

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 20 Août -1955-Skikda

Faculté des sciences - Département d'Informatique



Mémoire de fin d'études en vue de l'obtention du diplôme de

MASTER en informatique

Option : Systèmes informatiques

Thème

**La reconnaissance des images par CNN
(Convolutional Neural Network)**

Réalisé par :
Boufedah badissi Romayssa

Encadreur :
Dr. Mansoul A.

Juin 2023

Dédicace

Je dédie ce modeste travail :

A mon cher père que dieu lui fasse miséricorde qui m'a encouragé tout au long de mon parcours universitaire. Ma mère, la lumière de ma vie, qui a tout fait pour ma réussite et mon bonheur.

A mes frères AbdAraouf, AbdArahim, AbdElmalek, Brahim, AbdElhadi, AbdElaziz et mes sœurs Khadidja, Wafa, et à mon binôme Rahma, à mes chères sœurs à tous les membres de ma famille pour leur aide et courage.

A tous mes enseignants durant mes années d'études avec lesquels j'ai beaucoup appris, ainsi qu'à toutes les personnes que j'ai connues, qui m'ont aidé, soutenue et encouragé.

Romayssa,

Remerciements

Je remercie en premier lieu mon DIEU de nous avoir aidé à défier tous les obstacles, afin de compléter ce mémoire.

C'est avec un immense plaisir que j'exprime mes remerciements ; les plus chaleureux à notre encadreur Monsieur MANSOUL A, qui a toujours suivi ce travail avec intérêt et qui l'a soutenu par des encouragements de toutes sortes. Nous adressons aussi nos vifs remerciements à Melle BRAHIMI Roumayssa pour le temps qu'elle a consacré et pour les précieuses informations qu'ils nous ont prodigués avec intérêt.

Je remercie très vivement les membres du jury qui ont accepté de juger ce modeste travail.

Enfin, pour tous ceux qui, de près ou de loin, ont rendu ce travail possible, merci à tous.

Résumé

La reconnaissance d'images est considérée comme l'une des tâches les plus importantes et les plus complexes en intelligence artificielle et en traitement d'images. Elle implique l'utilisation d'algorithmes et de modèles pour analyser et interpréter le contenu visuel d'une image. La reconnaissance d'images trouve de nombreuses applications pratiques, notamment dans la classification d'images, la détection d'objets, la reconnaissance faciale, la reconnaissance de caractères, la vision par ordinateur, etc.

Cela a conduit de nombreux chercheurs à explorer plusieurs techniques basées sur des données brutes pour la reconnaissance d'objets les au vu des caractéristiques apparentes.

Ceci nous a incités à nous pencher vers une approche basée sur la reconnaissance à l'aide d'un CNN afin de reconnaître le contenu d'une image.

Cette approche est testé sur un data set d'images qui présentent plusieurs catégories d'animaux avec analyse de ces de différentes images, et pour un cas donné nous utiliserons le modèle CNN pour reconnaître l'animal en question.

Notre approche nous a permis d'atteindre un taux de reconnaissance de 98.93 %.

Mots clés : image, reconnaissance, CNN, réseau de neurones, convolution.

Abstract

Image recognition is considered one of the most important and complex asks in artificial intelligence and image processing. It involves the use of algorithms and models to analyze and interpret the visual content of an image. Image recognition has many practical applications, including image classification, object detection, facial recognition, character recognition, computer vision, and more.

This has led many researchers to explore various techniques based on raw data for object recognition based on their apparent features. This has prompted us to adopt an approach based on recognition using a Convolutional Neural Network (CNN) to identify the content of an image.

We have tested this approach on a dataset of images that contain various categories of animals, analyzing different images. For a given case, we use the CNN model to recognize the specific animal. Our approach has a chie veda recognition rate of 98.93%.

Keywords : Image, Recognition, CNN, Neural Network, Convolution.

Table des matières

| | Page |
|--|-------|
| Dédicace | / |
| Remerciements | / |
| Résumé | / |
| Abstract | / |
| Introduction générale | 01 |
| Chapitre 1 : L'image | 03-10 |
| 1.1. Introduction | 03 |
| 1.2 Les types d'image | 04 |
| 1.3 Le codage des couleurs | 07 |
| 1.4 Quelques caractéristiques de l'image | 08 |
| 1.5. Conclusion | 10 |
| Chapitre 2. Le traitement des images et la classification | 11-22 |
| 2.1 Introduction | 11 |
| 2.2 Les images | 11 |
| 2.3 Les traitement numérique des images | 12 |
| 2.4. Les méthodes de classification | 18 |
| 2.5 Les réseaux de neurones | 21 |
| 2.6 Conclusion | 22 |
| Chapitre 3 Les réseaux de neurones convolution(CNN) | 23-40 |
| 3.1. Introduction | 23 |
| 3.2.Le Neurone | 23 |
| 3.3. Le Réseau de neurones multicouches | 27 |
| 3.4 Le Réseau de neurones convolution(CNN) | 28 |
| 3.5 Les compositions | 32 |
| 3.6 Architecture générale du CNN | 38 |
| 3.7. Conclusion | 40 |
| Chapitre 4 La reconnaissance d'image par CNN | 41-53 |
| 4.1 Introduction | 41 |
| 4.2 Architecture générale du CNN | 41 |
| 4.3 Environnement de développement | 43 |
| 4.4 Le data set image | 47 |
| 4.5 Conclusion | 53 |
| Conclusion générale et perspectives | 54 |
| Bibliographie | 55-57 |

Liste des figures

| Figure | page |
|--|------|
| Figure 1 : Différence entre les images matricielles et images vectorielles. | 04 |
| Figure 2 : Représentation des pixels. | 05 |
| Figure 3 : Exemple d'histogramme d'une image. | 06 |
| Figure 4 : La transformation ponctuelle d'une image numérique. | 07 |
| Figure 5 : Les niveaux de gris. | 08 |
| Figure 6: Schéma explicatif de la résolution de l'image numérique. | 09 |
| Figure 7: Images compressées en PNG, WebP sans perte et avec perte. | 09 |
| Figure 8 : Binarisation d'une image | 14 |
| Figure 9 : Squelettisation d'une image | 14 |
| Figure 10: Principe du filtre médian. | 17 |
| Figure 11 : Exemple d'arborescence pour un problème à 5 classes. | 20 |
| Figure 12 : Exemple de classification par l'algorithme K-Plus Proches voisins. | 21 |
| Figure 13 : Les réseaux de neurones. | 22 |
| Figure 14 : Représentation d'un neurone biologique | 24 |
| Figure 15 : Le neurone artificiel | 25 |
| Figure 16 : Les modèles de fonctions d'activation | 26 |
| Figure 17 : Correspondance entre le neurone biologique et le neurone formel | 26 |
| Figure 18 : Perceptron multicouche | 27 |
| Figure 19 : L'architecture Le Net | 28 |
| Figure 20 : L'architecture Alex Net | 29 |
| Figure 21 : L'architecture ZF Net | 29 |
| Figure 22 : L'architecture Google Le Net/ Inception | 30 |
| Figure 23 : L'architecture VGG Net | 30 |
| Figure 24 : L'architecture Res Net | 31 |
| Figure 25 : Exemple d'une opération de convolution | 32 |
| Figure 26 : Exemple de réseau composé de nombreuses couches à convolution | 32 |
| Figure 27 : Différents types de convolution | 33 |
| Figure 28 : Pooling avec un filtre 2*2 et un pas de 2 | 34 |
| Figure 29: Example of pooling with a 2×2 filter and a stride of 2 | 35 |
| Figure 30 : Exemple de mise à plat des images finales en sortie (des filtres + simplifications). | 35 |
| Figure 31 : Application de RELU à partir d'une matrice et d'une image | 36 |
| Figure 32 : Un exemple d'opération entièrement connectée (FC). | 37 |
| Figure 33 : Architecture générale d'un CNN. | 38 |
| Figure 34 : Conception générale de système | 41 |
| Figure 35 : L'environnement Google Colaboratory | 43 |
| Figure 36 : Quelques bibliothèques dans le modèle | 45 |
| Figure 37 : Les 6 classes dans le data set. | 46 |
| Figure 38 : Création de classes (classe 'Cat'). | 46 |
| Figure 39 : Définir le nombre total d'images | 47 |
| Figure 40 : prétraitement des images + entraînement + validation | 47 |
| Figure 41: Augmentation des images d'entraînement | 47 |
| Figure 42 : Augmentation des images validation | 48 |
| Figure 43: Redimensionner les images | 48 |
| Figure 44 : Construction du modèle de réseau de neurones convolutif | 49 |
| Figure 45: Compilation du modèle | 49 |
| Figure 46 : résumé du modèle | 50 |
| Figure 47 : Résultat d'entraînement en fonction des Epochs | 50 |
| Figure 48 : Résultat de test en utilisant un exemple d'image de test. | 51 |

Liste des équations

| Equation | Page |
|----------|------|
| 1.1 | 05 |
| 2.1 | 14 |
| 2.2 | 15 |
| 2.3 | 17 |

Liste des tableaux

| Tableau | page |
|---|------|
| Tableau 1: Masque de convolution du filtre passe-bas. | 16 |
| Tableau 2 : Masque de convolution du filtre passe-haut. | 16 |
| Tableau 3 : Masque de convolution du filtre Gaussien. | 17 |
| Tableau 4 : Analogie entre neurone biologique et le neurone artificiel. | 27 |
| Tableau 5 : Résultats de test pour chaque classe. | 51 |

Introduction générale

L'utilisation de réseaux de neurones ne cesse de se généraliser à divers domaines de l'activité humaine. L'accroissement de ce type d'outils pourra être expliqué par deux facteurs :

L'évolution rapide des architectures matérielles permettant l'implémentation de modèles basés apprentissage automatique, jusqu'à récemment étaient difficiles de les déployer sur des architectures conventionnelles. Il s'agit notamment d'architecture permettant le déploiement des réseaux de neurones profonds, ce qui en a résulté en un type de processeurs, dit GPGPU pour "General-pur pose computing on graphic sprocessing units".

La disponibilité de données massives et diversifiées, produites dans un contexte Big Data. Cette explosion de données est occasionnée avec la généralisation quasi-totale de la population mondiale à l'internet notamment via les réseaux sociaux, les plateformes du commerce électronique et des gouvernements électroniques.

Dans ce réservoir de données, les images occupent une place d'excellence, de part de leur facilité d'acquisition, par la généralisation et la démocratisation de l'informatique, et aussi par leur richesse d'information (actuellement des dizaines voire des centaines de mégas octets par image). Les applications basées apprentissage automatiques, dont les données sont visuelles se créent dans les différents domaines. Il s'agit principalement d'applications en sécurité par reconnaissance du visage, en recherche par le contenu, en médecine par les différentes applications médicales dont le diagnostic automatique.

Pour toutes ces applications basées images, les systèmes procèdent par détection des objets d'intérêt dans les images, puis à la classification des objets ou des images selon les spécifications du problème à résoudre. Typiquement en diagnostique automatique basé image, que ce soit en santé humaine, animale ou végétale, le système procède par l'apprentissage automatique de modèles afin d'identifier des régions d'intérêt, 220 des anomalies générales dans les images, ou tout simplement la classification des images selon des critères physiologiques ou diagnostiques.

Dans notre mémoire de master, nous nous sommes intéressées à la reconnaissance des images. Pour ce faire, nous allons recourir aux réseaux de neurones convolutifs, qu'entraîne pour

reconnaitre le contenu des images, nous avons considéré un data set d'images de taille suffisante permettant de reconnaître leur contenu.

Notre mémoire est organisée comme suit :

Au chapitre 1, nous introduisons des notions générales de l'image en donnant leurs définitions et leurs caractéristiques.

Le chapitre 2, donne un aperçu général sur le traitement d'images, et les différentes méthodes utilisées pour la reconnaissance.

Le chapitre 3, sera dédié à la présentation des réseaux de neurones, nous nous penchons plus sur les CNN (les réseaux de neurones convolutifs). C'est ce dernier type de réseaux qui va vous servir pour la reconnaissance des images.

Le chapitre 4, sera consacré à la mise en œuvre de notre modèle, à son implémentation et à la présentation des résultats obtenus. Nous montrons à travers une expérimentation à l'aide d'une plateforme et de bibliothèques qui ont été utilisées pour concevoir et entraîner le modèle considéré.

En **conclusion générale**, nous résumons le travail réalisé, et nous soulignons quelques-unes de ses perspectives.

Chapitre 01 : L'image

1.1 Introduction

L'image constitue l'un des moyens les plus importants qu'utilise l'homme pour communiquer avec autrui. C'est un moyen de communication universel dont la richesse du contenu permet aux êtres humains de tout âge et de toute culture de se comprendre. C'est aussi le moyen le plus efficace pour communiquer, chacun peut analyser l'image à sa manière, pour en dégager une impression et d'en extraire des informations précises.

Dans ce chapitre, on présente quelques principaux concepts qui sont en relation avec notre sujet d'étude.

Définition de l'image

L'image est une représentation visuelle, voire mentale, de quelque chose (objet, être vivant ou concept). Elle peut être naturelle(ombre, reflet) ou artificielle (sculpture, peinture, photographie), visuelle ou non, tangible ou conceptuelle (métaphore), elle peut entretenir un rapport de ressemblance directe avec son modèle ou au contraire y être liée par un rapport plus symbolique.

Platon donne une des plus anciennes définitions de l'image : « J'appelle image d'abord les ombres ensuite les reflets qu'on voit dans les eaux, ou à la surface des opaques, polis et brillants et toutes les représentations de ce genre » [1].

Aussi une autre définition :l'image est une représentation d'une personne ou d'un objet par la peinture, la sculpture, le dessin, la photographie, le film...etc. C'est aussi un ensemble structuré d'informations qui, après l'affichage sur l'écran, ont une signification pour l'œil humain. Elle peut être décrite sous la forme d'une fonction $I(x, y)$ de brillance analogique continue, définie dans un domaine borné, tel que x et y sont les coordonnées spatiales d'un point de l'image et est une fonction d'intensité lumineuse et de couleur. Sous cet aspect ,l'image est inexploitable par la machine, ce qui nécessite sa numérisation [2].

1.2 Types d'images

On rencontre 2 types d'images en informatique : Les images vectorielles (en anglais vector) et les images matricielles (on parle aussi de rasters et de bitmaps)[3].

1.2.1 Images vectorielles

L'image est décrite par des figures géométriques, des calculs. C'est une représentation purement mathématique .Autrement dit, on stocke la succession d'opérations conduisant au tracé dans le cas d'une image vectorielle, alors qu'on mémorise une mosaïque de points élémentaires dans le cas d'image matricielle. Ces images présentent deux avantages : elles occupent peu de place en mémoire et peuvent être redimensionnées sans perte d'information.

1.2.2 Image matricielle (bitmap)

Une image matricielle est une image en mode point. Le système de codage le plus universel consiste en effet à décomposer la représentation graphique, l'image, en un certain nombre de points élémentaires caractérisés par leurs coordonnées spatiales et leur couleur.

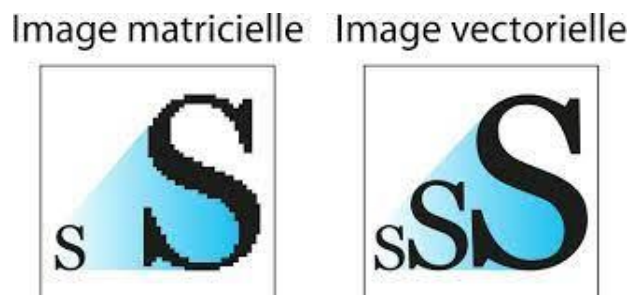


Figure 1 : Différence entre les images matricielles et images vectorielles[4].

L'image est représentée par une matrice (une grille) de pixels (des points de couleur). On peut rapprocher cela du pointillisme en peinture.

1.2.3 Concepts et définitions de base

- Pixel

Le pixel (abréviation venant de l'anglais : PICture Element) est l'élément de base d'une image ou d'un écran, c'est-à-dire un point. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions (largeur et hauteur) constituant l'image [5].



Figure 2 :Représentation des pixels[6].

