



République Algérienne Démocratique Et Populaire
Ministère De L'enseignement Supérieure Et De La Recherche
scientifique Université 20 Août 1955 –Skikda-
Faculté des sciences département de l'informatique

**Mémoire de fin d'études en vue de l'obtention du diplôme de Master en
Informatique
OPTION : Génie Logiciel Avancé et Applications,
Système Informatique**

THÈME

**Classification d'Images Numériques Couleurs :
Une Approche par Apprentissage Profond (Deep
Learning, CNN) – Application à la Reconnaissance
Automatique du Genre Humain**

Présenté par :

ACHOURI A. Karim.
DEROUICHE Fadi Ali.

Encadré par :

Pr. BOUCHEHAM Bachir.

Juin 2022

Remerciements

Premièrement, nous remercions Dieu tout puissant pour tous les bienfaits qu'il nous a accordés et pour nous avoir donné la force et la volonté nécessaires pour accomplir ce travail.

Nous remercions par la suite, notre encadreur, Professeur BOUCHEHAM Bachir, pour la liberté qu'il nous a donné et pour ses conseils précieux.

Nous exprimons notre respect aux membres du jury qui ont accepté d'y participer et de juger notre travail.

Nous remercions également tous les enseignants qui ont contribué à notre instruction durant nos années d'étude.

Sans oublier tous ceux qui nous ont aidés de près ou de loin à l'élaboration de ce modeste travail.

Dédicaces

Avec tout l'amour que nous portons dans nos cœurs, nous dédions ce mémoire de fin d'études universitaires à nos deux êtres respectifs qui nous sont les plus chers dans ce monde : Nos parents respectifs. Nous dédions ce modeste travail aussi à nos familles respectives, nos amis ainsi qu'à tous nos enseignants.

Sommaire

Acronymes	7
Liste des figures.....	8
Résumé	11
<i>Introduction Générale</i>	12
<i>Introduction générale : Problématique, Contexte, Objectifs et Outils</i>	13
Problématique et Contexte.....	13
Chapitre I :	15
Classification automatique des images numériques.....	15
1 <i>Introduction</i>	16
2 <i>Notion de classification d'image numérique</i>	16
3 <i>Types de classification d'image numérique</i>	16
3.1 <i>Classification supervisée</i>	17
3.2 <i>Classification non supervisée</i>	17
4 <i>Motivations et Applications de la classification d'images</i>	18
5 <i>Extraction de caractéristiques (features extraction) dans l'image numérique</i>	18
5.1 <i>Les caractéristiques de bas niveau</i>	19
5.1.1 <i>La texture de l'image</i>	19
5.1.2 <i>La forme de l'image</i>	20
5.2 <i>Caractéristiques de haut niveau</i>	20
5.3 <i>Caractérisation locale</i>	20
5.3.1 <i>Transformation de caractéristique visuelles invariantes à l'échelle (SIFT) :</i>	21
5.3.2 <i>Orienté rapide et tourné BRIEF (ORB) (Oriented FAST and RotatedBRIEF</i>	21
5.4 <i>Caractérisation globale</i>	21
5.4.1 <i>Histogramme de gradient Orienté (HOG)</i>	21
6 <i>Quelques algorithmes de classification</i>	21
6.1 <i>Le k-plus proches voisins (KPPV) ou KNN : K-NearestNeighbors</i> :.....	22
6.2 <i>Définition de KNN</i>	22
6.2.1 <i>Les différents types de distance utilisée dans le KNN</i>	22
6.2.2 <i>Choix de la valeur de K</i>	23
6.3 <i>Les arbres de décision</i>	23
6.3.1 <i>Définition de l'arbre de décision</i>	23
6.3.2 <i>Le principe de fonctionnement de l'arbre de décision</i>	23

6.4	Les Machines à vecteur de support (SVM : Support Vector Machines)	24
6.4.1	Définition d'un SVM	24
6.4.2	Le fonctionnement principal des machines à vecteurs support SVM	24
6.5	Les réseaux de neurones artificiels (RNA)	26
6.5.1	Définition de réseaux de neurones artificiels	26
6.5.2	Principe fonctionnement de RNA	27
7	Mesures de la qualité de classification	28
7.1	Mesures d'évaluation des algorithmes de classifications	28
7.1.1	Matrice de confusion multi classe	28
7.1.2	Précision (P)	30
7.1.3	Recall (True positive rate sensitivity)	31
7.1.4	F-Measure	31
7.1.5	Taux d'erreur (Accuracy)	32
8	Techniques d'amélioration des algorithmes de classification	32
8.1	Amélioration par bagging	32
8.2	Amélioration par boosting	33
9	Conclusion	34
	Chapitre II :	35
	Le Deep Learning pour la classification de l'image numérique	35
1	Introduction :	36
2	Définition de l'apprentissage profond (deep learning) :	36
3	Machine learning ET deep learning	37
4	Pourquoi le deep learning :	37
5	Architectures de réseaux de neurones profonds	37
6	Quelques explications sur les CNNs	38
6.1	Principe d'architecture d'un CNN	38
6.2	L'entraînement d'un nouveau CNN est difficile	38
6.3	Les blocs de construction :	39
6.3.1	La couche de convolution (CONV) :	39
6.3.2	Couche de pooling (POOL) :	40
6.3.3	Couche de correction (ReLU):	41
6.3.4	Le flattening :	42
6.3.5	La couche entièrement connectée :	43
7	Optimisation pour l'apprentissage en Deep Learning	43

7.1	<i>Les variantes de la descente du gradient</i>	43
7.2	<i>Algorithmes d'optimisation de la descente du gradient</i>	44
7.2.1	<i>Adagrad</i>	44
7.2.2	<i>Adam</i>	44
7.2.3	<i>RMSprop</i>	44
8	<i>Les applications du Deep-Learning</i>	44
8.1	<i>La reconnaissance faciale</i>	44
8.2	<i>Le traitement automatique de langage naturel</i>	45
8.3	<i>Voitures autonomes</i>	45
8.4	<i>Recherche vocale et assistants à commande vocale</i>	45
8.5	<i>Ajout automatique de sons à des films muets</i>	45
8.6	<i>Traduction automatique</i>	46
8.7	<i>Génération automatique de texte</i>	46
8.8	<i>Compréhension de l'Image (Image Understanding)</i>	46
9	<i>La description automatique d'image (Image Automatiqueannotation)</i>	46
10	<i>La classification d'image par les réseaux de neuronesconvolutionnels</i>	47
11	<i>Les problèmes liés à l'apprentissage</i>	47
11.1	<i>Sur-apprentissage et sous-apprentissage</i>	47
11.2	<i>Le problème de descente et l'explosion de gradients (gradientsVanishing et exploding)</i>	48
12	<i>Les solutions :</i>	48
12.1	<i>La solution du « Early-Stop »</i>	48
12.2	<i>Drop-Out</i>	49
13	<i>Conclusion :</i>	50
	Chapitre III :	51
	Langages et outils	51
	1. Introduction	52
	2. Langages et outils	52
	2.1 Le langage de programmation Python	52
	2.1.1 Cas d'utilisation Python	52
	2.1.4 Flexibilité Python	53
	2.1.5 Les parties les moins intéressantes de Python	54
	2.1.6 Python et IA	54
	2.2 Environnement de développement : Google Colab	55
	2.2.2 Fonctionnement général de Google-Colab	55

2.2.4 Les avantages de l'environnement Google-colab	56
3. Le DataSet d'images	56
4. Conclusion	58
Chapitre IV :	59
Proposition d'un Classificateur du genre humain d'images basé Deep Learning.....	59
1 Introduction	60
2 Description des étapes du processus :	60
2.1 Chargement des librairies nécessaires :	60
2.2 Préparation des données	61
2.3 Définition du modèle de détection de genre :	62
2.4 Compilation du modèle	63
2.5 Création du modèle de détection de voile :	64
2.6 Test du modèle :	65
3 Conclusion	67
<i>Conclusion Générale</i>	68
Conclusion générale.....	69
<i>Bibliographie</i>	70

Acronymes

AA : Apprentissage automatique.

AD : Arbre de décision.

ADAM : Adaptive Moments

AI : Artificial Intelligence.

CNN: Convolutional Neural Network.

DM: Data Mining.

DW : Data Warehousing.

ECD : Extraction des Connaissance dans les Données

HOG : Histogramme de gradient Orienté

IA : Intelligence Artificielle

KDD: Knowledge discovery in Data.

KNN: K-Nearest Neighbors.

K-PP: K-plus proche voisins

ML: Machine Learning.

MLP: Multi-Layer perception.

NoSQL : Not Only Sql.

ORB : Orienté rapide et tourné BRIEF.

ReLU : Les couches de Connexion

SIFT : Transformation de caractéristique visuelles invariantes à l'échelle.

SVM : Support Vector Machine.

TPU : Tensor Processing Units.

Liste des figures

Figure 1 type de classification.....	16
Figure 2 Classification Supervisée.....	17
Figure 3 Classification Non Supervisée.....	17
Figure 4 Exemple d'histogramme d'une image couleur (Modèle RGB : Red Green Blue ou RVB : Rouge Vert Bleu) [14].	19
Figure 5 Cinq textures d'image différentes [15].....	20
Figure 6 l'application de l'algorithme KNN pour $K=3$ et $K=5$ [20].....	22
Figure 7 Exemple d'un arbre de décision de régression (à droite) issu d'un ensemble d'apprentissage (Tableau à gauche) [24].....	24
Figure 8 Exemple des données linéairement séparable [25].	24
Figure 9 Le principe des SVM pour les cas linéairement séparables [25].	25
Figure 10 Exemple des données Non linéairement séparable [26].....	26
Figure 11 Projection des données dans un espace de plus grande dimension [27].	26
Figure 12 Structure d'un réseau de neurone artificiel [29].....	27
Figure 13 Exemple d'une matrice de confusion de N classes [32].....	29
Figure 14 Illustration d'un tirage aléatoire avec remise pour la formation d'un échantillon bootstrap [33]......	32
Figure 15 Des échantillons bootstrap d'une base de données [40]	33
Figure 16 Schéma résumant la place de deep learning dans le monde de l'informatique.....	36
Figure 17 Les réseaux de neurones convolutifs.	38
Figure 18 Ensemble de neurones (cercles) créant la profondeur d'une couche de convolution [37].....	40
Figure 19 Exemple couche de pooling [39].....	41
Figure 20 Tracé de la fonction ReLU [39]	41
Figure 21 La taille de l'image après le max pooling [39].....	42
Figure 22 Exemple de flattening [37].....	42
Figure 23 Exemple de sur apprentissage et de sous apprentissage [37].	48
Figure 24 Un exemple du point du earlystopping.	49
Figure 25 Un réseau de neurones avant et après l'application de la méthode « drop- out »	50
Figure 26 man dataset (17678) picture	57
Figure 27 Woman Dataset (9849) picture	57
Figure 28 Woman with Hijab Dataset (500) picture	58
Figure 29 librairies utilisées	60
Figure 30 préparation des données.....	61
Figure 31 définition du modèle de détection de genre	62
Figure 32 Compilation du modèle.....	63

Figure 33 graphe de précision du modèle genre.....	63
Figure 34 graphe de précision du modèle voile.....	64
Figure 35 principe de transfert d'apprentissage	64
Figure 36 exemple des résultats prédits.....	65
Figure 37 Model Testing	66
Figure 38 exemple live prédiction.....	67

Résumé

Nous nous intéressons dans ce travail de Master à la tâche de classification des images numériques. En particulier, notre étude s'oriente sur le plan pratique sur le cas spécifique de la classification du genre humain, à partir d'images numériques. Nous utilisons pour cela, les réseaux de neurones à convolution, (ou Convolutionnels, Convolutional Neural Networks, CNN, en anglais) sous la Plateforme GoogleColab.

Les CNN sont des réseaux de neurones multicouches spécialisés dans des tâches de reconnaissance de formes. Dans notre travail, hormis la partie étude théorique, nous avons proposé un modèle de classification basé CNN autour de la base d'images que nous avons collectée. Les résultats du classifieur du genre humain, ainsi réalisé, ont montré que le choix du nombre d'époques et la taille de la base d'images ainsi que la profondeur du réseau ont une grande influence pour avoir les meilleures performances de classification

Abstract

In this Master's work, the focus is on the task of digital images classification, in general. Particularly, we target the more specific objective of Human Gender Classification, through input images. For this aim, we use convolutional neural networks (CNN), under the Google Colab Platform.

CNNs are multilayer neural networks specialized in pattern recognition tasks. In our work, apart from the theoretical study, we have proposed a CNN-based Gender classification model, around the image base that we have collected. The results of the classifier thus realized showed that the choice of the number of epochs and the size of the image base as well as the depth of the network have a great influence to have the best classification performance.

Introduction Générale

Introduction générale : Problématique, Contexte, Objectifs et Outils

Problématique et Contexte

La reconnaissance des Formes est un domaine en plein essor depuis des dizaines d'années. Parmi les tâches les plus connues de ce domaine, la tâche de classification est, sans nul doute, des plus étudiées. Présentée de façon simple, cette tâche consiste à assigner une classe d'appartenance à un élément x en entrée (externe) une des n classes préexistantes. La connaissance extraite ici est justement la classe d'appartenance de x .

Dans ce cadre, nous nous intéressons dans ce travail de Master à la tâche de classification des images numériques. Plus particulièrement, nous intéressons à classifier les images de visages humains selon leurs genres : {Féminin, Masculin}.

Une telle Problématique-Application est, à la fois, intéressante, motivante et assez complexe à réaliser. En effet, le visage humain présente des variabilités multidimensionnelles : Genre, Race, Age, Couleur de la peau, Expression faciale, Port ou non de : Barbe, Lunettes, Chapeau/Casquette, Voile, etc., et même, ces dernières années, Masque de protection contre la COVID-19 !

Objectifs

Ce travail de Master impose deux objectifs majeurs :

Objectif 1 : *Une étude théorique sur les techniques de classification des données en général et des images en particulier. Les travaux sur la classification du genre humain constituent, bien sûr, une cible privilégiée dans cette étude théorique.*

Objectifs 2 : *Le deuxième objectif de ce travail de Master consiste en la réalisation d'un classifieur du genre humain capable de recevoir en entrée une image d'un visage humain et puis d'en déduire s'il s'agit d'un genre Masculin ou d'un genre Féminin. Le système doit, bien sûr, résister aux différentes variabilités citées plus haut, en plus des variabilités techniques, liées, elles, aux conditions d'acquisition des images. Cet objectif nécessite, quant à lui, de choisir les outils appropriés pour aboutir à un Classifieur du genre humain très performant. Sachant que tout système relevant de l'informatique computationnelle doit répondre à deux*

critères de performance : L'efficacité (efficiency) et la pertinence (effectiveness), nous parlons ici de deux critères, comme suit :

Critère de pertinence : dans le contexte de la classification de données, le critère de pertinence le plus utilisée est celui de la *Précision de Classification*.

Critère d'efficacité : Le classifieur doit être assez rapide pour donner la classe d'appartenance de l'élément entrée en un temps aussi court que possible, voire, en temps réel.

Outils

Pratiquement et suite aux études théoriques, nous avons opté pour l'utilisation des Réseaux de Neurones à Convolution, (ou Convolutionnels, Convolutional Neural Networks, CNN, en anglais) sous la Plateforme Google-Colab.

En bref, les CNN sont des réseaux de neurones multicouches spécialisés dans des tâches de reconnaissance des formes. Dans notre travail, hormis la partie étude théorique, nous avons proposé un modèle de classification basé CNN autour de la base d'images que nous avons collectées. Les résultats du classifieur ainsi réalisé ont montré que :

Critère de pertinence : Les taux de précisions obtenus sont de ...%.

Critère d'efficacité : Les temps de réponse du système réalisé sont de l'ordre de... Nous avons même introduit la classification du genre par caméra en ligne. Le temps de réponse ici aussi est quasi-réel.

Autres : L'évaluation générale du système montre aussi que le choix du nombre d'époques et la taille de la base d'images ainsi que la profondeur du réseau ont une grande influence pour avoir les meilleures performances de classification.

Chapitre I :

Classification automatique des images numériques

1 Introduction

Le traitement et l'analyse des images permettent l'extraction des caractéristiques des objets au sein d'une image et la classification des objets en fonction des données extraites. La forme, la texture et la couleur sont des caractéristiques typiques qui peuvent être extraites d'une image et utilisées pour la reconnaissance d'objets. Dans ce chapitre nous allons parler de la classification automatique des images numérique [1].

2 Notion de classification d'image numérique

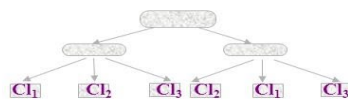
La classification automatique des images consiste à attribuer automatiquement une Image à une classe à l'aide d'un système de classification, elle consiste à répartir systématiquement des images selon des classes établies au préalable, classer une image lui fait correspondre une classe, marquant ainsi sa parenté avec d'autres images [2].

3 Types de classification d'image numérique

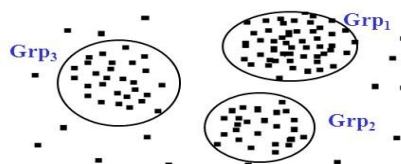
De nombreuses méthodes classiques de classification d'image ont été consacrées, elles peuvent être séparées en deux grandes catégories : les méthodes de classification supervisée et les méthodes de classification non supervisée.

