



# Classification des images avec les réseaux de neurones convolutifs

Projet préparé sous la direction de M. *Makrem MHADHBI*

Présenté par

*Sana RANNEN*

*Marwen GHODHBEN*

*Oussama MEZZI*

Année universitaire 2019/2020

## **REMERCIEMENT**

*Au terme de ce travail, nous adressons nos vifs remerciements à Monsieur **Makrem MHADHBI**, notre encadrant, pour son suivi, sa disponibilité, son aide précieux et ses conseils qui nous ont été d'une utilité indéniable.*

*Nous exprimons également toutes nos gratitude à tous nos enseignants de **SESAME** pour la formation qu'ils nous ont donnés.*

*Enfin, nous remercions ceux et celles qui ont participé de près ou de loin à l'élaboration du présent travail et principalement pour leur service et pour leur soutien moral tout au long de la préparation de ce projet.*

**Sana RANNEN**

**Marwen GHODHBEN**

**Oussama MEZZI**

## SOMMAIRE

REMERCIEMENT .....	2
SOMMAIRE .....	3
Liste des figures.....	5
Liste des tableaux.....	6
Acronymes.....	7
INTRODUCTION GENERALE .....	8
Etat de l’art.....	10
1. Introduction .....	11
2. Intelligence Artificielle.....	11
3. Machine Learning.....	12
3.1. Apprentissage supervisé .....	12
3.2. Apprentissage non supervisé .....	12
3.3. Apprentissage par renforcement .....	13
3.4. Algorithmes .....	13
4. Deep Learning .....	15
4.1. Algorithmes de Deep Learning .....	16
5. Réseaux de neurones artificiels.....	16
5.1. Principe de fonctionnement.....	17
6. Réseaux de neurones convolutifs .....	17
6.1. Architecture de réseaux de neurones convolutifs .....	18
7. Classification des images.....	19
8. Méthodologie de travail.....	20
8.1. Mode d’emploi .....	21
9. Conclusion .....	22
Cadre du projet .....	23
1. Introduction .....	24
2. Compréhension métier .....	24
2.1. Problématique étudiée .....	24
2.2. Présentation des outils.....	24
3. Compréhension donnée.....	28
3.1. Types de format d’image.....	29

3.2. Caractéristiques de l'image .....	29
4. Préparation des données .....	31
4.1. Choix des datasets.....	31
5. Conclusion .....	33
Implémentation et réalisation .....	34
9.1. Introduction .....	35
9.2. Modélisation et évaluation .....	35
9.2.1. Application des algorithmes de ML.....	35
9.2.2. Application de CNN .....	36
9.3. Déploiement.....	41
9.4. Conclusion .....	42
Conclusion générale .....	43
Neto graphie.....	44

## Liste des figures

Figure 1 : Relation entre IA ML et DL .....	15
Figure 2: Exemple de modèle de CNN [18] .....	18
Figure 3: Méthode CRISP.....	21
Figure 4: Logo python.....	24
Figure 5: Logo TensorFlow .....	25
Figure 6: Logo theano.....	25
Figure 7: Logo Keras .....	26
Figure 8: Logo Scikit-learn .....	26
Figure 9: Logo Flask .....	27
Figure 10: Logo Anaconda.....	28
Figure 11: Gender Dataset .....	31
Figure 12: MPEG-7 Dataset .....	32
Figure 13: Flowers-17 Dataset .....	33
Figure 14: Résultats de Classification.....	35
Figure 15: Résultats de prédiction .....	36
Figure 16: Modèle CNN appliqué .....	37
Figure 17: Précision et Erreur – Flowers-17 Dataset.....	38
Figure 18: Précision et Erreur – MPEG-7 Dataset .....	38
Figure 19: Précision et Erreur –Gender Dataset .....	38
Figure 20: Résultats de prédiction .....	39
Figure 21: Résultats de prédiction .....	40
Figure 22: Résultats de prédiction .....	41
Figure 23: Architecture de l'application Web .....	41
Figure 24: Interface Web.....	42

## Liste des tableaux

Tableau 1 : Caractéristiques des machines utilisées .....	28
Tableau 2: Résultats de classification.....	35
Tableau 3: Résultats de prédiction.....	36
Tableau 4 : Taux de précision .....	39
Tableau 5: Résultats de prédiction.....	39
Tableau 6: Résultats de prédiction.....	40

## **Acronymes**

**IA** – Intelligence Artificielle

**ML** – Machine Learning

**DL** – **Deep** Learning

**K-NN** – K-nearest neighbors

**CNN** – Convolutional Neural Network

**CRISP** – Cross Industry Standard Process

**SPYDER** – Scientific PYthon Development EnviRonment

# INTRODUCTION GENERALE

Nous vivons dans un monde numérique, où les informations sont stockées, traitées, indexées et recherchées par des systèmes informatiques, ce qui rend leur récupération une tâche rapide et pas cher. Au cours des dernières années, des progrès considérables ont été réalisés dans le domaine de classification d'images. Ce progrès est dû aux nombreux travaux dans ce domaine et à la disponibilité des bases d'images internationales qui ont permis aux chercheurs de signaler de manière crédible l'exécution de leurs approches dans ce domaine, avec la possibilité de les comparer à d'autres approches qu'ils utilisent les mêmes bases.

A la fin des années 80 Yan le Cun a développé un type de réseau particulier qui s'appelle le réseau de neurone convolutif, ces réseaux sont une forme particulière de réseau neuronal multicouche dont l'architecture des connexions est inspirée de celle du cortex visuel des mammifères. Par exemple, chaque élément n'est connecté qu'à un petit nombre d'éléments voisins dans la couche précédente. En 1995, Yan le cun et deux autres ingénieurs ont développé un système automatique de lecture de chèques qui a été déployé largement dans le monde. À la fin des années 90, ce système lisait entre 10 et 20 % de tous les chèques émis aux États-Unis. Mais ces méthodes étaient plutôt difficiles à mettre en œuvre avec les ordinateurs de l'époque, et malgré ce succès, les réseaux convolutifs et les réseaux neuronaux plus généralement ont été délaissés par la communauté de la recherche entre 1997 et 2012.

En 2011 et 2012 trois événements ont soudainement changé la situation. Tout d'abord, les GPU (Graphical Processing Unit) capables de plus de mille milliards d'opérations par seconde sont devenus disponibles pour un prix moins cher. Ces puissants processeurs spécialisés, initialement conçus pour le rendu graphique des jeux vidéo, se sont avérés être très performants pour les calculs des réseaux neuronaux. Deuxièmement, des expériences menées simultanément à Microsoft, Google et IBM avec l'aide du laboratoire de Geoff Hinton ont montré que les réseaux profonds pouvaient diminuer de moitié les taux d'erreurs des systèmes de reconnaissance vocale. Troisièmement plusieurs records en reconnaissance d'image ont été battus par des réseaux de neurones convolutifs. L'événement le plus marquant a été la victoire éclatante de l'équipe de Toronto dans la compétition de reconnaissance d'objets « ImageNet ». La diminution des taux d'erreurs était telle qu'une véritable révolution. Du jour au lendemain, la majorité des équipes de recherche en parole et en vision ont abandonné leurs méthodes préférées et sont passées aux réseaux de neurones convolutifs et autres réseaux neuronaux. L'industrie d'Internet a immédiatement saisi l'opportunité et a commencé à investir massivement dans des équipes de recherche et développements en apprentissage profond.

Dans notre projet on va utiliser les réseaux de neurones convolutifs pour la classification des images.



Pour ce faire, nous avons structuré notre rapport en trois chapitres :

- Dans le premier chapitre on va présenter les notions de base de l'Intelligence Artificielle, le Machine Learning, Le Deep Learning, les Réseaux de Neurones Convolutifs, la classification d'image ainsi que la méthodologie de travail.
- Le deuxième chapitre est consacré à présenter le cadre de notre projet, les outils utilisés ainsi qu'à la compréhension de la partie données.
- Dans le troisième chapitre, on va montrer la partie expérimentale de notre travail et on discute les différents résultats obtenus et à la fin on termine par une conclusion générale.

# **Etat de l'art**

# 1. Introduction

Dans ce chapitre, nous allons définir les notions de base de l'Intelligence Artificielle, le Machine Learning, le Deep Learning et les réseaux de neurones artificiels. Ainsi que la présentation de la méthodologie de travail suivie pour la réalisation de ce travail.

## 2. Intelligence Artificielle

Historiquement, les travaux en IA démarrent dans les années 1950 avec les travaux d'Alan TURING. L'IA est devenue un domaine de recherche à l'été 1956, lors de la première conférence des pionniers de cette discipline, notamment John MCCARTHY, Marvin MINSKY, Allen NEWELL, Herbert SIMON ou Donald MICHIE. Avant 2000, les limites imposées par les capacités de calculs et de stockage n'ont pas permis de réaliser des avancées significatives dans le domaine de l'IA. Il faut attendre le début des années 2000 pour voir apparaître les principaux facteurs de ruptures technologiques qui ont permis les avancées actuelles [1] :

- Le réseau internet et l'usage partagé de données qui ont permis de créer des technologies comme les moteurs de recherches ou les architectures décentralisées et hyper scalables.
- Une croissance exponentielle de la quantité de données : L'espace de stockage offert pour 1€ double tous les 14 mois.
- Une croissance exponentielle des capacités de calcul : La quantité totale de données créée chaque année double tous les 2 ans.
- La mobilité et le développement des objets connectés qui favorisent l'accès à des flux de données en temps réel : en 2020, il y aura 50 milliards d'objets connectés, qui produiront 10% du total des données créées.

En conséquence, l'intelligence artificielle s'est développée très fortement depuis plus de 10 ans avec une accélération dans les 5 dernières années, pour permettre notamment des usages comme [2] :

- La perception visuelle : reconnaissance d'un objet ou description de scènes.
- La compréhension du langage naturel écrit ou parlé : traduction automatique, production automatique d'articles de presse, analyse de sentiments.
- L'analyse automatique en « comprenant » une requête et en renvoyant des résultats pertinents, même si ce résultat ne contient pas les mots de la requête.
- La prise de décision autonome pour battre les humains aux échecs ou au Go.

L'IA, peut être définie comme « ensemble de techniques permettant à des machines d'accomplir des tâches et de résoudre des problèmes normalement réservés aux humains et à certains animaux ». Il s'agit donc de techniques qui visent à reproduire, imiter, simuler l'intelligence, ou en tout cas les capacités que l'on peut associer à ce terme : capacité à percevoir son environnement, à interagir avec lui, à communiquer au moyen d'un langage, à mettre en œuvre un raisonnement et à résoudre des problèmes, à planifier des actions afin d'atteindre un but, à représenter la connaissance, à appliquer cette connaissance au moment opportun [3].

