière, puis on garde au sein de chaque cellule la valeur maximale. En pratique, on utilise souvent des cellules carrées de petite taille pour ne pas perdre trop d'informations. Les choix les plus communs sont des cellules adjacentes de taille 2 × 2 pixels qui ne se chevauchent pas, ou des cellules de taille 3 × 3 pixels, distantes les unes des autres d'un pas de 2 pixels.

On obtient en sortie le même nombre de feature maps qu'en entrée, mais celles-ci sont bien plus petites. La couche de pooling permet de réduire le nombre de paramètres et de calculs dans le réseau. On améliore ainsi l'efficacité du réseau et on évite le sur-apprentissage.

Pourquoi le Max Pooling ?

Il y a plusieurs raisons pour lesquelles nous utilisons le Max Pooling. La première est qu'avec chaque ensemble de couches convolutives dans VGG16 nous augmentons la profondeur ou le nombre de canaux. Dans la couche avant d'aplatir notre tenseur pour l'utiliser dans une couche entièrement connectée, nous avons une profondeur de 512. La profondeur est importante car elle signale les informations structurées que le réseau a apprises sur l'entrée. Cependant, il ne serait pas efficace en mémoire de maintenir notre hauteur et largeur d'origine de 224 et de finir avec une profondeur de 512. Le max pooling nous permet de prendre une statistique récapitulative (dans ce

cas, le maximum) d'une fenêtre dans une couche convulitive et envoyez-le au niveau suivant. Essentiellement, nous prenons à peu près les activations les plus importantes de la couche précédente et les envoyons à la couche suivante, réduisant ainsi la hauteur et la largeur et diminuant les besoins en mémoire. A titre d'exemple, dans la couche de pooling finale, on se retrouve avec un tenseur de dimensions (7,7,512). Ceci est parfois appelé sous-échantillonnage.

La couche d'aplanissement (Flat Layer) :

Dans une couche d'aplanissement, nous prenons en entrée la couche de max pooling finale et l'aplatissons, pour obtenir en sortie une couche plate de dimensions (25,088,1). Le 25 088 provient de la multiplication de toutes les dimensions de la couche d'entrée ($7 * 7 * 512 = 25\,088$). La raison pour laquelle nous faisons cela est que les couches entièrement connectées prennent en entrée un vecteur ligne par opposition à un tenseur. Donc, dans cette couche, nous changeons la dimension pour nous préparer à utiliser les couches entièrement connectées.

La couche de correction ReLU :

ReLU (*Rectified Linear Units*) désigne la fonction réelle non-linéaire définie par ReLU(x)=max(0,x)

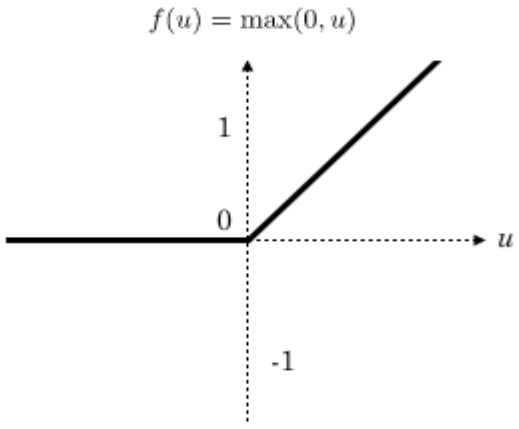


Figure III.8. : la fonction d'activation ReLU

Les couches entièrement connectée (fully connected layers) :

La couche *fully-connected* constitue toujours la dernière couche d'un réseau de neurones, convolutif ou non – elle n'est donc pas caractéristique d'un CNN.

Ce type de couche reçoit un vecteur en entrée et produit un nouveau vecteur en sortie. Pour cela, elle applique une combinaison linéaire puis éventuellement une fonction d'activation aux valeurs reçues en entrée.

La dernière couche *fully-connected* permet de classifier l'image en entrée du réseau : elle renvoie un vecteur de taille N, où N est le nombre de classes dans notre problème de classification d'images. Chaque élément du vecteur indique la probabilité pour l'image en entrée d'appartenir à une classe.

A partir de notre problème qui consiste à distinguer les individus, le vecteur final sera de taille 10 , Ainsi, le vecteur [0.9 0.1 … 0.01] signifie que l'image à 90% de chances de représenter l'un des individus .

Pour calculer les probabilités, la couche *fully-connected* multiplie donc chaque élément en entrée par un poids, fait la somme, puis applique une fonction d'activation (logistique si N=2, *softmax* si N>2).

Ce traitement revient à multiplier le vecteur en entrée par la matrice contenant les poids. Le fait que chaque valeur en entrée soit connectée avec toutes les valeurs en sortie explique le terme *fully-connected*.

La couche *fully-connected* détermine le lien entre la position des *features* dans l'image et une classe. En effet, le tableau en entrée étant le résultat de la couche précédente, il correspond à une carte d'activation pour une *feature* donnée : les valeurs élevées indiquent la localisation (plus ou moins précise selon le *pooling*) de cette *feature* dans l'image. Si la localisation d'une *feature* à un certain endroit de l'image est caractéristique d'une certaine classe, alors on accorde un poids important à la valeur correspondante dans le tableau.

La couche d'activation softmax :

Afin de faire la transition finale de la couche entièrement connectée à la couche softmax, nous utilisons la fonction softmax.