La classification

- **Supervisée : on connaît les classes**



- **Non supervisée : on ne connaît pas les classes**



Ansaf SALLEB - Décembre 2004

9

Figure 1 type de classification

3.1 Classification supervisée

L'objectif de la classification supervisée de l'image numérique est principalement de définir des règles permettant de classer des images dans des classes à partir de variables caractérisant les images [3]. En effet, la classification supervisée d'image permet à l'algorithme d'apprentissage d'apprendre une règle de classification fondée sur des classes d'image bien définies [4].

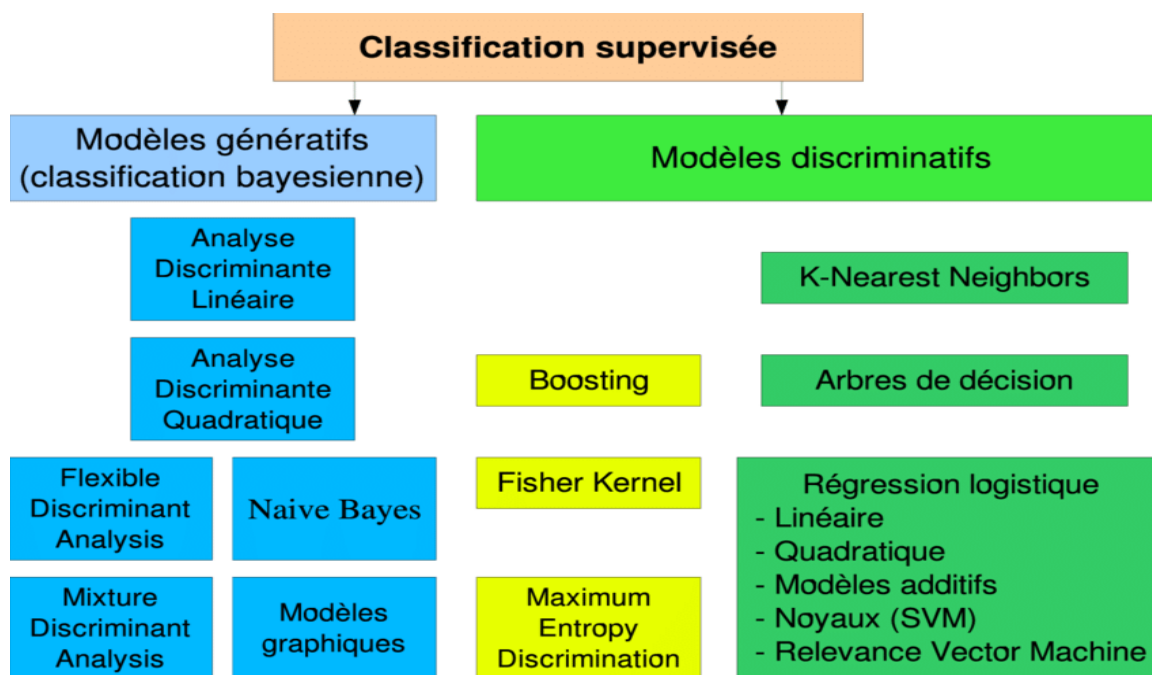


Figure 2 Classification Supervisée

3.2 Classification non supervisée

La classification non supervisée de l'image numérique 'image clustering' consiste à diviser un ensemble des images en sous-ensembles appelés classes, de sorte que les images de chaque classe soient le plus homogènes possible avec les autres images de la même classe suivant des critères définis. Les images sont regroupées selon le critère qu'utilise la méthode de classification employée [5].

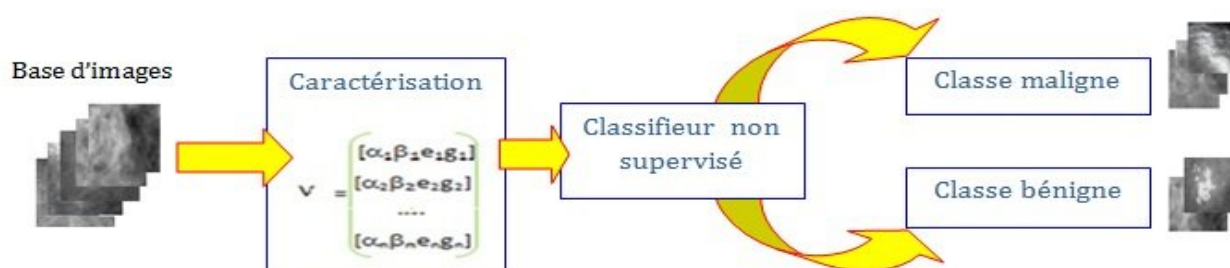


Figure 3 Classification Non Supervisée

4 Motivations et Applications de la classification d'images

L'objectif de la classification d'images est d'élaborer un système capable d'affecter une image automatiquement à une classe prédéfinie. Ainsi, ce système permet d'effectuer une tâche d'expertise qui peut s'avérer coûteuse pour un être humain en raison des contraintes physiques comme la concentration, la fatigue ou le temps nécessaire par un volume important d'image [5].

Les applications de la classification automatique d'images sont nombreuses comme [6] :

- Domaine médical comme la reconnaissance de cellules ou de tumeurs dans les mammographies ;
- Dans l'agriculture comme la classification de pollen la reconnaissance du type de sol et des grains, la classification d'herbes ;
- Dans le domaine du document comme la reconnaissance de l'écriture manuscrite pour les chèques, les codes postaux, les cartes ;
- Dans le domaine urbain comme la reconnaissance de panneaux de signalisation, la reconnaissance de piétons, la détection de véhicules, la reconnaissance de bâtiments pour aider à la localisation ;
- Dans le domaine de la biométrie comme la reconnaissance de visage, d'empreintes.

Le point commun à toutes ces motivations est qu'elles nécessitent la mise en vicinité d'une chaîne de traitement à partir des photos disponibles composée de plusieurs étapes afin de fournir en sortie une décision. Chaque étape de la mise en place d'un tel système de classification nécessite la recherche de méthode appropriées pour une performance globale optimale savoir le segment d'extraction de caractéristiques et la section d'apprentissage [6].

5 Extraction de caractéristiques (features extraction) dans l'image numérique

Une caractéristique est une fonction qui représente la valeur quantifiée d'une image. Elle représente les caractéristiques importantes de l'image telles que la couleur, la texture et la forme. Dans le processus de l'image mining, la phase d'extraction de caractéristiques constitue généralement l'une des phases les plus importantes dans l'élaboration du système de classification [7]. Il s'agit en effet de déterminer un espace numérique de description dans lequel les données images seront projetées. Nous trouvons des descripteurs de bas niveau s'intéressant à l'information contenue dans l'image au niveau du pixel et des descripteurs de plus haut niveau

nécessitant une représentation intermédiaire de l'image plus adaptée [8]. Cette description peut être locale ou globale selon la nature de l'information à prélever et se fait à l'aide d'opérateurs ou de descripteurs [9].

5.1 Les caractéristiques de bas niveau

Une caractéristique bas niveau est un ensemble de valeurs et qui sont extraites directement de l'image permettant de la caractériser. L'extraction des caractéristiques de bas niveau représente une première abstraction par rapport à l'image brute, le but est que cette caractéristique soit discriminante par rapport aux entités visuelles que l'on cherche à caractériser. Ces descripteurs bas niveau peuvent être utilisés pour l'image entière comme ils peuvent être utilisés pour une partie de l'image [10].

5.1.1 La couleur de l'image

La couleur est l'une des plus importantes caractéristiques de bas niveau et la plus attirante dès la première vue de l'image, Il existe plusieurs espaces colorimétriques pour représenter la couleur tels que RGB, HSL, LAB [11]. Quel que soit l'espace de couleur utilisée, il existe plusieurs façons de caractériser la couleur tels que L'histogramme couleur qui a été proposé pour la première fois par SWAIN [12]. Cette méthode consiste à calculer l'histogramme de couleur de chaque image c'est-à-dire le nombre de pixels pour chaque couleur ensuite la similarité entre les images est calculée à partir d'intersection entre histogrammes [13].

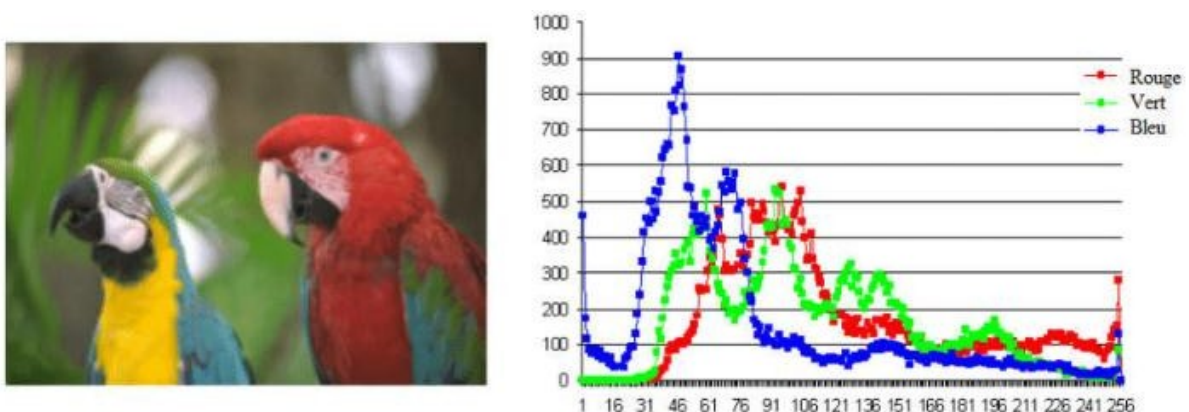


Figure 4 Exemple d'histogramme d'une image couleur (Modèle RGB : Red Green Blue ou RVB : Rouge Vert Bleu) [14].

5.1.1 La texture de l'image

La définition littéraire de la texture est la suivante : « répétition spatiale d'un même motif dans

différentes directions de l'espace. Cette définition est limitative car elle caractérise l'objet indépendamment d'un observateur humain. La notion de texture est utilisée pour traduire un aspect homogène de la surface d'un objet sur une image [15].

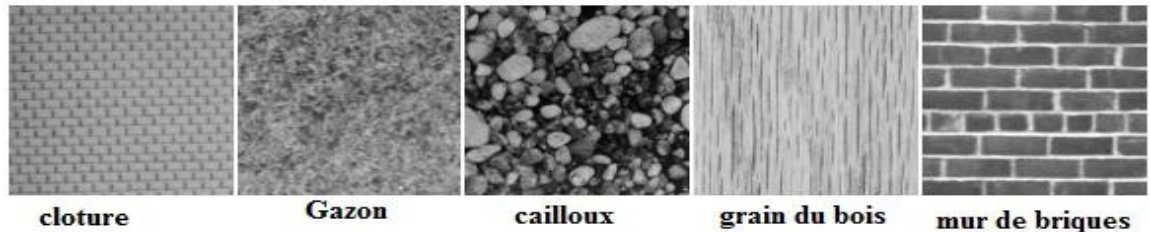


Figure 5 Cinq textures d'image différentes [15].

5.1.2 La forme de l'image

La forme est un descripteur très important pour l'image, elle désigne l'aspect général d'un objet, et son contour [16].

5.2 Caractéristiques de haut niveau

Une caractéristique de haut niveau, généralement appelée caractéristique sémantique, représente le contenu sémantique qui se définit par les attributs abstraits de l'image tels que : les scènes sémantique (bord de mer, rue, champ d'oiseaux, etc.), comportement sémantique (la performance, agression...) et l'émotion sémantique (calme, l'harmonie, l'inspiration, etc.). Cependant la sémantique de l'image est la caractéristique la plus complexe, floue et abstraite est difficile à extraire de l'image ; généralement, les caractéristiques haut niveau dépendent des caractéristiques bas niveaux [13].

Les méthodes d'extraction de plus haut niveau tiennent compte des formes et des structures dans l'image et des relations spatiales entre les pixels. Les propriétés les plus recherchées dans ces extracteurs sont, outre leur pouvoir descriptif, l'invariance et la robustesse des différentes transformations pouvant affecter l'image [17].

5.3 Caractérisation locale

Les caractéristiques locales d'un ensemble d'images se réfèrent aux caractéristiques de l'image qui adhère à la détection des contours, la segmentation de l'image et les caractéristiques calculées sur les résultats de la subdivision de l'image. Les méthodes les plus courantes utilisées pour l'extraction de caractéristiques locales sont les suivantes :

5.3.1 Transformation de caractéristique visuelles invariantes à l'échelle (SIFT) :

En anglais Scale Invariant Feature Transformation (SIFT). SIFT est utilisé pour détecter les caractéristiques locales de l'image pour des applications telles que la reconnaissance des gestes, le suivi vidéo, la modélisation 3D, etc. Un ensemble de caractéristiques clés de l'image est extrait et stocké dans une base de données. Pour une nouvelle image, les entités sont comparées à la base de données existante afin de déterminer le meilleur appariement candidat en utilisant la distance euclidienne des vecteurs d'entités [17]

5.3.2 Orienté rapide et tourné BRIEF (ORB) (Oriented FAST and Rotated BRIEF)

ORB est l'une des méthodes d'extraction de caractéristiques locales qui surmontent l'inconvénient de SIFT et réduit le temps de calcul. Elle est basée sur le BRIEF (Binary robust elementary features) et le détecteur de points FAST [18].

5.4 Caractérisation globale

Une fois que les entités locales telles que les bords des images sont extraites de l'image, les entités globales sont calculées pour l'image entière. Ces fonctionnalités globales aident à la classification finale de l'image à l'aide des éléments calculés des fonctionnalités locales [19].

5.4.1 Histogramme de gradient Orienté (HOG)

Une méthode courante utilisée pour extraire les entités globales de l'ensemble des images Histogramme de gradient Orienté (HOG), en anglais Histogram of Oriented Gradients (HOG), est utilisé comme descripteur de fonctionnalités dans le traitement d'image et la vision par ordinateur pour la détection d'objets. Il est basé sur l'histogramme des gradients qui agissent comme une caractéristique globale pour la détection d'objets. Dans les parties locales de l'image, il compte les occurrences de l'orientation du gradient pour produire l'histogramme [19].

6 Quelques algorithmes de classification

Le data mining utilise beaucoup de techniques pour l'extraction de connaissance, telles que les arbres de décision, les réseaux de neurones, les k plus proches voisins, les SVM (Support Vector Machines), etc.

6.1 Le k -plus proches voisins (KPPV) ou KNN : K-NearestNeighbors) :

6.2 Définition de KNN

L'algorithme KNN figure parmi les plus simples algorithmes d'apprentissage automatique. Cet algorithme est qualifié comme paresseux (Lazy Learning) car il n'apprend rien pendant la phase d'entraînement. Pour prédire la classe d'une nouvelle donnée d'entrée, il va chercher ses K voisins les plus proches (en utilisant la distance euclidienne, ou autres) et choisira la classe des voisins majoritaires. Pour pouvoir effectuer une prédiction, K-NN se base sur le jeu de données pour produire un résultat[20].

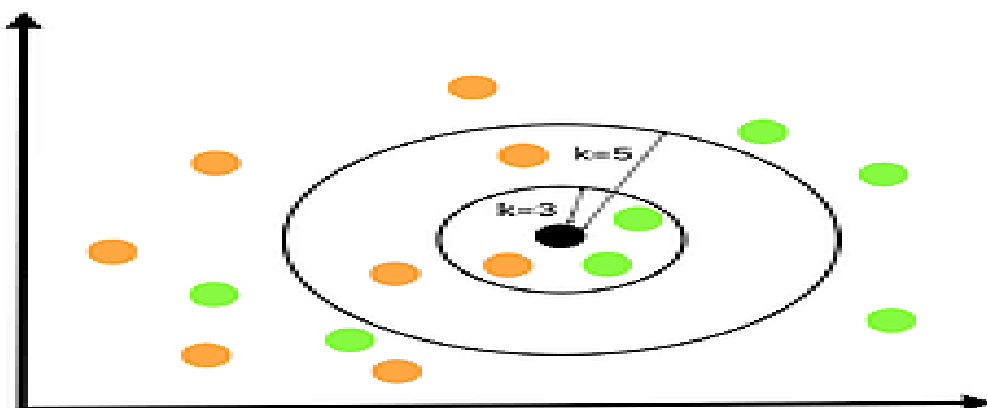


Figure 6 l'application de l'algorithme KNN pour $K=3$ et $K=5$ [20].

6.2.1 Les différents types de distance utilisée dans le KNN

Il existe plusieurs fonctions de calcul de distance, notamment, la distance euclidienne, la distance de Manhattan [21].

a. La distance euclidienne

Distance calcule la racine carrée de la somme des différences carrées entre les coordonnées de deux vecteurs $x(n)$, $y(n)$:

$$D_e(x, y) = \sqrt{\sum_{j=1}^n (x_j - y_j)^2} \quad (1)$$

b. Distance Manhattan :

La distance de Manhattan : calcule la somme des valeurs absolues des différences entre les coordonnées de deux vecteurs $x(n)$, $y(n)$:

$$D_m(x, y) = \sum_{i=1}^k |x_i - y_i| \quad (2)$$

6.2.2 Choix de la valeur de K

Le paramètre k doit être déterminé par l'utilisateur : $k \in \mathbb{N}$. En classification binaire, il est utile de choisir k impair pour éviter les votes égalitaires. Le meilleur choix de k dépend du jeu de données. En général, les grandes valeurs de k réduisent l'effet du bruit sur la classification et donc le risque de sur-apprentissage, mais rendent les frontières entre classes moins distinctes. Un bon k peut être sélectionné par diverses techniques heuristiques, par exemple, de validation croisée. Nous choisirons la valeur de k qui minimise l'erreur de classification [22].