### 3. Machine Learning

Le Machine Learning correspond au fait de donner la capacité d'apprendre à la machine, par elle-même, quelle que soit la situation, sans que l'on ait formellement à écrire (ni même à connaître) toutes les règles [4].

La phase d'apprentissage repose sur un algorithme d'apprentissage c'est-à-dire la mise en place d'une architecture et d'un programme informatiques qui permettent à une machine de recevoir des données d'entrée, d'effectuer une série de traitements utilisant ces données d'entrée, de produire un résultat en sortie et, surtout, de s'améliorer pour produire ce résultat. Ce dernier point suppose qu'un objectif de résultat (et éventuellement une tolérance par rapport à l'atteinte de cet objectif) soit communiqué au système et que des données d'entraînement et les moyens de mesurer ses performances lui soient fournis. C'est en cherchant à améliorer ses performances sur les données d'entraînement que la machine va apprendre. Une fois que l'apprentissage sera terminé, la « machine » pourra produire le résultat de façon autonome sur des données d'entrée qu'elle n'aura encore jamais rencontrées.

Le ML permet de résoudre différents types de tâches telles que :

- Le regroupement (de choses ou de personnes) via des méthodes de classification automatique (clustering) ;
- La prédiction (prédire une valeur à partir d'autres valeurs) via des méthodes de régression par exemple ;
- La détection d'anomalies ou la recherche de signaux faibles ;
- La recommandation (recommander des items en se basant sur des comportements) via du filtrage collaboratif.

Selon le résultat visé, les données dont on dispose au départ et le contexte, il existe plusieurs façons de faire apprendre à une machine. On distingue notamment l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

#### 3.1. Apprentissage supervisé

Dans le cas de l'apprentissage supervisé, le système est guidé dans son apprentissage. On lui indique le type de résultat à atteindre en le nourrissant d'exemples. Pour cela, on lui fournit des données d'entrée pour lesquelles le résultat est connu et communiqué au système. Le but est qu'il puisse ensuite généraliser ce qu'il a appris pour des données non connues. Par exemple, si le système doit apprendre à reconnaître des feuilles de vigne dans une image, on lui fournit des images où la feuille est signalée et où le label « feuille de vigne » est associé. On parle ainsi de données étiquetées ou annotées. Le jeu de données d'entraînement annotées permet au système de calculer ses erreurs en comparant ses résultats avec les résultats connus et ainsi d'ajuster le modèle pour progresser. Une partie des données annotées pourra également servir à vérifier l'efficacité du modèle, une fois l'apprentissage terminé [3].

#### 3.2. Apprentissage non supervisé

Dans le cas de l'apprentissage non supervisé, on ne donne pas d'exemples de résultats attendus au système. Seules les données d'entrée sont fournies et le système doit apprendre, de façon autonome, la meilleure façon d'explorer les données. Il doit chercher à identifier dans le jeu de données une façon de les structurer ou encore à extraire des caractéristiques particulières. La

performance permettant l'ajustement du modèle est alors appréciée grâce à des indicateurs objectifs, comme, par exemple, des calculs de variabilité intra ou interclasses [3].

### 3.3. Apprentissage par renforcement

Le système apprend cette fois par interaction avec l'environnement. A chaque instant, le système connaît l'état actuel et toutes les actions possibles. Il effectue une des actions et reçoit un signal de retour qui lui notifie son nouvel état et la récompense associée. A force d'itération, le système doit pouvoir déterminer automatiquement le comportement idéal pour un contexte spécifique. Ce type d'apprentissage est également dit « semi-supervisé » dans la mesure où la récompense donne une indication sur le bon résultat à atteindre [3].

### 3.4. Algorithmes

#### 3.4.1. La régression linéaire

L'algorithme de régression linéaire est un algorithme d'apprentissage supervisé c'est-à-dire qu'à partir de la variable cible ou de la variable à expliquer (Y), le modèle a pour but de faire une prédiction grâce à des variables dites explicatives (X) ou prédictives [5].

#### 3.4.2. Les *k* plus proches voisins

L'algorithme K-NN est une méthode d'apprentissage supervisé. Il peut être utilisé aussi bien pour la régression que pour la classification. Son fonctionnement peut être assimilé à l'analogie suivante “*dis-moi qui sont tes voisins, je te dirais qui tu es...*”.

Pour effectuer une prédiction, l'algorithme K-NN ne va pas calculer un modèle prédictif à partir d'un *Training Set* comme c'est le cas pour la régression logistique ou la régression linéaire. En effet, K-NN n'a pas besoin de construire un modèle prédictif. Ainsi, pour K-NN il n'existe pas de phase d'apprentissage proprement dite. C'est pour cela qu'on le catégorise parfois dans le Lazy Learning. Pour pouvoir effectuer une prédiction, K-NN se base sur le jeu de données pour produire un résultat [6].

#### 3.4.3. K-means

L'algorithme k-means mis au point par McQueen en 1967, un des plus simples algorithmes d'apprentissage non supervisé, appelée algorithme des centres mobiles, il attribue chaque point dans un cluster dont le centre (centroïde) est le plus proche. Le centre est la moyenne de tous les points dans le cluster, ses coordonnées sont la moyenne arithmétique pour chaque dimension séparément de tous les Points dans le cluster c'est à dire chaque cluster est représentée par son centre de gravité [7].

#### 3.4.4. Les arbres de décision

Les arbres de décision sont l'une des structures de données majeures de l'apprentissage statistique. Leur fonctionnement repose sur des heuristiques qui, tout en satisfaisant l'intuition, donnent des résultats remarquables en pratique. Leur structure arborescente les rend également lisibles par un être humain, contrairement à d'autres approches où le prédicteur construit est une « boîte noire ».

Un arbre de décision modélise une hiérarchie de tests sur les valeurs d'un ensemble de variables appelées *attributs*. À l'issue de ces tests, le prédicteur produit une valeur numérique ou choisit

un élément dans un ensemble discret de conclusions. On parle de *régression* dans le premier cas et de *classification* dans le second [8].

#### 3.4.5. Les forêts aléatoires

L'algorithme des « forêts aléatoires » (ou Random Forest parfois aussi traduit par forêt d'arbres décisionnels) est un algorithme de *classification* qui réduit la *variance* des prévisions d'un *arbre de décision* seul, améliorant ainsi leurs performances. Pour cela, il combine de nombreux arbres de décisions dans une approche de type *bagging*.

L'algorithme des « forêts aléatoires » a été proposé par Leo Breiman et Adèle Cutler en 2001. Dans sa formule la plus classique, il effectue un apprentissage en parallèle sur de multiples arbres de décision construits aléatoirement et entraînés sur des sous-ensembles de données différents. Le nombre idéal d'arbres, qui peut aller jusqu'à plusieurs centaines voire plus, est un paramètre important : il est très variable et dépend du problème. Concrètement, chaque arbre de la forêt aléatoire est entraîné sur un sous-ensemble aléatoire de données selon le principe du bagging, avec un sous-ensemble aléatoire de features (caractéristiques variables des données) selon le principe des « projections aléatoires ». Les prédictions sont ensuite moyennées lorsque les données sont quantitatives ou utilisés pour un vote pour des données qualitatives, dans le cas des arbres de classification [9].

#### 3.4.6. Les machines à vecteur de support

Parmi les méthodes à noyaux, inspirées de la théorie statistique de l'apprentissage de Vladimir Vapnik, les SVM constituent la forme la plus connue. SVM est une méthode de classification binaire par apprentissage supervisé, elle fut introduite par Vapnik en 1995. Cette méthode est donc une alternative récente pour la classification. Cette méthode repose sur l'existence d'un classificateur linéaire dans un espace approprié. Puisque c'est un problème de classification à deux classes, cette méthode fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle. Elle est basée sur l'utilisation de fonction dites noyau (kernel) qui permettent une séparation optimale des données [10].

#### 3.4.7. Le perceptron multicouche

Le perceptron est l'un des tout premiers algorithmes de Machine Learning, et le réseau de neurones artificiels le plus simple. C'est un système artificiel capable d'apprendre par l'expérience [11].

Introduit en 1957 par Franck Rosenblatt, il n'est véritablement utilisé que depuis 1982 après son perfectionnement. Grâce à la puissance de calcul des années 2000, le perceptron s'est largement démocratisé et est de plus en plus utilisé.

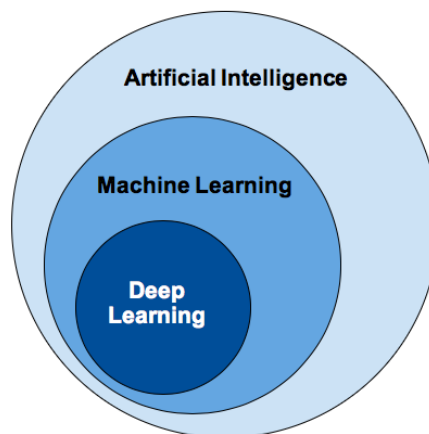
Lors de sa création, le perceptron apparaissait comme une avancée majeure pour le développement de l'intelligence artificielle. Malheureusement, les limites techniques ont rapidement été atteintes. Pour cause, un perceptron à une seule couche peut uniquement séparer les classes si elles sont séparables de façon linéaire.

Toutefois, par la suite, il a été découvert qu'un perceptron multicouche pouvait permettre de classer des groupes qui ne sont pas séparables de façon linéaire. Il permet donc de résoudre des problèmes que les algorithmes à une seule couche ne peuvent résoudre.

Ainsi, un perceptron multicouche est un type de réseau neuronal formel qui s'organise en plusieurs couches. L'information circule de la couche d'entrée vers la couche de sortie. Au contraire un modèle monocouche ne dispose que d'une seule sortie pour toutes les entrées. Il est donc un réseau à propagation directe. Chaque couche se compose d'un nombre de neurones variables. Les neurones de la dernière couche sont les sorties du système global [12].

## 4. Deep Learning

Le Deep Learning n'en est qu'un sous ensemble de l'Intelligence Artificielle. Il fait partie des méthodes d'apprentissage automatique (ou Machine Learning), constituant elles-mêmes l'une des branches de l'IA. Il concerne les algorithmes inspirés par la structure et le fonctionnement du cerveau. Ils peuvent apprendre plusieurs niveaux de représentation dans le but de modéliser des relations complexes entre les données.