La transition de la couche entièrement connectée $a^{[15]}$ à la couche softmax $a^{[16]}$ commence comme n'importe quelle couche entièrement connectée. Nous appliquons une multiplication matricielle en utilisant $W^{[16]}$ et ajoutons un biais $b^{[16]}$ pour atteindre $z^{[16]}$.

$$z^{[16]}=W^{[16]}a^{[15]}+b^{[16]}$$

Nous désignerons la fonction softmax avec le variable (σ). Pour calculer la probabilité d'un i^{ème} élément dans la couche $a^{[16]}$ Nous faisons ce qui suit:

$$a_{i,1}^{[16]} = \sigma_{(i,1)}(z^{[16]})$$

$$a_{i,1}^{[16]} = \frac{e^{z}}{\sum_{j=1}^{10} e^{z}}$$

Donc, pour calculer le i^{ème} élément de $a^{[16]}$, nous prenons *e* à la puissance de $z^{[16]}$et on divisons par la somme de *e* à la puissance de tous les éléments de z. Et après avoir appliqué cette fonction d'activation, nous obtenons un vecteur $a^{[16]}$ dont la somme deces éléments est 1. Notons que $a^{[16]}$ est égal à ŷ. Auparavant, nous avons discuté de la façon dont nous voulions que notre sortie (ŷ) soit la probabilité que l'image d'exemple d'entraînement provienne de l'une des 10 classes.

$$\hat{y} = \begin{bmatrix} \hat{y}_1 \\ . \\ . \\ . \\ . \\ \hat{y}_{10} \end{bmatrix}$$

Tel que chaque $\hat{y}_i$ est égal à la probabilité que y soit égal à la classe i étant donné l'image d'entrée ou :

$$\hat{y}_i = P(y=i|x)$$

Après avoir appliqué la fonction d'activation softmax, nous avons maintenant un vecteur de probabilités qui fait le totale de 1.

- Organigramme pour la phase d'apprentissage :

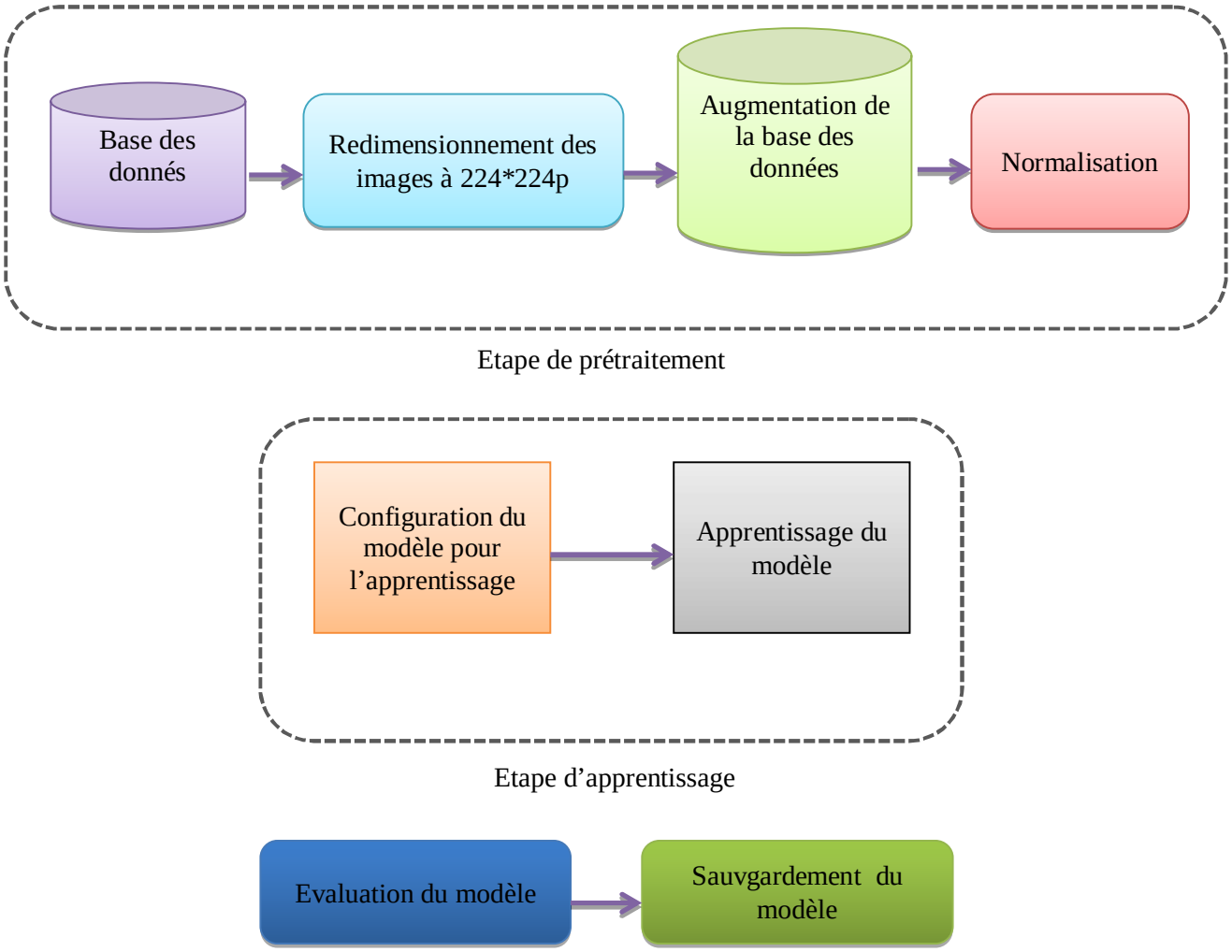


Figure III.9 : Principaux modules de la phased'apprentissage.