- **Le codage de pixel**

Pratiquement toujours, la valeur d'un pixel est un mot binaire de longueur K bits, par conséquent un pixel peut prendre l'une des valeurs de l'intervalle $[0...2^k-1]$. La valeur K est appelée profondeur de l'image. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions (une matrice) constituant l'image finalement obtenue [7].

- **La taille d'une image**

La taille d'une image est le nombre de pixels de cette image, la taille d'une image qui est représenté par (328×456) dont 328 est le nombre de lignes, et 456 est le nombre de colonnes, est égale à : $328 \times 456 = 149568$ pixels [7].

- **La luminance**

Le mot luminance est substituée au mot brillance, qui correspond à l'éclat d'un objet. La luminance est le degré de luminosité des pixels de l'image. Elle est définie aussi comme l'intensité d'une source étendue dans une direction donnée, divisée par l'aire apparente de cette source dans cette même direction [8].

- **Le contraste**

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux régions d'une image.

Si L_1 et L_2 sont les degrés de luminosité respectivement de deux régions voisines A_1 et A_2 d'une image, le contraste C est défini par le rapport :

$$C = \frac{L1-L2}{L1+L2} \quad (1.1)$$

- **Le bruit**

Un système d'enregistrement d'images ne restitue pas l'image de manière parfaite. En effet des informations parasites s'ajoutent de manière aléatoire aux détails de la scène d'origine, c'est ce qu'on appelle : bruit.

Le bruit n'a pas les mêmes origines, mais provoque des effets assez semblable scommela perte de netteté sur les détails ou l'apparition de grains.

- **L'histogramme**

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui associe à chaque valeur d'intensité le nombre de pixels de l'image ayant cette valeur.

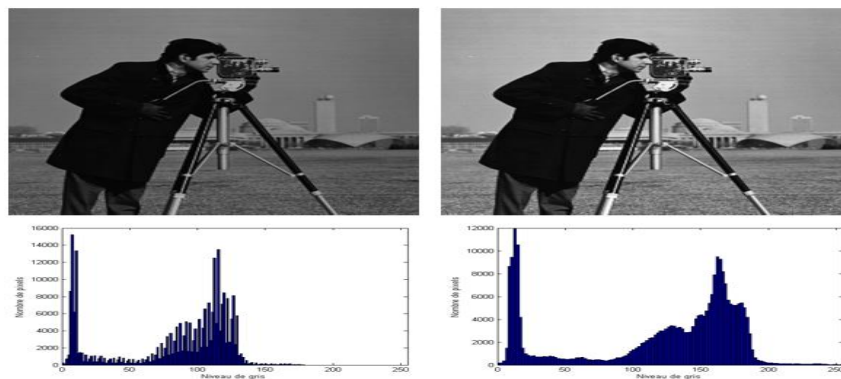


Figure 3 : Exemple d'histogramme d'une image [9].

- **La palette des couleurs**

On appelle la palette des couleurs, la liste des couleurs que peut contenir une image.

La valeur de chaque pixel représente le rang d'une couleur dans cette liste.

Il est fréquent de voir des images qui n'utilisent jamais certaines couleurs, il devient dès lors intéressant de limiter la palette de couleur en ne sélectionnant que la ou les couleurs utilisées réellement par l'image (Figure 4).

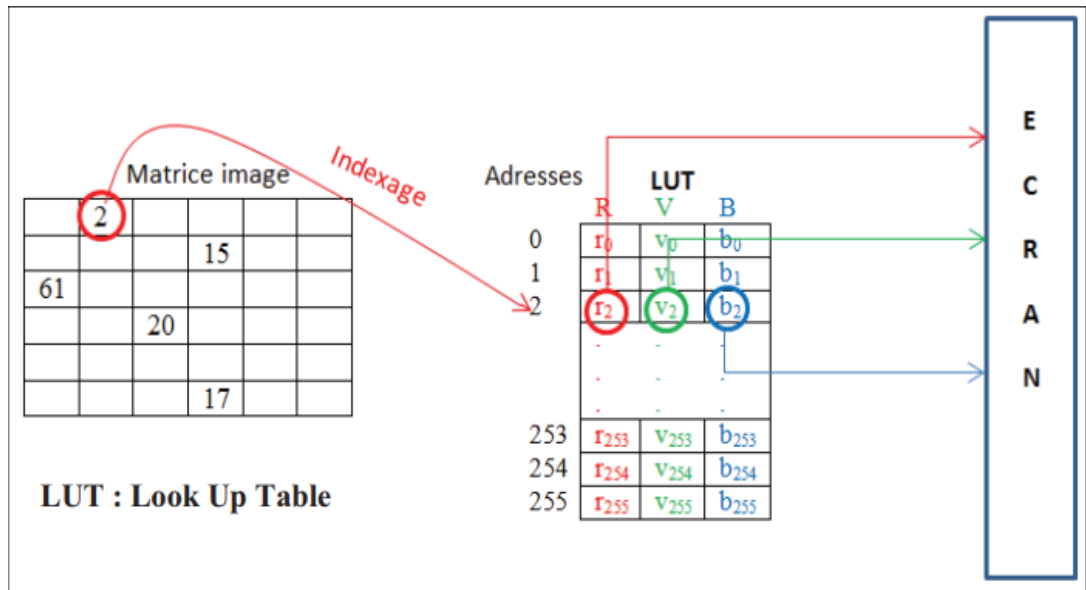


Figure 4 : La transformation ponctuelle d'une image numérique [7].

1.3 Le codage des couleurs

1.3.1 Les images binaires (noir ou blanc)

Les images binaires sont des images de profondeur $K=1$ bit, donc un pixel peut prendre l'une des valeurs : noir ou blanc (0 ou 1). C'est typiquement le type d'images que l'on utilise pour scanner du texte quand celui-ci est composé d'une seule couleur [10].

1.3.2 Les images en niveaux de gris

En général, les images en niveaux de gris sont des images de profondeur $k=8$ bits, donc chaque pixel peut prendre l'une des valeurs de l'intervalle $[0...255]$, où la valeur 0 représente la brillance minimale (le noir) et 255 la brillance maximale (le blanc). Ce type d'image est fréquemment utilisé pour reproduire des photos en noir et blanc ou du texte.

Dans plusieurs applications professionnelles de photographie et d'impression ainsi qu'en médecine et astronomie, 8 bits par pixel n'est pas suffisant, pour cela il existe d'autres types d'images en niveaux de gris de profondeur $K=12$, $K=14$ ou $K=16$ bits.

Une image où chaque pixel est représenté par une seule valeur correspondant à l'intensité de gris. La valeur d'intensité varie de 0 (noir) au niveau maximum (blanc), avec des valeurs intermédiaires représentant différentes nuances de gris. Les images en niveaux de gris sont couramment utilisées dans

diverses tâches de traitement d'images, telles que l'amélioration d'image, la segmentation et l'extraction de caractéristiques [11].

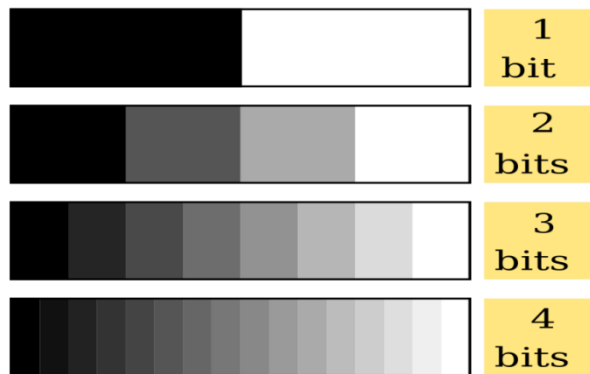


Figure 5 : Les niveaux de gris [12].

1.3.3 Les images couleurs

L'espace couleur est basé sur la synthèse additive des couleurs, c'est-à-dire que le mélange entre différentes couleurs (trois, quatre...) donne une couleur. La plupart des images couleurs sont basées sur trois couleurs primaires : Rouge, Vert et Bleu (RVB) (RGB en anglais), et utilisent typiquement 8 bits pour chaque composante de couleur, donc chaque pixel nécessite $3 \times 8 = 24$ bits pour coder les trois composantes, et chaque composante de couleur peut prendre l'une des valeurs de l'intervalle [0...255].

1.4 Quelques caractéristiques de l'image

Les caractéristiques d'une image peuvent varier en fonction du contexte et des besoins spécifiques, mais voici quelques éléments clés que l'on peut considérer lors de l'analyse d'une image :

1.4.1 La résolution

La résolution d'une image numérique fait référence au nombre de pixels qui composent l'image. Elle est souvent exprimée en termes de largeur x hauteur, par exemple, 1920x1080 pixels. Une résolution plus élevée permet une image plus détaillée et de meilleure qualité.

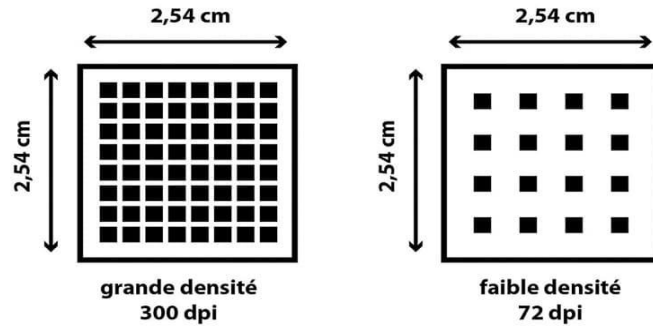


Figure 6: Schéma explicatif de la résolution de l'image numérique [13].

1.4.2 La profondeur de bits

La profondeur de bits fait référence au nombre de bits utilisés pour représenter la couleur ou l'intensité de chaque pixel. Une profondeur de bits plus élevée permet de représenter plus de nuances ou de couleurs, ce qui se traduit par une plus grande profondeur de couleur ou une plus grande gamme de niveaux de gris [14].

1.4.3 La compression

Les techniques de compression d'images réduisent la taille des fichiers image sans perte significative de qualité visuelle. Les méthodes de compression incluent la compression sans perte et la compression avec perte, chacune présentant des compromis entre la taille du fichier et la fidélité de l'image.



| PNG | WebP-lossless | WebP-lossy (with alpha) |
|---|--|---|
| "Yellow Rose" ¹ | | |
|  |  |  |
| PNG file size: 118.5 KB | WebP-lossless file size: 88.1 KB | WebP-lossy (with alpha) file size: 23.4 KB |

Figure7: Images compressées en PNG, WebP sans perte et avec perte [15].

1.4.4 Les métadonnées

Les métadonnées sont des informations supplémentaires associées à l'image numérique, telles que la date de création, la résolution, l'appareil photo utilisé, les coordonnées GPS, etc. Elles fournissent des détails sur l'image et peuvent être utilisées pour l'organisation, la recherche et la gestion des images.

1.5 Conclusion

Dans ce chapitre, nous avons présenté les différents concepts liés à l'image numérique, on a parlé sur les types d'une image, les caractéristiques et quelques formats de fichiers d'images. Dans le chapitre suivant nous allons aborder le traitement des images et la reconnaissance des images.

Chapitre 02 : Le traitement d'images et la classification

2.1 Introduction

Dans ce chapitre, nous indiquons les différents principes de traitement d'images et la classification, ils sont des concepts clés de la vision par ordinateur et de la reconnaissance de formes. Le traitement d'images implique la manipulation et l'analyse d'images numériques pour améliorer la qualité visuelle, extraire des informations pertinentes et permettre une compréhension approfondie du contenu de l'image. La classification consiste à attribuer des catégories prédéfinies à des images en utilisant des modèles d'apprentissage automatique ou statistique qui ont été entraînés sur des données annotées. Ces concepts sont largement utilisés dans de nombreux domaines tels que l'imagerie médicale, l'imagerie satellite, la surveillance et la sécurité, la robotique, l'agriculture, l'automobile, etc. Ils permettent d'automatiser l'analyse d'images et d'extraire des informations précieuses à partir de données visuelles, ouvrant ainsi de nombreuses possibilités pour les systèmes et les applications intelligentes.

2.2 Les images

2.2.1 Définition de l'image numérique

Le terme d'image numérique désigne, dans son sens le plus général, toute image qui a été acquise, traitée et sauvegardée sous une forme codée représentable par des nombres (valeurs numériques). La numérisation est le processus qui permet de passer de l'état d'image physique (image optique par exemple) qui est caractérisée par l'aspect continu du signal qu'elle représente (une infinité de valeur de l'intensité lumineuse par exemple), à l'état d'image numérique qui est caractérisée par l'aspect discret (l'intensité lumineuse ne peut prendre que des valeurs quantifiées en un nombre fini de points distincts). C'est cette forme numérique qui permet une exploitation ultérieure par des outils logiciels sur ordinateur [16, 17].

2.2.2 Les différents types d'images numériques

Il existe différents types de formats d'images numériques, chacun ayant ses propres caractéristiques et utilisations spécifiques. Voici quelques-uns des types les plus courants[18]:

- **JPEG (Joint Photographic Experts Group)**

Le format JPEG est largement utilisé pour les photographies et les images à couleurs riches. Il utilise une compression avec perte, ce qui réduit la taille du fichier mais peut entraîner une perte de qualité lors de la compression excessive.

- **PNG (Portable Network Graphics)**

Le format PNG est couramment utilisé pour les images comportant des éléments graphiques, des illustrations et des images avec transparence. Il prend en charge la compression sans perte, ce qui signifie que la qualité de l'image n'est pas altérée lors de la compression.

- **GIF (Graphics Interchange Format)**

Le format GIF est utilisé pour les images animées, telles que les petits clips ou les images en boucle. Il utilise une palette de couleurs limitée et prend en charge la transparence, ce qui en fait un choix populaire pour les graphiques simples et les animations.

- **TIFF (Tagged Image File Format)**

Le format TIFF est souvent utilisé pour les images de haute qualité, en particulier dans les applications professionnelles. Il prend en charge la compression sans perte, peut stocker plusieurs images dans un même fichier, et offre une gamme étendue de profondeur de bits et de modes de couleur.

- **BMP (Bitmap)**

Le format BMP est un format d'image non compressé qui stocke les données pixel par pixel, sans perte de qualité. Cependant, il a tendance à produire des fichiers volumineux, ce qui le rend moins couramment utilisé.

- **RAW**

Le format RAW est utilisé dans les appareils photo numériques pour enregistrer les données brutes capturées par le capteur d'image, sans aucune compression ni traitement. Les fichiers RAW offrent une flexibilité maximale pour le post-traitement, mais nécessitent un logiciel spécifique pour les visualiser et les convertir en d'autres formats.

2.3 Le traitement numérique des images

Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information.

Il s'agit d'un sous-ensemble du traitement du signal dédié aux images et aux données dérivées comme la vidéo (par opposition aux parties du traitement du signal consacrées à d'autres types de données : son et autres signaux monodimensionnels notamment), tout en opérant dans le domaine numérique (par opposition aux techniques analogiques de traitement du signal, comme la photographie ou la télévision traditionnelles).

Dans le contexte de la vision artificielle, le traitement d'images se place après les étapes d'acquisition et de numérisation, assurant les transformations d'images et la partie de calcul permettant d'aller vers une interprétation des images traitées. Cette phase d'interprétation est d'ailleurs de plus en plus intégrée dans le traitement d'images, en faisant appel notamment à l'intelligence artificielle pour manipuler des connaissances, principalement sur les informations dont on dispose à propos de ce que représentent les images traitées (connaissance du « domaine »).

La compréhension du traitement d'images commence par la compréhension de ce qu'est une image. Le mode et les conditions d'acquisition et de numérisation des images traitées conditionnent largement les opérations qu'il faudra réaliser pour extraire de l'information. En effet, de nombreux paramètres entrent en compte, les principaux étant[19]. :

- la résolution d'acquisition et le mode de codage utilisé lors de la numérisation, qui déterminent le degré de précision des éventuelles mesures de dimensions ;
- les réglages optiques utilisés, (dont la mise au point) qui déterminent par exemple la netteté de l'image ;
- les conditions d'éclairage, qui déterminent une partie de la variabilité des images traitées ;
- le bruit de la chaîne de transmission d'image.