*Figure 1 : Relation entre IA ML et DL*

Le Deep Learning est basé sur l'idée des réseaux de neurones artificielles et il est taillé pour gérer de larges quantités de données en ajoutant des couches au réseau. Un modèle de DL a la capacité d'extraire des caractéristiques à partir des données brutes grâce aux multiples couches de traitement composé de multiples transformations linéaires et non linéaires et apprendre sur ces caractéristiques petit à petit à travers chaque couche avec une intervention humaine minime.

Sur les cinq dernières années, le DL est passé d'un marché de niche ou seulement une poignée de chercheurs s'y intéressait au domaine le plus prisé par les chercheurs. Les recherches en relation avec le DL apparaissent maintenant dans les tops journaux comme Science, Nature et Nature Methods pour ne citer que quelques-uns. Le DL a coquerie le GO, appris à conduire une voiture, diagnostiquer le cancer et l'autisme et même devenu un artiste.

Le terme « Deep Learning » a été introduit pour la première fois au ML par Dechter (1986), et aux réseaux neuronaux artificiels par Aizenberg et al (2000) [13].

Les algorithmes de ML fonctionnent bien pour une grande variété de problèmes. Cependant ils ont échoué à résoudre quelques problèmes majeurs de l'IA telle que la reconnaissance vocale et la reconnaissance d'objets.

Le développement du DL fut motivé en partie par l'échec des algorithmes traditionnels dans de telle tâche de l'IA.

#### 4.1. Algorithmes de Deep Learning

Il existe différents algorithmes de Deep Learning. Nous pouvons ainsi citer [14] :

5. **Les réseaux de neurones profonds** (Deep Neural Networks). Ces réseaux sont similaires aux réseaux MLP mais avec plus de couches cachées. L'augmentation du nombre de couches, permet à un réseau de neurones de détecter de légères variations du modèle d'apprentissage, favorisant le sur-apprentissage ou sur-ajustement (« overfitting »).
6. **Les réseaux de neurones convolutifs** (Convolutional Neural Networks). Le problème est divisé en sous parties, et pour chaque partie, un « cluster » de neurones sera créer afin d'étudier cette portion spécifique. Par exemple, pour une image en couleur, il est possible de diviser l'image sur la largeur, la hauteur et la profondeur (les couleurs).
7. **La machine de Boltzmann profonde** (Deep Belief Network) : Ces algorithmes fonctionnent suivant une première phase non supervisée, suivi de l'entraînement classique supervisé. Cette étape d'apprentissage non-supervisée, permet, en outre, de faciliter l'apprentissage supervisé.

### 5. Réseaux de neurones artificiels

Le concept des réseaux de neurones artificiels fut inventé en 1943 par deux chercheurs de l'Université de Chicago : le neuro physicien Warren McCullough, et le mathématicien Walter Pitts. Dans un article publié dans le journal Brain Theory, les deux chercheurs présentent leur théorie selon laquelle l'activation de neurones est l'unité de base de l'activité cérébrale.

En 1957, le Perceptron fut inventé. Il s'agit du plus ancien algorithme de Machine Learning, conçu pour effectuer des tâches de reconnaissance de patterns complexes. C'est cet algorithme qui permettra plus tard aux machines d'apprendre à reconnaître des objets sur des images.

Malheureusement, à l'époque, les réseaux de neurones étaient limités par les ressources techniques. Par exemple, les ordinateurs n'étaient pas assez puissants pour traiter les données nécessaires au fonctionnement des réseaux de neurones. C'est la raison pour laquelle la recherche dans le domaine des Neural Networks est restée en sommeil durant de longues années.

Il aura fallu attendre le début des années 2010, avec l'essor du Big Data et du traitement massivement parallèle, pour que les Data Scientists disposent des données et de la puissance de calcul nécessaires pour exécuter des réseaux de neurones complexes.

En 2012, lors d'une compétition organisée par ImageNet, un Neural Network est parvenu pour la première fois à surpasser un humain dans la reconnaissance d'image. C'est la raison pour laquelle cette technologie est de nouveau au cœur des préoccupations des scientifiques. A présent, les réseaux de neurones artificiels ne cessent de s'améliorer et d'évoluer de jour en jour [15].



## 5.1. Principe de fonctionnement

En règle générale, un réseau de neurones repose sur un grand nombre de processeurs opérant en parallèle et organisés en tiers. Le premier tiers reçoit les entrées d'informations brutes, un peu comme les nerfs optiques de l'être humain lorsqu'il traite des signaux visuels.

Par la suite, chaque tiers reçoit les sorties d'informations du tiers précédent. On retrouve le même processus chez l'Homme, lorsque les neurones reçoivent des signaux en provenance des neurones proches du nerf optique. Le dernier tiers, quant à lui, produit les résultats du système.

Par le biais d'un algorithme, le réseau de neurones artificiels permet à l'ordinateur d'apprendre à partir de nouvelles données. L'ordinateur doté du réseau de neurones apprend à effectuer une tâche en analysant des exemples pour s'entraîner. Ces exemples ont préalablement été étiquetés afin que le réseau puisse savoir ce dont il s'agit.

Par exemple, un réseau de neurones peut être utilisé pour apprendre à l'ordinateur à reconnaître des objets. Un grand nombre d'objets d'une même catégorie est présenté au réseau de neurones, et l'ordinateur apprend à reconnaître cet objet sur de nouvelles images en analysant les patterns récurrents au sein des images d'exemple. Ainsi, en analysant des milliers de photos de chats, le Neural Network apprendra à reconnaître un chat sur n'importe quelle photo.

Contrairement à d'autres types d'algorithmes, les réseaux de neurones ne peuvent pas être programmés directement pour effectuer une tâche. A la manière du cerveau en développement d'un enfant, la seule instruction qu'ils ont est d'apprendre [16].

On distingue différents types de réseaux de neurones. En règle générale, les Neural Networks sont catégorisés en fonction du nombre d'épaisseurs qui séparent l'entrée de données de la production du résultat, en fonction du nombre de nœuds cachés du modèle, ou encore du nombre d'entrées et de sorties de chaque nœud.

En fonction du type de réseau, la propagation des informations entre les différents tiers de neurones peut varier. Dans la variante la plus simple, celle du réseau de neurones dit « feed-forward », les informations passent directement de l'entrée aux nœuds de traitement puis aux sorties [15].

## 6. Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs sont une puissante application de l'intelligence artificielle au traitement des images, qui s'appuie sur l'apprentissage profond pour effectuer des tâches descriptives et génératives. Ils exploitent souvent la vision artificielle, notamment la reconnaissance de la vidéo et des images, les systèmes de recommandation et le traitement automatique du langage naturel.

Un réseau de neurones est un système logiciel ou matériel qui est calqué sur le fonctionnement des neurones humains. Sous leur forme classique, les réseaux neuronaux se prêtent mal au traitement des images, car celles-ci leur sont transmises à résolution réduite.

Dans les CNN, les « neurones » sont disposés comme ceux du lobe frontal, siège du traitement des stimuli visuels chez l'homme et d'autres animaux. Les couches de neurones sont organisées de façon à couvrir tout le champ visuel afin d'éviter le problème du traitement d'images fragmentées des réseaux neuronaux classiques [17].

## 6.1. Architecture de réseaux de neurones convolutifs

Un CNN utilise un système comparable au perceptron multicouche, mais conçu pour réduire le nombre de calculs. La structure d'un CNN consiste en une succession de couches :

7. La couche de convolution (CONV) qui traite les données d'un champ récepteur.
8. La couche de correction (ReLU), souvent appelée par abus 'ReLU' en référence à la fonction d'activation (Unité de rectification linéaire).
9. La couche de pooling (POOL), qui permet de compresser l'information en réduisant la taille de l'image intermédiaire (souvent par sous-échantillonnage).
10. La couche « entièrement connectée » (FC), qui est une couche de type perceptron.
11. La couche de perte (LOSS).

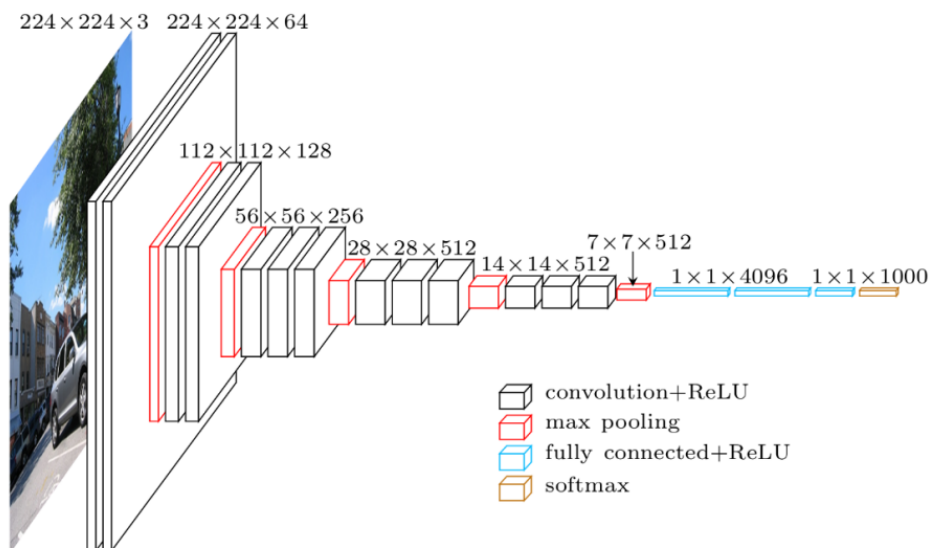


Figure 2: Exemple de modèle de CNN [18]

#### 6.1.1. Couche de convolution

Elle qui consiste à appliquer un **filtre de convolution** à l'image pour détecter des caractéristiques de l'image. Une image passe 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 [19].

#### 6.1.2. Couche de correction

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.

La fonction ReLU :  $F(x) = \max(0, x)$ . Cette fonction force les neurones à retourner des valeurs positives.

#### 6.1.3. Couche de pooling

Cette couche consiste à réduire progressivement la taille de l'image en ne gardant que les informations les plus importantes, par exemple pour chaque groupe de 4 pixels, le pixel ayant la valeur maximale (*Max Pooling, le plus populaire*) ou la moyenne des pixels (AVG pooling).

Avec la couche pooling, la quantité de paramètres et de calcul dans le réseau sont réduites, et cela va permettre de contrôler le sur-apprentissage.

#### 6.1.4. Couche entièrement connectée

Après plusieurs couches de convolution et de max-pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées. Les neurones dans une couche entièrement connectée ont des connexions vers toutes les sorties de la couche précédente. Leurs fonctions d'activations peuvent donc être calculées avec une multiplication matricielle suivie d'un décalage de polarisation.