Etape de prétraitement :

- Redimensionnement et normalisation des images :

  Les images sont préalablement redimensionnées à la taille 224*224 ensuite, elles sont normalisées en soustrayant le nombre de pixel moyen de la base ImageNet, ceci parce que le modèles profond utilisés ont été entrainés sur cette base.

- Augmentation de la base des données :

  L'augmentation des données est une technique puissante pour augmenter la diversité de notre ensemble d'entraînement en appliquant des transformations aléatoires (mais réalistes) telles que la rotation d'image et le décalage .

  Voici donc la fonction utilisé pour faire l'augmentation ainsi que leur paramètres :

```
train_datagen = ImageDataGenerator(horizontal_flip=True, width_shift_range
= 0.2, height_shift_range = 0.2, shear_range = 0.2, zoom_range = 0.2,
fill_mode ='nearest', rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)
```

Etape d'apprentissage :

- Module de configuaration (model.compile) :

Le module de configuration du modèle définit la fonction de perte, l'optimiseur et les métriques.

La fonction de compilation pour notre modèle depuis la bibliothèque keras :

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

- Module d'apprentissage (model.fit) :

Après la création de CNN, l'étape suivante consiste à faire l'apprentissage. Pour l'apprentissage du réseau, nous utilisons la fonction «fit ()» de notre modèle avec les paramètres suivants:

```
model.fit_generator(training_set,
                            validation_data=test_set,
                            epochs=120,
                            steps_per_epoch=len(training_set),
                            validation_steps=len(test_set),
verbose=1)
```

```
Epoch 72/75
120/120 [==============================] - 2s 17ms/step - loss: 0.3710 - accuracy: 0.9000
Epoch 73/75
120/120 [==============================] - 2s 17ms/step - loss: 0.1787 - accuracy: 0.9417
Epoch 74/75
120/120 [==============================] - 2s 17ms/step - loss: 0.1150 - accuracy: 0.9583
Epoch 75/75
120/120 [==============================] - 2s 18ms/step - loss: 0.1799 - accuracy: 0.9417
```

Figure III.10 : le contenu de la fenêtre de sortie pendant la phase d'apprentissage.

Le nombre d'époques c'est le nombre de fois où le modèle parcourt les données. Plus nous avons d'époques, plus le modèle s'améliorera, jusqu'à un certain point. Ensuite, le modèle cessera de s'améliorer à chaque époque. La figure représente notre modèle qui entraîne les données, puis les valide. Une époque est le nombre de fois que le modèle s'entraîne sur l'ensemble de nos données. Un lot peut s'expliquer comme prenant de petites quantités, s'entraînant et prenant un peu plus. Le modèle soi meilleur aussi si la perte est faible que possible.

Organigramme pour la phase de prédiction :