6.3 Les arbres de décision

6.3.1 Définition de l'arbre de décision

L'apprentissage par arbre de décision est une méthode classique en apprentissage automatique. Les arbres de décision (AD) sont une catégorie d'arbres utilisés dans l'exploration de données et en informatique décisionnelle. Son but est de créer un modèle qui prédit la valeur d'une variable-cible depuis la valeur de plusieurs variables d'entrée [23].

Il existe deux principaux types d'arbre de décision en fouille de données [23].

- Les arbres de classification (Classification Tree) permettent de prédire à quelle classe la variable-cible appartient. Dans ce cas, la prédiction est une étiquette de classe ;
- Les arbres de régression (Regression Tree) permettent de prédire une quantité réelle (par exemple le prix d'une maison ou la durée de séjour d'un patient dans un hôpital). Dans ce cas, la prédiction est une valeur numérique.

6.3.2 Le principe de fonctionnement de l'arbre de décision

Les arbres de décision, sont une classe d'algorithmes d'apprentissage se basant sur la représentation des choix sous la forme graphique d'un arbre avec les différentes décisions de classification placées dans les feuilles. Les arbres de décision emploient une représentation hiérarchique de la structure des données sous forme des séquences de décisions (tests) en vue de la prédiction d'un résultat ou d'une classe. Chaque individu (ou observation), qui doit être attribué(e) à une classe, est décrit(e) par un ensemble de variables qui sont testées dans les nœuds de l'arbre. Les tests s'effectuent dans les nœuds internes et les décisions sont prises dans les nœuds feuilles [23].

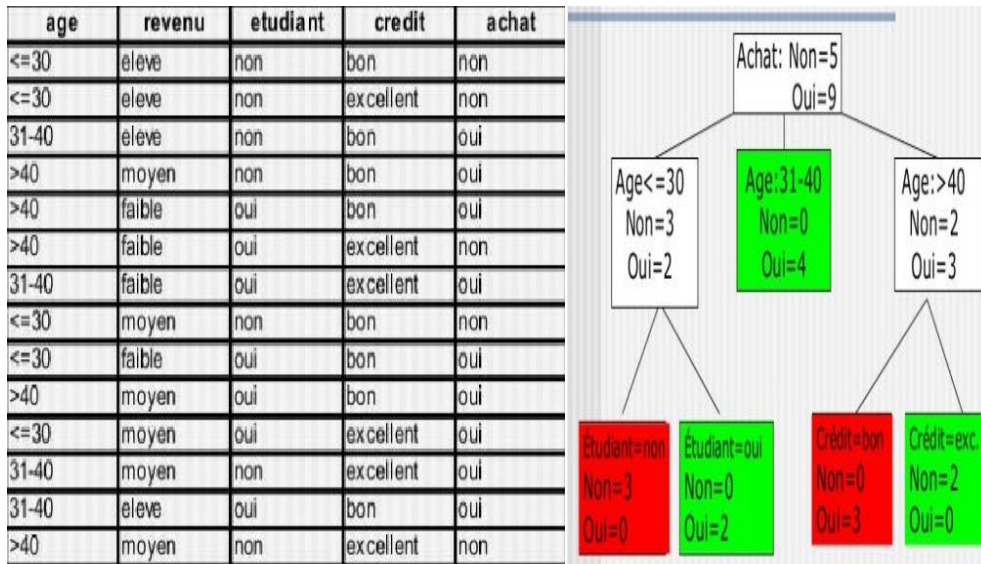


Figure 7 Exemple d'un arbre de décision de régression (à droite) issu d'un ensemble d'apprentissage (Tableau à gauche) [24].

6.4 Les Machines à vecteur de support (SVM : Support Vector Machines)

6.4.1 Définition d'un SVM

Les machines à vecteurs de support ou séparateurs à vaste marge (en anglais support vector machine) ont été développées en 1995. Il s'agit d'un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes tant de classification, notamment dans le cadre de la classification binaire, que, de régression ou de détection d'anomalies et de discrimination. Les SVM reposent sur deux notions : celle de marge maximale et celle de fonction noyau [25].

6.4.2 Le fonctionnement principal des machines à vecteurs support SVM

Pour un problème linéairement séparable, deux classes d'exemples donnés, le but de SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur de type linéaire appelé hyperplan. Dans la figure suivante, on détermine un hyperplan qui sépare les deux ensembles [25].

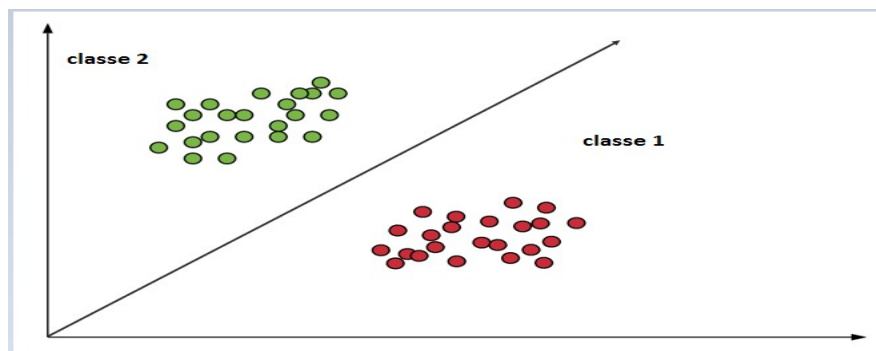


Figure 8 Exemple des données linéairement séparable [25].

Les points les plus proches, qui sont seuls utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support. Nous allons en plus chercher parmi les hyperplans valides, celui qui est optimal, dont la distance minimale aux exemples d'apprentissage est maximale. On appelle cette distance « marge » entre l'hyperplan et les exemples. Comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste marge [25].

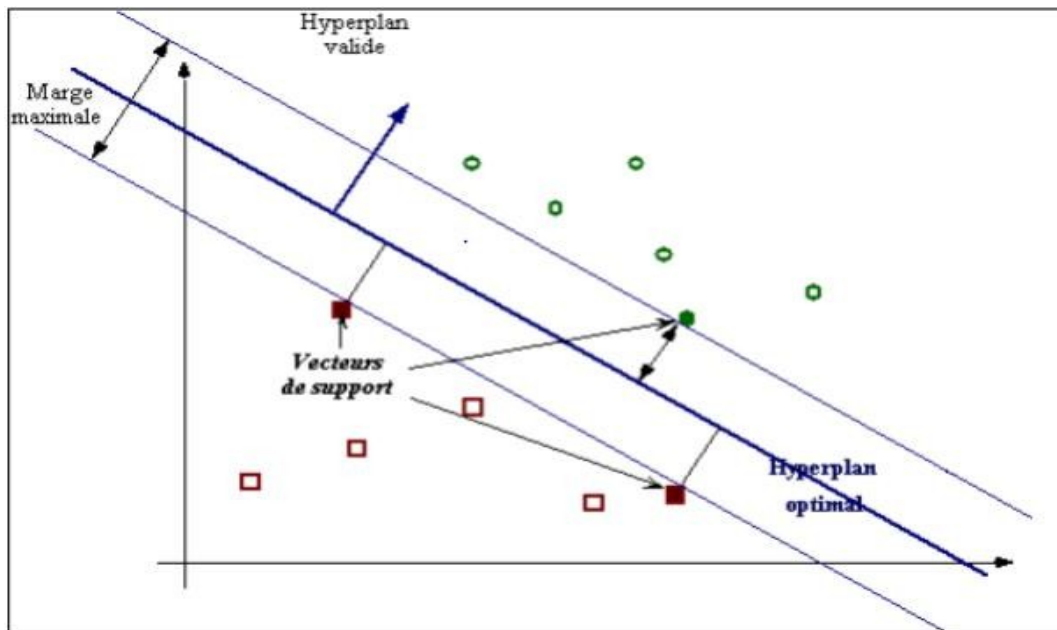


Figure 9 Le principe des SVM pour les cas linéairement séparables [25].

Dans le cas des données non linéairement séparables, l'idée des SVM est de changer l'espace des données. La transformation non linéaire des données peut permettre une séparation linéaire des exemples dans un nouvel espace. On va donc avoir un changement de dimension. Cette nouvelle dimension est appelée « espace de re-description ». En effet, intuitivement, plus la dimension de l'espace de re-description est grande, plus la probabilité de pouvoir trouver un hyperplan séparateur entre les exemples est élevée [25].

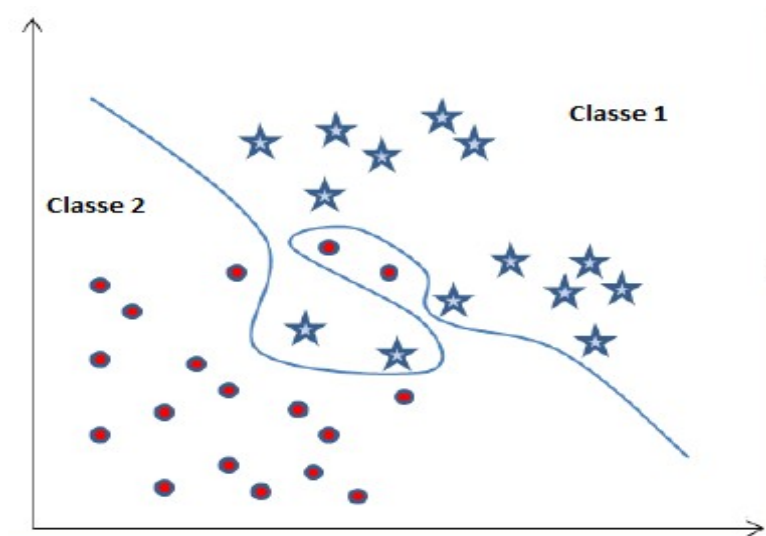


Figure 10 Exemple des données Non linéairement séparable [26].

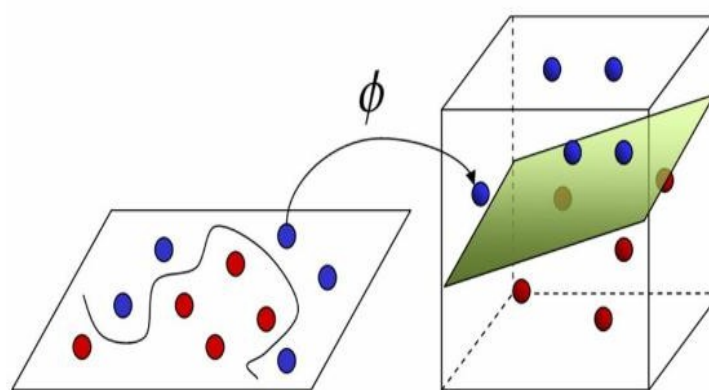


Figure 11 Projection des données dans un espace de plus grande dimension [27].

Cette transformation non linéaire est réalisée via une fonction noyau. En pratique, quelques familles des fonctions noyau paramétrables sont connues et il revient à l'utilisateur de SVM d'effectuer des tests pour déterminer celle qui convient le mieux pour son application. On peut citer les exemples de noyaux suivants : polynomiale, gaussien, sigmoïde [25].

6.5 Les réseaux de neurones artificiels (RNA)

6.5.1 Définition de réseaux de neurones artificiels

Les RNA sont des modèles mathématiques non linéaires, de type « boîte noire », capables de déterminer des relations entre données par la présentation (l'analyse) répétée d'exemples (à savoir des couples constitués par une information d'entrée et une valeur de sortie que l'on voudrait approcher par le modèle). La modélisation à l'aide de RNA appelé phase

d'apprentissage suppose l'adaptation des paramètres du réseau, afin de mettre en évidence les relations qui portent sur les exemples présentés. Les premiers modèles de réseaux de neurones ont été introduits en 1943 par les Neurologues McCulloch et Pitts. Toutefois, la technologie de l'époque ne leur a pas permis d'obtenir beaucoup de progrès. D'autres chercheurs comme Donald Hebb, qui a présenté en 1949 une série d'idées sur la structure et le fonctionnement des systèmes biologiques de neurones [28].

6.5.2 Principe fonctionnement de RNA

Un réseau de neurones artificiels est un système dont la conception est à l'origine inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques [29]. Ils sont constitués d'un ensemble d'éléments de calcul (neurones artificiels), organisés dans une structure spécifique. Les paramètres du réseau (les poids) étant représentés par les valeurs associées aux connexions de ces éléments de calcul. Un élément de calcul du RNA comporte une ou plusieurs entrées et des sorties. La valeur de sortie est obtenue par l'application d'une relation mathématique (fonction d'activation) sur la somme pondérée d'entrées. Dans la modélisation à l'aide de réseaux de neurones artificiels, on peut choisir le type de fonctions d'activation, le nombre de neurones et l'arrangement de leurs connexions (à savoir la structure du réseau).

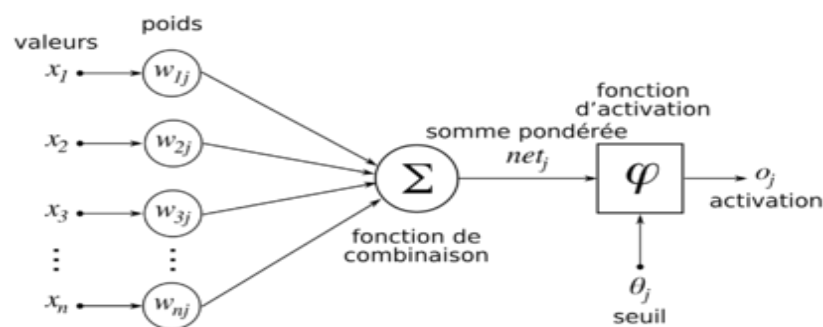


Figure 12 Structure d'un réseau de neurone artificiel [29].

Un réseau de neurones s'exprime sous forme d'un graphe composé de trois éléments [30] :

- Une architecture : Elle concerne le nombre et la disposition des neurones, le nombre de couches d'entrées de sorties et intermédiaires ainsi que les caractéristiques (pondération et direction) des arcs du réseau. Le nombre de neurones des différentes couches dépend du contexte d'application. Par ailleurs, la détermination du nombre de neurones à y associer demeure dans la plupart du temps arbitraire ;

- Une fonction de transfert : Elle traduit le niveau d'activation d'un neurone en un état donné. Le niveau d'activation d'un neurone est obtenu en cumulant l'état de l'ensemble des entrées qui agissent sur lui. Par la suite, la fonction de transfert transforme le niveau d'activation en une valeur binaire ou continue, identifiant ainsi l'état du neurone. La fonction de transfert peut être linéaire, à seuil, stochastique et le plus souvent sigmoïde ;
- Une règle d'apprentissage : C'est le processus d'ajustement des poids associés aux arcs lorsque le réseau est en situation d'apprentissage. La réduction de l'erreur entre la valeur de sortie du réseau et la valeur initiale dans l'ensemble d'apprentissage permet de déterminer les paramètres (poids) du réseau.

Nous avons présenté une brève définition des réseaux de neurones, dans le chapitre qui suit nous parlerons de ces derniers plus en détails et plus exactement des réseaux de neurones convolutifs (CNN).

7 Mesures de la qualité de classification

L'évaluation des méthodes de classification (supervisée ou non supervisée) exige des mesures indépendantes et fiables. Il s'avère qu'on trouve toujours un gap entre la théorie et la pratique. La masse de données d'une part et les détails subtils de la représentation des données et des algorithmes de classifications d'autre part rendent impossible un jugement intuitif. Il n'existe pas une mesure absolue pour l'évaluation des méthodes de classification, mais une variété des méthodes selon les caractéristiques des données ou des algorithmes, nous présenterons les mesures d'évaluation les plus fréquemment spécifiques [31].

7.1 Mesures d'évaluation des algorithmes de classifications

7.1.1 Matrice de confusion multi classe

La matrice de confusion indique le nombre de prédictions correctes pour chaque **classe** et le nombre de prédictions incorrectes pour chaque classe organisée en fonction de la classe prédite. Chaque ligne de tableau correspond à une classe prédite, et chaque colonne correspond à une classe réelle [32].

Figure 13 Exemple d'une matrice de confusion de N classes [32].

		Classes prédictives			
		Classe 1	Classe2	...	Classe n
Classes	Classe 1	X ₁₁	X ₁₂	...	X _{1n}
	Classe 2	X ₂₁	X ₂₂	...	X _{2n}
Effectives

	Classe n	X _{n1}	X _{n2}	...	X _{nn}

Dans les lignes sous les classes réelles, les prédictions ou les résultats sont inscrits. Ces résultats peuvent être l'indication correcte d'une prédiction positive comme vraie positive (true positive) et d'une prédiction négative comme vraie négative (true negative), ou une prédiction positive incorrecte comme fausse positive (False positive) et une prédiction négative incorrecte comme fausse négative (false negative).