#### 6.1.5. Couche de perte

La couche de perte spécifie comment l'entraînement du réseau pénalise l'écart entre le signal prévu et réel. Elle est normalement la dernière couche dans le réseau. Diverses fonctions de perte adaptées à différentes tâches peuvent y être utilisées. La fonction « **Softmax** » permet de calculer la distribution de probabilités sur les classes de sortie.

## 7. Classification des images

La classification des images 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.

En général reconnaître une image est une tâche aisée pour un humain au fil de son existence, il a acquis des connaissances qui lui permettent de s'adapter aux variations qui résultent de conditions différentes d'acquisition. Il lui est par exemple relativement simple de reconnaître un

objet dans plusieurs orientations partiellement caché par un autre de près ou de loin et selon diverses illuminations.

Toutefois les progrès technologiques en termes d'acquisition d'images (microscopes, caméras, capteurs) et de stockage engendrent des bases de données riche en information et multiplient les domaines d'applications, il devient alors difficile pour l'humain d'analyser le nombre important d'images, le temps requis le caractère répétitif de la tâche et la concentration nécessaire sont problématiques. Toutefois celle-ci n'est pas forcément aisée pour un programme informatique pour lequel une image est un ensemble de valeur numérique

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

Les applications de la classification automatique d'images sont nombreuses et vont de l'analyse de documents à la médecine en passant par le domaine militaire. Ainsi on retrouve des applications dans le domaine médical comme la reconnaissance de cellules et de tumeurs, la reconnaissance d'écriture manuscrite pour les chèques les codes postaux. 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, d'iris.

Le point commun à toutes ces applications est qu'elles nécessitent la mise en place d'une chaîne de traitement à partir des images 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 méthodes appropriées pour une performance globale optimale à savoir la phase d'extraction de caractéristiques et la phase d'apprentissage. Typiquement, nous disposons de données images desquelles il nous faut extraire des informations pertinentes traduites sous formes de vecteurs numériques. Cette phase d'extraction nous permet de travailler dans un espace numérique. Il s'agit ensuite d'élaborer dans la phase d'apprentissage, à partir de ces données initiales, une fonction de décision pour décider de l'appartenance d'une donnée nouvelle à l'une des classes en présence [20].

## 8. Méthodologie de travail

La méthode CRISP (initialement connue comme CRISP-DM) a été au départ développée par IBM dans les années 60 pour réaliser les projets Datamining. Elle reste aujourd'hui la seule méthode utilisable efficacement pour tous les projets Data Science.

Cette méthode est agile et itérative, c'est-à-dire que chaque itération apporte de la connaissance métier supplémentaire qui permet de mieux aborder l'itération suivante. C'est d'ailleurs pour cette raison que, même si nous la vendons comme un projet, la Data Science est plus une démarche globale qu'un simple projet.

La méthode CRISP a été officiellement adoptée par Business & Decision et son utilisation constitue donc un facteur déterminant à la réussite des projets Data Science [21].

## 8.1. Mode d'emploi

La méthode CRISP se décompose en 6 étapes allant de la compréhension du problème métier au déploiement et la mise en production [22].

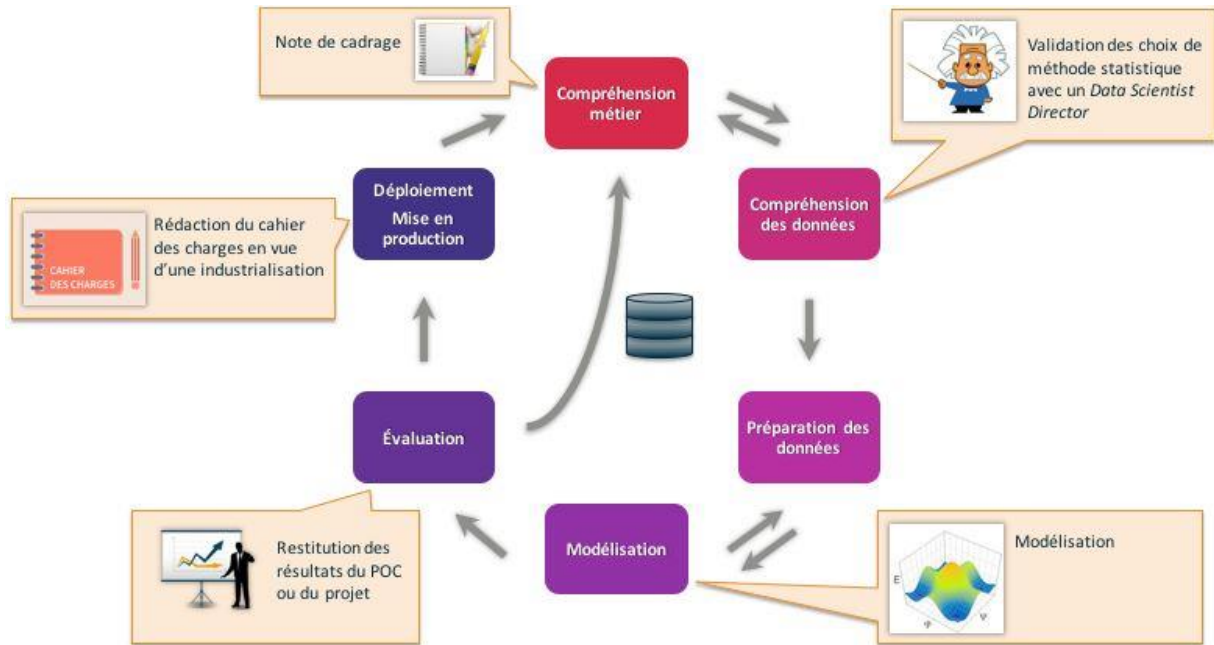


Figure 3: Méthode CRISP

### 8.1.1. La compréhension du problème métier

La première étape consiste à bien comprendre les éléments métiers et problématiques que la Data Science vise à résoudre ou à améliorer.

### 8.1.2. La compréhension des données

Cette phase vise à déterminer précisément les données à analyser, à identifier la qualité des données disponibles et à faire le lien entre les données et leur signification d'un point de vue métier. La Data Science étant basée sur les données seules, les problèmes métiers relatifs à des données existantes, qu'elles soient internes ou externes, peuvent ainsi être résolus par la Data Science.

### 8.1.3. La construction du Data Hub

Cette phase de préparation des données regroupe les activités liées à la construction de l'ensemble précis des données à analyser, faite à partir des données brutes. Elle inclut ainsi le classement des données en fonction de critères choisis, le nettoyage des données, et surtout leur recodage pour les rendre compatibles avec les algorithmes qui seront utilisés.

La paramétricité des données numériques et leur recodage en données catégorielles sont extrêmement importantes et à réaliser avec soin afin d'éviter que les algorithmes utilisés donnent

des résultats faux dans la phase suivante. Toutes ces données doivent en effet être centralisées dans une base de données structurée et qui porte le nom de Data Hub.

#### *8.1.4. La modélisation*

C'est la phase de Data Science proprement dite. La modélisation comprend le choix, le paramétrage et le test de différents algorithmes ainsi que leur enchaînement, qui constitue un modèle. Ce processus est d'abord descriptif pour générer de la connaissance, en expliquant pourquoi les choses se sont passées. Il devient ensuite prédictif en expliquant ce qu'il va se passer, puis prescriptif en permettant d'optimiser une situation future.

#### *8.1.5. L'évaluation*

L'évaluation vise à vérifier le(s) modèle(s) ou les connaissances obtenues afin de s'assurer qu'ils répondent aux objectifs formulés au début du processus. Elle contribue aussi à la décision de déploiement du modèle ou, si besoin est, à son amélioration. A ce stade, on teste notamment la robustesse et la précision des modèles obtenus.

#### *8.1.6. Le déploiement*

Il s'agit de l'étape finale du processus. Elle consiste en une mise en production pour les utilisateurs finaux des modèles obtenus. Son objectif : mettre la connaissance obtenue par la modélisation, dans une forme adaptée, et l'intégrer au processus de prise de décision.

Le déploiement peut ainsi aller, selon les objectifs, de la simple génération d'un rapport décrivant les connaissances obtenues jusqu'à la mise en place d'une application, permettant l'utilisation du modèle obtenu, pour la prédiction de valeurs inconnues d'un élément d'intérêt.

## 9. Conclusion

Ce chapitre a été consacré pour la présentation des notions de bases de l'Intelligence Artificielle, Machine Learning, Deep Learning ainsi que les Réseaux de Neurones Convolutifs. Nous présenterons le cadre de notre projet dans le chapitre suivant.

# **Cadre du projet**

# 1. Introduction

Ce chapitre est consacré à la compréhension métier et données du projet. Ainsi que la préparation des données que nous allons l'utiliser pour la réalisation de notre travail. En effet, nous présentons la problématique, les outils software et hardware et les caractéristiques des données.

## 2. Compréhension métier

Le but de notre projet est de réaliser une application web pour la classification des images.

### 2.1. Problématique étudiée

Nous allons nous intéresser à la problématique de la classification d'image qui est la tâche d'attribuer à une image d'entrée  $x$  un label  $y$  à partir d'un ensemble fixe de catégories. C'est l'un des problèmes fondamentaux de la vision par ordinateur qui, malgré sa simplicité, a une grande variété d'applications pratiques.

### 2.2. Présentation des outils

Dans cette partie, nous allons présenter les outils software et hardware que nous avons utilisé pour la réalisation de notre projet.

#### 2.2.1. Software

Plusieurs framework open sources sont disponibles dans la littérature, la grande majorité supporte le langage Python.

##### 2.2.1.1. Python

Python est un langage de programmation open source créé par le programmeur Guido van Rossum en 1991. Il tire son nom de l'émission Monty Python's Flying Circus. Le logo python est présenté par la figure 4.



Figure 4: Logo python

Il s'agit d'un langage de programmation interprété, qui ne nécessite donc pas d'être compilé pour fonctionner. Un programme « interpréteur » permet d'exécuter le code Python sur



n'importe quel ordinateur. Ceci permet de voir rapidement les résultats d'un changement dans le code. En revanche, ceci rend ce langage plus lent qu'un langage compilé comme le C.

En tant que langage de programmation de haut niveau, Python permet aux programmeurs de se focaliser sur ce qu'ils font plutôt que sur la façon dont ils le font. Ainsi, écrire des programmes prend moins de temps que dans un autre langage. Il s'agit d'un langage idéal pour les débutants.