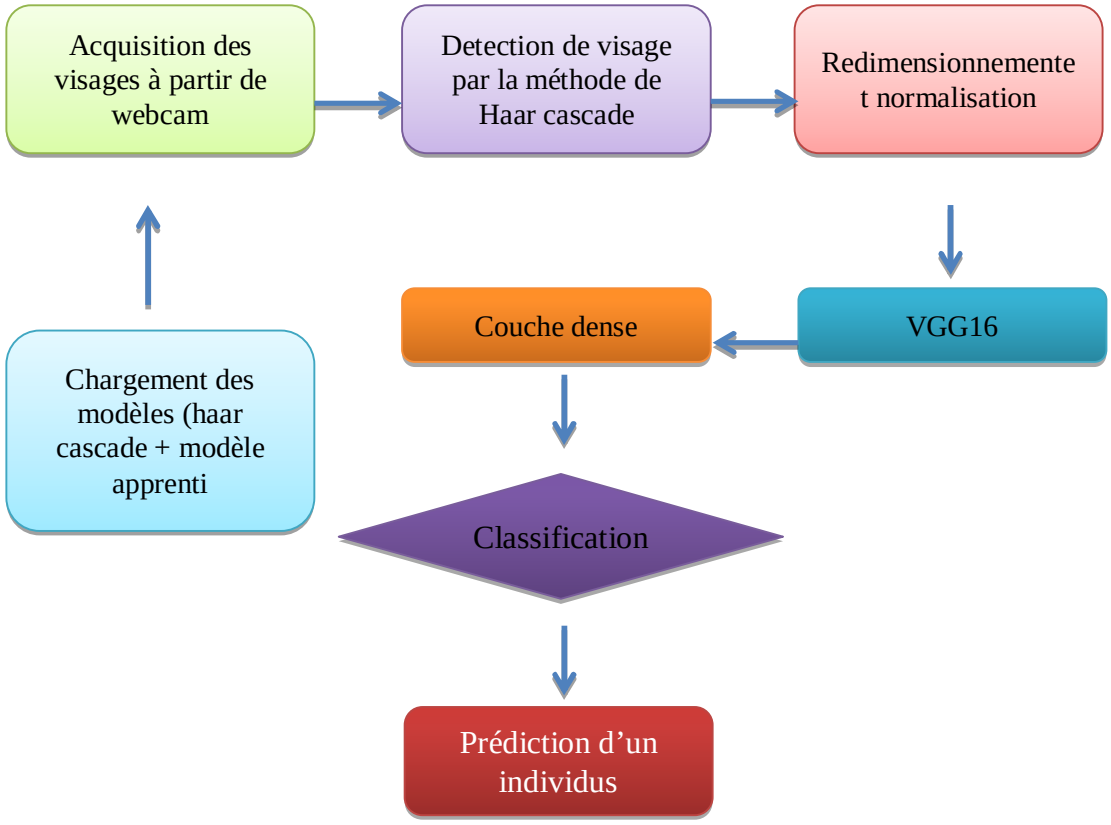


Figure III.11. : Principaux modules de la phasede prédiction.

– Chargement des modèles:

Avant tout, nous devons charger le modèle que nous avons sauvegardé sur le format .h5 après l'apprentissage, nous pouvons le faire en utilisant ces instructions :

```
#loading the model
      model = load_model('facefeatures_new_model.h5')
      # Loading the cascades
      face_cascade =cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

– Acquisition des visages à partir de webcam :

Pour ce faire, nous utilisons la fonction de capture vidéo de la bibliothèque OpenCV

```
video_capture = cv2.VideoCapture(0)
```

– Detection de visage par la méthode de Haar cascade :

Après le chargement des modèles nous avons créé une fonction qui faire l'extraction des visages depuis le vidéo en utilisons le modèle pré-apprenti « haar cascade » :

```
    def face_extractor(img):
        faces = face_cascade.detectMultiScale(img, 1.3, 5)
if faces is():
returnNone
for(x,y,w,h)in faces:
cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,255),2)
cropped_face = img[y:y+h, x:x+w]
return cropped_face
```

– Redimensionnement et normalisation :

Dans cette étape on doit redimensionner en 128x128 car nous avons formé le modèle avec cette taille d'image. En plus notre modèle keras utilise un tenseur 4D, (images x hauteur x largeur x canal) Donc, on doit changer la dimension 128x128x3 en 1x128x128x3

```
    whileTrue:
        _, frame = video_capture.read()

        face=face_extractor(frame)
    if type(face)is np.ndarray:
            face = cv2.resize(face,(224, 224))
            im = Image.fromarray(face, 'RGB')
            img_array = np.array(im)
            img_array = np.expand_dims(img_array, axis=0)
```

– Prédiction :

C'est là que nous allons appeler la fonction de prédiction , la prédiction s'affiche sur les visages détecter et dans le terminal de sortie sous forme d'un vecteur de 10 valeurs

```
pred = model.predict(img_array)
print(pred)
       name="None matching"

if(pred[0][0]>0.8):

        name='farouk'
    cv2.putText(frame,name,(50, 50), cv2.FONT_HERSHEY_COMPLEX, 1,(0,255,0),2)
```

```
        if(pred[0][1]>0.8):
            name='miloud'
    cv2.putText(frame,name,(50, 50), cv2.FONT_HERSHEY_COMPLEX, 1,(0,255,0),2)