- Quelques exemples types d'informations qu'il est possible d'obtenir d'une image numérique :
- la luminance moyenne ;
- le contraste moyen ;
- la couleur prédominante ;
- le taux d'acuité moyen (précis ou flou) ;
- le taux d'uniformité des couleurs ;

— la présence ou l'absence de certains objets.

2.3.1 Quelques traitements d'images

Il existe une variété de traitements d'images, nous allons présenter quelques-uns [7] :

- **Binarisation**

La binarisation (le seuillage) est la technique de classification la plus simple, où les pixels de l'image sont partagés par un seul seuil S en deux classes : ceux qui appartiennent au fond et ceux qui appartiennent à la scène (l'objet). L'image est alors séparée en deux classes de façon à ce que l'information comprise entre 0 et S est retenue et l'autre non, ou vice-versa.

Soit l'image $I(M \times N)$, supposons que $f(x, y)$ représente le niveau de gris du pixel aux coordonnées (x, y) , $0 \leq x < M$, $0 \leq y < N$ et S est le seuil choisi, les pixels de l'objet sont ceux ayant le niveau de gris inférieur à S et les autres ayant le niveau de gris supérieur à S sont des pixels du fond. Alors, l'image binarisée G est déterminée par les pixels (x, y) dont la valeur est :

$$g(x, y) = \begin{cases} 0 & \text{si } f(x, y) < S \\ 255 & \text{si } f(x, y) > S \end{cases} \quad (2.1)$$

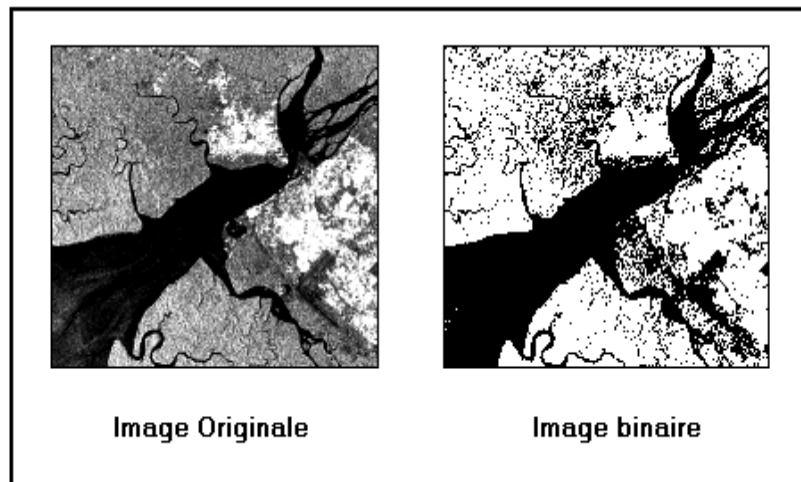


Figure 8 : Binarisation d'une image [20].

- **Squelettisation**

La procédure de squelettisation s'effectue sur une image binaire, et a pour but de réduire l'épaisseur du tracé à un pixel seulement, tout en conservant la continuité de celui-ci (Figure 11) [21].

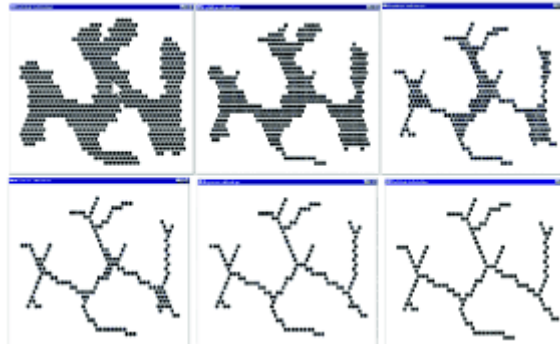


Figure 9: Squelettisation d'une image [22].

- **Segmentation**

La segmentation d'images est une opération qui a pour but de rassembler des pixels entre eux suivant des critères prédéfinis, et peut-être accomplie selon plusieurs méthodes. Les pixels ainsi regroupés en régions constituent un pavage ou une partition de l'image. La segmentation est une étape primordiale en traitement d'images[23].

- **Convolution**

La convolution est le remplacement de la valeur d'un pixel par une combinaison de ses avoisinants. Elle consiste à faire balayer une fenêtre d'analyse (masque) sur l'ensemble des pixels de l'image. L'opération de convolution est calculée en tout point de l'image source en 3 étapes:

1. On centre le masque (par exemple un masque carré 3*3) sur le pixel courant.
2. On calcule les 9 produits entre la valeur de l'image et la valeur du masque superposé.
3. Puis on somme les 9 produits pour obtenir la valeur du pixel de l'image filtrée.

On peut résumer l'opération de convolution par la formule suivante:

$$S(i, j) = \sum_{k=-(N-1)/2}^{(N-1)/2} \sum_{l=-(N-1)/2}^{(N-1)/2} E_{i+k, j+l} \cdot h_{k, l} \quad (2.2)$$

Où E est l'image d'entrée, h le masque de convolution avec h (0, 0) le centre du masque et N la taille du filtre (dans notre exemple N=3).

— **Filtrage**

La notion de filtre est empruntée à la physique et aux techniques de traitement de signal. Si un signal (électrique, radio, image,...etc.) présente des composantes de fréquences très différentes, il peut être avantageux d'en éliminer certaines, en parle dans ce cas de filtrage. Le filtrage des images est un traitement local utilisé principalement pour réaliser une analyse spatiale d'une image. Son

objectif est d'accentuer les variances d'intensité de l'image, ou de détecter les contours et de réduire les bruits existants.

Il existe un grand nombre de filtres, on peut les classer en deux grandes catégories : les filtres linéaires et les filtres non linéaires.

— Filtres linéaires

Un filtre linéaire transforme un ensemble de données d'entrée en un ensemble de données de sortie selon une opération mathématique appelée convolution. Il permet pour chaque pixel de la zone à laquelle il s'applique, de modifier sa valeur en fonction des valeurs des pixels avoisinants. Il existe plusieurs filtres linéaires, nous allons présenter quelques-uns.

a. Les filtres passe-bas (lissage)

Consistent à atténuer les pixels de l'image ayant une fréquence haute (pixels foncés). Ce type de filtrage est généralement utilisé pour atténuer le bruit de l'image, c'est la raison pour laquelle on parle habituellement de lissage.

Les filtres moyens sont des types de filtres passe-bas dont le principe est de faire la moyenne des valeurs des pixels avoisinants. Le résultat de ce filtre est une image plus floue.

Tableau 1: Masque de convolution du filtre passe-bas.

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 4 | 1 |
| 1 | 1 | 1 |

b. Les filtres passe-haut (accentuation)

À l'inverse des filtres passe-bas, les filtres passe-haut atténuent les pixels de basse fréquence de l'image et permettent notamment d'accentuer les détails et le contraste, c'est la raison pour laquelle le terme de "filtre d'accentuation" est parfois utilisé.

Tableau 2 : Masque de convolution du filtre passe-haut.

| | | |
|----|----|----|
| 0 | -1 | 0 |
| -1 | 5 | -1 |
| 0 | -1 | 0 |

c. Filtre Gaussien

Le filtre Gaussien est un filtre isotrope spécial avec des propriétés mathématiques bien précises. La fonction gaussienne $G(x)$ est souvent utilisée dans les distributions statistiques. Dans le traitement d'images, on traite des données à deux dimensions (x, y) , on introduit alors une fonction gaussienne à deux dimensions :

$$G(x, y) = \frac{1}{2\pi\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.3)$$

Etant données que les images numériques sont composées de valeurs discrètes, le noyau gaussien sera composé aussi de valeurs discrètes :

Tableau 3 : Masque de convolution du filtre Gaussien.

| | | | |
|----------------------|---|---|---|
| $\frac{1}{6} \times$ | 1 | 2 | 1 |
| | 2 | 4 | 2 |
| | 1 | 2 | 1 |

— Filtres non linéaires

Ils sont conçus pour régler les problèmes des filtre linéaires, surtout ce qui concerne la mauvaise conservation des contours. Leur principe est le même que les filtres linéaires, il s'agit toujours de remplacer la valeur de chaque pixel par la valeur d'une fonction calculée dans son voisinage. La différence est que cette fonction n'est plus linéaire mais une fonction quelconque. Les filtres non linéaires les plus connus sont:

a. Le filtre médian

Le filtre médian appartient à la famille des filtres d'ordre. Les filtres d'ordre procèdent en remplaçant la valeur de chaque pixel par la valeur du pixel qui occupe un certain rang lorsqu'on trie les valeurs des pixels rencontrés dans un certain voisinage de ce pixel. La valeur du pixel central est remplacée par la valeur médiane de tous les pixels de la fenêtre d'analyse centrée sur ce pixel. Le filtrage médian est très robuste à différents types de bruit.

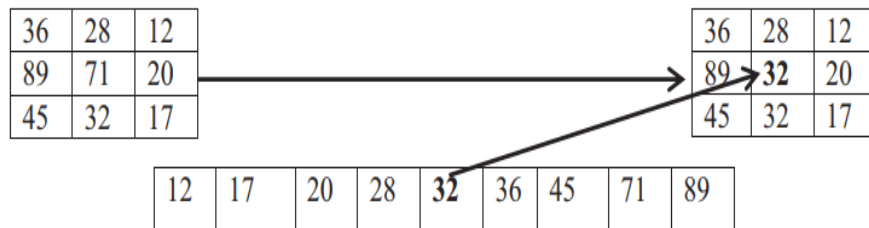


Figure 10: Principe du filtre médian.

b. Le filtre maximum

On applique le même traitement que celui du filtre médian mais la valeur du pixel du centre va être remplacée par le maximum.

c. Le filtre minimum

On applique le même traitement que celui du filtre médian mais, cette fois, la valeur du pixel du centre va être remplacée par le minimum [24].

— **Morphologie mathématique**

La morphologie mathématique est une théorie et technique mathématique et informatique d'analyse de structures, elle est liée avec l'algèbre et s'effectue sur une image binaire. Une des idées de base de la morphologie mathématique est d'étudier ou de traiter un ensemble à l'aide d'un autre ensemble, appelé élément structurant (masque binaire constitué de pixels blancs et noirs), qui sert de sonde.

A chaque position de l'élément structurant, on regarde s'il touche ou s'il est inclus dans l'ensemble initial. En fonction de la réponse, on construit un ensemble de sortie. On obtient ainsi des opérateurs de base qui sont relativement intuitifs [25].

Parmi les outils les plus importants de la morphologie mathématique on cite : érosion et dilatation.

— **Erosion** : qui consiste à enlever un pixel sur les pourtours de la forme, agrandir les trous.

— **Dilatation** : consiste à ajouter un pixel sur les pourtours de la forme, à boucher les trous.

— **La détection de contours**

Le but de la détection de contours est de repérer les points d'une image numérique qui correspondent à un changement brutal de l'intensité lumineuse. La détection des contours d'une image réduit de manière significative la quantité de données et élimine les informations qu'on peut juger moins pertinentes, tout en préservant les propriétés structurelles importantes de l'image.

2.4 Les méthodes de classification

- **Définition de la classification**

D'un point de vue général, les méthodes de classification ont pour but de regrouper les éléments d'un ensemble $X = \{X_1, \dots, X_n, \dots, X_N\}$ en un nombre C optimal de classes selon leurs ressemblances [26].

- **Les méthodes :**

De manière générale, les problèmes de classification s'attachent à déterminer des procédures permettant d'associer un objet (individu) à une classe. Ces problèmes se déclinent essentiellement en deux variantes selon Bezdek [27] : la classification dite " supervisée " et la classification dite " non supervisée ".

La classification, supervisée ou non, en tant que discipline scientifique, n'a été automatisée et massivement appliquée que relativement récemment. Comme la plupart des activités scientifiques, l'essor des différentes techniques de classification a largement bénéficié de l'avènement et du perfectionnement des outils informatiques. De nos jours, la classification est une démarche qui est appliquée dans d'innombrables domaines.

Les méthodes de classification ont pour but de regrouper les éléments d'un ensemble X , de nature quelconque, en un nombre restreint de classes.

La qualité de la classification peut être jugée sur la base des deux critères suivants :

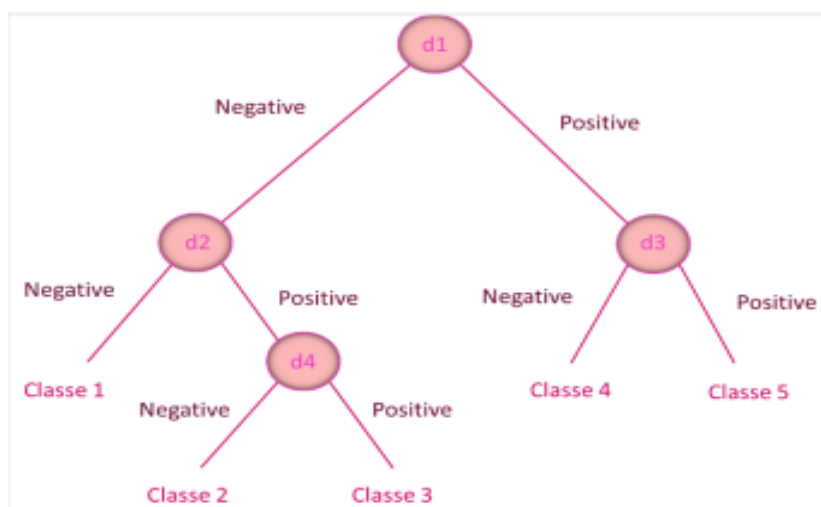
- Les classes générées doivent être les plus différentes possibles les unes des autres vis-à-vis de certaines caractéristiques.

Chaque classe doit être la plus homogène possible vis-à-vis de ces caractéristiques[28]. Si l'on se place dans un espace métrique, en termes d'inertie, le premier critère peut être traduit par le fait que l'inertie inter- classe doit être la plus élevée possible, et le deuxième critère traduit la préoccupation d'obtenir une inertie intra-classe la plus petite possible.

Nous allons citer quelques méthode utiliser du classification :

2.4.1. L'arbre de décision

C'est un classifieur présenté sous forme d'une structure arborescente (Figure 13). Comporte trois types de nœuds : 1) un nœud racine correspondant à l'ensemble des données d'apprentissage, 2) des nœuds internes dont les exemples vont être testés, 3) les nœuds terminaux qui représentent les classes. L'arbre est parcouru de la racine jusqu'à un nœud terminal. Les exemples seront testés dans les nœuds internes généralement sur un seul attribut dont le but d'obtenir plusieurs sous-ensembles ne contenant que des exemples appartenant à des classes homogènes. Une fois arriver à un nœud terminal qui correspond à une classe l'exemple sera prédit comme faisant partie de cette classe [29].



| |
|-------------------------------------|
| $d1$: Nœud racine. |
| $d2, d3, d4$: Nœuds internes. |
| Négative, Positive : les attributs. |

Figure 11 : Exemple d'arborescence pour un problème à 5 classes [29].

2.4.2. le classifieur bayésien

Naive Bayes est un algorithme de classification probabiliste basé sur le théorème de Bayes et l'hypothèse de naïveté conditionnelle. Il est largement utilisé pour la classification de textes, mais il peut également être appliqué à d'autres types de données, y compris les images.

Dans le contexte des images, Naive Bayes peut être utilisé pour classer des objets, des scènes ou des motifs visuels en fonction de leurs caractéristiques extraites. Les caractéristiques d'une image, telles que les histogrammes de couleur, les textures ou les formes, peuvent être utilisées comme variables pour l'algorithme Naive Bayes.

L'idée principale de Naive Bayes est de calculer la probabilité qu'une image appartienne à une classe donnée en se basant sur les probabilités conditionnelles des caractéristiques de cette classe. L'hypothèse de naïveté conditionnelle suppose que les caractéristiques sont indépendantes les unes des autres, simplifiant ainsi le calcul des probabilités [28].