Le nombre total de faux négatifs (TFN), et le nombre total de faux positifs (TFP) et le nombre total de vrais négatifs (TTN) pour chaque classe sera calculé sur la base des équations généralisées 3, 4 et 5, respectivement. Le total des vrais positifs dans le système sera obtenu par l'équation 6 [32].

$$TFN_i = \sum_{j \neq i}^n X_{ij} \quad (3)$$

$$TFP_i = \sum_{j \neq i}^n X_{ji} \quad (4)$$

$$TTN_i = \sum_{j \neq i}^n \sum_{k \neq i}^n X_{jk} \quad (5)$$

$$TTP_{\text{tout}} = \sum_{j=1}^n X_{jj} \quad (6)$$

7.1.2 Précision (P)

Cette mesure d'évaluation décrit le taux de réussite par rapport à une classe. Pour calculer la précision généralisée (P) on applique l'équation suivante [32] :

$$P_i = \frac{TTP_{tout}}{TTP_{tout} + TFP_i} \quad (7)$$

7.1.3 Recall (True positive rate sensitivity)

Le principe de la mesure d'évaluation Recall (R) est très simple : De toutes les classes positives, combien le modèle a prédit correctement. Il doit être aussi élevé que possible. Il est exprimé par l'équation suivante [32].

$$Ri = \frac{TTP_{tout}}{TTP_{tout} + TFNi} \quad (8)$$

7.1.4 F-Measure

Il est difficile de comparer deux modèles avec une faible précision et un Recall élevé ou vice versa. Donc, pour les rendre comparables, nous utilisons F-measure. Le F- measure permet de mesurer simultanément le Recall et la précision. Il utilise la moyenne harmonique à la place de la moyenne arithmétique en punissant davantage les valeurs extrêmes [32].

$$F\text{- Measure} = \frac{2 * \text{Recall} * \text{precision}}{\text{Recall} + \text{precision}} \quad (9)$$

7.1.5 Taux d'erreur (Accuracy)

Le principe de cette mesure est de toutes les classes, combien le modèle a prédit correctement [32].

$$ACCURACY = \frac{TP_{tout}}{N} \quad (10)$$

Tels que N= Nombre total d'entrées de test

8 Techniques d'amélioration des algorithmes de classification

8.1 Amélioration par bagging

Le Bagging est une méthode d'ensemble introduite par Breiman (1996). Le mot Bagging est la contraction des mots Bootstrap et Aggregating. Le Bootstrap est un principe de ré-échantillonnage statistique, traditionnellement utilisé pour l'estimation de grandeurs ou de propriétés statistiques. L'idée du bootstrap est d'utiliser plusieurs ensembles de données ré-échantillonnées à partir de l'ensemble des données observées et ceci à l'aide d'un tirage aléatoire avec remise. Supposons que l'on dispose d'un ensemble $T = \{x_1, x_2, \dots, x_n\}$ de N données observées de notre population et que l'on s'intéresse à une statistique notée $S(T)$. Le bootstrap va consister à former L échantillons $T^*k = \{x^*1, x^*2, \dots, x^*N\}$ $k = 1..L$, où chaque T^*k est constitué par tirage aléatoire avec remise de N' données dans T. Ces L échantillons sont usuellement appelés les échantillons bootstrap [33].

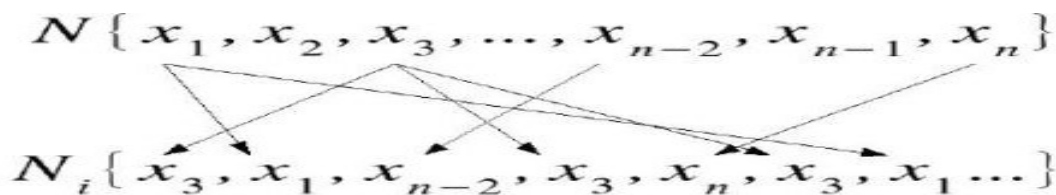


Figure 14 Illustration d'un tirage aléatoire avec remise pour la formation d'un échantillon bootstrap [33].

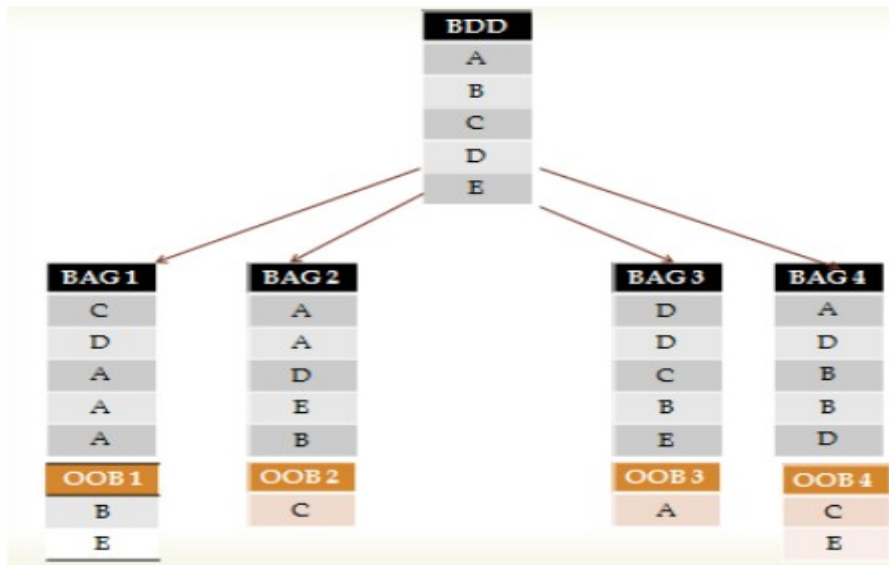


Figure 15 Des échantillons bootstrap d'une base de données [40]

D'après Breiman la statistique $S(T)$ que l'on cherche à étudier est un algorithme d'apprentissage noté $H(x)$. Il a appliqué alors le principe de bootstrap. Ainsi chaque classifieur élémentaire $h(x)$ de l'ensemble sera entraîné sur un des L échantillons bootstrap de sorte qu'ils soient tous entraînés sur un ensemble d'apprentissage différent [34].

8.2 Amélioration par boosting

Les différents classifieurs dans l'apprentissage automatique (k-pp, arbre de décision, réseaux de neurones...) sont pondérés de manière à ce qu'à chaque prédiction, les classifieurs ayant prédit correctement auront un poids plus fort que ceux dont la prédiction est incorrecte. Adaboost est un algorithme de boosting qui s'appuie sur ce principe, avec un paramètre de mise à jour adaptatif permettant de donner plus d'importance aux valeurs difficiles à prédire, donc en boostant les classifieurs qui réussissent quand d'autres ont échoué. Des variantes permettent de l'étendre à la classification multi classes. Adaboost s'appuie sur des classifieurs existants et cherchent à leur affecter les bons poids vis à vis de leurs performances [34].

- L'algorithme Adaboosting

C'est un méta-algorithme de boosting introduit par Yoav Freund et Robert Schapire. Il peut être utilisé en association avec de nombreux types d'algorithmes d'apprentissage afin d'en améliorer les performances. Les sous-classeurs utilisés peuvent être faibles tant qu'ils proposent une performance au moins un peu supérieure à celle d'un classifieur aléatoire, auquel cas il peut être prouvé que le modèle final converge vers un classifieur fort [35].

9 Conclusion

Dans ce chapitre, nous avons discuté de la notion de classification d'images numériques (définition, type...). Nous avons également évoqué la notion de features extraction (extraction de caractéristique dans l'image), puis on a abordé les algorithmes de classification, les mesures de qualité de classification. Nous avons terminé par les techniques d'amélioration des algorithmes par la suite. Nous avons conclu que les algorithmes de classification d'image sont très différents concernant le principe l'un de l'autre et dont le choix d'utilisation dépend strictement de la nature du problème à traiter. Dans le prochain chapitre nous allons nous approfondir dans les réseaux de neurones artificiels, plus précisément les réseaux de neurones convolutifs, un outil très utilisé en matière de classification de l'image numérique.

Chapitre II :

Le Deep Learning pour la classification de l'image numérique

1 Introduction :

L'intelligence artificielle est une discipline scientifique recherchant des méthodes de solution de problèmes à forte complexité logique ou algorithmique. L'apprentissage automatique champ d'étude de l'intelligence artificielle. Par conséquent, l'apprentissage profond (en Anglais deep learning, deep structured learning, hierarchical learning) est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires.

Dans ce chapitre nous allons présenter tout d'abord les notions en relation avec l'apprentissage profond. [36]

2 Définition de l'apprentissage profond (deep learning) :

L'apprentissage profond (« *deep learning* ») est un ensemble de techniques d'apprentissage automatique qui a permis des avancées importantes en intelligence artificielle dans les dernières années. Dans l'apprentissage automatique, un programme analyse un ensemble de données afin de tirer des règles qui permettront de tirer des conclusions sur de nouvelles données. L'apprentissage profond est basé sur ce qui a été appelé, par analogie, des « réseaux de neurones artificiels », composés de milliers d'unités (les « neurones ») qui effectuent chacune de petites opérations simples. Les résultats d'une première couche de « neurones » servent d'entrée aux calculs d'une deuxième couche et ainsi de suite. Par exemple, pour la reconnaissance visuelle, des premières couches d'unités identifient des lignes, des courbes, des angles... des couches supérieures identifient des formes, des combinaisons de formes, des objets, des contextes...

Les progrès de l'apprentissage profond ont été possibles notamment grâce à l'augmentation de la puissance des ordinateurs et au développement de grandes bases de données (« *big data* »).[36]

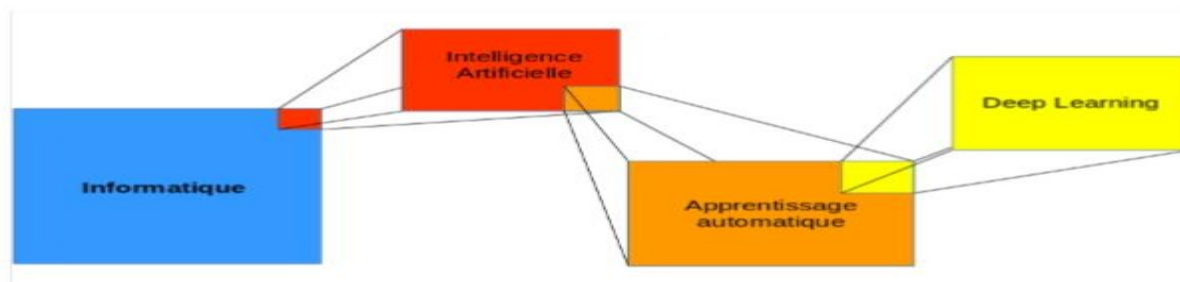


Figure 16 Schéma résumant la place de deep learning dans le monde de l'informatique

3 *Machine learning ET deep learning*

L'apprentissage machine (ML Machine Learning) est un sous-domaine de l'intelligence artificielle, qui donne à un système une capacité de compréhension grâce à ses algorithmes. Il est basé sur l'idée de faire apprendre des algorithmes à partir de données et de faire des prédictions avec ces données et par cela les ordinateurs apprennent à résoudre des tâches spécifiques, sans avoir besoin de les programmer [41]. Le Deep Learning est un sous-ensemble des méthodologies et techniques de Machine Learning qui utilisent le réseau neuronal artificiel (ANN). C'est l'adaptation des réseaux neuronaux qui imite la structure du cerveau humain. La force de DL réside dans le fait que la machine peut extraire des caractéristiques et apprendre toute seule, indépendamment de l'intervention d'un expert. Il a été appliqué dans de nombreux domaines différents (traitement d'images, textes, paroles et vidéos). Le succès de DL appartient à la disponibilité de plus de données d'entraînement. Google, Facebook et Amazon ont déjà commencé à l'utiliser pour faire l'analyse de leurs énormes quantités de données [37].

4 *Pourquoi le deep learning :*

Une des grandes différences entre le Deep Learning et les algorithmes de ML traditionnelles c'est qu'il s'adapte bien, plus la quantité de données fournie est grande plus les performances d'un algorithme de Deep Learning sont meilleures. Contrairement à plusieurs algorithmes de ML classiques qui possèdent une borne supérieure à la quantité de données qu'ils peuvent recevoir des fois appelée "plateau de performance", les modèles de Deep Learning n'ont pas de telles limitations (théoriquement) et ils sont même allés jusqu'à dépasser la performance humaine dans des domaines comme l'image processing. Autre différence entre les algorithmes de ML traditionnelles et les algorithmes de Deep Learning c'est l'étape de l'extraction de caractéristiques. Dans les algorithmes de ML traditionnelles l'extraction de caractéristiques est faite manuellement, c'est une étape difficile et coûteuse en temps et requiert un spécialiste en la matière alors qu'en Deep Learning cette étape est exécutée automatiquement par l'algorithme [37].

5 *Architectures de réseaux de neurones profonds*

Il existe un grand nombre de variantes d'architectures profondes. La plupart d'entre elles sont dérivées de certaines architectures originales. Nous allons choisir les réseaux de neurones convolutifs (CNNs).

6 Quelques explications sur les CNNs

6.1 Principe d'architecture d'un CNN

Les réseaux de neurones convolutifs sont à ce jour les modèles les plus performants pour classer des images. Désignés par l'acronyme CNN, de l'anglais Convolutional Neural Network, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a deux dimensions pour une image aux niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu].

La première partie d'un CNN est la partie convolutive à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers d'une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions (Figure 2 et 17). Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. Enfin, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN. [36]

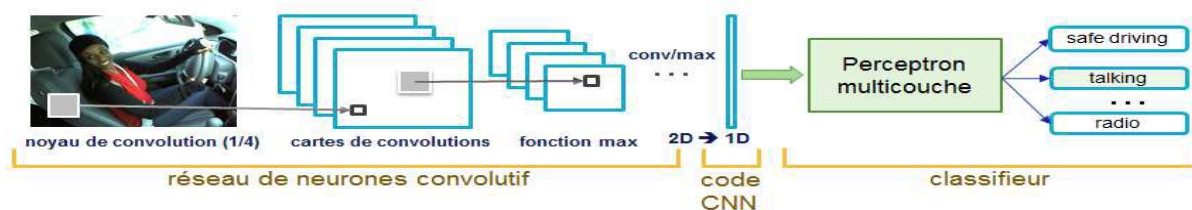


Figure 17 Les réseaux de neurones convolutifs.

Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée de couches entièrement connectées. Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image.

La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories. [36]

6.2 L'entraînement d'un nouveau CNN est difficile

Créer un nouveau réseau de neurones convolutif est coûteux en termes d'expertise, de matériel et de quantité de données annotées nécessaires.

Il s'agit d'abord de fixer l'architecture du réseau, c'est-à-dire le nombre de couches, leurs tailles et les opérations matricielles qui les connectent. L'entraînement consiste alors à optimiser les coefficients du réseau pour minimiser l'erreur de classification en sortie. Cet entraînement peut prendre plusieurs semaines pour les meilleurs CNN, avec de nombreux GPU travaillant sur des centaines de milliers d'images annotées.

Des équipes de recherche se spécialisent dans l'amélioration des CNN. Elles publient leurs innovations techniques, ainsi que le détail des réseaux entraînés sur des bases de données de références. [36]

6.3 Les blocs de construction :

Une architecture CNN est formée par un empilement de couches de traitement indépendantes:

6.3.1 La couche de convolution (CONV) :

Trois hyper paramètres permettent de dimensionner le volume de la couche de convolution (aussi appelé volume de sortie) : la 'profondeur', le 'pas' et la 'marge'.

1. 'Profondeur' de la couche : nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).
2. 'Le pas' contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.
3. 'La marge (à 0)' ou 'zero padding' : parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. La taille de ce 'zero-padding' est le troisième hyperparamètre. Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d'entrée [38].

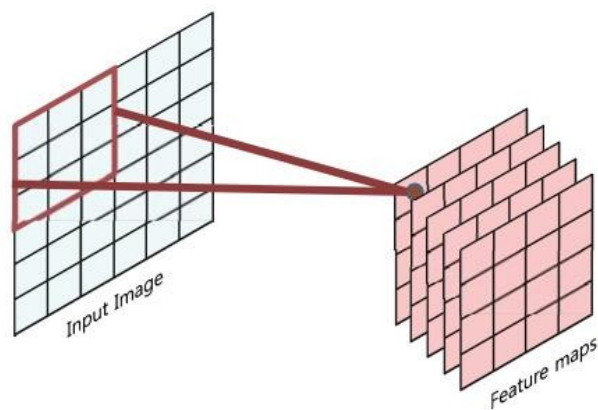


Figure 18 Ensemble de neurones (cercles) créant la profondeur d'une couche de convolution [37]

6.3.2 Couche de pooling (POOL) :

Un autre concept important des CNNs est le pooling (« mise en commun »), ce qui est une forme de sous-échantillonnage de l'image. L'image d'entrée est découpée en une série de rectangles de n pixels de côté ne se chevauchant pas (pooling). Chaque rectangle peut être vu comme une tuile. Le signal en sortie de tuile est défini en fonction des valeurs prises par les différents pixels de la tuile.

Le pooling réduit la taille spatiale d'une image intermédiaire, réduisant ainsi la quantité de paramètres et de calcul dans le réseau. Il est donc fréquent d'insérer périodiquement une couche de pooling entre deux couches convolutives successives d'une architecture CNN pour contrôler l'overfitting (sur-apprentissage). L'opération de pooling crée aussi une forme d'invariance par translation.