else:
        cv2.putText(frame,"No face found",(50, 50), cv2.FONT_HERSHEY_COMPLEX,
     1,(0,255,0), 2)
        cv2.imshow('Video', frame)
```

III.5. Reconnaissance des visages et résultats

C'est là où nous testant notre modèle apprenti, est c'est l'interface de logiciel spyder ou on acquit les visages via une webcam puis le système présent les visages détectés et les reconnaître dans une fenêtre , la figure III.12 représente les résultats obtenue .

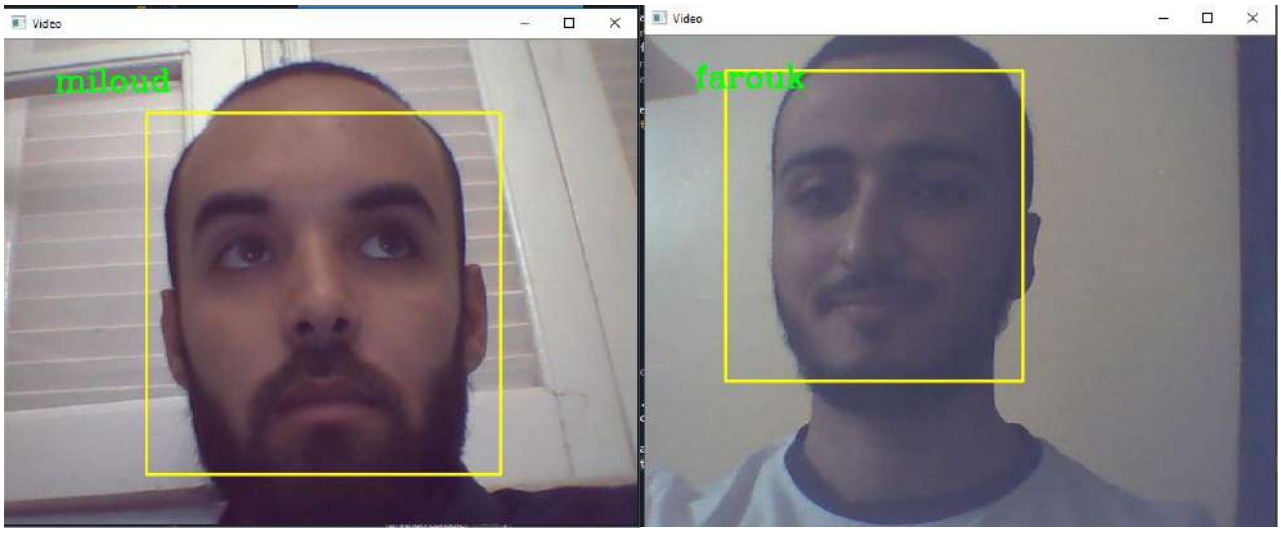


Figure III.12 : Aperçu sur les fenêtres de sortie (reconnaissance de deux visages étique par leurs noms

III.5.1. Résultats expérimentaux et discussions

Du point de vue des résultats, nous obtenons après l'apprentissage du modèle VGG16 une précision de test de 93%, Au moyen de deux graphes de figure 3.9 et figure 3.10, nous présentons le taux de précision et celui de perte obtenus :

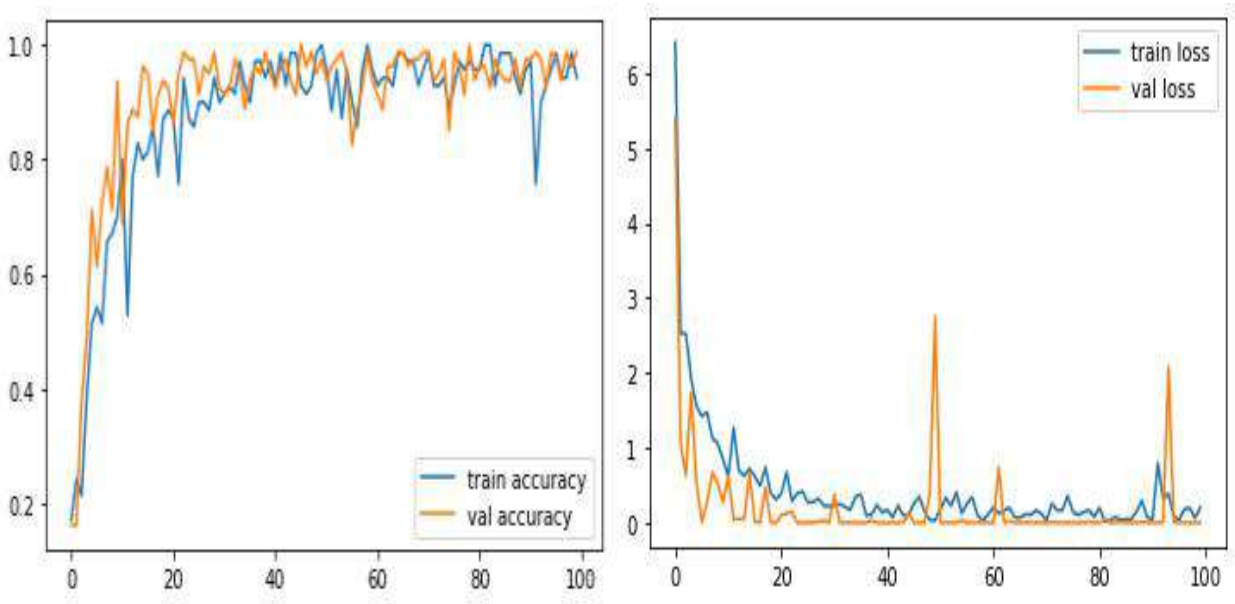


Figure III.13. : Apprentissage et validation de la précision et perte dans VGG16

III.6. Conclusion

Dans ce chapitre, nous avons présenté notre application de reconnaissance faciale basée sur les réseaux de neurones de types CNN, nous avons utilisé la technique de transfert learning à base de l'architecture VGG16 avec explication détailléedes différentes couches dans ce modèle.Nous avons présenté également les résultats obtenus pour cettearchitecture. Notre système a été testé sur la base de données « *The FEI face database* ».Nous pouvons noter que la technique de transfert learning est efficace et puissante lorsqu'on n'arrive pas à former une large base des données. Cependant, nous avons également constaté que de nombreux facteurs extérieurs influent sur la qualité de la reconnaissance.

Conclusion générale

La biométrie est un domaine à la fois passionnant et complexe. Elle tente, par des outils mathématiques souvent très évolués, de faire la distinction entre des individus, nous obligeant à travailler dans un contexte de très grande diversité. Cette diversité se retrouve également dans le nombre considérable d'algorithmes qui ont été proposés en reconnaissance faciale. Dans ce mémoire, nous nous sommes intéressés au problème de reconnaissance faciale. Notre travail consiste à la mise au point d'un algorithme robuste destiné à reconnaître un individu par son visage en temps réel en utilisant une méthode parmi les méthodes la plus utilisé dans ce domaine la premier technique c'est la détection des visages par l'utilisation de l'algorithme de viola et jones «haar cascade» c'est une approche basée sur l'apparence, qui consiste à parcourir l'ensemble de l'image en calculant un certain nombre de caractéristique dans des zones rectangulaires qui se chevauchent. Et pour la prédiction nous avons utilisé un modèle générique de VGG16 et nous avons lui apprenti sur notre base des donnés utiliser . cette méthode d'apprentissage profond appelé transfert learning est utilisée pour représenter efficacement les images de visages, qui peuvent être approximativement reconstruites à partir d'un petit ensemble de poids et d'une image de visage standard.

Nous estimons avoir réalisé un système répondant à l'objectif que nous nous sommesfixés au départ, à savoir la mise en oeuvre d'un système permettant la reconnaissanced'individus et le contrôle d'accès.En guise de perspectives, dans un premier temps une extension de ce travail peut êtreenvisager par l'étude et la réalisation d'un système de détection et de localisation du visageavec des performances assez hautes , une autre consiste à appliquer ce système sur d'autresbases de visages présentant de fortes variations dans l'éclairage et de la pose ainsi qued'envisager la possibilité d'employer une approche basée sur les éléments locale du visage.Ensuite, un des grands challenges serait de pouvoir mieux maîtriser les variationsd'environnement, qui perturbent encore trop les systèmes de reconnaissance, Les dernièresavancées de la technologie de capture 3D des images de visage ont permis de mettre en placedes systèmes de reconnaissance assez robustes par rapport à la 2D.

Si la biométrie est un enjeu important au niveau économique, la recherche, en particulierdans le domaine de la reconnaissance Des visages offre encore un champ d'investigations trèsouvert.

Bibliographie

[1] Jafri, R., and Arabnia, H. R. A survey of face recognition techniques. Jips 5, 2 (2009), 41–68.

[2]Sharif, M., Naz, F., Yasmin, M., Shahid, M. A., and Rehman, A. Face recognition : Asurvey. Journal of Engineering Science & Technology Review 10, 2 (2017).

[3]Rajanna, U., Erol, A., and Bebis, G. A comparative study on feature extraction for fingerprintclassification and performance improvements using rank-level fusion. PatternAnalysis and Applications 13, 3 (2010), 263–272.