2.4.3 La méthode des k plus proches voisins (kppv ou knn)

Dans un contexte de classification d'une nouvelle observation , l'idée fondatrice est de supposer qu'une observation est similaire à celle de ses plus proches voisins. On suppose par ailleurs qu'il existe une notion de distance, la classe de x_i est déterminée en fonction de la classe majoritaire parmi les k plus proches voisins de l'observation. La méthode KNN est donc une méthode à base de voisinage, non paramétrique, ceci signifiant que l'algorithme permet de faire une classification sans faire d'hypothèse sur la fonction $y_i = H(x_i)$ [30].

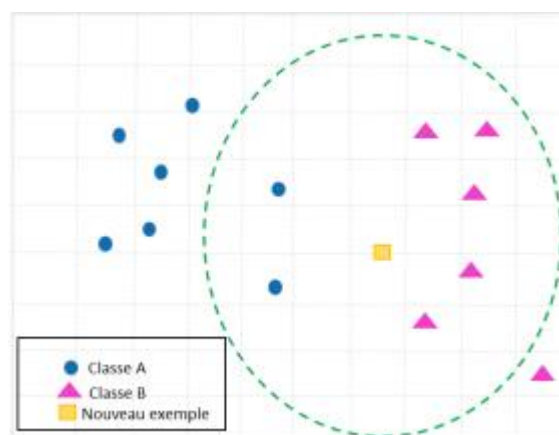


Figure 12 : Exemple de classification par l'algorithme K-Plus Proches voisins [30]

2.5. Les réseaux de neurones

Les réseaux neuronaux, également appelés réseaux de neurones artificiels, sont des modèles d'apprentissage automatique inspirés par le fonctionnement du cerveau humain. Ils sont largement utilisés dans divers domaines, y compris la vision par ordinateur, la reconnaissance vocale, la traduction automatique et bien d'autres applications.

Un réseau neuronal est composé de plusieurs couches de neurones artificiels interconnectés. Chaque neurone reçoit des entrées pondérées, les combine à l'aide d'une fonction d'activation et produit une sortie. Les couches intermédiaires entre l'entrée et la sortie sont appelées couches cachées, et elles permettent au réseau d'apprendre des représentations hiérarchiques et complexes des données.

L'apprentissage dans un réseau neuronal se fait généralement en deux phases principales : la phase d'apprentissage et la phase de prédiction. Pendant la phase d'apprentissage, le réseau est entraîné sur un ensemble de données d'entraînement étiquetées, où les poids des connexions entre les neurones sont ajustés afin de minimiser une fonction de perte ou d'erreur. Ce processus d'ajustement des poids est généralement réalisé à l'aide d'algorithmes d'optimisation tels que la rétro propagation du gradient.

Une fois que le réseau est entraîné, il peut être utilisé pour faire des prédictions sur de nouvelles données non étiquetées pendant la phase de prédiction. Les entrées sont propagées à travers le réseau, et les sorties produites par le réseau fournissent les prédictions pour les différentes classes ou catégories [31].

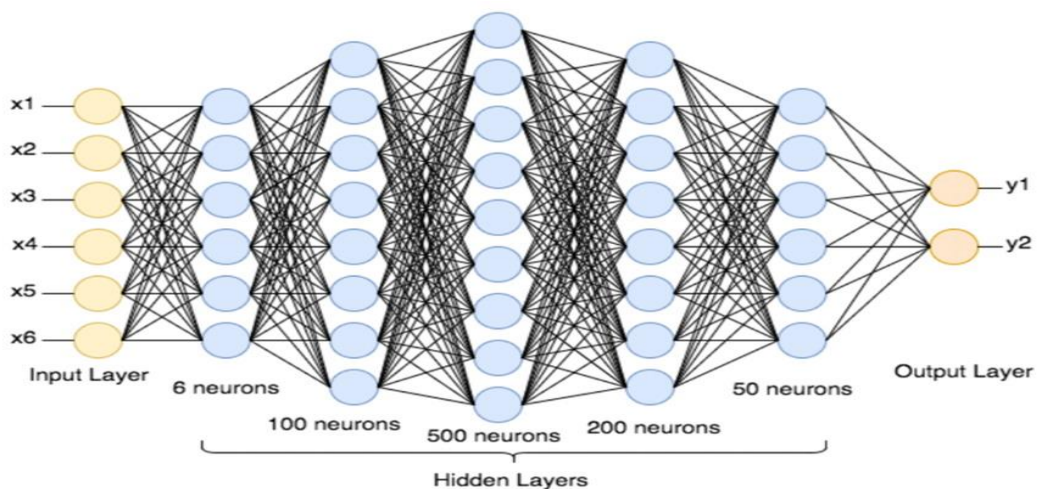


Figure13 : Les réseaux de neurones [32].

2.6. Conclusion

Dans ce chapitre, nous avons présenté quelques traitements et les méthodes utilisées dans les images, ensuite nous avons donné un aperçu sur les différentes approches utilisées pour la classification : supervisées et non supervisées.

Chapitre 03 : Les réseaux de neurones convolution(CNN)

3.1. Introduction

Dans ce chapitre, nous indiquons Les réseaux neuronaux, également connus sous le nom de réseaux de neurones artificiels, sont des modèles d'apprentissage automatique qui s'inspirent du fonctionnement du cerveau humain. Ces modèles ont révolutionné de nombreux domaines de l'intelligence artificielle et ont permis des avancées significatives dans des tâches telles que la reconnaissance d'images, la traduction automatique, le traitement du langage naturel et bien d'autres.

Le concept de base des réseaux neuronaux repose sur l'idée de simuler le fonctionnement des neurones biologiques et de leurs connexions synaptiques. Un réseau neuronal est constitué de plusieurs couches de neurones interconnectés, chaque neurone étant un élément de traitement de l'information.

Le Deep Learning (apprentissage profond en français) est une branche de l'intelligence artificielle (IA) qui vise à développer des modèles d'apprentissage automatique capables d'apprendre des représentations hiérarchiques et complexes des données. Il s'agit d'une approche basée sur les réseaux neuronaux profonds, qui sont des modèles composés de plusieurs couches de neurones interconnectés

3.2. Le Neurone

3.2.1 Le modèle biologique

Un neurone est une unité de base du système nerveux, que ce soit dans le cerveau humain ou dans les modèles de réseaux neuronaux artificiels. Il joue un rôle essentiel dans le traitement et la transmission de l'information [33].

La structure d'un neurone biologique se compose de plusieurs parties clés :

- **Le corps cellulaire (ou soma)**

C'est la partie centrale du neurone qui contient le noyau et les organites cellulaires nécessaires à son fonctionnement.

- **Les dendrites**

Ce sont de fines extensions ramifiées qui se propagent à partir du corps cellulaire. Les dendrites reçoivent les signaux provenant d'autres neurones et les transmettent vers le corps cellulaire.

- **L'axone**

C'est une longue extension unique qui émerge du corps cellulaire. L'axone est responsable de la transmission des signaux électriques (potentiels d'action) générés par le neurone.

- **Les synapses**

Ce sont les connexions spécialisées situées à l'extrémité des axones du neurone. Les synapses permettent la transmission des signaux électriques d'un neurone à l'autre, ou vers d'autres cellules cibles telles que les muscles ou les glandes.

Le fonctionnement d'un neurone repose sur la transmission de signaux électriques et chimiques. Lorsque des signaux électriques atteignent les dendrites, ils sont sommés et intégrés au niveau du corps cellulaire. Si le seuil d'excitation est atteint, un potentiel d'action est généré le long de l'axone, permettant la transmission du signal vers d'autres neurones via les synapses.

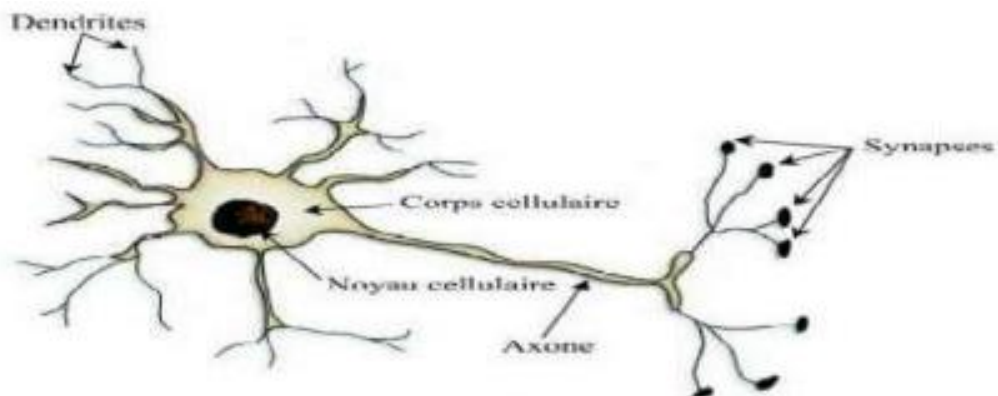


Figure 14 : Représentation d'un neurone biologique [34].

2.3.2 Modèle formel (artificiel)

Le neurone artificiel est une fonction algébrique non linéaire et bornée, dont la valeur dépend des paramètres appelés coefficients ou poids. Les variables de cette fonction sont habituellement appelées entrées du neurone, et la valeur de la fonction est appelée sa

sortie .Le neurone Artificiel est donc un modèle mathématique simplifié du neurone biologique [35].

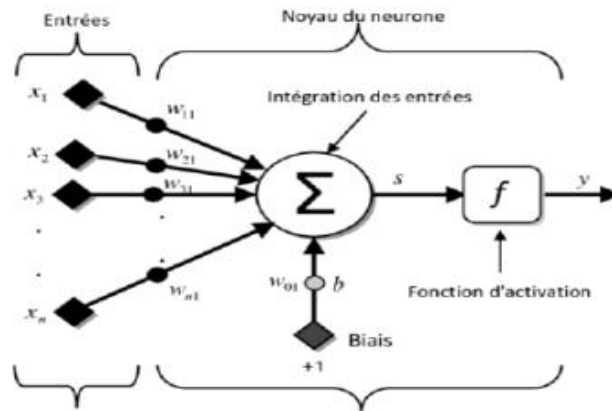


Figure 15 : Le neurone artificiel[36].

- Les X_i représentent les vecteurs d'entrées
- Les W_{ij} sont les poids synaptiques du neurone j .
- **Biais W_0** : entrée prend souvent les valeurs -1 ou $+1$ qui permet d'ajouter de la flexibilité
- **Noyau** : intègre toutes les entrées et le biais et calcul la sortie du neurone.
- **La fonction d'activation f** est la fonction d'activation du neurone. Il existe denombreuses formes possibles pour la fonction d'activation, les plus courantes sontprésentées sur le tableau suivant. On remarquera qu'à la différence des neurones biologiques dont l'état est binaire, la plupart des fonctions d'activations sont des fonctions continues, offrant une infinité de valeurs possibles comprises dans l'intervalle $[0, +1]$ (ou $[-1, +1]$).

| Nom de la fonction | Relation d'entrée/sortie | Icône |
|-----------------------------|---|-------|
| seuil | $a = 0$ si $n < 0$ $a = 1$ si $n \geq 0$ | |
| seuil symétrique | $a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$ | |
| linéaire | $a = n$ | |
| linéaire saturée | $a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$ | |
| linéaire saturée symétrique | $a = -1$ si $n < -1$ $a = n$ si $-1 \leq n \leq 1$ $a = 1$ si $n > 1$ | |
| linéaire positive | $a = 0$ si $n < 0$ $a = n$ si $n \geq 0$ | |
| sigmoïde | $a = \frac{1}{1+\exp^{-n}}$ | |
| tangente hyperbolique | $a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$ | |
| compétitive | $a = 1$ si n maximum $a = 0$ autrement | |

Figure 16 : Les modèles de fonctions d'activation [37].

Donc le calcul de la sortie se fait en deux étapes :

1. Une combinaison linéaire des entrées : $S = \sum_{i=1}^n (W_i X_i) \pm b$
2. Application d'une fonction d'activation : $f(x) = f(S)$

Analogie entre le neurone biologique et le neurone formel

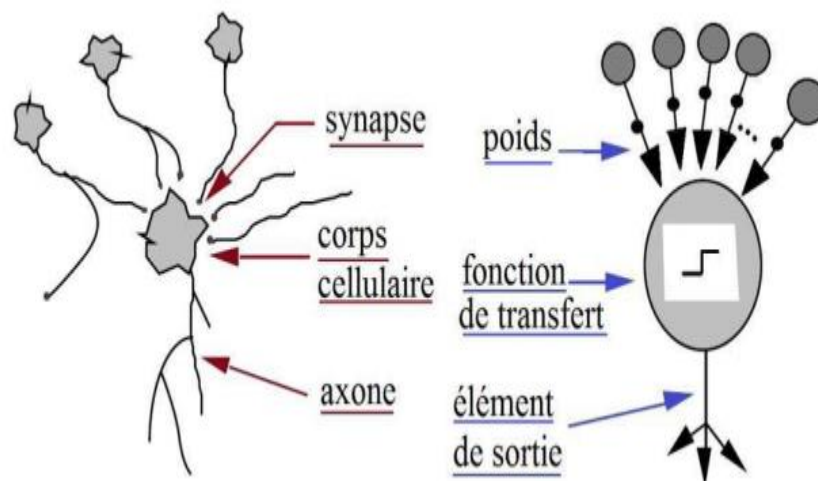


Figure 17 : Correspondance entre le neurone biologique et le neurone formel [38].

Le tableau suivant montre l'analogie entre le neurone biologique et le neurone artificiel.

Tableau 4 : Analogie entre neurone biologique et le neurone artificiel.

| Neurone biologique | Neurone artificiel : |
|--------------------|-----------------------|
| Synapses | Poids de connexions |
| Axones | Signal de sortie |
| Dendrite | Signal d'entrée |
| Somma | Fonction d'activation |

3.3. Le Réseau de neurones multicouche

Le perceptron multicouche (Multi Layer Perceptron) est une amélioration du perceptron comprenant une ou plusieurs couches cachées qui font le réseau MLP un outil robuste pour les tâches complexes. Il est largement utilisé pour la décision dans le domaine de reconnaissance faciale. Les réseaux MLP sont généralement des réseaux entièrement connectés. Les neurones de la première couche reçoivent le vecteur d'entrée, ils calculent leurs sorties qui sont transmises aux neurones de la seconde couche qui calculent eux même leurs sorties et ainsi de suite de couche en couche jusqu'à celle de sortie [46].

Dans le réseau MLP il n'y a aucune connexion entre les cellules d'une même couche. Les perceptrons multicouches sont utilisés avec apprentissage supervisé et aussi avec la technique de rétro propagation (back-propagation) pour la correction de l'erreur.

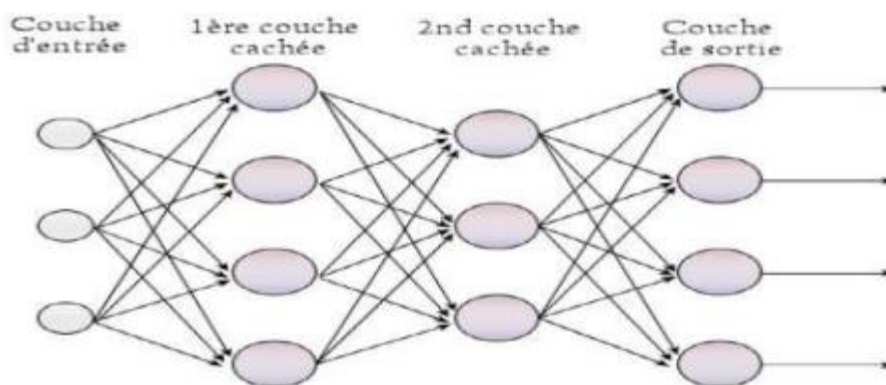


Figure 18 : Perceptron multicouches [39].