Python est utilisé dans le domaine de la science des données et du Machine Learning. Avec l'essor de l'analyse de données dans toutes les industries, c'est d'ailleurs devenu l'un de ses principaux cas d'usage.

La grande majorité des bibliothèques utilisées pour la science des données ou le Machine Learning ont des interfaces Python. Ainsi, ce langage est devenu l'interface de commande de haut niveau la plus populaire pour les bibliothèques de Machine Learning et autres algorithmes numériques [23].

### TensorFlow

Développé par Google Brain, TensorFlow est une bibliothèque de Machine Learning. Le logo TensorFlow est présenté par la figure 5. Ses graphiques de data flow et son architecture flexible permettent d'effectuer des opérations et des calculs de données à l'aide d'une API unique sur de multiples CPU ou GPU depuis un PC, un serveur ou même un appareil mobile [23].



*Figure 5: Logo TensorFlow*

### Theano

Theano est une bibliothèque basée sur Python permettant d'effectuer des opérations mathématiques sur des tableaux multidimensionnels. Le logo theano est présenté par la figure 6.



*Figure 6: Logo theano*

La structure utilise le processeur graphique au lieu du processeur, ce qui conduit à des niveaux de productivité plus élevés.

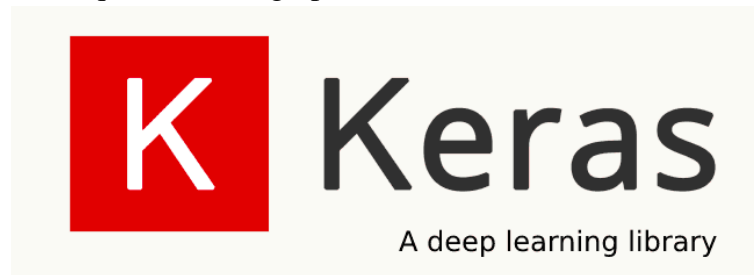
Avec Theano, on peut créer des modèles d'apprentissage approfondi ou des bibliothèques de wrapper.

Un autre avantage est la capacité du cadre de gérer les types de calcul nécessaires pour les algorithmes de réseau neuronal de grande taille.

### Keras

Keras est une librairie de réseaux de neurones minimaliste et modulaire. Elle utilise Theano ou TensorFlow comme backend. Grâce à sa facilité et sa simplicité, Keras permet de mettre en place des projets rapidement. Le logo Keras est présenté par la figure 7.

Keras possède des algorithmes pour les couches d'optimisation, de normalisation et d'activation. Il traite également des réseaux de neurones de convolution et permet de construire des réseaux à base de séquences et de graphes [24].



*Figure 7: Logo Keras*

### Scikit-learn

Scikit-learn ou Sklearn est une bibliothèque basée sur Python permettant de créer des modèles d'apprentissage automatique. Le logo Scikit-learn est présenté par la figure 8.



*Figure 8: Logo Scikit-learn*

Il fournit de nombreux algorithmes d'apprentissage pour la régression, le regroupement et la classification.

Sklearn est compatible avec NumPy et SciPy. Cela signifie que vous pourrez interagir facilement avec différentes bibliothèques pour Python. Plus d'informations sur cette bibliothèque seront fournies dans les sections suivantes [25].

### Flask

Flask a été créé initialement par Armin Ronacher comme étant un poisson d'avril. Le souhait de Ronacher était de réaliser un Framework web contenu dans un seul fichier Python mais pouvant maintenir des applications très demandées. Le logo Flask est présenté par la figure 9.



*Figure 9: Logo Flask*

Flask, un « micro Framework » en Python permettant de créer des service REST en quelques lignes [26].

Il est donc recommandé d'utiliser des distributions Python qui proposent un packaging cohérent de toutes ces librairies et offrent un minimum d'optimisations pour les machines. Parmi les plus répandues on trouve Anaconda.

### Anaconda

Anaconda est peut-être la distribution la plus répandue en raison de la diversité des plateformes qu'elle supporte et d'une plus grande ouverture des outils sur lesquels elle se base.

Elle propose une distribution communautaire et partiellement optimisée ainsi qu'une version payante mieux optimisée offrant de nombreux outils complémentaires. Elle est « cross-platform » et s'appuie sur l'outil de virtualisation d'environnement « conda ». Elle inclue plus de 100 librairies préinstallées et en propose 600 de plus en téléchargement [27] [28]. Le logo Anaconda est présenté par la figure 9.



*Figure 10: Logo Anaconda*

Anaconda est fourni avec l'environnement de travail SPYDER que nous allons utiliser pour la réalisation de notre projet.

### *2.2.2. Hardware*

Le Deep Learning est un domaine avec des exigences en calculs intenses et la disponibilité des ressources (surtout en GPU) dédiés à cette tâche vont fondamentalement influencer sur l'expérience de l'utilisateur car sans ses ressources, il faudra trop de temps pour apprendre de ses erreurs ce qui peut être décourageant.

Les expérimentations ont tous été effectuées sur des machines qui offre des performances acceptables dont les caractéristiques sont présentées dans la table 1.

*Tableau 1 : Caractéristiques des machines utilisées*

CPU	Intel Core i7- 6500U (3.1 GHz)		
GPU	Intel® HD Graphics 520 et NVIDIA GeForce 920MX		
RAM	8 GB		

## 3. Compréhension donnée

Une image est une représentation visuelle voire mentale de quelque chose (objet, être vivant et/ou concept). Elle peut être naturelle (ombre, reflet) ou artificielle (peinture, photographie), visuelle ou non, elle peut entretenir un rapport de ressemblance directe avec son modèle ou au contraire y être liée par un rapport plus symbolique. Elle correspond à une surface 2D, dont chaque point de la surface absorbe la lumière selon une longueur d'onde particulière.

Une image numérique est une image disposant d'un nombre fini de points appelés pixels, codant une information lumineuse dans une gamme finie d'une longueur d'onde [29].

### 3.1. Types de format d'image

- **Image couleur RVB** : L'œil humain analyse la couleur à l'aide de trois types de cellules photo 'les cônes'. Ces cellules sont sensibles aux basses, moyennes, ou hautes fréquences (rouge, vert, bleu). Pour représenter la couleur d'un pixel, il faut donc donner trois nombres, qui correspondent au dosage de trois couleurs de base : Rouge, Vert, Bleu. On peut ainsi représenter une image couleur par trois matrices chacune correspondant à une couleur de base.
- **Image d'intensités** : C'est une matrice dans laquelle chaque élément est un réel compris entre 0 (noir) et 1 (blanc). On parle aussi d'image en niveaux de gris, car les valeurs comprises entre 0 et 1 représentent les différents niveaux de gris.
- **Image binaire** : Une image binaire est une matrice rectangulaire dans l'élément valent 0 ou 1. Lorsque l'on visualise une telle image, les 0 sont représentés par du noir et les 1 par du blanc [30].

### 3.2. Caractéristiques de l'image

L'image est un ensemble structuré d'information caractérisé par les paramètres suivants [31] [32] :

#### 3.1. Pixel

Le pixel est l'abréviation du mot « Picture élément », est une unité de surface permettant de définir la base d'une image numérique. Il matérialise un point donné (x, y) du plan de l'image. L'information présentée par le pixel est le niveau de gris (ou la couleur) prélevée à l'emplacement correspondant dans l'image réelle. La différence entre image monochrome et image couleur réside dans la quantité d'informations contenue dans chaque pixel, par exemple dans une image couleur (RVB : Rouge, Vert, Bleu) la valeur d'un pixel est représentée sur trois octets pour chaque couleur.

#### 3.2. Dimension & Résolution

La dimension est la taille de l'image. Elle se présente sous forme d'une matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multipliée par le nombre de colonnes nous donne le nombre total de pixels dans une image.

Par contre, la résolution est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateur, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels horizontaux et verticaux sur un moniteur. Plus ce nombre est grand, plus la résolution est meilleure.

#### 3.3. Voisinage

Le plan de l'image est divisé en termes de formes rectangulaires ou hexagonales permettant ainsi l'exploitation de la notion de voisinage (voir figure 1). Le voisinage d'un pixel est formé par l'ensemble des pixels qui se situent autour de ce même pixel. On définit aussi l'assiette comme étant l'ensemble de pixels définissant le voisinage pris en compte autour d'un pixel.

On distingue deux types de voisinage :

Voisinage à 4 : On ne prend en considération que les pixels qui ont un coté commun avec le pixel considéré.

Voisinage à 8 : On prend en compte tous les pixels qui ont au moins un point en liaison avec le pixel considéré.

### 3.4. Niveau de gris

C'est la valeur d'intensité lumineuse d'un pixel. Cette valeur peut aller du noir (0) jusqu'au blanc (255) en passant par les nuances qui sont contenues dans l'intervalle [0, 255]. Elle correspond en fait à la quantité de la lumière réfléchie.

Pour 8 bits, on dispose de 256 niveaux de gris dont 40 sont reconnus à l'œil nue. Plus le nombre de bit est grand plus les niveaux sont nombreux et plus la représentation est fidèle.

### 3.5. Contraste

C'est l'opposition marquée entre deux régions d'une image. Une image contrastée présente une bonne dynamique de la distribution des valeurs de gris sur tout l'intervalle des valeurs possibles, avec des blancs bien clairs et des noirs profonds. Au contraire une image peu contrastée a une faible dynamique, la plupart des pixels ayant des valeurs de gris très proches.

### 3.6. Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet.

Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes) ;
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir ; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites.

### 4.7. Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur. C'est un parasite qui représente certains défauts (poussière, petits nuages, baisse momentanée de l'intensité électrique sur les capteurs, ...etc.). Il se traduit par des taches de faible dimension et dont la distribution sur l'image est aléatoire.

### 4.8. Contour

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentant une différence significative.

Dans une image numérique, les contours se situent entre les pixels appartenant à des régions ayant des intensités moyennes différentes ; il s'agit de contours de type « saut d'amplitude ». Un contour peut également correspondre à une variation locale d'intensité présentant un maximum ou un minimum ; il s'agit alors de contour « en toit ».

## 4. Préparation des données

Il existe plusieurs repositories en ligne d'ensembles de données spécifiques au Machine Learning. Ces ensembles de données sont généralement nettoyés au préalable et permettent de tester les algorithmes très rapidement.

### 4.1. Choix des datasets

Pour la réalisation de notre travail, nous avons choisi de travailler sur trois datasets différentes.