[4]Vlachos, M., and Dermatas, E. Finger vein segmentation in infrared images using supervisedand unsupervised clustering algorithms. Pattern recognition and image analysis23, 2 (2013), 328–334.

[5]Bhuiyan, A., Nath, B., and Ramamohanarao, K. Detection and classification of bifurcationand branch points on retinal vascular network. In International Conference on DigitalImage Computing Techniques and Applications, DICTA 2012 (2012).

[6]Bhuiyan, A., Nath, B., and Ramamohanarao, K. Detection and classification of bifurcationand branch points on retinal vascular network. In International Conference on DigitalImage Computing Techniques and Applications, DICTA 2012 (2012).

[7]Mailah, M., and Lim, B. H. Biometric signature verification using pen position, time,velocity and pressure parameters. University Technology. Malaysia (2008).

[8]Shah, H. N. M., Ab Rashid, M. Z., Abdollah, M. F., Kamarudin, M. N., Chow, K. L., andKamis, Z. Biometric voice recognition in security system. Indian Journal of Science andTechnology 7, 2 (2014), 104–112.

[9]Shah, H. N. M., Ab Rashid, M. Z., Abdollah, M. F., Kamarudin, M. N., Chow, K. L., andKamis, Z. Biometric voice recognition in security system. Indian Journal of Science andTechnology 7, 2 (2014), 104–112.

[10]Pal, A., and Singh, Y. N. Ecg biometric recognition. In Mathematics and Computing(Singapore, 2018), D. Ghosh, D. Giri, R. N. Mohapatra, E. Savas, K. Sakurai, and L. P.Singh, Eds., Springer Singapore, pp. 61–73.

[11]Sonkamble, S., Thool, D. R., and Sonkamble, B. Survey of biometric recognition systemsand their applications. Journal of Theoretical & Applied Information Technology (2010).

[12]Bhatia, R. Biometrics and face recognition techniques. International Journal of Advanced Research in Computer Science and Software Engineering 3, 5 (2013).

[13]Pierre Bonazza,Système de sécurité biométrique multimodal par imagerie, dédié au contrôle d’accès (2019)

[14] Pierre Bonazza,Système de sécurité biométrique multimodal par imagerie, dédié au contrôle d’accès (2019)

[15] Pierre Bonazza,Système de sécurité biométrique multimodal par imagerie, dédié au contrôle d’accès (2019)

[16]Pierre Bonazza,Système de sécurité biométrique multimodal par imagerie, dédié au contrôle d'accès (2019)

[17] Pierre Bonazza, Système de sécurité biométrique multimodal par imagerie, dédié au contrôle d'accès (2019)

[18]Pierre Bonazza, Système de sécurité biométrique multimodal par imagerie, dédié au contrôle d'accès (2019)

[19]Pierre Bonazza, Système de sécurité biométrique multimodal par imagerie, dédié au contrôle d'accès (2019)

[20] Think excellence, choose multimodality. Tech. rep. https://www.morpho.com/sites/ morpho/files/multimodality_en-5_0.pdf.

[21]Morizet, N. Reconnaissance Biométrique par Fusion Multimodale du Visage et del'Iris. Theses, Télécom ParisTech, 2009.

[22]Lupu, E., and Pop, P. G. Multimodal biometric systems overview.

[23]Lupu, E., and Pop, P. G. Multimodal biometric systems overview.

[24] Y. Bengio, A. Courville, and P. Vincent, "Representation learning : A review and new perspectives,"

IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.

[25] J. Schmidhuber, "Deep learning in neural networks : An overview," Neural networks, vol. 61, pp. 85– 117, 2015.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[27] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning-a new frontier in artificial in- telligence research [research frontier]," IEEE Computational Intelligence Magazine, vol. 5, no. 4, pp. 13–18, 2010.

[28] J. Schmidhuber, "Deep learning.," Scholarpedia, vol. 10, no. 11, p. 32832, 2015.

[29] H. Y. Xiong, B. Alipanahi, L. J. Lee, H. Bretschneider, D. Merico, R. K. Yuen, Y. Hua, S. Gue- roussov, H. S. Najafabadi, T. R. Hughes, et al., "The human splicing code reveals new insights into the genetic determinants of disease," Science, vol. 347, no. 6218, p. 1254806, 2015.

[30] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser,

[31] F. Buggenthin, F. Buettner, P. S. Hoppe, M. Endele, M. Kroiss, M. Strasser, M. Schwarzfischer,

D. Loeffler, K. D. Kokkaliaris, O. Hilsenbeck, et al., “Prospective identification of hematopoietic lineage choice by deep learning,” Nature Methods, vol. 14, no. 4, pp. 403–406, 2017.

[32] E. Gibney, “Google reveals secret test of ai bot to beat top go players.,” Nature, vol. 541, no. 7636, p. 142, 2017.

[33] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Mon- fort, U. Muller, J. Zhang, et al., “End to end learning for self-driving cars,” arXiv preprint arXiv :1604.07316, 2016.

[34] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, “Dermatologist- level classification of skin cancer with deep neural networks,” Nature, vol. 542, no. 7639, pp. 115–118, 2017.