3.4 Le Réseau de neurones convolution (CNN)

3.4.1 Fonctionnement d'un CNN :

Le fonctionnement d'un CNN se déroule en plusieurs étapes :

- **Prétraitement des données** : Les images d'entrée sont généralement prétraitées pour les normaliser et les mettre à l'échelle. Cela peut inclure la réduction de la résolution, la normalisation des intensités de couleur, etc.
- **Propagation avant** : Les images prétraitées sont introduites dans le réseau. La propagation avant consiste à passer les données d'entrée à travers les différentes couches du CNN, en appliquant les opérations de convolution, de pooling et de non-linéarité.
- **Calcul de la perte** : Une fois que les données ont été propagées à travers le réseau, la sortie finale est comparée avec les étiquettes (labels) réelles pour calculer une mesure de perte (par exemple, l'erreur quadratique moyenne ou l'entropie croisée) qui évalue la différence entre les prédictions du réseau et les véritables valeurs attendues.
- **Rétro propagation du gradient**: Le gradient de la perte par rapport aux poids du réseau est calculé à l'aide de la rétro propagation du gradient. Ce gradient est utilisé pour ajuster les poids du réseau en utilisant des algorithmes d'optimisation.

3.4.2 les différentes architectures d'un CNN

Il existe différentes architectures de réseaux neuronaux convolutifs (CNN) qui ont été développées pour répondre à des problèmes spécifiques de vision par ordinateur. Voici quelques-unes des architectures les plus couramment utilisées :

- **LeNet (1990)**

Les premières applications réussies des réseaux convolutifs ont été développées par Yann Le Cun dans les années 1990. Parmi ceux-ci, le plus connu est l'architecture Le Net utilisée pour lire les codes postaux, les chiffres, etc [48].

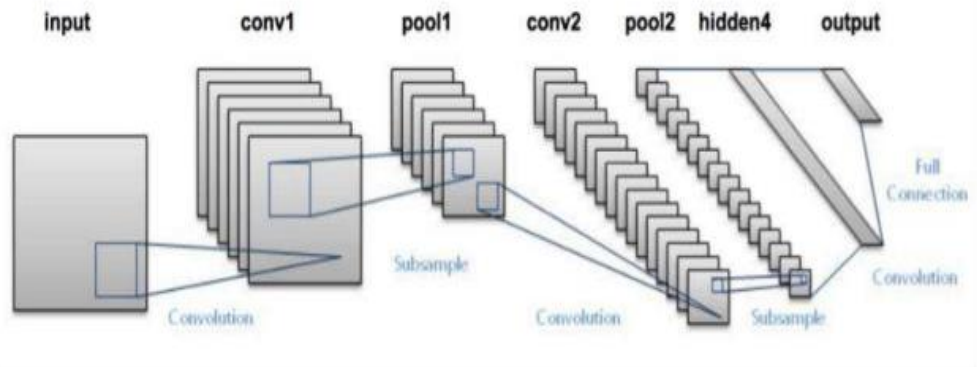


Figure 19 : L'architecture Le Net [49].

- AlexNet(2012)

Le premier travail qui a popularisé les réseaux convolutifs dans la vision par ordinateur était Alex Net, développé par Alex Krizhevsky, Ilya Sutskever et Geoff Hinton. Ce CNN a été soumis au défi de la base Image Net en 2012 et a nettement surpassé ses concurrents. Le réseau avait une architecture très similaire à Le Net, mais était plus profond, plus grand et comportait des couches convolutives empilées les unes sur les autres (auparavant, il était commun de ne disposer que d'une seule couche convolutives toujours immédiatement suivie d'une couche de pooling) [48].

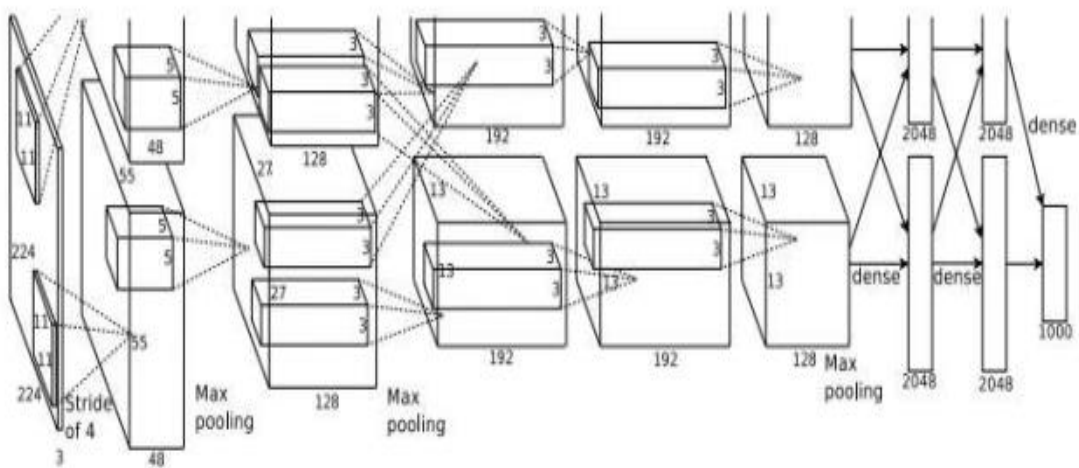


Figure 20 :L'architecture AlexNet [49].

- ZFNet (2013)

C'était une amélioration d'AlexNet en ajustant les hyper-paramètres de l'architecture, en particulier en élargissant la taille des couches convolutifs et en réduisant la taille du noyau sur la première couche [50].

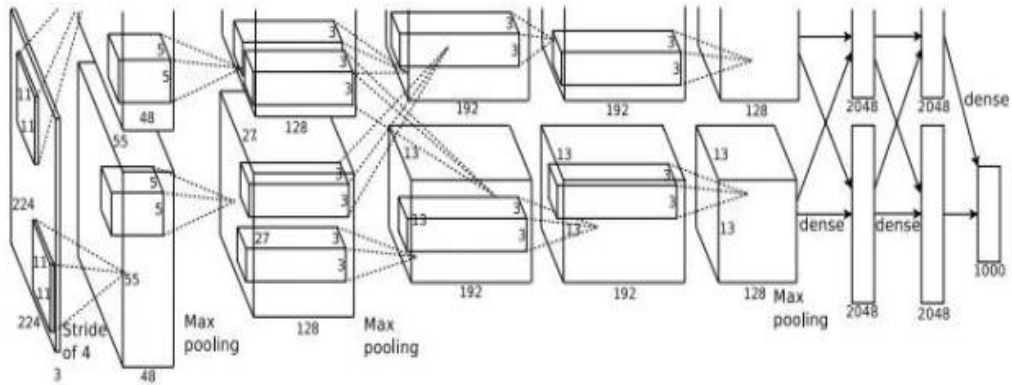


Figure 21 : L'architecture ZFNet [49].

- GoogLeNet (2014)

C'est un modèle de Google. Sa principale contribution a été le développement d'un module inception qui a considérablement réduit le nombre de paramètres dans le réseau (4M, par rapport à Alex Net avec 60M). En outre, ce module utilise le global Average pooling ce qui élimine une grande quantité de paramètres. Il existe également plusieurs versions de GoogLeNet, parmi elles, Inception-v4 et X Inception ce dernier est l'un des modèles auxquels notre système s'inspire, plus de détails dans le chapitre de conception [50].

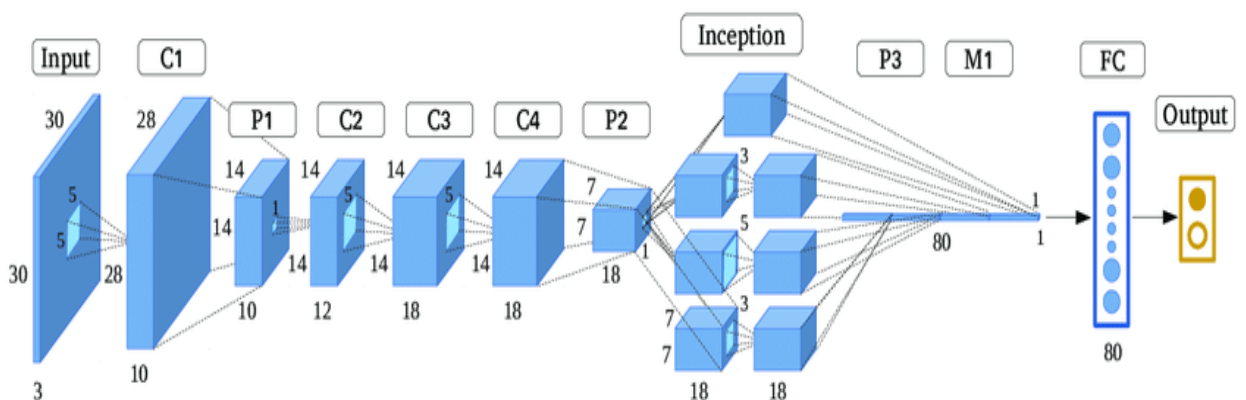


Figure 22 : L'architecture GoogLeNet/ Inception [49].

- VGGNet (2014)

Il s'agit d'une structure du Visual Geometry Group d'Oxford réalisée par Andrea Vedaldi et Andrew Zisserman, Sa principale contribution a été de montrer que la profondeur du réseau est un composant essentiel pour de bonnes performances. Son inconvénient est qu'il est plus coûteux à évaluer et utilise beaucoup plus de mémoire et de paramètres (140M).

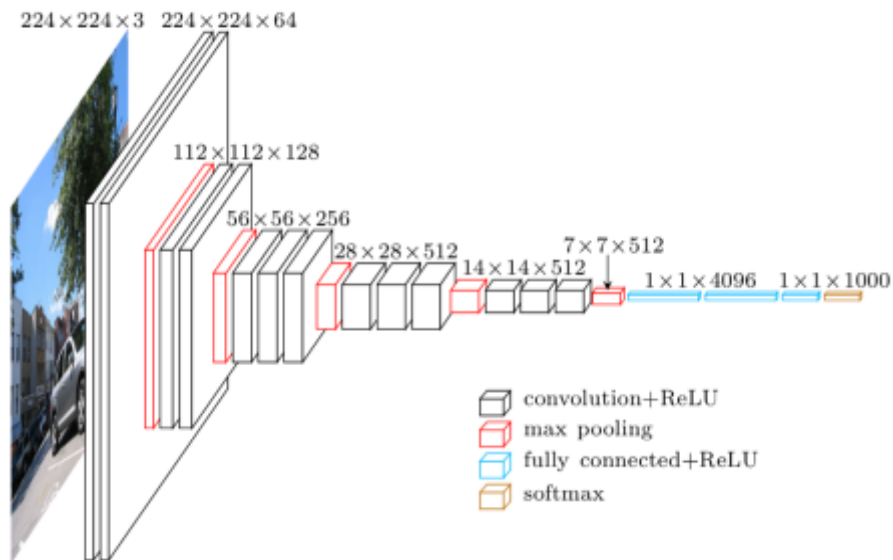


Figure 23: L'architecture VGGNet[49].

- ResNet (Residual Network)(2015)

Residual network développé par Kaiming He et al, il a été le vainqueur de ILSVRC 2015. Il présente des sauts de connexion et une forte utilisation de la batch normalisation. Il utilise aussi le global AVG pooling au lieu du PMC à la fin [48].

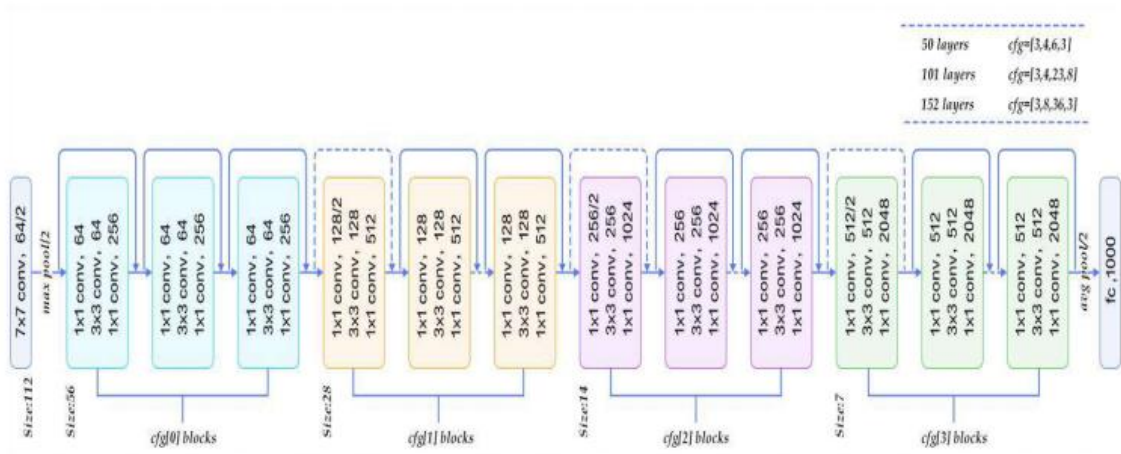


Figure 24 :L'architecture ResNet[49].

3.5 Les compositions

3.5.1 La couche de convolution(conv)

La convolution est le cœur du réseau de neurones convolutifs. A l'origine, une convolution est un outil mathématique (on parle de produit de convolution) très utilisé en retouche d'image, car il permet d'en faire ressortir l'extraction des caractéristiques à partir des images d'entrées, afin d'appliquer un bon filtre. En fait, une convolution prend simplement en entrée une image et un filtre (qui est une autre image), effectue un calcul, puis renvoie une nouvelle image (généralement plus petite) [40].

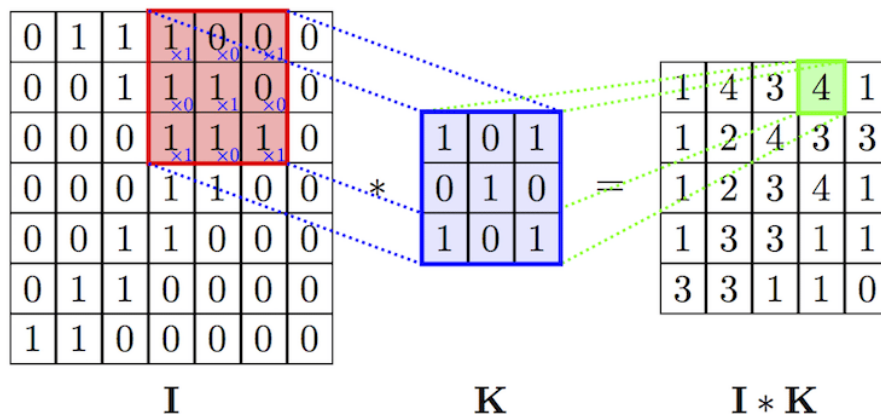


Figure 25 : Exemple d'une opération de convolution [41].

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

- **Profondeur de la couche** : nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).
- **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.
- **La marge (à 0) ou zéro padding**: parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. La taille de ce 'zéro-padding' est le troisième hyper paramètre. 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.

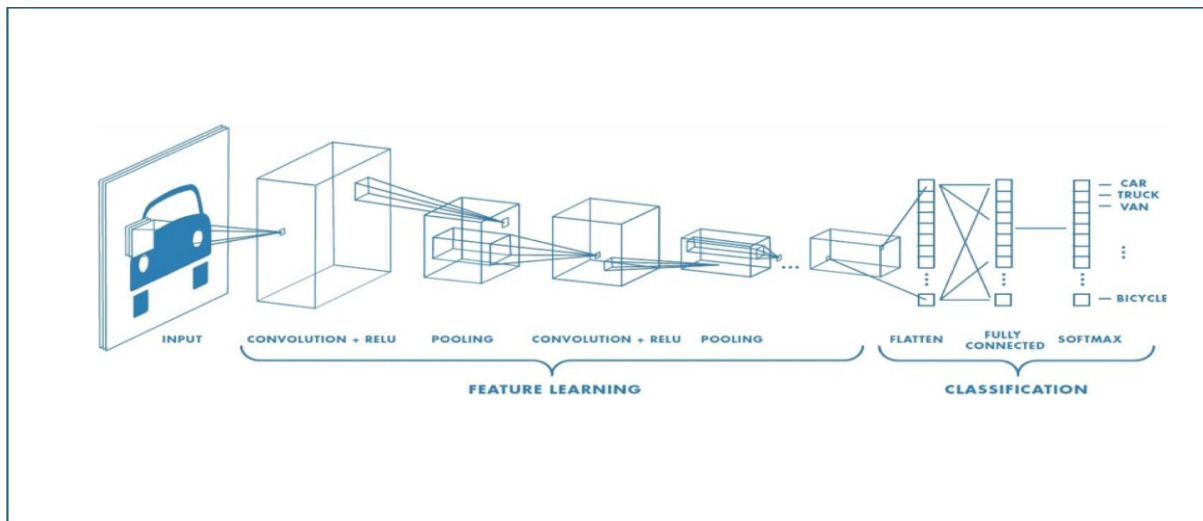


Figure 26 : Exemple de réseau composé de nombreuses couches à convolution[42].