La couche de pooling fonctionne indépendamment sur chaque tranche de profondeur de l'entrée et la redimensionne uniquement au niveau de la surface. La forme la plus courante est une couche de mise en commun avec des tuiles de taille 2×2 (largeur/hauteur) et comme valeur de sortie la valeur maximale en entrée (cf. schéma). On parle dans ce cas de « Max-Pool 2×2 » (compression d'un facteur 4).

Il est possible d'utiliser d'autres fonctions de pooling que le maximum. On peut utiliser un «average pooling» (la sortie est la moyenne des valeurs du patch d'entrée), du «L2-norm pooling». Dans les faits, même si initialement l'average pooling était souvent utilisé il s'est avéré que le max-pooling était plus efficace car celui-ci augmente plus significativement l'importance des activations fortes. En d'autres circonstances, on pourra utiliser un pooling stochastique (voir « Méthodes de régularisation » plus bas dans ce document).

Le pooling permet de gros gains en puissance de calcul. Cependant, en raison de la réduction agressive de la taille de la représentation (et donc de la perte d'information associée), la tendance actuelle est d'utiliser de petits filtres [en] (type 2x2) (figure (2.3)). Il est aussi possible d'éviter la couche de pooling [en] mais cela implique un risque sur-apprentissage plus important. [39]

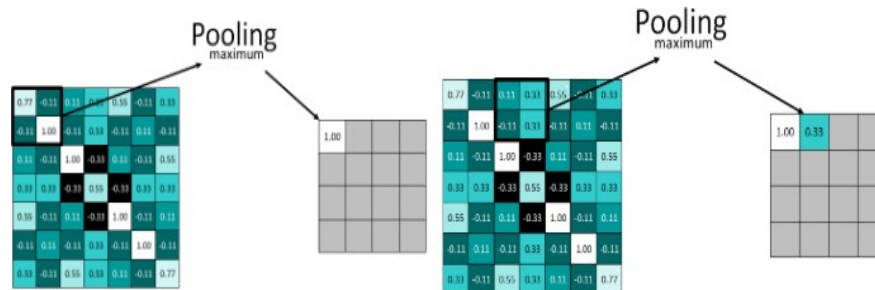


Figure 19 Exemple couche de pooling [39]

6.3.3 Couche de correction (ReLU):

Souvent, il est possible d'améliorer l'efficacité du traitement en intercalant entre les couches de traitement une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie. On a notamment :

- La correction ReLU (abréviation de Unités Rectifié linéaires) : $f(x) = \max(0, x)$.

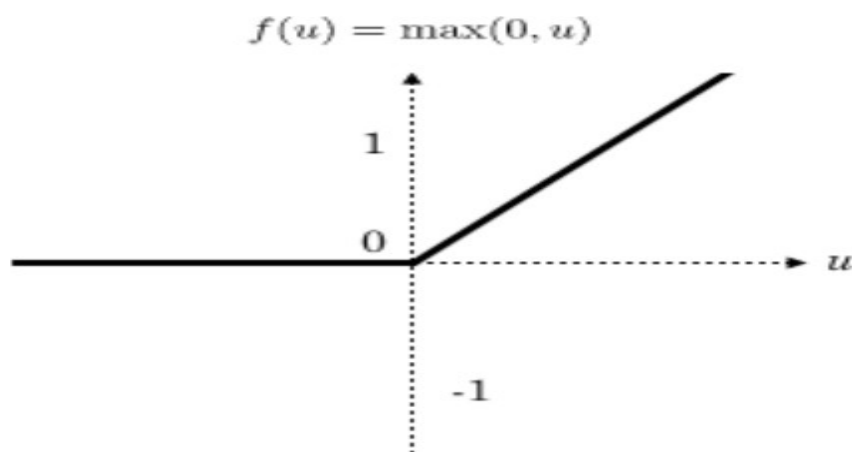


Figure 20 Tracé de la fonction ReLU [39]

Cette Fonction, appelée aussi « fonction d'activation non saturante », augmente les propriétés non linéaires de la fonction de décision et de l'ensemble du réseau sans affecter les champs récepteurs de la couche de convolution.

- La correction par tangente hyperbolique $f(x)=\tanh(x)$.
- La correction par la tangente hyperbolique saturante : $f(x)=|\tanh(x)|$.
- La correction par la fonction sigmoïde.

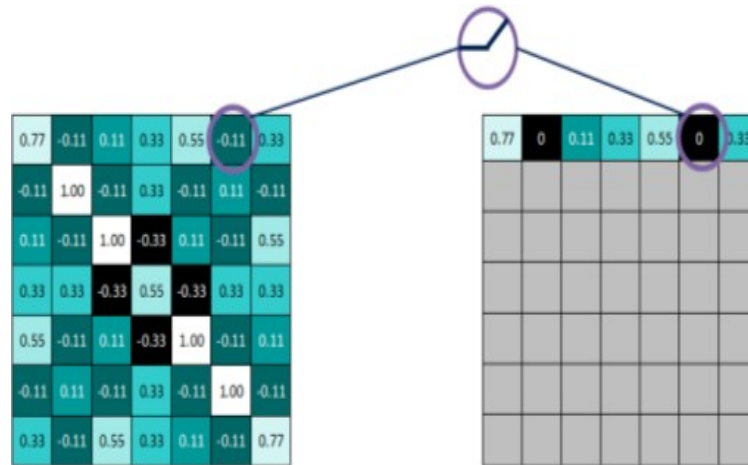


Figure 21 La taille de l'image après le max pooling [39]

Souvent, la correction Relu est préférable, car il en résulte la formation de réseau neuronal plusieurs fois plus rapide, sans faire une différence significative à la généralisation de précision.[39]

6.3.4 Le flattening :

Le flattening ou mise à plat consiste simplement à mettre bout à bout toutes les images (matrices) que nous avons pour en faire un (long)vecteur. Les pixels (en réalité ce ne sont plus des images ou des pixels, mais des matrices de nombres, donc les pixels sont ces nombres) sont récupérés ligne par ligne et ajoutés au vecteur final



Figure 22 Exemple de flattening [37]

6.3.5 La couche entièrement connectée :

Après plusieurs couches de convolution et de max-pooling, 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. Elle prend en entrée un vecteur, applique une combinaison linéaire, éventuellement une fonction d'activation aux valeurs reçues en entrée et renvoie le vecteur de taille N . 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 [37].

7 Optimisation pour l'apprentissage en Deep Learning

7.1 Les variantes de la descente du gradient

Il existe trois principaux types de variantes de l'algorithme de descente de gradient. La principale différence entre eux est la quantité de données que nous utilisons lorsque nous calculons le gradient pour chaque étape d'apprentissage. Les algorithmes d'optimisation qui utilisent tout l'ensemble de l'apprentissage sont appelés les méthodes de Deep Learning gradients déterministes ou batch descente de gradient, car ce sont des méthodes où tous les exemples d'apprentissage sont traités simultanément dans un grand batch. Le terme "batch" est utilisé ici pour décrire un groupe d'exemples. Les algorithmes d'optimisation qui n'utilisent qu'un seul exemple à la fois sont parfois appelés méthodes stochastiques ou online descente de gradient en anglais. La plupart des algorithmes utilisés pour l'apprentissage profond se situent quelque part entre les deux, utilisant plus d'un mais moins que tous les exemples d'entraînement. Ils sont appelés les méthodes stochastiques de mini batch ou de mini batch descente de gradient [37].

7.2 Algorithmes d'optimisation de la descente du gradient

7.2.1 Adagrad

Adagrad est un algorithme d'optimisation basé sur la descente de gradient qui ne fait qu'adapter le taux d'apprentissage aux paramètres, effectuant de plus grandes mises à jour pour les caractéristiques peu fréquentes et de plus petites pour les caractéristiques plus fréquentes. Un des avantages de Adagrad est qu'il élimine le besoin de modifier manuellement le taux d'apprentissage, vu que Adagrad modifie automatiquement le taux d'apprentissage en se basant sur la somme des carrés de gradient. L'inconvénient majeur de Adagrad est l'accumulation des carrés des gradients dans le dénominateur. Comme chaque terme ajouté est positif, l'accumulation des sommes continue d'augmenter au cours de l'apprentissage ce qui va baisser le taux d'apprentissage de façon considérable [37].

7.2.2 Adam

Adaptive Moments optimizer (ADAM) est une autre méthode qui calcule un taux d'apprentissage adaptatif pour chaque paramètre. En plus de stocker une moyenne décroissante exponentielle des précédents carrés des gradients (v_t) comme RMSprop, ADAM conserve aussi la moyenne décroissante exponentielle des précédents gradients (m_t) [42].

7.2.3 RMSprop

RMSprop est une méthode de taux d'apprentissage adaptative. Au lieu d'accumuler tous les carrés des gradients précédents, on restreint la fenêtre des gradients accumulés à une taille fixée w . Au lieu de stocker les w carrés des gradients précédents, on applique une moyenne mobile exponentielle des carrés des gradients précédents. La moyenne courante à l'étape (t) dépend uniquement de la moyenne précédente et du gradient courant [37].

8 Les applications du Deep-Learning

Le deep learning est utilisé dans des différents domaines tels que [37] :

8.1 La reconnaissance faciale

Les yeux, le nez, la bouche, tout autant de caractéristiques qu'un algorithme de DL va apprendre à détecter sur une photo. Il va s'agir en premier lieu de donner un certain nombre d'images à l'algorithme, puis à force d'entraînement, l'algorithme va être en mesure de détecter un visage sur une image.

8.2 Le traitement automatique de langage naturel

Le traitement automatique de langage naturel est une autre application du DL. Son but étant d'extraire le sens des mots, voire des phrases pour faire de l'analyse de sentiments. L'algorithme va par exemple comprendre ce qui est dit dans un avis Google, ou va communiquer avec des personnes via des chatbots. La lecture et l'analyse automatique de textes est aussi un des champs d'application du DL avec le Topic Modeling tel texte aborde tel sujet.

8.3 Voitures autonomes

Les entreprises qui construisent de tels types de services d'aide à la conduite, doivent apprendre à un ordinateur à maîtriser certaines parties essentielles de la conduite à l'aide de systèmes de capteurs numériques au lieu de l'esprit humain. Pour ce faire, les entreprises commencent généralement par entraîner des algorithmes de DL utilisant une grande quantité de données. Vous pouvez imaginer comment un enfant apprend grâce à des expériences constantes et à la réplication.

8.4 Recherche vocale et assistants à commande vocale

L'un des domaines d'utilisation les plus populaires de DL est la recherche vocale et les assistants intelligents à commande vocale. Avec les grands géants de la technologie ont déjà fait d'importants investissements dans ce domaine, des assistants à commande vocale peuvent être trouvés sur presque tous les Smartphones. Le Siri d'Apple est sur le marché depuis octobre 2011. Google aujourd'hui, l'assistant à commande vocale pour Android, a été lancé moins d'une année après Siri. Le plus récent des assistants intelligents à commande vocale est Microsoft Cortana.

8.5 Ajout automatique de sons à des films muets

Dans cette tâche, le système doit synthétiser des sons pour correspondre à une vidéo silencieuse. Le système est formé à l'aide de 1 000 exemples de vidéos avec le son d'une baguette frappant différentes surfaces et créant différents sons. Un modèle DL associe les images vidéo à une base de données de sons pré-enregistrés afin de sélectionner le son à reproduire qui correspond le mieux à ce qui se passe dans la scène. Le système a ensuite été évalué à l'aide d'un test de contrôle, comme une configuration dans laquelle les humains devaient déterminer quelle vidéo comportait le son réel ou le son factice (synthétisé). Ceci utilise à la fois les réseaux de neurones convolutionnels et les réseaux de neurones récurrents à mémoire à court terme.

8.6 Traduction automatique

Il s'agit d'une tâche dans laquelle des mots, expressions ou phrases donnés dans une langue sont automatiquement traduits dans une autre langue. La traduction automatique existe depuis longtemps, mais DL permet d'obtenir les meilleurs résultats dans deux domaines spécifiques :

- Traduction automatique de texte ;
- Traduction automatique d'images.

La traduction de texte peut être effectuée sans aucun traitement préalable de la séquence, ce qui permet à l'algorithme d'apprendre les dépendances entre les mots et leur correspondance avec une nouvelle langue.

8.7 Génération automatique de texte

C'est une tâche intéressante, où un corpus de texte est appris et à partir de ce modèle, un nouveau texte est généré, mot par mot ou caractère par caractère. Le modèle est capable d'apprendre comment épeler, ponctuer, Deep Learning former des phrases et même capturer le style du texte dans le corpus. Les grands réseaux de neurones récurrents sont utilisés pour apprendre la relation entre les éléments dans les séquences de chaînes d'entrée, puis pour générer du texte.

8.8 Compréhension de l'Image (Image Understanding)

Un autre domaine populaire en matière de DL est la compréhension de l'image. Son objectif est de reconnaître et d'identifier les personnes et les objets dans les images, ainsi que de comprendre le contenu et le contexte. La reconnaissance d'image est déjà utilisée dans plusieurs secteurs tels que les jeux, les médias sociaux, la vente au détail, le tourisme, etc. Cette tâche nécessite la classification des objets d'une photo parmi un ensemble d'objets connus auparavant. Une variante plus complexe de cette tâche, appelée détection d'objet, consiste à identifier spécifiquement un ou plusieurs objets dans la scène de la photo et à dessiner un cadre autour d'eux.

9 La description automatique d'image (Image Automatique annotation)

Le sous-titrage automatique des images est la tâche pour laquelle le système doit générer une

légende décrivant le contenu de l'image. Une fois que vous pouvez détecter des objets sur des photographies et générer des étiquettes pour ces objets, vous pouvez voir que l'étape suivante consiste à transformer ces étiquettes en description de phrase cohérente. Généralement, les systèmes impliquent l'utilisation de très grands réseaux de neurones convolutifs pour la détection d'objets sur les photographies, puis d'un réseau de neurones récurrent comme une mémoire à court terme à long terme pour transformer les étiquettes en une phrase cohérente.

10 La classification d'image par les réseaux de neurones convolutionnels

Les réseaux de neurones convolutionnels sont à ce jour les modèles les plus performants pour classer des images. Désignés par l'acronyme CNN, de l'anglais Convolutional Neural Network, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a 2 dimensions pour une image en niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur pour représenter les couleurs fondamentales [Rouge, Vert, Bleu]. La première partie d'un CNN est la partie convolutive à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN [37].

11 Les problèmes liés à l'apprentissage

11.1 Sur-apprentissage et sous-apprentissage

Le but d'un algorithme d'apprentissage automatique est de capter la tendance d'un échantillon et les relations entre la variable réponse et les covariables. Le sur-apprentissage (overfitting) désigne le fait que le modèle prédictif produit par l'algorithme d'apprentissage automatique s'adapte trop bien aux données de l'échantillon d'apprentissage. Par conséquent, le modèle prédictif capturera tous les détails qui caractérisent les données d'apprentissage. Dans ce sens, il capturera toutes les fluctuations et variations aléatoires des données d'apprentissage. En d'autres termes, le modèle prédictif capturera non seulement les corrélations généralisables mais aussi les bruits propres aux données d'apprentissage. On dit que la fonction prédictive se généralise mal. De la même manière, lorsque l'on n'arrive pas à capter les relations entre la variable réponse et les covariables, le modèle ne sera pas très performant. Cela sous-entend que

le modèle prédictif généré lors de la phase d'apprentissage, s'adapte mal aux données d'apprentissage. Dans ce cas on parle alors de sous-apprentissage (underfitting). Le sur-apprentissage et le sous-apprentissage

Sont les causes principales des mauvaises performances des modèles prédictifs générés par les algorithmes d'apprentissage automatique. Le but, dans la mise en place de n'importe quel algorithme, est de trouver le meilleur équilibre entre sous-apprentissage et sur-apprentissage. En d'autres termes, il ne souffre ni d'un grand biais ni d'une grande variance [43].

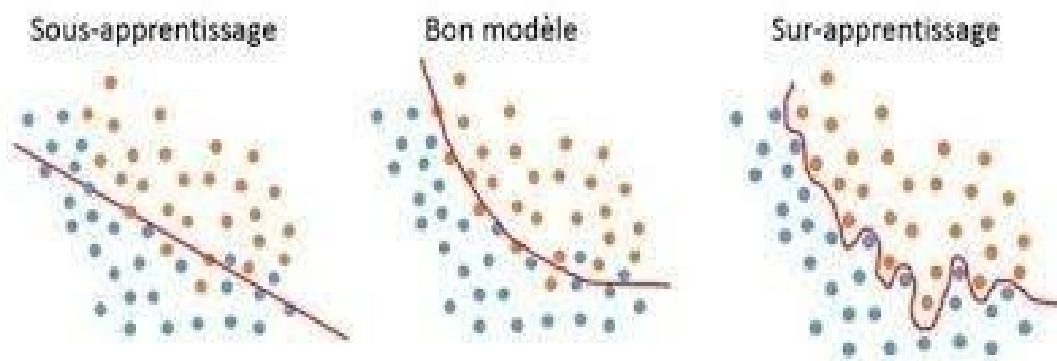


Figure 23 Exemple de sur apprentissage et de sous apprentissage [37].