#### 4.1.1. Gender Dataset

Il s'agit d'un ensemble de données collecté et nettoyé manuellement contenant 3.354 photos (jpg) d'hommes (1414 fichiers) et de femmes (1940 fichiers). Les catégories peuvent être vues dans la figure ci-dessous. Cet ensemble est disponible sur Kaggle. C'est une plateforme web organisant des compétitions en science des données. Sur cette plateforme, les entreprises proposent des problèmes en science des données et offrent un prix aux data logistes obtenant les meilleures performances. L'entreprise a été créée en 2010 par Anthony Goldbloom [33].



Figure 11: Gender Dataset

#### 4.1.2. MPEG-7 Dataset

MPEG-7 CE Shape-1 Part B est une base de données d'images binaires développée pour tester les descripteurs de forme MPEG-7, et est disponible gratuitement en ligne. Il est utilisé pour tester les descripteurs de contour / image et basés sur le squelette. Les classes d'images varient considérablement et incluent des classes de formes similaires. Il y a 20 instances de chaque classe et 70 classes au total [34]. Les classes peuvent être vues dans la figure 12.

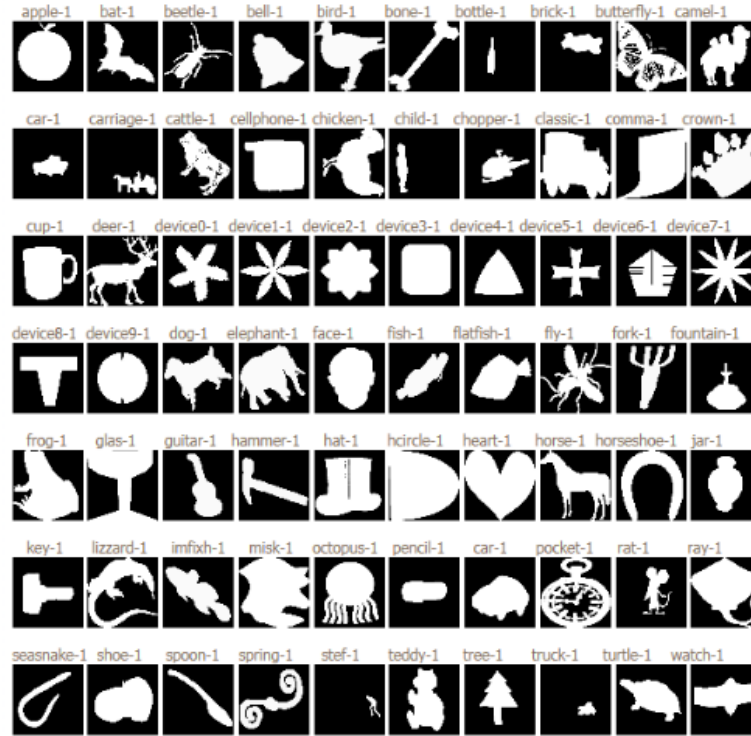


Figure 12: MPEG-7 Dataset

#### 4.1.3. Flowers-17 Dataset

C'est un ensemble de 17 catégories de fleurs avec 80 images pour chaque classe. Les fleurs choisies sont des fleurs communes au Royaume-Uni. Les images ont des variations à grande échelle, de pose et de lumière et il existe également des classes avec de grandes variations d'images au sein de la classe et une similitude proche avec d'autres classes. Les catégories peuvent être vues dans la figure ci-dessous [35].



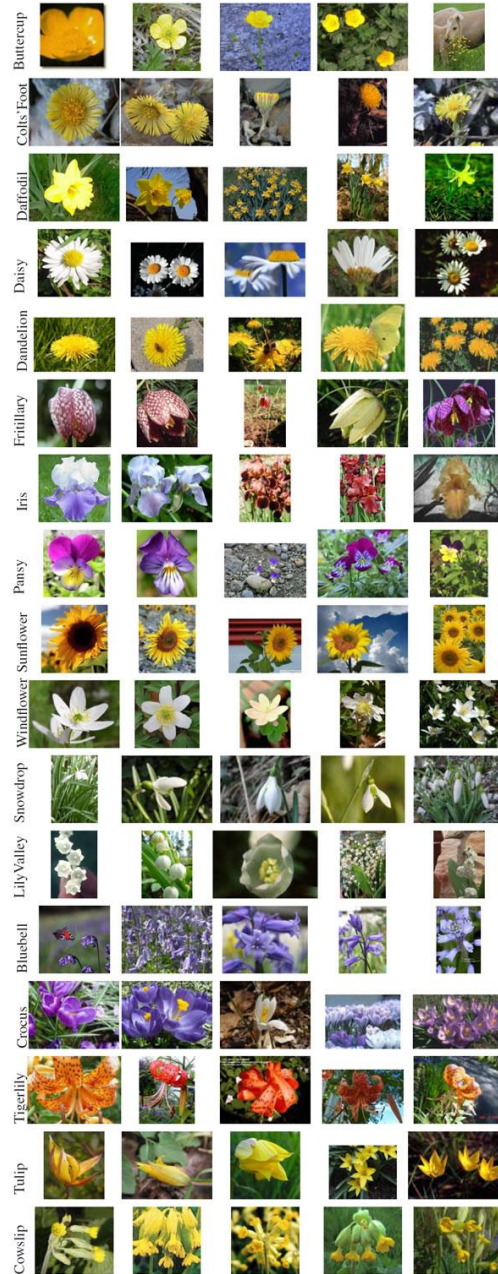


Figure 13: Flowers-17 Dataset

## 5. Conclusion

Ce chapitre a été consacré pour la partie métier et données de notre travail. En effet, nous avons commencé par présenter la problématique étudiée. Ensuite, nous avons cité les outils hardware et software utilisés. Ainsi que les caractéristiques et les types de nos données. Et finalement, nous avons présenté les datasets exploitées. Nous présenterons, le travail réalisé dans le chapitre qui suit.

# **Implémentation et réalisation**

## 9.1. Introduction

Dans ce chapitre, nous avons commencé par présenter les résultats de classification des images en appliquant les algorithmes de ML. Ainsi que les résultats obtenus en utilisant les réseaux de neurones convolutifs.

## 9.2. Modélisation et évaluation

### 9.2.1. Application des algorithmes de ML

Nous avons commencé par appliquer les algorithmes de Machine Learning ; Régression Linéaire, K-plus proches voisins, Arbres de décision, Forêts aléatoires et Machine à Vecteur de Support ; sur la base des données Flowers-17 afin de classer les fleurs.

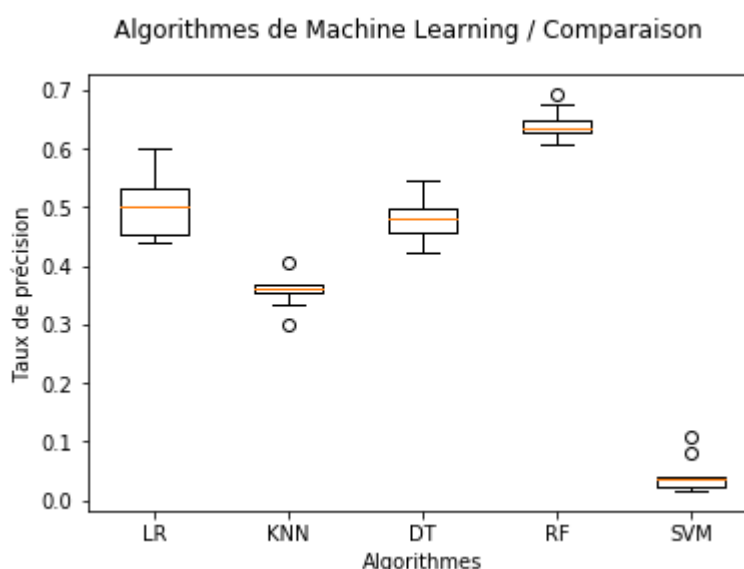


Figure 14: Résultats de Classification

Comme nous pouvons le voir dans le tableau 2 et la figure 14, les précisions ne sont pas si bonnes. L'algorithme des forêts aléatoires (RF) donne une précision maximale de **64,38%**.

Tableau 2: Résultats de classification

Algorithme	Précision en %
Régression Linéaire	<b>50%</b>
K-plus proches voisins	<b>35,7%</b>
Arbres de décision	<b>48,12%</b>
Forêts Aléatoires	<b>64,38%</b>
Machines à Vecteurs de Support	<b>4.33%</b>

Dans cette partie, nous avons choisi aléatoirement des images de fleurs qui n'existent pas dans notre dataset afin de les tester avec le modèle de forêts aléatoires obtenu.

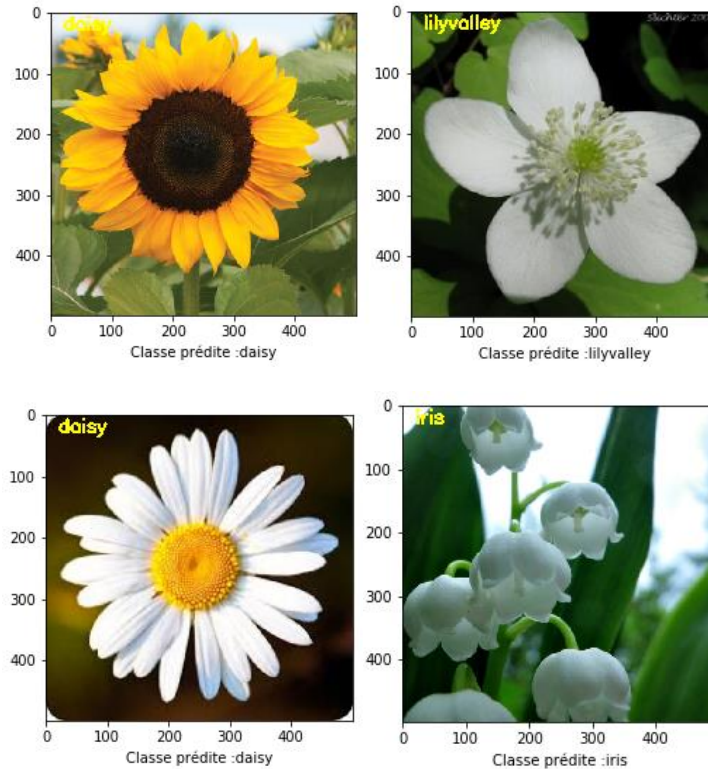


Figure 15: Résultats de prédiction

Nous remarquons que les résultats de prédiction sont mauvais. Notre model a réussi de classifier une seule classe correctement et il a échoué pour les trois autres.