Des filtres sont appliqués à chaque image utilisée pour l'apprentissage à différentes résolutions, et la sortie de chaque image convoluée est utilisée comme entrée de la couche suivante.

- **Les différentes convolutions**

Il existe plusieurs types de convolutions, même si en général on utilise celle de base, il peut s'avérer utile de connaître les outils à notre disposition [43].

- **La convolution classique**: qui représente le décalage du noyau entre chaque calcul, et le padding qui est la manière dont on peut « dépasser » de l'image pour appliquer la convolution.
- **La dilated convolution** : identique à la convolution à ceci près que le kernel est éclaté (on prend, par exemple, un pixel sur deux pour calculer la convolution). Il y a un paramètre supplémentaire : la dilation rate, qui est le nombre de pixels à ignorer.

- **La transposed convolution:** qui construit la sortie comme si on inversait une convolution sur l'image.
- **La séparable convolution:** qui est une convolution décomposable en convolutions plus simples.

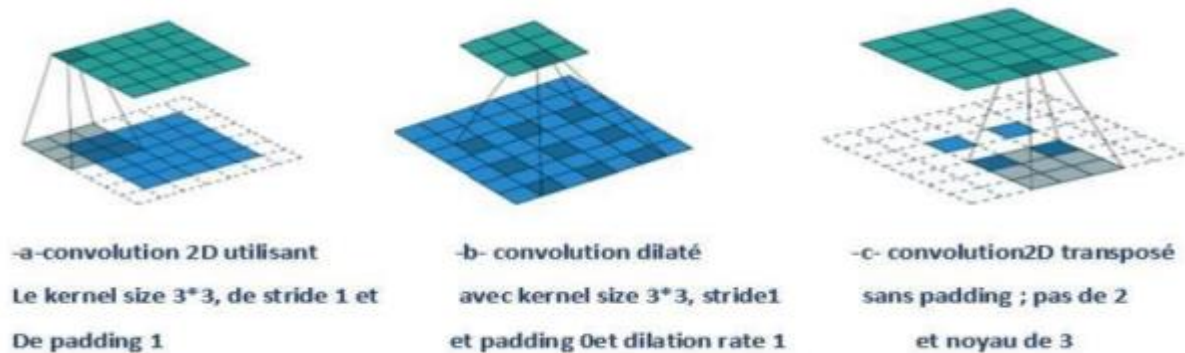


Figure 27 : Différents types de convolution.

3.5.2 La couche de pooling (pool)

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs feature maps, et applique à chacune d'entre elles l'opération de pooling. L'opération de pooling (ou sub-sampling) consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes. Pour cela, on découpe l'image en cellules régulières, puis on garde au sein de chaque cellule la valeur maximale.

En pratique, on utilise souvent des cellules carrées de petite taille pour ne pas perdre trop d'informations. Les choix les plus communs sont des cellules adjacentes de taille 2 x 2 pixels qui ne se chevauchent pas, ou des cellules de taille 3 x 3 pixels, distantes les unes des autres d'un pas de 2 pixels (qui se chevauchent donc).

On obtient en sortie le même nombre de feature maps qu'en entrée, mais celles-ci sont bien plus petites. La couche de pooling permet de réduire le nombre de paramètres et de calculs dans le réseau. On améliore ainsi l'efficacité du réseau et on évite le sur-apprentissage [40].

Il existe plusieurs types de pooling : Max, Moyenne, Somme, etc. • Max pooling : qui revient à prendre la valeur maximale de la sélection. C'est le type le plus utilisé, car il est rapide à calculer (immédiat), et permet de simplifier efficacement l'image.

- **Le meanpooling** : (ou average pooling), soit la moyenne des pixels de la sélection : on calcule la somme de toutes les valeurs et on divise par le nombre de valeurs. On obtient ainsi une valeur intermédiaire pour représenter ce lot de pixels.
- **Le sumpooling** : c'est la moyenne sans avoir divisé par le nombre de valeurs(on ne calcule que leur somme).

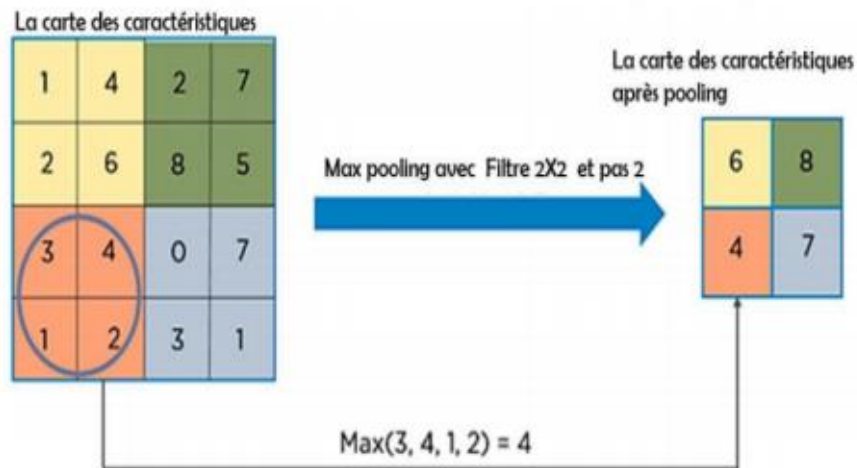


Figure 28 : Pooling avec un filtre 2*2 et un pas de 2 [44]

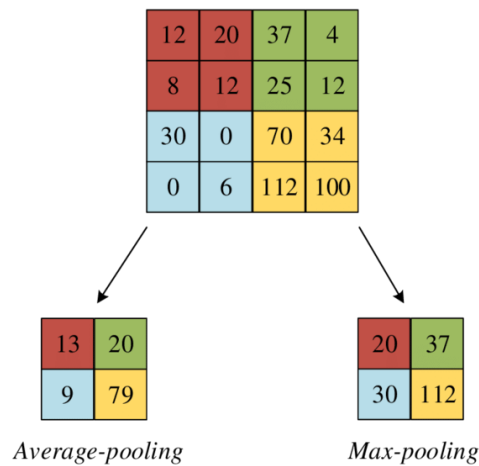


Figure 29: Example of pooling with a 2×2 filter and a stride of 2 [44].

- **Le Flattening (La mise à plat)** :
Dernière étape de la partie "extraction des informations", le flattening consiste simplement à mettre bout à bout toutes les images (matrices) que nous avons (résultat de pooling) pour en faire un (long) vecteur. Les pixels (en réalité ce ne sont plus des images ou des pixels, mais des matrices de

nombre, donc les pixels sont ces nombres) sont récupérés ligne par ligne et ajoutés au vecteur final [43].

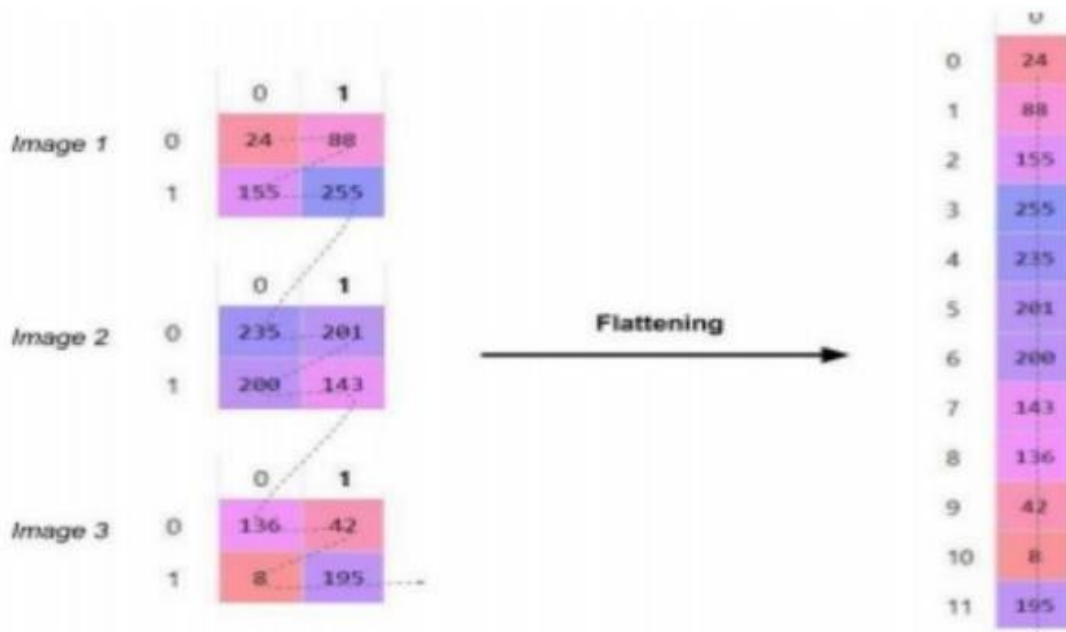


Figure 30 : Exemple de mise à plat des images finales en sortie (des filtres + simplifications).

3.5.3 La couche de correction(ReLU)

C'est une couche pour 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 [43].

La fonction d'activation est souvent une fonction non-linéaire. Leur but est de permettre aux réseaux de neurones d'apprendre des fonctions plus complexes qu'une simple régression linéaire car le fait de multiplier les poids d'une couche cachée est juste une transformation linéaire [45].

Rectified Linear Unit (ReLU)

Elle est utilisée après chaque opération de convolution, ou toutes les valeurs de pixels négatifs sont mises à zéro. Le but de ReLU est d'introduire la non-linéarité dans notre CNN, puisque la plupart des données du monde réel, puisque la plupart de caractéristiques appliquées à l'une des cartes d'entrée donne une carte de sortie qui est également appelée carte de caractéristiques rectifiées. [40]

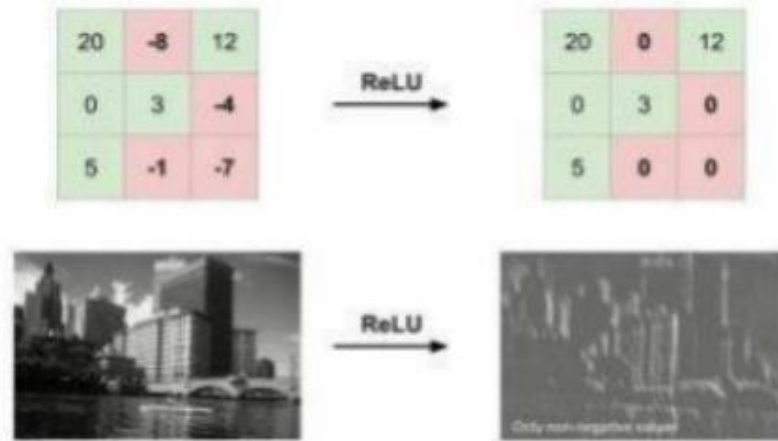


Figure 31 : Application de ReLU à partir d'une matrice et d'une image.

3.5.4. La couche entièrement connectée(FC)

Après les couches de convolution et pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches totalement connectées. Dans les réseaux de neurones convolutifs, chaque couche agit comme un filtre de détection pour la présence de caractéristiques spécifiques ou de motifs présents dans les données d'origine. Les premières couches d'un réseau convolutif détectent des caractéristiques qui peuvent être reconnues et interprétées facilement. Les couches ultérieures détectent de plus en plus des caractéristiques plus abstraites

La dernière couche du réseau convolutif est capable de faire une classification ultra spécifique en combinant toutes les caractéristiques spécifiques détectées par les couches précédentes dans les données d'entrée .Les couches totalement connectées font les mêmes tâches que celles des ANN (réseaux de neurone artificielle) standard et tenteront de produire des notes de classe à partir des activations, pour les utiliser pour la classification.

Il est également suggéré d'utiliser ReLU entre ces couches pour améliorer les performances. Le but de la couche entièrement connectée est de pouvoir utiliser ces fonctions pour classer l'image d'entrée dans différentes classes en fonction de l'ensemble de données d'apprentissage. [43]

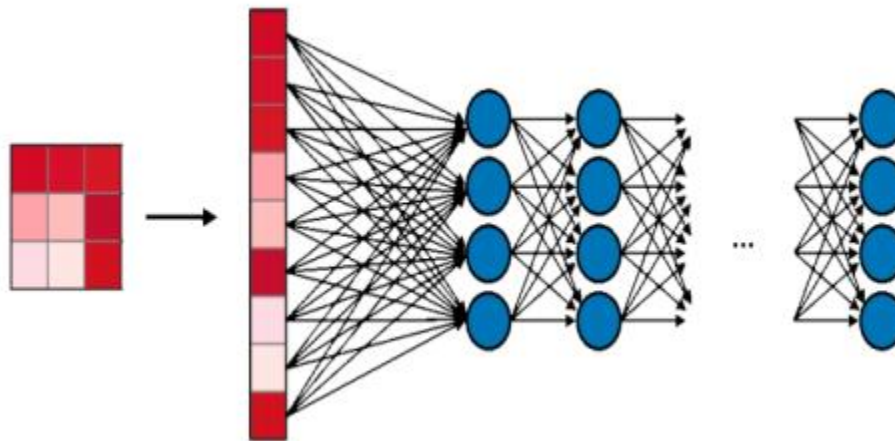


Figure 32 :Un exemple d'opération entièrement connectée (FC).

3.6 Architecture générale du CNN

L'architecture générale d'un réseau neuronal convolutif (CNN) comprend plusieurs couches qui sont conçues pour extraire et représenter les caractéristiques des images de manière progressive. Voici une description des principales couches qui composent l'architecture d'un CNN :

- **Couche d'entrée (Input Layer)** : Cette couche reçoit les images en tant que données d'entrée. Les images sont généralement représentées sous forme de tenseurs multidimensionnels, où chaque dimension correspond à une dimension spatiale de l'image (largeur, hauteur) et au nombre de canaux (par exemple, les canaux RVB pour les images en couleur).
- **Couches de convolution (Convolutional Layers)**: Les couches de convolution sont responsables de l'extraction des caractéristiques à partir des images. Chaque couche de convolution applique un ensemble de filtres (noyaux de convolution) à l'image d'entrée, produisant des cartes d'activation qui mettent en évidence les motifs et les structures présents dans l'image. Ces filtres sont appris par le réseau pendant la phase d'entraînement.
- **Couches de pooling (Pooling Layers)** : Les couches de pooling réduisent la dimension spatiale des cartes d'activation produites par les couches de convolution. Elles regroupent les activations voisines en calculant, par exemple, la valeur maximale (max pooling) ou la valeur moyenne (average pooling) dans chaque région. Cela permet de réduire la complexité du modèle et d'obtenir une représentation plus compacte des caractéristiques.
- **Couches de non-linéarité (Activation Layers)** : Après chaque couche de convolution ou de pooling, une fonction d'activation non linéaire est appliquée aux cartes d'activation. La fonction

d'activation la plus couramment utilisée est la fonction ReLU (Rectified Linear Unit), qui remplace les valeurs négatives par zéro et garde les valeurs positives inchangées. Cela introduit de la non-linéarité dans le modèle, ce qui lui permet de capturer des relations complexes entre les caractéristiques.

- **Couches entièrement connectées (Fully Connected Layers):** Les couches entièrement connectées sont situées à la fin du CNN. Elles prennent les caractéristiques extraites par les couches précédentes et les transforment en une représentation finale pour effectuer la classification ou d'autres tâches. Les neurones de ces couches sont connectés à tous les neurones de la couche précédente, ce qui permet au réseau d'apprendre des combinaisons complexes de caractéristiques.
- **Couche de sortie (Output Layer) :** La couche de sortie est la dernière couche du CNN. Elle produit les prédictions finales du modèle, généralement sous la forme de probabilités pour chaque classe ou d'un vecteur de sortie.