11.2 Le problème de descente et l'explosion de gradients (gradients Vanishing et exploding)

Le problème d'explosion de gradients est une difficulté qui peut survenir lors de la formation de réseaux de neurones artificiels en utilisant la descente de gradient par rétro propagation. Le changement de poids du modèle peut créer un réseau instable. Les valeurs des poids peuvent devenir si grandes et provoquer un débordement. Un gradient est la direction et l'amplitude calculées pendant l'entraînement d'un réseau de neurones, il est utilisé pour enseigner les poids du réseau dans la bonne direction par la bonne quantité. Lorsqu'il y a un gradient d'erreur, l'explosion des composants peut croître de façon exponentielle [37].

12 Les solutions :

12.1 La solution du « Early-Stop »

Dans l'apprentissage automatique, le Early-Stop est une forme de régularisation utilisée pour éviter le sur-apprentissage lors de la formation d'un apprenant avec une méthode itérative, comme la descente de gradient. Ces méthodes mettent à jour l'apprenant afin de mieux l'adapter aux données de formation à chaque itération. Jusqu'à un certain point, cela améliore les

performances de l'apprenant sur les données en dehors de l'ensemble de formation. Au-delà de ce point, cependant, l'amélioration de l'adéquation de l'apprenant aux données de formation se fait au détriment de l'erreur de généralisation accrue. Les règles d'Early-Stop fournissent des indications sur le nombre d'itérations pouvant être exécutées avant que l'apprenant ne commence à sur-ajuster. Des règles d'Early-Stop ont été utilisées dans de nombreuses méthodes d'apprentissage automatique différentes, avec des quantités variables de fondements théoriques [37].

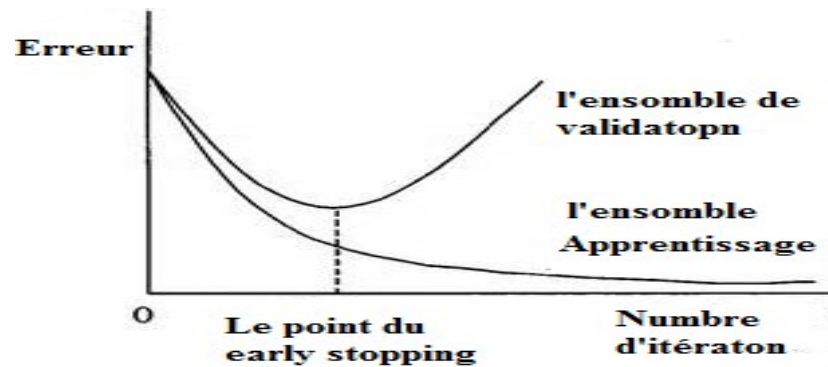


Figure 24 Un exemple du point du early stopping.

12.2 Drop-Out

Les couches "FC" (Fully Connected) occupent la majeure partie de la mémoire du CNN. D'ailleurs le concept de FC crée un problème exponentiel de mémoire appelé "overfitting" ("surconnexion" conduisant au sur-apprentissage) ralentissant le traitement de l'information. Pour prévenir cela, la méthode du dropout est utilisée pour "éteindre" les neurones aléatoirement (avec une probabilité prédéfinie, souvent un neurone sur deux) ainsi que les neurones périphériques. Ainsi, avec moins de neurones, le réseau est plus réactif et peut donc apprendre plus rapidement. À la fin de la séance d'apprentissage, les neurones "éteints" sont "rallumés" (avec leurs poids Originaux). Plus la couche FC est proche de l'image source, moins on éteindra de neurones. L'objectif est d'éteindre et rallumer les neurones aléatoirement, dans le cadre d'entraînements successifs. Une fois les séries d'entraînements terminées, on rallume tous les neurones et on utilise le réseau. Cette technique a montré non seulement un gain dans la vitesse d'apprentissage, mais en déconnectant les neurones, on a aussi limité des effets marginaux, rendant le réseau plus robuste et capable de mieux généraliser les concepts appris [37].

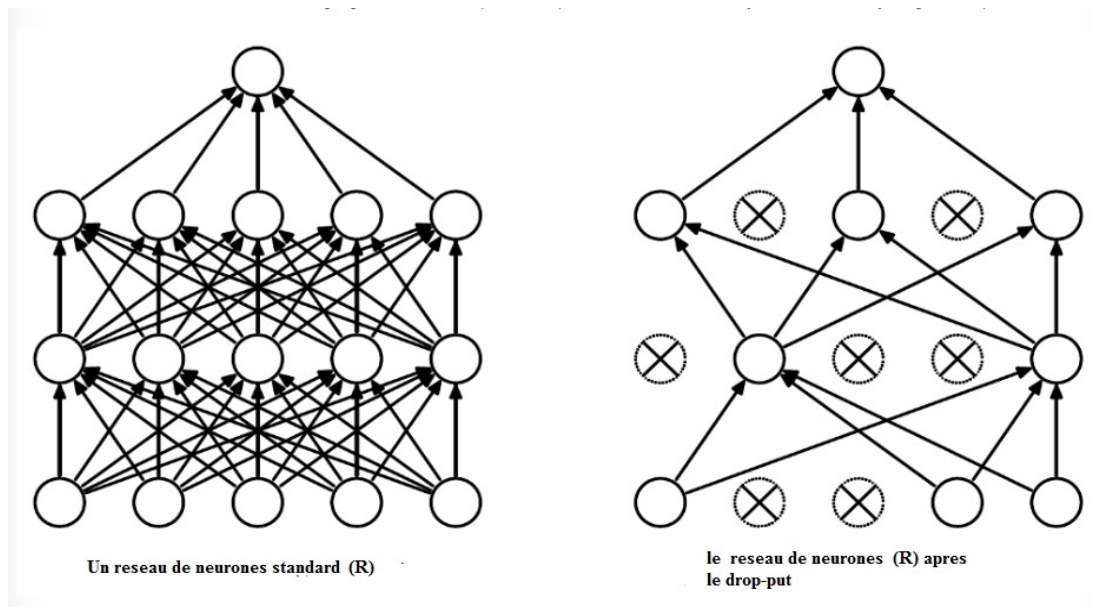


Figure 25 Un réseau de neurones avant et après l'application de la méthode « drop-out » [37].

13 Conclusion :

Dans ce chapitre, nous avons passé en revue l'essentiel pour notre travail de Master en ce qui concerne le Deep Learning, les CNNs et les outils qui vont avec, dans le contexte de la classification d'images numériques. Nous avons opté particulièrement pour les CNNs, vus l'intérêt, l'attractivité et les résultats rapportés par les chercheurs dans différents domaines se basant sur ces outils récents et pertinents.

Chapitre III :

Langages et outils

1. Introduction

L'objectif de ce chapitre est de présenter les outils de l'implémentation de l'approche proposée dans le cadre d'un système de classification d'images par les réseaux de neurones convolutionnels. Nous commençons tout d'abord par la présentation des langages et de l'environnement de développement que nous avons utilisés. Puis une description de la base qu'on a exploitée.

2. *Langages et outils*

2.1 *Le langage de programmation Python*

Python est un langage de programmation interprété, orienté objet et de haut niveau avec une sémantique dynamique développé par Guido van Rossum. Il a été initialement publié en 1991. Conçu pour être facile et amusant, le nom "Python" est un clin d'œil au groupe de comédie britannique Monty Python. Python a la réputation d'être un langage convivial pour les débutants, remplaçant Java en tant que langage d'introduction le plus largement utilisé, car il gère une grande partie de la complexité pour l'utilisateur, permettant aux débutants de se concentrer sur la compréhension complète des concepts de programmation plutôt que sur les moindres détails.

Python est utilisé pour le développement Web côté serveur, le développement de logiciels, les mathématiques et les scripts système, et est populaire pour le développement rapide d'applications et comme langage de script ou de collage pour lier les composants existants en raison de ses structures de données intégrées de haut niveau, typage dynamique et liaison dynamique. Les coûts de maintenance du programme sont réduits avec Python en raison de la syntaxe facile à apprendre et de l'accent mis sur la lisibilité. De plus, la prise en charge des modules et des packages par Python facilite les programmes modulaires et la réutilisation du code. Python est un langage communautaire open source, de sorte que de nombreux programmeurs indépendants créent continuellement des bibliothèques et des fonctionnalités pour celui-ci.[46]

2.1.1 *Cas d'utilisation Python*

- Création d'applications Web sur un serveur
- Création de flux de travail pouvant être utilisés conjointement avec des logiciels

- Connexion aux systèmes de base de données
- Lecture et modification de fichiers
- Effectuer des mathématiques complexes
- Traitement de mégadonnées
- Prototypage rapide
- Développement de logiciels prêts pour la production

Professionnellement, Python est idéal pour le développement Web backend, l'analyse de données, l'intelligence artificielle et le calcul scientifique. Les développeurs utilisent également Python pour créer des outils de productivité, des jeux et des applications de bureau.[46]

2.1.2 Fonctionnalités et avantages de Python

- Compatible avec une variété de plates-formes, y compris Windows, Mac, Linux, Raspberry Pi et autres
- Utilise une syntaxe simple comparable à la langue anglaise qui permet aux développeurs d'utiliser moins de lignes que les autres langages de programmation
- Fonctionne sur un système d'interpréteur qui permet au code d'être exécuté immédiatement, accélérant le prototypage
- Peut être manipulé de manière procédurale, orientée objet ou fonctionnelle[46]

2.1.3 Syntaxe Python

- Quelque peu similaire à la langue anglaise, avec une influence mathématique, Python est conçu pour la lisibilité
- Contrairement à d'autres langages qui utilisent des points-virgules et/ou des parenthèses pour compléter une commande, Python utilise de nouvelles lignes pour la même fonction
- Définit la portée (c'est-à-dire les boucles, les fonctions, les classes) en s'appuyant sur l'indentation, en utilisant des espaces plutôt que des accolades[46]

2.1.4 Flexibilité Python

Python, un langage typé dynamiquement, est particulièrement flexible, éliminant les règles strictes pour la création de fonctionnalités et offrant une plus grande flexibilité de résolution de problèmes avec une variété de méthodes. Il permet également aux utilisateurs de compiler et d'exécuter des

programmes jusqu'à une zone problématique car il utilise la vérification de type au moment de l'exécution plutôt que la vérification au moment de la compilation. [46]

2.1.5 Les parties les moins intéressantes de Python

En revanche, Python n'est pas facile à entretenir. Une commande peut avoir plusieurs significations selon le contexte, car Python est un langage à typage dynamique. Et, maintenir une application Python à mesure qu'elle grandit en taille et en complexité peut être de plus en plus difficile, en particulier pour trouver et corriger les erreurs. Les utilisateurs auront besoin d'expérience pour concevoir du code ou écrire des tests unitaires qui facilitent la maintenance.

La vitesse est une autre faiblesse de Python. Sa flexibilité, car elle est typée dynamiquement, nécessite une quantité importante de référencement pour atterrir sur une définition correcte, ralentissant les performances. Cela peut être atténué en utilisant une implémentation alternative de Python (par exemple, PyPy). [46]

2.1.6 Python et IA

Les chercheurs en IA sont fans de Python. Google TensorFlow, ainsi que d'autres bibliothèques (scikit-learn, Keras), établissent une base pour le développement de l'IA en raison de la convivialité et de la flexibilité qu'il offre aux utilisateurs de Python. Ces bibliothèques, et leur disponibilité, sont essentielles car elles permettent aux développeurs de se concentrer sur la croissance et la construction. [46]

2.1.7 Les bibliothèques Deep learning de Python utilisées

Dans le cadre de ce projet, nous avons choisi un environnement python 3.7.6 qui offre un large éventail de bibliothèques incontournables pour implémenter des algorithmes d'apprentissage automatique : La bibliothèque keras qui a l'avantage d'être spécialisée dans les réseaux de neurones. Parmi la large gamme de bibliothèques disponibles, nous avons utilisé massivement **Numpy**, **ScikitLearn** et **Matplotlib**.

2.2 Environnement de développement : Google Colab

L'environnement de développement de notre projet est l'outil de Cloud fourni par Google «Google Colab».

2.2.1 Présentation générale de La plateforme Google Colab

L'outil que nous choisissons pour la réalisation de notre modèle est Colaboratory, ou "Colab" pour faire court, est un produit de Google Research, un service en nuage (cloud) gratuit qui prend désormais en charge les GPU gratuits, permettant d'améliorer les compétences de codage en langage de programmation Python, de développer des applications de deep-learning à l'aide de bibliothèques populaires, telles que Keras, TensorFlow, PyTorch et OpenCV.[47]

2.2.2 Fonctionnement général de Google-Colab

Google-colab est une nouvelle technologie qui permet d'entraîner des modèles de Machine Learning directement dans le cloud, sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur. Plus techniquement, Colab est un service de notebook Jupyter hébergé qui ne nécessite aucune configuration à utiliser. Jupyter étant le projet open source sur lequel Colab est basé. Colab vous permet d'utiliser et de partager des blocs-notes Jupyter avec d'autres sans avoir à télécharger, installer ou exécuter quoi que ce soit, tout en offrant un accès gratuit aux ressources informatiques, y compris les GPU. Le code est exécuté dans une machine virtuelle privée du compte que nous avons ouvert. Les machines virtuelles sont supprimées lorsqu'elles sont inactives pendant un certain temps et ont une durée de vie maximale appliquée par le service Colab. [47]

2.2.3 Tensor Processing Unit (TPU)

Les TPU (Tensor-Processing-Units) sont des circuits intégrés spécifiques aux applications (Application-Specific Integrated Circuit ou ASIC), développés spécifiquement par Google. Ils ont récemment été ajoutés au portefeuille Google Colab. Ils permettant d'accélérer les charges de travail de machine learning. Leur conception repose sur la vaste expérience de Google et sur son leadership en matière de machine learning, TPU est conçu pour offrir des performances et une flexibilité maximale afin d'aider les chercheurs, les développeurs et les entreprises.[48]

2.2.4 Les avantages de l'environnement Google-colab

l'environnement Google-colab et les TPU Google-colab et les TPU sont tous gratuits ils sont très utilisés dans le domaine du machine learning. Les TPU permettent d'atteindre une précision plus rapidement lors de l'entraînement de modèles de réseaux de neurones vastes et complexes. Ces services du cloud donnent une non-conformité avec les appareils et leurs capacités et limitées.[47]

3. Le DataSet d'images

Le réseau a été évalué sur un ensemble de données de 28,167 images de visage avec une grande variation dans l'arrière-plan, la couleur de la peau, la pose de la tête. Le jeu de données est divisé en deux parties, chaque partie elle-même divisée en deux catégories, la première partie est consacrée à la détection de voile, elle est constituée de 1000 images, 500 avec voile et 500 sans. La deuxième partie quant à elle est consacrée à la détection de sexe, constituée d'environ 17000 images d'hommes et 10000 images de femmes.

La raison pour laquelle le nombre d'images choisi pour la détection de voile est bien moins élevé par rapport au nombre d'images pour la détection de genre est le fait que la détection de voile est bien plus basique est a besoin de beaucoup moins de données pour apprendre à faire la différence comparativement à la détection de genre qui est beaucoup plus complexe et requiert le maximum de données possible pour avoir un degré acceptable de précision.