Tableau 3: Résultats de prédiction

Classe réelle	Classe prédite
Sunflower	Daisy
Windflower	Lilyvalley
<b>Daisy</b>	<b>Daisy</b>
Lilyvalley	Iris

### 9.2.2. Application de CNN

Comme nous avons cité dans le premier chapitre, les algorithmes de ML fonctionnent bien pour une grande variété de problèmes. Cependant ils ont échoué à résoudre quelques problèmes majeurs de l'IA telle que la reconnaissance vocale et la reconnaissance d'objets. Donc, nous allons appliquer l'algorithme CNN de DL afin d'obtenir une meilleure précision.

#### 9.2.2.1. Architecture du modèle CNN appliqué

Notre modèle, présenté par la figure ci-dessous, est composé de trois couches de convolution, de trois couches de maxpooling et de deux couches de fully connected. Nous avons appliqué ce modèle sur trois datasets différentes ; Flowers-17, MPEG-7 et Gender.

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 148, 148, 32)	896
activation_1 (Activation)	(None, 148, 148, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_2 (Conv2D)	(None, 72, 72, 32)	9248
activation_2 (Activation)	(None, 72, 72, 32)	0
max_pooling2d_2 (MaxPooling2D)	(None, 36, 36, 32)	0
conv2d_3 (Conv2D)	(None, 34, 34, 64)	18496
activation_3 (Activation)	(None, 34, 34, 64)	0
max_pooling2d_3 (MaxPooling2D)	(None, 17, 17, 64)	0
flatten_1 (Flatten)	(None, 18496)	0
dense_1 (Dense)	(None, 64)	1183808
activation_4 (Activation)	(None, 64)	0
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 17)	1105
activation_5 (Activation)	(None, 17)	0
Total params: 1,213,553		

Figure 16: Modèle CNN appliqué

#### 9.2.2.2. Résultats

Comme nous pouvons le voir dans les figures 17, 18 et 19, les résultats obtenus en appliquant l'algorithme CNN sur les différents Datasets sont très prometteurs.

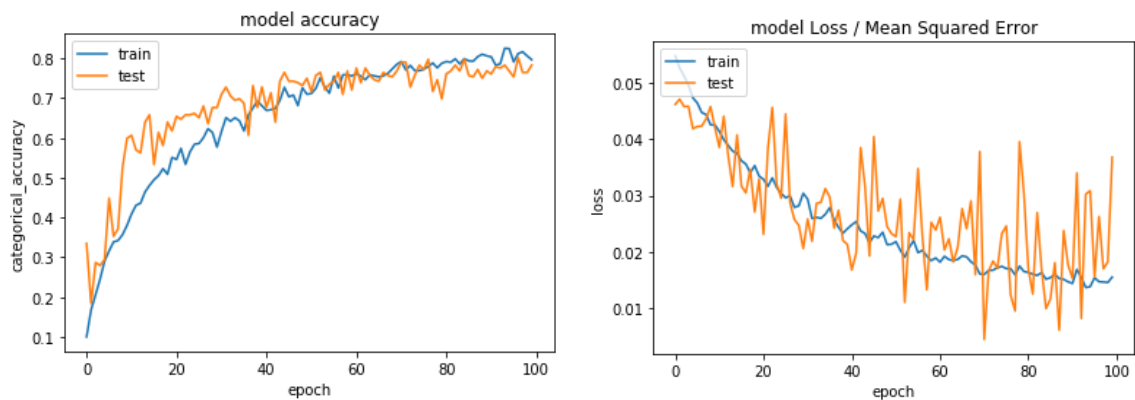


Figure 17: Précision et Erreur – Flowers-17 Dataset

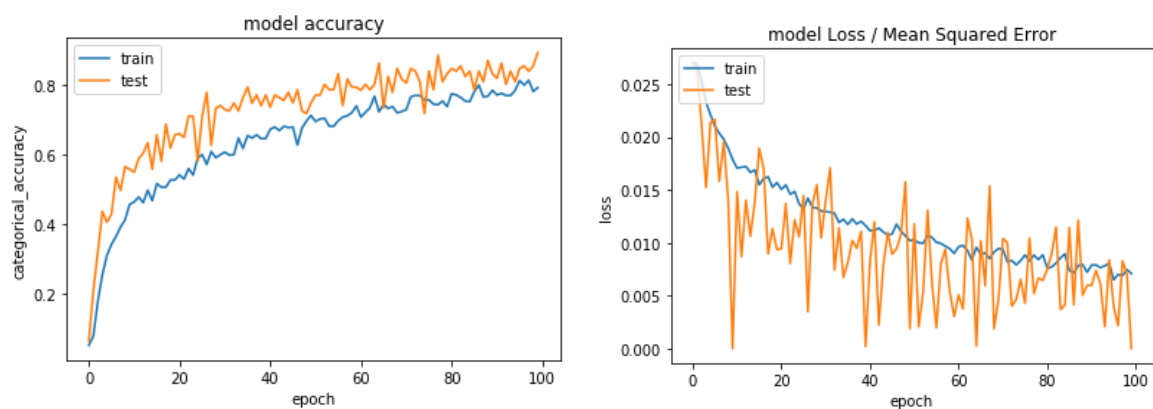


Figure 18: Précision et Erreur – MPEG-7 Dataset

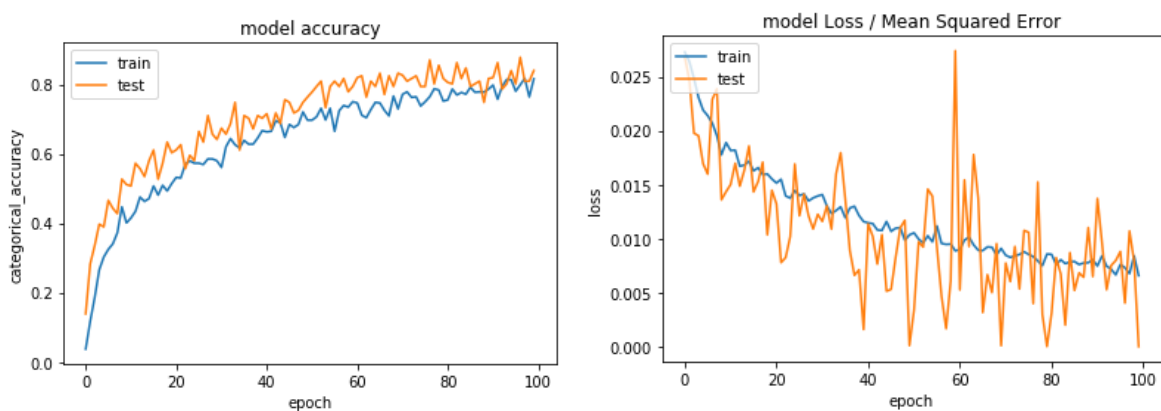


Figure 19: Précision et Erreur – Gender Dataset

Tableau 4 : Taux de précision

Dataset	Taux de précision
Flowers-17	95.93%
MPEG-7	95.17%
Gender	97.78%

Dans cette partie, nous avons utilisé les mêmes images choisies pour le test du modèle de forêts aléatoires afin de tester notre modèle de CNN.

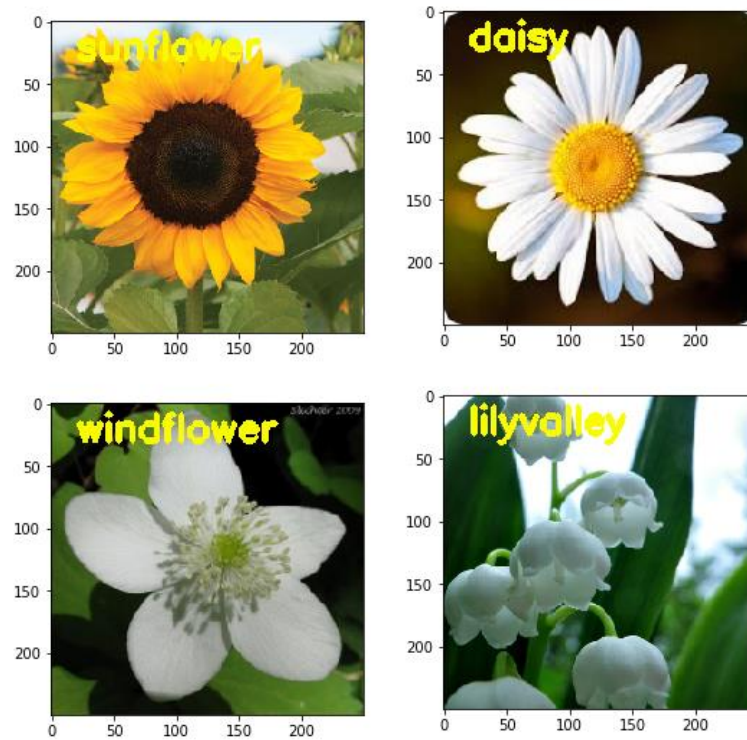


Figure 20: Résultats de prédiction

D'après la figure 20 et le tableau 5 nous concluons que notre model a réussi de classifier toutes les classes correctement.

Tableau 5: Résultats de prédiction

Classe réelle	Classe prédite
Sunflower	Sunflower
Windflower	Windflower
Daisy	Daisy
Lilyvalley	Lilyvalley



Pour évaluer notre modèle pour la dataset MPEG-7, nous avons aussi choisi aléatoirement des images qui n'existent pas dans notre dataset. Les résultats obtenus sont présentés par la figure 21.

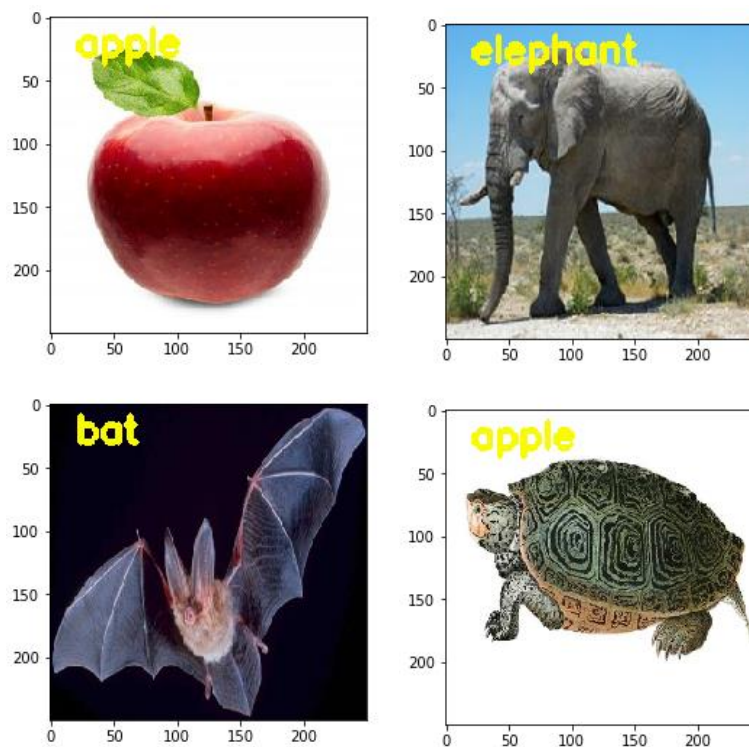


Figure 21: Résultats de prédiction

Nous remarquons par les résultats présentés dans le tableau 6 que notre model a échoué pour classifier une seule classe et a réussi pour les trois autres.