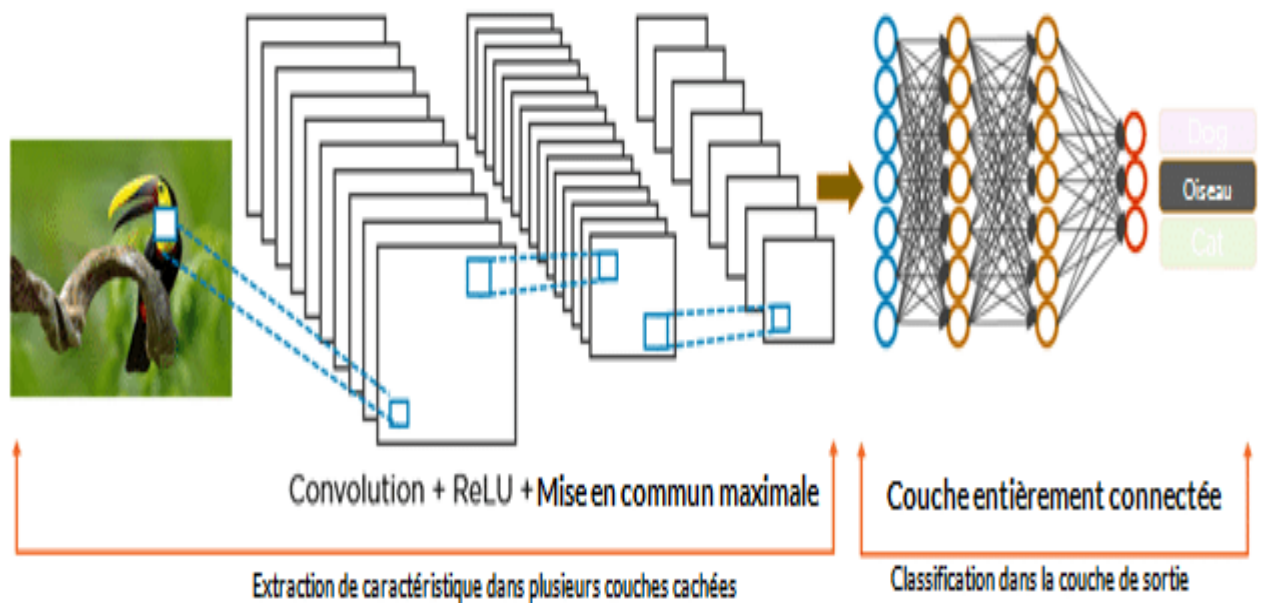


Figure 33 : Architecture générale d'un CNN.

[47].

3.7. Conclusion

Dans ce chapitre nous avons présenté des notions importantes qui sont en relation avec l'apprentissage profond (définition, types....etc.). Aussi qu'un aperçue général sur l'apprentissage profond, toute en donnant en détail le modèle choisi dans notre travail de recherche qui est le CNN. Le prochain chapitre, traite les détails de la conception, ainsi que la méthode et les outils utilisés pour la réalisation de notre application pour la diagnostic basé image.

Chapitre 04 :

La reconnaissance d'images par CNN

4.1 Introduction

La reconnaissance d'image est l'une des tâches les plus importantes et courantes dans le domaine de la vision par ordinateur. Elle consiste à entraîner un modèle informatique pour identifier et classer automatiquement les objets, les motifs ou les caractéristiques présents dans une image.

Les réseaux neuronaux convolutifs (CNN) ont révolutionné la reconnaissance d'image en fournissant des performances exceptionnelles. Les CNN sont spécialement conçus pour extraire automatiquement des caractéristiques pertinentes à partir des images brutes, sans nécessiter une extraction manuelle de ces caractéristiques.

Dans ce chapitre nous allons définir notre architecture de CNN mise en point pour la reconnaissance des images d'animaux, via le langage de programmation python. Nous commençons tout d'abord par la présentation des ressources, du langage et de l'environnement de développement que nous avons utilisé. Puis les étapes d'implémentation du système modèle et on termine par les tests et les résultats effectués.

4.2 Architecture générale du CNN

Notre modèle est basé sur l'architecture séquentielle, Il convient de noter que, bien que les modèles séquentiels de CNN soient linéaires dans leur structure, ils peuvent être suffisamment profonds et complexes pour capturer des motifs visuels et des hiérarchies de caractéristiques de plus en plus abstraites. En ajustant l'architecture, la taille des filtres, le nombre de couches et d'autres hyper paramètres, il est possible de construire des modèles séquentiels de CNN puissants pour la reconnaissance d'images.

Voici ces principales étapes :

- La recherche des data sets puis la préparation de notre data set.
- La recherche et la réalisation d'un modèle en rapport avec le domaine de la classification des images.

- L'entraînement et la préparation du modèle pour la prédiction.
- Estimation du succès et teste de la performance du modèle.

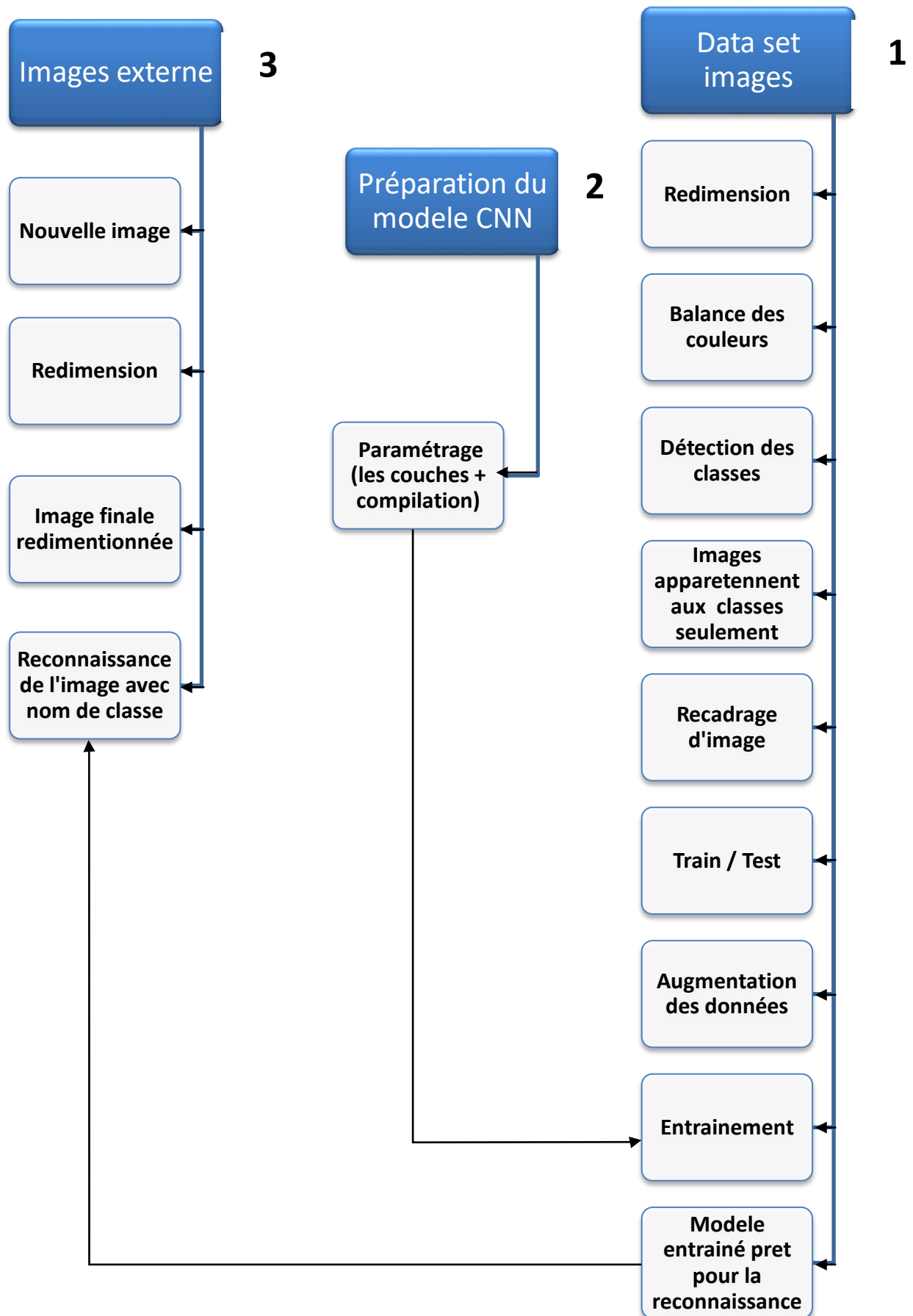


Figure 34 : Conception général de système.

4.3 Environnement de développement

4.3.1 Matériel

Le matériel réalisé est PC personnel Hp avec un 4GB capacité mémoire, et un processeur Intel® core™ i7- CPU @ 1.8GHz, avec Windows 10 édition intégral, service pack 1 64 bits type système.

4.3.2 Logiciel

4.3.2.1 Google Colab, Python, Bibliothèque

- **Google-Colaboratory**

Pendant de nombreuses années, Google a développé un outil de développement appelé Colab oratory (Google colab). Aujourd'hui, Google a rendu Colab oratory gratuit pour une utilisation publique. Google colab est un environnement de bloc-notes gratuit qui fonctionne entièrement dans le cloud. Il permet, ainsi qu'aux membres de l'équipe, de modifier des documents, de la même manière que travaillez avec Google Docs.

Colab prend en charge de nombreuses bibliothèques d'apprentissage automatique populaires qui peuvent être facilement chargées dans le carnet de notes. Avec Google colab, il suffit de quelques lignes de code pour importer un ensemble de données d'images, entraîner un classificateur d'images et évaluer le modèle.

Les notebooks Colab exécutent le code sur les serveurs en nuage de Google, ce qui signifie que les utilisateurs peuvent exploiter la puissance du matériel Google, notamment les GPU et les TPU, quelle que soit la puissance de leur machine. En utilisant un langage de programmation soit : C++ /Python [51].

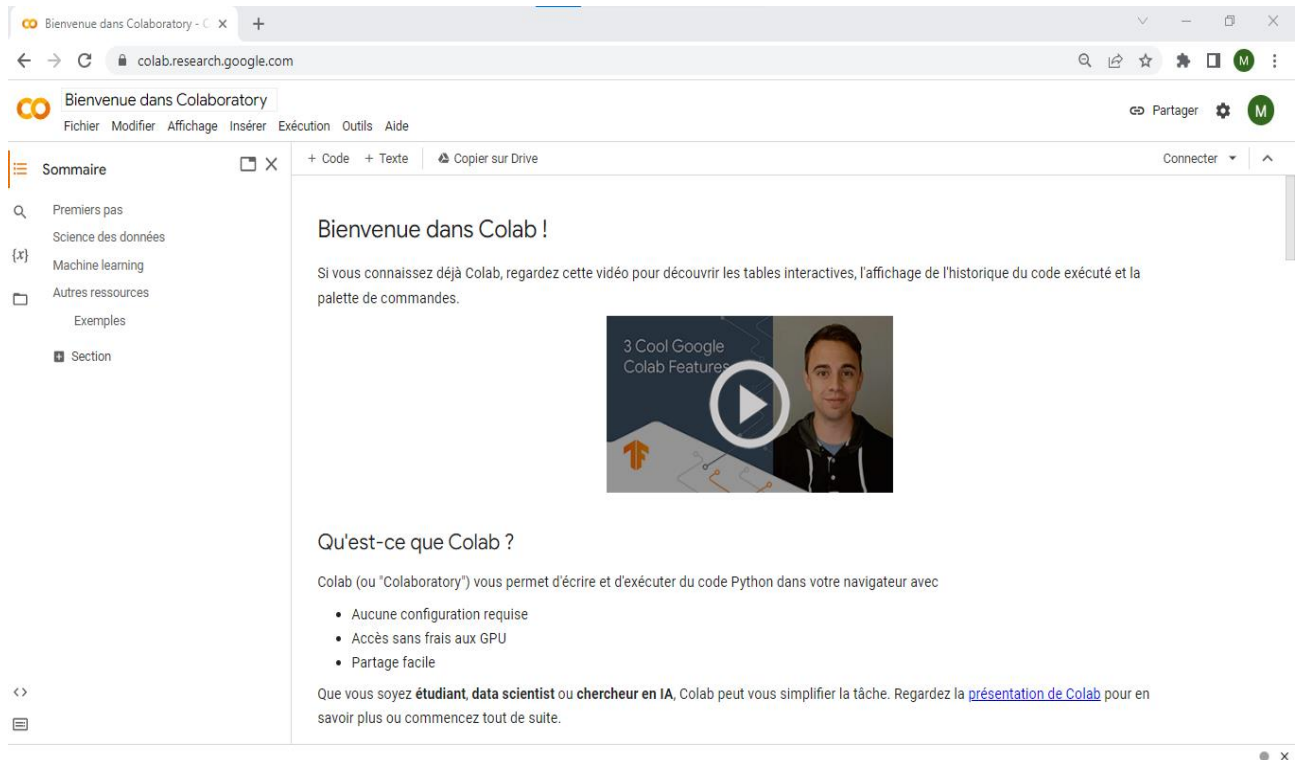


Figure 35 : L'environnement Google Colaboratory.

• Pourquoi utiliser Google Colab ?

Il y a plusieurs raisons pour l'adopter Google Colab au lieu d'une simple instance de Jupyter Notebook :

❖ Bibliothèques préinstallées

La distribution Anaconda de Jupyter Notebook est livrée avec plusieurs bibliothèques de données préinstallées, telles que Pandas, NumPy, Matplotlib, ce qui est génial. Google Colab, quant à lui, fournit encore plus de bibliothèques d'apprentissage automatique préinstallées, telles que Keras, TensorFlow et PyTorch. Sauvegardé dans le Cloud.

Lorsque vous choisissez d'utiliser un simple carnet Jupyter comme environnement de développement, tout est enregistré sur votre machine locale. Si vous êtes prudent en matière de confidentialité, il s'agit peut-être d'une fonctionnalité que vous préférez. Toutefois, si vous souhaitez que vos carnets soient accessibles à partir de n'importe quel appareil avec une simple connexion Google, alors Google Colab est la solution.

Tous vos carnets GoogleColab sont enregistrés sous votre compte Google Drive, tout comme vos fichiers GoogleDocs et Google Sheets.

❖ Collaboration

La fonction de collaboration est une autre fonctionnalité intéressante de Google Colab. Si vous travaillez avec plusieurs développeurs sur un projet, l'utilisation de Google Colab note book est idéale. Tout comme vous collaborez sur un document Google Docs, vous pouvez coder avec plusieurs développeurs à l'aide d'un bloc-notes Google Colab. En outre, vous pouvez également partager votre travail terminé avec d'autres développeurs.

❖ Utilisation gratuite des GPU et TPU

Google Research vous permet d'utiliser ses GPU et TPU dédiés pour vos projets personnels d'apprentissage automatique. D'après mon expérience, pour certains projets, l'accélération des GPU et TPU fait une énorme différence, même pour de petits projets.

En outre, comme il utilise les ressources de Google, les opérations d'optimisation du réseau neuronal ne perturbent pas mes processeurs [52].

❖ Python3

Python est un langage de programmation interprété, orienté objet, de haut niveau et doté d'une sémantique dynamique. Ses structures de données intégrées de haut niveau, combinées au typage dynamique et à la liaison dynamique, le rendent très attrayant pour le développement rapide d'applications, ainsi que pour une utilisation en tant que langage de script.

Python supporte les modules et les packages, ce qui encourage la modularité des programmes et la réutilisation du code. L'interpréteur Python et la bibliothèque standard étendue sont disponibles gratuitement sous forme de source ou de binaire pour toutes les principales plates-formes et peuvent être distribués librement [53].

❖ TensorFlow

TensorFlow est une plateforme open source de bout en bout pour l'apprentissage automatique. Elle dispose d'un système complet et flexible d'outils, de bibliothèques et de ressources communautaires qui permet aux chercheurs de faire avancer l'apprentissage automatique et aux développeurs de créer et de déployer facilement des applications basées sur l'apprentissage automatique [54].