[35] H. C. Hazlett, “Early brain development in infants at high risk for autism spectrum disorder,” in Biological Psychiatry, vol. 73, pp. 115S–115S, ELSEVIER SCIENCE INC 360 PARK AVE SOUTH, NEW YORK, NY 10010-1710 USA, 2013.

[36] L. A. Gatys, A. S. Ecker, and M. Bethge, “A neural algorithm of artistic style,” arXiv preprint arXiv :1508.06576, 2015.

[37] R. Dechter and J. Pearl, The cycle-cutset method for improving search performance in AI applica- tions. University of California, Computer Science Department, 1986.

[38] I. Aizenberg, N. N. Aizenberg, and J. P. Vandewalle, Multi-Valued and Universal Binary Neurons : Theory, Learning and Applications. Springer Science & Business Media, 2013.

[39] A. Van den Oord, S. Dieleman, and B. Schrauwen, “Deep content-based music recommendation,” in Advances in neural information processing systems, pp. 2643–2651, 2013.

[40] R. Collobert and J. Weston, “A unified architecture for natural language processing : Deep neural networks with multitask learning,” in Proceedings of the 25th international conference on Machine learning, pp. 160–167, ACM, 2008.

[41] M. Matsugu, K. Mori, Y. Mitari, and Y. Kaneda, “Subject independent facial expression recognition with robust face detection using a convolutional neural network,” Neural Networks, vol. 16, no. 5, pp. 555–559, 2003.

[42] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016. http://www. deeplearningbook.org.

[43] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, pp. 1097–1105, 2012.

[45] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla,

M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,”

International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015.

[46] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in European conference on computer vision, pp. 818–833, Springer, 2014.

[47] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Ra- binovich, “Going deeper with convolutions,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9, 2015.

[48] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” arXiv preprint arXiv :1602.07261, 2016.

[49] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, 2016.

[50] M. C. Mozer, “A focused backpropagation algorithm for temporal,” Backpropagation : Theory, architectures, and applications, p. 137, 1995.

[51] P. J. Werbos, “Generalization of backpropagation with application to a recurrent gas market model,”

Neural networks, vol. 1, no. 4, pp. 339–356, 1988.

[52] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks.,”

ICML (3), vol. 28, pp. 1310–1318, 2013.

[53] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[54] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” arXiv preprint arXiv :1406.1078, 2014.

[55] T. Mikolov, M. Karafiát, L. Burget, J. Cernocky`, and S. Khudanpur, “Recurrent neural network based language model.,” in Interspeech, vol. 2, p. 3, 2010.

[56] T. Mikolov, S. Kombrink, L. Burget, J. Černocky`, and S. Khudanpur, “Extensions of recurrent neural network language model,” in Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pp. 5528–5531, IEEE, 2011.

[57] I. Sutskever, J. Martens, and G. E. Hinton, “Generating text with recurrent neural networks,” in Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 1017–1024, 2011.

[58] S. Liu, N. Yang, M. Li, and M. Zhou, “A recursive recurrent neural network for statistical machine translation,” 2014.

[59] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in

Advances in neural information processing systems, pp. 3104–3112, 2014.

[60] M. Auli, M. Galley, C. Quirk, and G. Zweig, “Joint language and translation modeling with recurrent neural networks.,” in EMNLP, vol. 3, p. 0, 2013.

[61] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks.,” in ICML, vol. 14, pp. 1764–1772, 2014.

[62] A. Karpathy and L. Fei-Fei, “Deep visual-semantic alignments for generating image descriptions,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3128–3137, 2015.

[63] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell : A neural image caption generator,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3156– 3164, 2015.

[64] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and

T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2625–2634, 2015.

[65] A. Y. Ng and M. I. Jordan, “On discriminative vs. generative classifiers : A comparison of logistic regression and naive bayes,” Advances in neural information processing systems, vol. 2, pp. 841–848, 2002.

[66] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, “A learning algorithm for boltzmann machines,”

Cognitive science, vol. 9, no. 1, pp. 147–169, 1985.