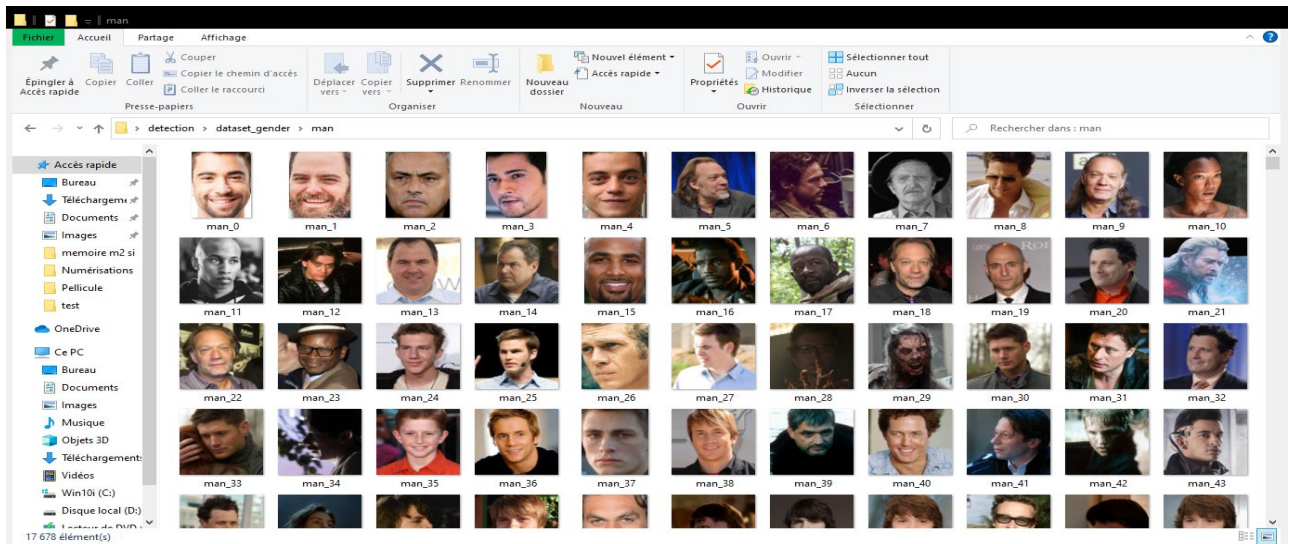


Figure 26 man dataset (17678) picture

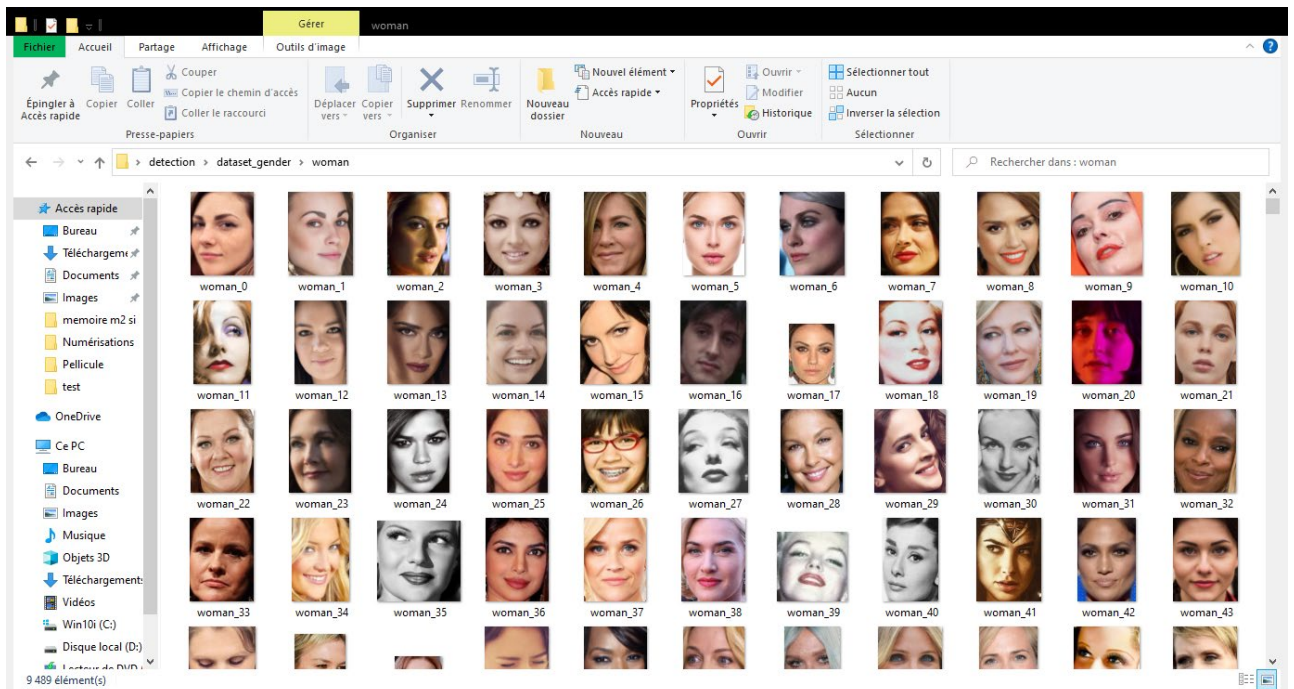


Figure 27 Woman Dataset (9849) picture

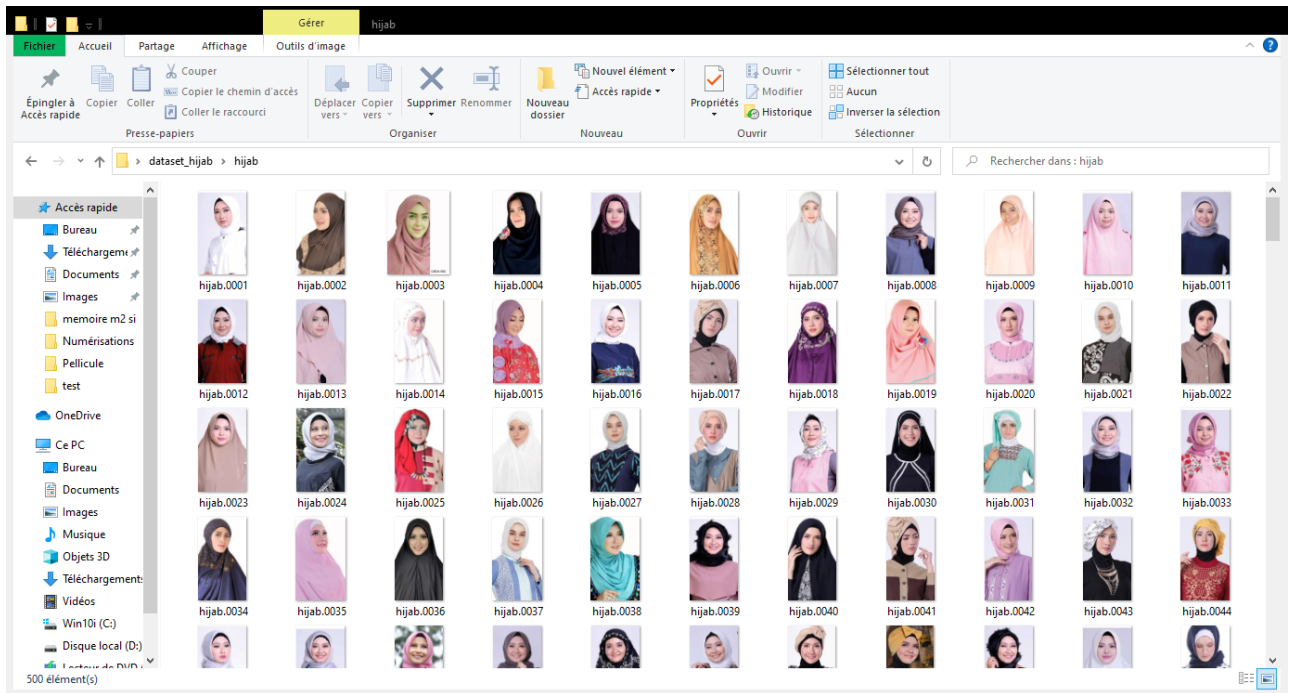


Figure 28 Woman with Hijab Dataset (500) picture

4. Conclusion

Nous avons présenté dans ce chapitre les outils, les langages et l'environnement de développement que nous avons exploités pour la réalisation de notre modèle, ainsi que la base d'images. Dans le prochain chapitre nous allons présenter notre modèle de classification basé sur les réseaux de neurones à convolutions Les (CNN).

Chapitre IV :

Proposition d'un Classificateur du genre humain d'images basé Deep Learning

1 Introduction

Nous allons présenter dans cette partie notre modèle de CNN pour la classification d'images du genre humain et les résultats.

2 Description des étapes du processus :

Le processus de construction de notre modèle est divisé en plusieurs parties à savoir :

2.1 Chargement des bibliothèques nécessaires :

La majeure bibliothèque qu'on utilise est « tensorflow » qui nous offre les fondations nécessaires pour créer, entraîner et utiliser nos propres modèles convolutionnels, les autres bibliothèques utilisées ne sont pas aussi importantes ou indispensables mais sont là pour faciliter quelques tâches ou améliorer la qualité du travail présenté (matplotlib par exemple nous permet de facilement créer des graphes pour analyser la précision des modèles)

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import BatchNormalization, Conv2D, MaxPooling2D, Activation, Flatten, Dropout, Dense
from tensorflow.keras import backend as K
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.layers import AveragePooling2D
from tensorflow.keras.layers import Input
from tensorflow.keras.models import Model
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.models import load_model
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import numpy as np
import random
import cv2
import os
import glob
import random
from sklearn.preprocessing import LabelBinarizer
from sklearn.metrics import classification_report
```

Figure 29 bibliothèques utilisées

2.2 Préparation des données

Avant la création du modèle nous devons d'abord charger les données nécessaires et les préparer pour traitement, nous commençons d'abord par obtenir les images que nous allons utiliser et les séparer en deux parties : hommes et femmes, les images ont auparavant été étiquetées selon leurs catégories respectives pour faciliter l'étape de triage, tout ce qu'il faut faire maintenant est de parcourir la liste d'images et lire le nom du fichier et attacher l'étiquette "1" s'il contient "femme" sinon l'étiquette "0" est attachée au fichier qui correspond à la catégorie "homme".

Après ça, nous prenons environ 20% du jeu total de données et nous le réservons pour la phase de validation pour s'assurer du bon apprentissage du modèle.

Pour artificiellement augmenter le nombre d'images sans avoir à rajouter de nouvelles, nous prenons les images existantes et nous appliquons aléatoirement certains effets sur elles, telles que des rotations, changement de résolution et zoom et nous rajoutons les résultats au dataset.

```
image_files = [f for f in glob.glob(r"C:\\Users\\Fader\\Dev\\memoire\\detection\\dataset_gender" + "**/*", recursive=True) if not os.path.isdir(f)]
random.shuffle(image_files)

for img in image_files:
    image = cv2.imread(img)
    image = cv2.resize(image, (img_dims[0],img_dims[1]))
    image = img_to_array(image)
    data.append(image)
    label = img.split(os.path.sep)[-2]
    if label == "woman":
        label = 1
    else:
        label = 0
    labels.append([label])

data = np.array(data, dtype="float") / 255.0
labels = np.array(labels)

(trainX, testX, trainY, testY) = train_test_split(data, labels, test_size=0.2,
                                                random_state=42)

trainY = to_categorical(trainY, num_classes=2)
testY = to_categorical(testY, num_classes=2)

aug = ImageDataGenerator(rotation_range=25, width_shift_range=0.1,
                        height_shift_range=0.1, shear_range=0.2, zoom_range=0.2,
                        horizontal_flip=True, fill_mode="nearest")
```

Figure 30 préparation des données

2.3 Définition du modèle de détection de genre :

Nous commençons d'abord par choisir le type de modèle que nous allons créer. Il existe plusieurs types de modèles possibles, chacun ayant ses propres avantages et inconvénients. Pour notre cas, le modèle « séquentiel » est la meilleure option, dû à la nature des données traitées. Séquentiel est le moyen le plus simple de créer un modèle. Il permet de construire un modèle couche par couche.

Nous utilisons la fonction 'add()' pour ajouter des couches à notre modèle. Nos 2 premières couches sont des couches Conv2D. Ce sont des couches de convolution qui traiteront nos images d'entrée, qui sont considérées comme des matrices bidimensionnelles.

```
def build(width, height, depth, classes):
    model = Sequential()
    inputShape = (height, width, depth)
    chanDim = -1
    if K.image_data_format() == "channels_first":
        inputShape = (depth, height, width)
        chanDim = 1
    model.add(Conv2D(32, (3,3), padding="same", input_shape=inputShape))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(MaxPooling2D(pool_size=(3,3)))
    model.add(Dropout(0.25))
    model.add(Conv2D(64, (3,3), padding="same"))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(Conv2D(64, (3,3), padding="same"))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.25))
    model.add(Conv2D(128, (3,3), padding="same"))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(Conv2D(128, (3,3), padding="same"))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.25))
    model.add(Flatten())
    model.add(Dense(1024))
    model.add(Activation("relu"))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))
    model.add(Dense(classes))
    model.add(Activation("sigmoid"))
    return model
```

Figure 31 définition du modèle de détection de genre

2.4 Compilation du modèle

A l'aide de la structure définie précédemment, nous pouvons finalement lancer la compilation du modèle avec les paramètres suivants :

- Fonction de perte : vu que nous attendons seulement deux résultats possibles, la fonction la plus optimale est la fonction binaire
- Optimiseur : Adam
- Critère d'évaluation : notre principal critère est la précision

```
model = build(width=img_dims[0], height=img_dims[1], depth=img_dims[2],
              classes=2)

opt = Adam(lr=lr, decay=lr/epochs)
model.compile(loss="binary_crossentropy", optimizer=opt, metrics=["accuracy"])

H = model.fit_generator(aug.flow(trainX, trainY, batch_size=batch_size),
                       validation_data=(testX, testY),
                       steps_per_epoch=len(trainX) // batch_size,
                       epochs=epochs, verbose=1)
```

Figure 32 Compilation du modèle

Après la compilation, nous pouvons voir un graphe qui représente l'évolution du modèle après chaque époque et à la fin on se retrouve avec un modèle doté d'une précision d'environ 95% ce qui est amplement acceptable

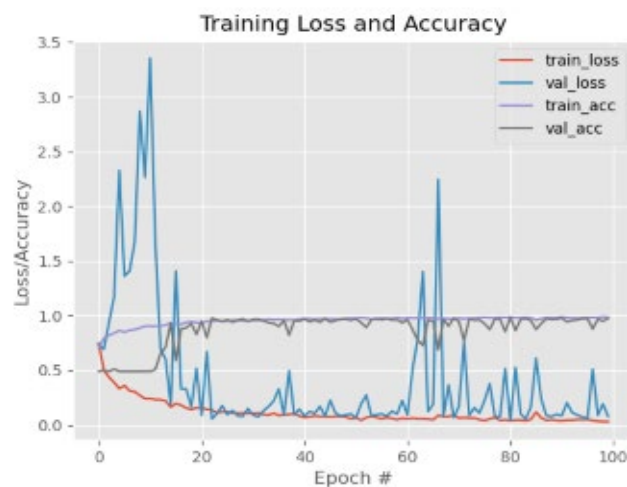


Figure 33 graphe de précision du modèle genre

2.5 Création du modèle de détection de voile :

Après la compilation, on se retrouve avec un modèle avec une précision de presque 100%. Vu la similarité des sujets traités, au lieu de créer de zéro un tout nouveau modèle nous pouvons tirer profit du concept de transfert d'apprentissage « transfer learning » qui consiste à utiliser un modèle pré-entraîné existant comme base et ne changer que la couche d'entrée pour correspondre aux nouvelles données à traiter, cela nous permet d'utiliser un nombre réduit de données d'entraînement sans sacrifier la précision.

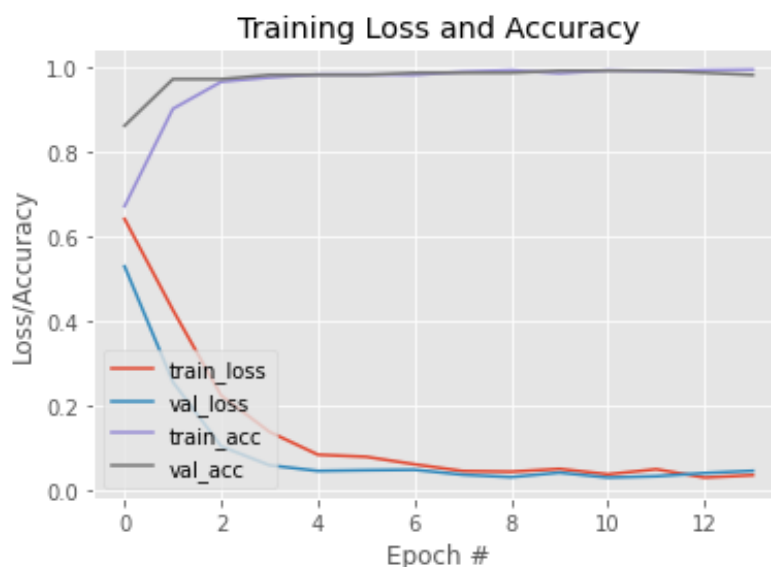


Figure 34 graphe de précision du modèle voile

La création de ce modèle suit quasiment les mêmes étapes que celles du modèle précédent, la différence la plus notable est le fait qu'un nombre inférieur de couches est requis pour avoir un résultat acceptable dû à la relative simplicité de différenciation entre un voile et un visage et la différenciation de genre qui a besoin de beaucoup plus de données.

Transfer learning: idea

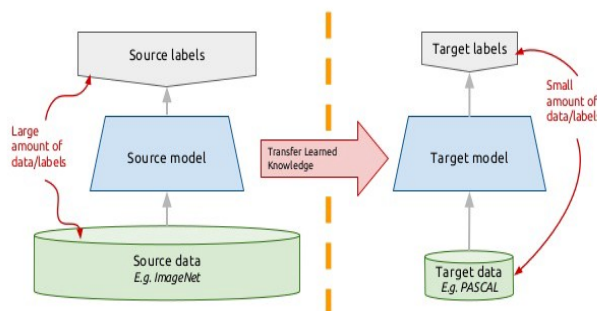


Figure 35 principe de transfert d'apprentissage

2.6 Test du modèle :

Pour s'assurer que les modèles fonctionnent comme prévu, un groupe d'images a été préparé pour un test de précision, à chaque exécution une image est choisie aléatoirement et évaluée par les deux modèles et nous rend la prédiction la plus probable.

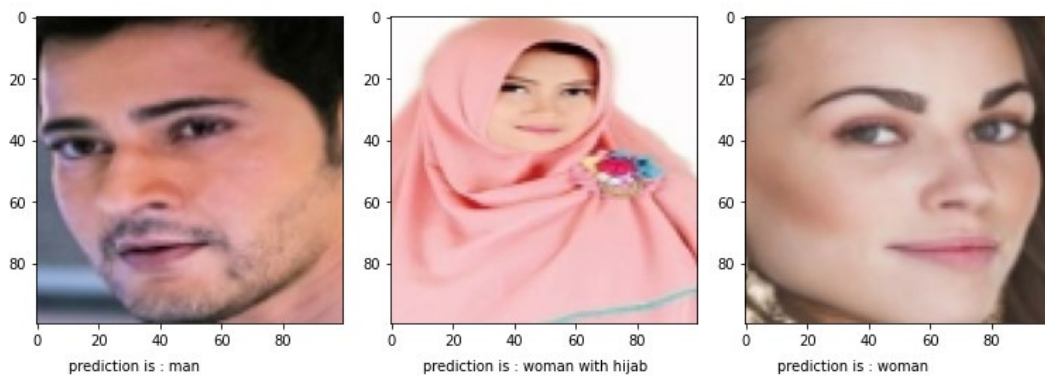
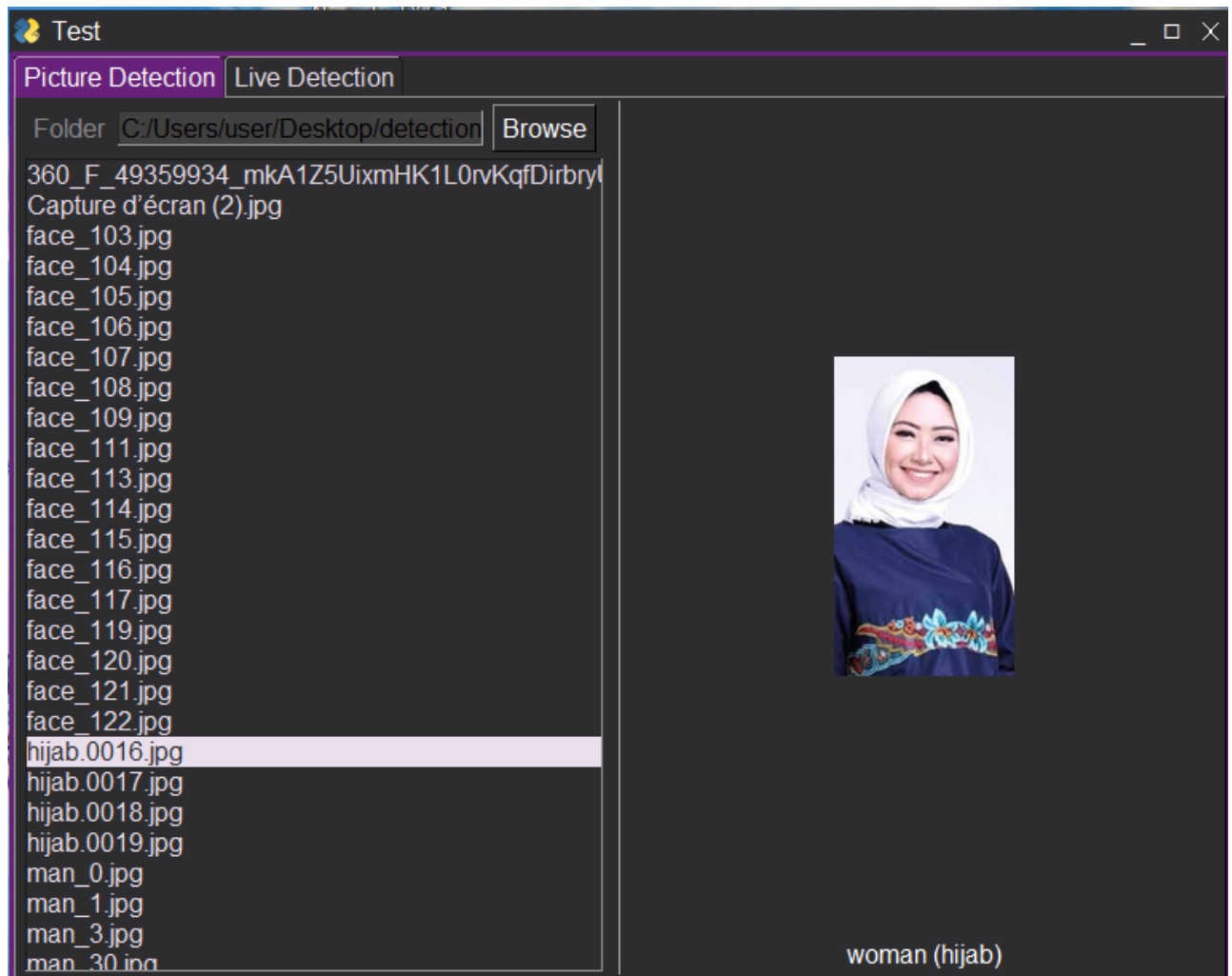


Figure 36 exemple des résultats prédits