Tableau 6: Résultats de prédiction

Classe réelle	Classe prédite
Apple	Apple
Elephant	Elephant
Bat	Bat
Turtle	Apple

Pour évaluer notre modèle pour la dataset Gender, nous avons aussi choisi aléatoirement des images qui n'existent pas dans notre dataset. Les résultats obtenus sont présentés par la figure 22.





Figure 22: Résultats de prédiction

Nous remarquons par les résultats présentés par la figure ci-dessus que notre model a échoué pour classer une seule personne sur dix personnes.

#### 9.2.2.3. Conclusion

Les résultats obtenus pour les trois datasets en appliquant l'algorithme CNN sont parfaits. Le taux de précision est supérieur à 95%.

Comparés à d'autres algorithmes de classification d'image, les réseaux de neurones convolutifs utilisent relativement peu de pré-traitement. Cela signifie que le réseau est responsable de faire évoluer tout seul ses propres filtres, ce qui n'est pas le cas d'autres algorithmes plus traditionnels. L'absence de paramétrage initial et d'intervention humaine est un atout majeur des CNN.

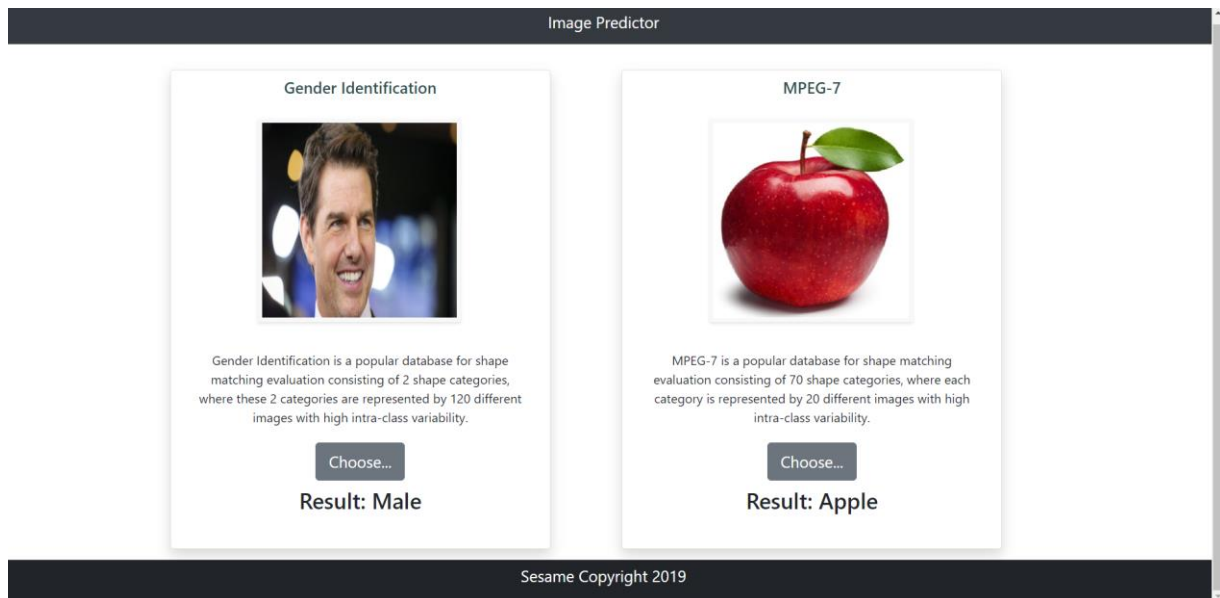
Grâce aux Deep Learning, l'avenir de l'intelligence artificielle est prometteur.

### 9.3. Déploiement

Nous avons tenu à introduire notre réseau de neurones dans une interface web afin de pouvoir visualiser les performances de notre modèle facilement sans être obligé de passer à chaque fois par les lignes de codes. L'architecture globale de notre application web est présenté par la figure 23 et l'interface web par la figure 24.



Figure 23: Architecture de l'application Web



*Figure 24: Interface Web*

## 9.4. Conclusion

Ce chapitre a été consacré pour la présentation de notre travail. Nous avons commencé par appliquer les algorithmes de Machine Learning pour la classification des images. Ensuite, nous testé l'algorithme CNN sur différentes datasets. Et finalement, nous avons présenté l'interface web réalisé pour exploiter et visualiser les résultats obtenus.

# Conclusion générale

Dans ce travail, nous avons exploré le domaine de la classification d'image qui comme tous les autres domaines de l'intelligence artificielle ont connu une évolution majeure depuis l'apparition du Deep Learning.

Nous avons commencé par présenter les notions fondamentales de l'Intelligence Artificielle, Machine Learning, Deep Learning, les Réseaux de Neurones Convolutifs ainsi que la méthodologie de travail adoptée. Nous avons détaillé la problématique de notre travail, présenté les outils software et hardware ainsi que les données utilisées. Et finalement, nous avons présenté le travail réalisé.

Les résultats de classification des images par les réseaux de neurones convolutifs nous permettent de conclure que grâce au Deep Learning, l'avenir de l'intelligence artificielle est prometteur.

Ce projet nous a permis d'approfondir nos connaissances théoriques, acquises tout le long de notre formation, par la pratique des nouvelles technologies. Il nous a permis de maîtriser le langage Python ainsi que la manipulation des bibliothèques de Machine Learning.

Comme perspectives nous pouvons citer :

- Tester autres architectures d'algorithme CNN sur des nouvelles bases de données
- La segmentation des images

## Netographie

- [1] [https://www.entreprises.gouv.fr/files/files/directions\\_services/etudes-et-statistiques/prospective/Intelligence\\_artificielle/2019-02-intelligence-artificielle-etat-de-l-art-et-perspectives.pdf](https://www.entreprises.gouv.fr/files/files/directions_services/etudes-et-statistiques/prospective/Intelligence_artificielle/2019-02-intelligence-artificielle-etat-de-l-art-et-perspectives.pdf)
- [2] <http://penseeartificielle.fr/tout-pour-bien-demarrer-en-ia/>
- [3] [https://www.agrotic.org/wp-content/uploads/2018/12/2018\\_ChairAgroTIC\\_DeepLearning\\_VD2.pdf](https://www.agrotic.org/wp-content/uploads/2018/12/2018_ChairAgroTIC_DeepLearning_VD2.pdf)
- [4] <https://www.lebigdata.fr/machine-learning-et-big-data>
- [5] <https://www.actuia.com/tutoriel/concepts-mathematiques-derriere-le-machine-learning-la-regression-lineaire/>
- [6] <https://mrmint.fr/introduction-k-nearest-neighbors>
- [7] <http://dSPACE.univ-tlemcen.dz/bitstream/112/6391/1/Application-de-k-means.pdf>
- [8] <https://scaron.info/doc/intro-arbres-decision/>
- [9] <https://dataanalyticspost.com/Lexique/random-forest/>
- [10] [http://georges.gardarin.free.fr/Surveys\\_DM/Survey\\_SVM.pdf](http://georges.gardarin.free.fr/Surveys_DM/Survey_SVM.pdf)
- [11] <https://www.lebigdata.fr/perceptron-machine-learning>
- [12] <http://penseeartificielle.fr/focus-reseau-neurones-artificiels-perceptron-multicouche/>
- [13] <http://dSPACE.univ-tlemcen.dz/bitstream/112/12583/1/Deep-Learning-pour-la-classification..pdf>
- [14] <http://dSPACE.univ-tlemcen.dz/bitstream/112/12235/1/Classification-des-images-avec-les-reseaux-de-neurones.pdf>
- [15] <https://www.lebigdata.fr/reseau-de-neurones-artificiels-definition>
- [16] [https://ml4a.github.io/ml4a/fr/neural\\_networks/](https://ml4a.github.io/ml4a/fr/neural_networks/)
- [17] <https://www.lemagit.fr/definition/Reseau-neuronal-convolutif>

- [18] <https://www.jeremyjordan.me/convnet-architectures/>
- [19] <https://mc.ai/comprendre-les-reseaux-de-neurones-convolutifs-cnn/>
- [20] <https://tel.archives-ouvertes.fr/tel-00879356/document>
- [21] <https://www.linkedin.com/pulse/chapter-1-introduction-crisp-dm-framework-data-science-anshul-roy/>
- [22] <https://fr.blog.businessdecision.com/methode-crisp-la-cle-de-la-reussite-en-data-science/>
- [23] <https://www.lebigdata.fr/python-langage-definition>
- [24] <https://le-datascientist.fr/top-10-des-librairies-de-deep-learning-sur-python>
- [25] [https://fr.bitdegree.org/tutos/bibliotheque-python/#AI\\_et\\_Python\\_Bibliotheques\\_utiles](https://fr.bitdegree.org/tutos/bibliotheque-python/#AI_et_Python_Bibliotheques_utiles)
- [26] <https://www.kaherecode.com/tutorial/demarrer-avec-flask-un-micro-framework-python>
- [27] <https://makina-corpus.com/blog/metier/2017/presentation-de-lecosysteme-python-scientifique>
- [28] <https://makina-corpus.com/blog/metier/2017/presentation-de-lecosysteme-python-scientifique>
- [29] <http://imagenumerique.50webs.com/ch1preliminaire.html>
- [30] [https://www.sites.univ-rennes2.fr/arts-spectacle/cian/image\\_numFlash/pdf/chap3\\_tout.pdf](https://www.sites.univ-rennes2.fr/arts-spectacle/cian/image_numFlash/pdf/chap3_tout.pdf)
- [31] <https://www.apprendre-en-ligne.net/info/images/images.pdf>
- [32] <https://www.maxicours.com/se/cours/caracteristiques-d-une-image-numerique/>
- [33] <https://www.kaggle.com/playlist/men-women-classification/data#>
- [34] <http://www.timeseriesclassification.com/description.php?Dataset=ShapesAI>  
!
- [35] <http://www.robots.ox.ac.uk/~vgg/data/flowers/17/>