[67] R. Salakhutdinov, A. Mnih, and G. Hinton, “Restricted boltzmann machines for collaborative filte- ring,” in Proceedings of the 24th international conference on Machine learning, pp. 791–798, ACM, 2007.

[68] G. Hinton, “A practical guide to training restricted boltzmann machines,” Momentum, vol. 9, no. 1, p. 926, 2010.

[69] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” Neural computation, vol. 18, no. 7, pp. 1527–1554, 2006.

[70] R. Salakhutdinov and G. Hinton, “Deep boltzmann machines,” in Artificial Intelligence and Statis- tics, pp. 448–455, 2009.

[71] N. Srivastava and R. R. Salakhutdinov, “Multimodal learning with deep boltzmann machines,” in

Advances in neural information processing systems, pp. 2222–2230, 2012.

[72] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and

Y. Bengio, “Generative adversarial nets,” in Advances in neural information processing systems, pp. 2672–2680, 2014.

ملخص

يوجد في السنوات الأخيرة اهتمام متزايد في القياسات الحيوية. أصبح التعرف على الوجه ، كتقنية بيومترية أساسية ، مهما بشكل متزايد في البحث العلمي ، بسبب طبيعته غير المتطفلة وغير التلامسية. ولكن على الرغم من العديد من الأساليب التي تم اقتراحها لحل مشكلة التعرف على الوجه البشري ، إلا أنها تظل مشكلة صعبة للغاية ، ويرجع ذلك إلى حقيقة أن وجوه الأشخاص المختلفين لها عموما نفس الشكل وتختلف بسبب الظروف. الإضاءة ، والاختلاف في التشكيل ، وتعبيرات الوجه. في الوقت الحاضر ، يبدو أن أنظمة التحقق من الهوية هي ناقل مثير للاهتمام لاستغلاله ، نظرا للعديد من التطبيقات التي تتطلب منها التحكم في الوصول إلى المواقع الحساسة ، والرصد عن بعد ، وما إلى ذلك. يركز العمل المنجز لهذه الأطروحة على التعرف التلقائي على الوجوه باستخدام طريقة التعرف على الوجه باستخدام التعلم العميق والشبكات العصبونية ، أظهرت النتائج التي تم الحصول عليها أن طريقة نقل التعلم تعطي نتائج أفضل بشكل واضح عندما لا تكون قاعدة البيانات كبيرة بما يكفي. للتحقق من صحة هذا العمل اختبرنا هذه التقنية على الوجوه الأمامية المكتسبة في الوقت الحقيقي عبر كاميرا ويب.

كلماتالبحث :

التعرف على الوجوه ، القياسات الحيوية ، التعلم العميق، الشبكات العصبية ، استخراج الميزات .

Résumé

Au cours de ces dernières années, on observe un intérêt croissant autour de la biométrie. La reconnaissance faciale en tant qu'une technologie biométriques de base, a pris une part de plus en plus importante dans le domaine de la recherche, du fait de son caractère non intrusif et sans contact. Mais malgré les nombreuses approches et méthodes qui ont été proposées pour résoudre le problème de reconnaissance du visage humains, il demeure un problème extrêmement difficile, ceci est dû au fait que le visage de personnes différentes ont généralement la même forme et varie du fait des conditions d'éclairage, de la variation de pose, et des expressions faciales. De nos jours les systèmes de vérification d'identité apparaît être un vecteur intéressant à exploiter, vu la multitude des applications qui leurs font appel contrôle d'accès aux sites sensibles, télésurveillance...etc. Le travail effectué dans le cadre de ce mémoire porte sur l'identification automatique de visages avec une méthode de reconnaissance faciale en utilisant l'apprentissage profond et le réseau neurones convolutif de VGG16 . Les résultats, obtenue ont montré que la méthode de transfert learning donne nettement des meilleurs résultats lorsque les base des données sont pas assez large .Pour validé ce travail nous avons testé cette technique sur des visages frontales acquis en temps réel via une webcam .

Mots-clés:

Reconnaissance de visages, Biométrie, apprentissage profond , VGG16, extraction de caractéristiques.

Abstract

In recent years, there has been a growing interest in biometrics. Facial recognition, as a basic biometric technology, has become increasingly important in research, due to its non-intrusive and contactless nature. But despite the many approaches and methods that have been proposed to solve the problem of human face recognition, it remains an extremely difficult problem, this is due to the fact that the faces of different people generally have the same shape and vary due to the conditions. lighting, variation in pose, and facial expressions. Nowadays, identity verification systems appear to be an interesting vector to exploit, given the multitude of applications which require them to control access to sensitive sites, remote monitoring, etc. The work carried out for this dissertation focuses on the automatic identification of faces with a facial recognition method using deep learning and the convolutional neural network of VGG16. The results obtained showed that the learning transfer method clearly gives better results when the database is not large enough. To validate this work we tested this technique on frontal faces acquired in real time via a webcam.

Keywords:

Face recognition, Biometrics, deep learning, VGG16, feature extraction.