```

path = random.choice(os.listdir("test"))
img = "test\\" + path

gender_model = load_model('gender.model')
hijab_model = load_model('hijab.model')
gender_classes = ['man', 'woman']
hijab_classes = ['hijab', 'no hijab']

def detect_gender(image):
    img = cv2.imread(image)
    dim = (96, 96)
    resized = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)
    cv2.imwrite("face.jpg", resized)
    img = cv2.imread('face.jpg')
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    plt.imshow(img, cmap=plt.cm.binary)
    prediction = gender_model.predict(np.array([img]) / 255)
    index = np.argmax(prediction)
    prediction = gender_classes[index]
    print(f'prediction is {gender_classes[index]}')
    os.remove('face.jpg')
    return prediction

def detect_hijab(image):
    img = cv2.imread(image)
    dim = (224, 224)
    resized = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)
    cv2.imwrite("face.jpg", resized)
    img = cv2.imread('face.jpg')
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    plt.imshow(img, cmap=plt.cm.binary)
    prediction = hijab_model.predict(np.array([img]) / 255)
    index = np.argmax(prediction)
    prediction = hijab_classes[index]
    print(f'prediction is {hijab_classes[index]}')
    os.remove('face.jpg')
    return prediction

prediction = detect_hijab(img)
if prediction == 'hijab':
    print("prediction is woman")
else:
    detect_gender(img)

```

Figure 37 Model Testing

Notre modèle est aussi assez performant pour une détection en temps réel, que ce soit une vidéo pré-enregistrée ou par le biais d'une webcam la procédure reste similaire, on lit le contenu vidéo image par image qu'on passe à notre réseau pour traitement qui nous rend un résultat presque instantané.

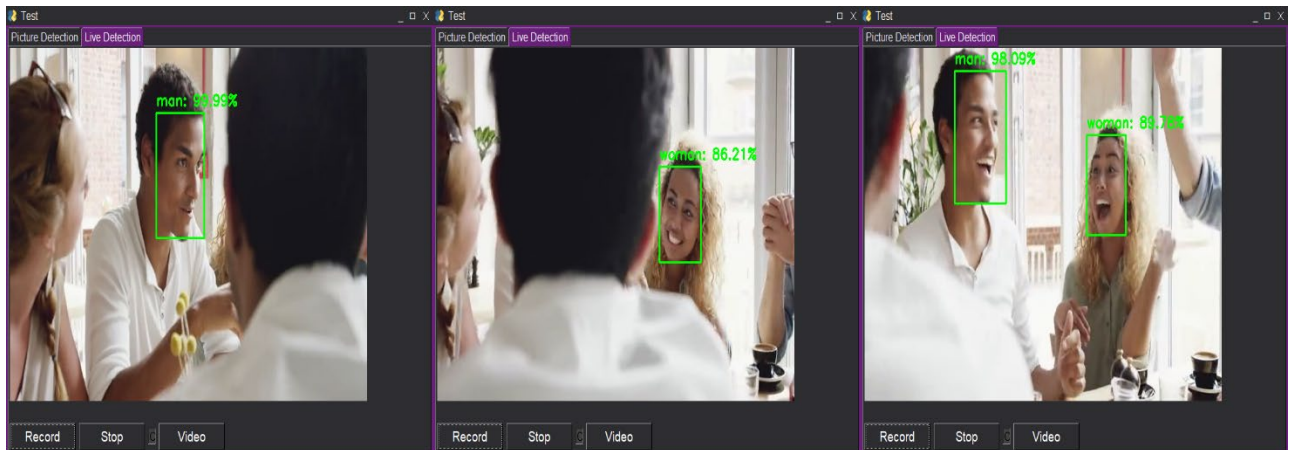


Figure 38 exemple live prédiction

3 Conclusion

Nous avons présenté dans ce chapitre une approche de classification basée sur les réseaux de neurones convolutionnels pour une classification du genre humain (Homme-Femme). Le data-set représente une excellente ressource pour en apprendre plus sur les réseaux de neurones. Il est très délicat d'améliorer ce modèle avec l'augmentation de données et la multiplication du nombre d'époques, mais cela reste assez délicat car on en court un énorme risque de sur-apprentissage. D'autres réseaux de neurones pré-entraînés tels que ResNet offrent une architecture plus profonde mais ne donnent pas un meilleur résultat.

Conclusion Générale

Conclusion générale

Dans ce mémoire de Master, nous nous sommes intéressés à l'importante tâche de classification des images numériques. Nous avons ainsi eu l'occasion d'investir un domaine (des domaines plus précisément) intéressant, motivant et utile. En effet, la tâche de classification des données (en général) est en elle-même intéressante. Cependant, la classification d'images est encore (peut-être) plus motivante et plus pratique (du moins pour notre humble point de vue de nouveaux explorateurs des domaines d'imagerie, de reconnaissance des formes et de classification d'images, en particulier).

Nous avons ainsi passé en revue l'essentiel en matière de classification, classification d'images et techniques d'apprentissage.

En particulier, nous avons visé une application pratique (et aussi motivante) qu'est la classification du genre humain, à partir d'images de visages humains. Problème, dont la résolution pourrait avoir un impact sur la réalisation d'autres tâches intéressantes : Recherche accélérée et plus pertinente (potentiellement) de visages humains par similarité (biométrie, criminalité, etc.), filtrage de publicité, selon que l'internaute est homme ou femme, etc.

Pour réaliser cet objectif, nous avons utilisé une technique basée sur le deep learning (apprentissage profond). Nous avons spécifiquement choisi l'outil des CNNs (Convolutional Neural Networks) comme méthode de classification. Ce choix est justifié par la simplicité et l'efficacité de la méthode.

Les résultats obtenus du segment de test confirment l'efficacité de notre approche.

Pour les perspectives, nous croyons que notre travail n'est que dans son model initial et peut être amélioré en y rajoutant de nouvelles classes à comparer à l'aide du principe de transfert d'apprentissage pour un résultat rapide et optimal. Nous avançons alors que ce travail reste ouvert pour des travaux de comparaison et/ou d'hybridation avec d'autres méthodes de classification.

Bibliographie

- [1] Sheikhpour,R, Sarram, M, A Survey on semi-supervised feature selection methods, memoire de master informatique, 2016.
- [2] Mokri, M, Classification des images avec les réseaux de neurones convolutionnels, mémoire Master en informatique, Université Abou BakrBelkaid Tlemcen, 2017.
- [3] la classificatonsuperviséé
www.math.univangers.fr/~labatte/enseignement%20UFR/master%20MIM/classificatio nsupervisee.pdf.
- [4] Zubiolo , A, extraction de caractéristiques et apprentissage statistique pour l'imagerie biomédicale tissulaire cellulaire, Thèse de Doctorat en sciences , Université Nice Sophia, Spécialité : Automatique, 2014
- [5] Haliche, A., classification supervisée a base de KNN avec pondération d'attributs par les algorithmes génétiques, Mémoire de Magistère, Université des Sciences et Technologie Houari Boumediene (USTHB), 2015.
- [6] Chesner DESIR, Classification Automatique d'Images, Application a` l'Imagerie du Poumon Profond, thèse de doctorat en science, Laboratoire d'Informatique, de Traitement de l'Information et des SystemesUniversité de Rouen ,2011.
- [7] Duin, R, A noté on comparing classifiers. Pattern Recognition Letters, 17(5) :529– 536. (Cf. 112.), 1996.
- [8] [Srinivasan, S et Srinivasan, G, Statistical texture analysis.In Proceedings of world academy of science, engineering and technology, volume 36, pages 1264–1269. (Cf.18.)], 2008.
- [9] Mikolajczyk et Schmid, 2004, Mikolajczyk and Schmid, 2005, Bay et al., 2006
- [10] Xiaohong , y et jinhua , X , The related techniques of content –based image retrieval», international symposium on Computer Science and Computational Technology, IEEE computer society, 2008.
- [11] V.chtkara, Color based image retrieval using compact binary signatures , Technical report, Department of Computer Science , University of Alberta, Admonton, 2001.
- [12] Swain, M. et Ballard, D., Color indexing , International Journal of Vision, 2012.
- [13] Meziane, A, Tanil , A, Techniques d'indexation d'images Médicales par contenu, Mémoire de Master en Informatique , Université Tlemcen, 2014.
- [14] l'histogramme de couleurs, www.researchgate.net/figure/6-Exemple-d-histogramme-dune-image-couleur-Une-des-premieres-approches-decrivant_fig4_331929280. Format non respecté
- [15] la texture de l'image,
www.tsi.telecomparistech.fr/pages/enseignement/ressources/beti/textures_Fourier/textur e.html.
- [16] Houari, K, recherche d'images par le contenu, Thèse De Doctorat ,2010. Incomplet
- [17] Lindeberg,T, Scale invariant feature transform , 2012. Incomplet
- [18] Rublee, E. et Rabaud, K., An efficient alternative to SIFT or SURF, en 2011 IEEE International Conference on Computer Vision (ICCV) (IEEE, 2011), pp. 2564–2571, 2011.

- [19] Dalal, N. & Triggs, B., Histograms of oriented gradients for human detection, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, CVPR 2005, vol. 1 (IEEE, 2005), pp. 886–893, 2005.
- [20] Les concept de KNN, www.kraj3.com.np/blog/2019/06/basic-concepts-of-knn-algorithm/
- [21] Les methods de KNN, www.wikipedia.org/wiki/Méthode_des_k_plus_proches_voisins.
- [22] Mathieu-Dupas, E, Algorithme des k plus proches voisins pondérés et application endiagnostic. Marseille, 2010.
- [23] les arbres de décisions, [www.fr.wikipedia.org/wiki/Arbre_de_décision_\(apprentissage\)](http://www.fr.wikipedia.org/wiki/Arbre_de_décision_(apprentissage)).
- [24] Les arbres de décision, www.slideshare.net/mariemchaaben/les-arbres-de-dcision-ou-de-regression.
- [25] Mohamadally, H, SVM : Machines à Vecteurs de Support ou Séparateurs à Vastes Marges, mémoire du master, 2006.
- [26] Les SVM WWW.dataanalyticspost.com/Lexique/svm/
- [27] Les SVM www.webs.iiitd.edu.in/raghava/rbpred/algorithm.html
- [28] Klein, M, “The hebb legacy,” Can. J. Exp. Psychol. 2013.
- [29] Les reseaux de neurones artificiels, www.wikipedia.org/wiki/Réseau_de_neurones_artificiels
- [30] EL AKADI, A, Contribution à la sélection de variables pertinentes en classification supervisée : Application à la sélection des gènes pour les puces à ADN et des caractéristiques faciales, (2012)
- [31] Alassan, Y, Amélioration du système de recueils d’information de l’entreprise Semantic Group Company grâce à la constitution de ressources sémantiques, / École doctorale Connaissance, langage, modélisation, 2017
- [32] Garillos, M, Generalized Confusion Matrix for Multiple Classes, mémoire de, article, 2016 .
- [33] Bekaddour, F, Amélioration des Performances des Classifieurs à base de Métaheuristiques, MAGISTER EN INFORMATIQUE, De l’Ecole Doctorale des systèmes et technologies d’information et de communication STIC, 2014.
- [34] HAMMYANI, A et ALLIOUA, S, Amélioration des forêts aléatoires : Application au diagnostic médical, Mémoire de Master en Informatique, Université Abou Bekr Belkaid, 2013.
- [35] Adabost, www.wikipedia.org/wiki/AdaBoost.
- [36] Boughaba Mohammed et Boukhris Brahim, L’apprentissage profond (Deep Learning) Pour la classification et la recherche d’images par le contenu, Master Professionnel, 2017.
- [37] Medjdoubi, A, ANALYSE Des SENTIMENTs UTILISANT LE DEEP LEARNING, MEMOIRE DE MASTER en informatique, Université Dr. TAHAR MOULAY SAIDA, , 2019
- [38] A. Krizhevsky, I. Sutskever et G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks », Advances in neural Processing Systems de traitement. 2012.
- [39] Graham, Benjamin. Fractional Max-Pooling. 18 décembre 2014.

- [40] Bekaddour, F, Amélioration des Performances des Classifieurs à base de Métaheuristiques, MAGISTER EN INFORMATIQUE, De l'Ecole Doctorale des systèmes et technologies d'information et de communication STIC, 2014.
- [41] Pradhan, V. Pradhan, V, A survey on Sentiment Analysis Algorithms for opinion mining International Journal of Computer Applications.2016.
- [42] Szeliski,R, ,Introduction to machine learning, pp. 1–25, 2010.
- [43] Ayodélé, A, Contributions au provisionnement en assurance de personnes et à la gestion des risques,mémoire de master en informatique, l'Université Claude Bernard Lyon , 2019.
- [44] www.ai.facebook.com/tools, 22 Avril 2020.
- [45] Kevin McGuiness Transfer Learning (D2L4 Insight@DCU Machine Learning Workshop 2017)
- [46] teradata. what is python. *teradata.com*.
- [47] colab.research.google.com
- [48] <https://www.tensorflow.org>