• Pourquoi TensorFlow ?

TensorFlow a facilité l'apprentissage automatique : grâce aux modèles pré-entraînés, aux données et aux API de haut niveau, il est devenu facile pour tout le monde de construire des modèles ML. Principalement utilisé par les chercheurs : La plupart des chercheurs et des étudiants utilisent TensorFlow dans leurs recherches et la construction de modèles.

Modèles prêts à l'emploi à des fins de production : TensorFlow prend en charge des modèles pré-entraînés qui peuvent être utilisés instantanément pour la production et l'expérimentation [55].

❖ **OpenCV**

OpenCV (Open Source Computer Vision Library) est une bibliothèque logicielle open source de vision par ordinateur et d'apprentissage automatique. OpenCV a été construit pour fournir une infrastructure commune pour les applications de vision par ordinateur et pour accélérer l'utilisation de la perception artificielle dans les produits commerciaux. Étant un produit sous licence BSD, OpenCV permet aux entreprises d'utiliser et de modifier facilement le code [56].

• **Pourquoi OpenCV ?**

La bibliothèque compte plus de 2500 algorithmes optimisés, ce qui inclut un ensemble complet d'algorithmes de vision par ordinateur et d'apprentissage automatique classiques et de pointe. Pour produire une image haute résolution d'une scène entière, trouver des images similaires dans une base de données d'images, Ces algorithmes peuvent être utilisés pour détecter et reconnaître des visages, identifier des objets, classer des actions humaines dans des vidéos, suivre les mouvements de la caméra, suivre des objets en mouvement, assembler des images supprimer les yeux rouges des images prises au flash, suivre les mouvements des yeux, reconnaître des paysages et établir des marqueurs pour les superposer à la réalité augmentée, etc [56].

❖ **Keras**

Keras est une API d'apprentissage profond écrite en Python, fonctionnant au-dessus de la plateforme d'apprentissage automatique TensorFlow. Elle a été développée dans le but de permettre une expérimentation rapide. La capacité de passer de l'idée au résultat aussi rapidement que possible est essentielle pour mener de bonnes recherches [57].

❖ **Kaggle**

Kaggle, une filiale de Google LLC, est une communauté en ligne de spécialistes des données et de l'apprentissage automatique. Kaggle permet aux utilisateurs de trouver et de publier des ensembles de données, d'explorer et de construire des modèles dans un environnement de science des données basé sur le Web, de travailler avec d'autres scientifiques des données et ingénieurs d'apprentissage automatique, et de participer à des concours pour résoudre des défis de science des données [58].

```
# Importer les bibliothèques nécessaires
import tensorflow as tf # Importer la bibliothèque TensorFlow
from tensorflow.keras.preprocessing.image import ImageDataGenerator #
from tensorflow.keras.models import Sequential # Importer le module Se
from tensorflow.keras.layers import Conv2D, MaxPool2D, Dense, Flatten
from tensorflow.keras.optimizers import RMSprop # Importer l'optimiseu
```

Figure 36 : Quelques bibliothèques dans le modèle.

4.4. Le data set image

Afin de développer une application à base de réseaux de neurones il est nécessaire de disposer de deux data sets : un data set pour effectuer l'apprentissage et une autre data set pour tester le réseau obtenu et déterminer ses performances.

Dans notre cas, le data set est acquis à partir d'un site Web bien connu (Kaggle), il contient un total de 8000 images elles sont ordonnées dans 6 classes.

Nous allons maintenant présenter les résultats obtenus grâce à des expériences réalisées par le modèle de notre approche de classification des images d'animaux en six classes (class 'Butterfly', class 'Cat', class 'Eagle', class 'Fish' et class 'Monkey'). On utilise deep learning et la méthode de classification CNN. Le schéma de notre système est le suivant :

1. Création des classes.
2. La classification et l'apprentissage.
3. Le test.

Voici les catégories utilisées dans notre base de données :

📁 > Ce PC > Disque local (D:) > dossiers > email_soutenance > base_de_données

| Nom | Modifié le | Type | Taille |
|-------------|------------------|---------------------|--------|
| 📁 Butterfly | 11/06/2023 07:33 | Dossier de fichiers | |
| 📁 Cat | 11/06/2023 07:33 | Dossier de fichiers | |
| 📁 Eagle | 11/06/2023 07:33 | Dossier de fichiers | |
| 📁 Fish | 11/06/2023 07:33 | Dossier de fichiers | |
| 📁 Monkey | 11/06/2023 07:33 | Dossier de fichiers | |
| 📁 Tiger | 11/06/2023 07:33 | Dossier de fichiers | |

Figure 37 : Les 6 classes dans le dataset.

a) Création des classes

Nous allons rechercher les images des mêmes classes d'animaux, à savoir Butterfly, Cat, Eagle, Fish, Monkey et Tiger. Nous allons ensuite classer les fichiers en vue de la préparation de l'apprentissage.

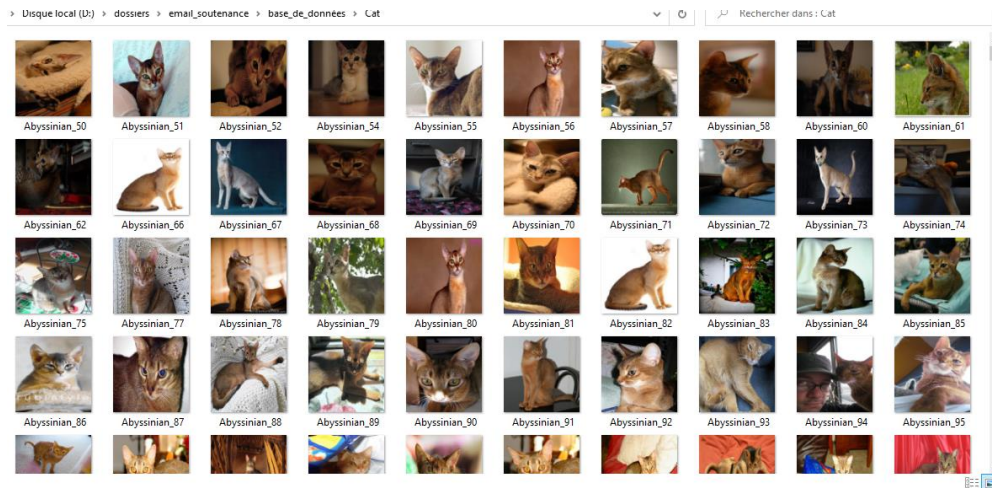


Figure 38 : Création de classes (classe 'Cat').

b) Data set d'apprentissage et sata set de test

L'entraînement sera réalisé sur 80% des données, soit 6000 images, tandis que les tests seront effectués sur 20% des données, soit 1500 images.

```
# Définir le nombre total d'images d'entraînement et de validation
total_train = 6000
total_val = 1500

batch_size = 20 # Définir la taille du lot pour l'entraînement
# batch_size : nombre d'échantillons d'entraînement utilisés dans un

epochs = 20 # Définir le nombre d'époques pour l'entraînement
```

Figure 39 : Définir le nombre total d'images.

c) Phase de prétraitement

La phase de prétraitement dans ce modèle est effectuée à l'aide de la classe ImageDataGenerator de la bibliothèque tensorflow.keras.preprocessing.image. Voici les étapes de prétraitement :

Mise à l'échelle des valeurs de pixels des images d'entraînement et de validation entre 0 et 1 à l'aide de l'instance `train_datagen` et `validation_datagen` respectivement :

```
train_datagen = ImageDataGenerator(rescale=1./255) # Mise à
validation_datagen = ImageDataGenerator(rescale=1./255) # M
# rescale : option de prétraitement courante qui normalise
```

Figure 40 : prétraitement des images + entraînement + validation.

Génération des lots de données d'entraînement augmentées à partir du répertoire d'entraînement :

```
train_generator = train_datagen.flow_from_directory(train_dir,
                                                    target_size=(img_width,img_height),
                                                    batch_size=batch_size,
                                                    class_mode='categorical')
```

Figure 41 :Augmentationdes images d'entraînement.

Génération des lots de données de validation à partir du répertoire de validation :

```
validation_generator = validation_datagen.flow_from_directory(validation_dir,
                                                             target_size=(img_width,img_height),
                                                             batch_size=batch_size,
                                                             class_mode='categorical')
```

Figure 42 :Augmentation des images validation.

La mise à l'échelle des valeurs de pixels permet de normaliser les images et de les rendre compatibles avec le modèle CNN. L'augmentation des données d'entraînement est réalisée par `ImageDataGenerator` pour générer des variations aléatoires des images, ce qui aide à augmenter la diversité des données d'entraînement et à améliorer la capacité du modèle à généraliser.

Veillez noter que le code fourni ne montre pas les étapes spécifiques de prétraitement des images individuelles, telles que le redimensionnement des images à la taille spécifiée (150x150) ou la conversion des images en tenseurs, car ces étapes sont effectuées automatiquement par les générateurs de données lors de l'utilisation des méthodes `flow_from_directory` et `predict` du modèle.

d) Phase d'apprentissage

Nous préparons les classes que nous choisissons est exécuter le code de l'apprentissage en Colab.

Le lancement de l'apprentissage renversée les données et remodelée toutes les images à la même taille.

```
# Définir la taille de l'image  
# ( largeur et hauteur de chaque image est :150 )  
img_width, img_height = 150, 150
```

Figure 43: Redimensionner les images.

e) Création du réseau

[1] La Couche de convolution(CONV)

La convolution est un outil mathématique simple qui est très largement utilisé pour le traitement d'image, ce qui explique que les réseaux de neurones à convolution soient particulièrement bien adaptés à la reconnaissance d'image. La convolution agit comme un filtrage. On définit une taille de fenêtre qui va se balader à travers toute l'image (rappelez-vous qu'une image peut être vue comme étant un tableau).

Au tout début de la convolution, la fenêtre sera positionnée tout en haut à gauche de l'image puis elle va se décaler d'un certain nombre de cases (c'est ce que l'on appelle le pas) vers la droite et lors qu'elle arrivera au bout de l'image, elle se décalera d'un pas vers le bas ainsi de suite jusqu'à ce que le filtre est parcourue la totalité de l'image [57].

[2] La représentation finale du modèle du réseau de neurones convolutif

```

# Créer un modèle séquentiel
model = Sequential()
# Ajouter une couche Conv2D avec 32 filtres, un noyau de taille 3x3, une activation 'relu' et une forme d'entrée (150, 150, 3)
model.add(Conv2D(32, (3,3), activation='relu', input_shape=(150,150,3)))
# Ajouter une couche MaxPool2D avec une taille de pool de 2x2
model.add(MaxPool2D((2,2)))

model.add(Conv2D(64, (3,3), activation='relu'))
model.add(MaxPool2D((2,2)))

model.add(Conv2D(128, (3,3), activation='relu'))
model.add(MaxPool2D((2,2)))

model.add(Conv2D(128, (3,3), activation='relu'))
model.add(MaxPool2D((2,2)))

# Appliquer l'aplatissement des données
model.add(Flatten())

# Ajouter une couche Dense avec 512 neurones et une activation 'relu'
model.add(Dense(512, activation='relu'))

# Ajouter une couche Dense finale avec 6 neurones correspondant aux classes de sortie et une activation 'softmax'
model.add(Dense(6, activation='softmax'))

```

Figure 44 : Construction du modèle de réseau de neurones convolutif.

Nous présentons dans la figure ci-après un résumé de notre modèle :

f) Compilation du Modèle

La compilation du modèle en utilisant **RMSprop** comme optimiseur, **categorical_crossentropy** comme fonction de perte et spécifions **accuracy** comme la métrique que nous voulons suivre :

```

model.compile(loss='categorical_crossentropy', optimizer=RMSprop(lr=0.001), metrics=['accuracy'])

```

Figure 45: Compilation do modèle.

4.6 Résultats et discussions

Les résultats obtenus à la fin de l'entraînement en fonction du nombre des Epochs. Nous présentons dans la figure ci-après un résumé de notre modèle :

```

Epoch 17/20
300/300 [=====] - 23s 78ms/step - loss: 0.1329 - accuracy: 0.9637 - val_loss: 0.1656 - val_accuracy: 0.9540
Epoch 18/20
300/300 [=====] - 25s 83ms/step - loss: 0.1273 - accuracy: 0.9660 - val_loss: 0.0425 - val_accuracy: 0.9847
Epoch 19/20
300/300 [=====] - 24s 79ms/step - loss: 0.1230 - accuracy: 0.9653 - val_loss: 0.0611 - val_accuracy: 0.9813
Epoch 20/20
300/300 [=====] - 24s 80ms/step - loss: 0.1244 - accuracy: 0.9657 - val_loss: 0.0297 - val_accuracy: 0.9907
<ipython-input-9-482c8ea97567>:63: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future version. Please
scores = model.evaluate_generator(validation_generator, steps=total_val//batch_size)
accuracy: 98.93%
    
```

98,93 %

Figure 46 : résumé du modèle.

Courbes de perte :

La courbe de perte d'entraînement diminue régulièrement et atteint une valeur basse, cela indique que le modèle apprend efficacement et converge vers une solution optimale.

Courbes d'exactitude :

La courbe d'exactitude d'entraînement croissante indique que le modèle s'améliore et apprend à effectuer des prédictions correctes sur les données d'entraînement.

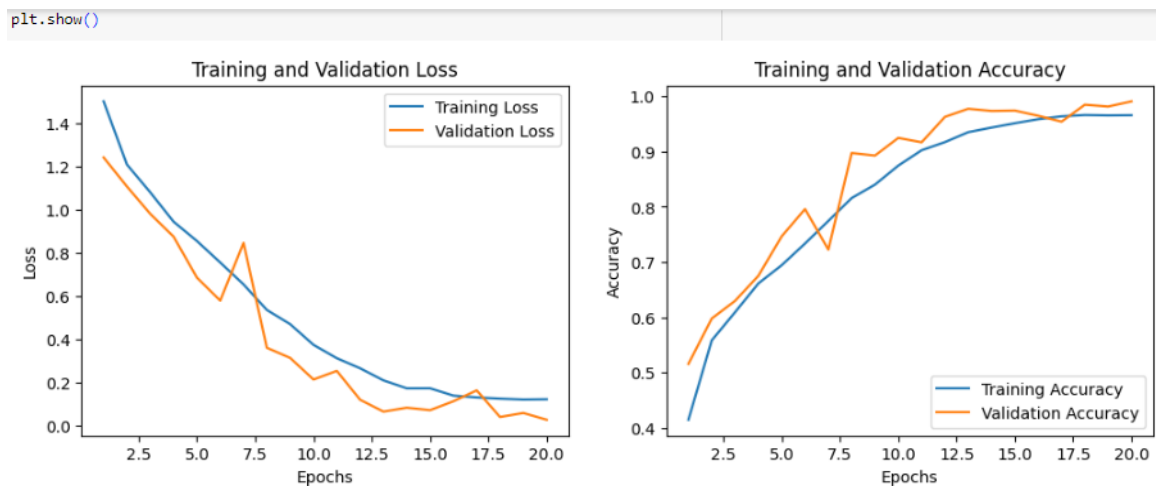


Figure 47: Résultat d'entraînement en fonction des Epochs.

a) Le test

Pour tester notre modèle et juger son performance nous avons testé sur une bases d'images. La relation que nous avons utilisée pour calculer le pourcentage de notre classification est: $P=R/T$

P : Le pourcentage de classification

R : Nombre d'image reconnue

T : Nombre total d'image de test.

Nous avons obtenu les résultats suivants :

Tableau 5 : Résultats de test pour chaque classe.

| Nom du classe: | Nombre des images reconnues | Nombre des images non reconnues | Nombre des images de test totale : | Taux de reconnaissance |
|----------------|-----------------------------|---------------------------------|------------------------------------|------------------------|
| Fish | 94 | 2 | 96 | 98 % |
| Betterfly | 120 | 0 | 120 | 100 % |
| Cat | 47 | 1 | 48 | 98% |
| Monkey | 99 | 2 | 101 | 99 % |
| Eagle | 76 | 6 | 82 | 93.5% |
| Tiger | 50 | 1 | 49 | 98% |

Le modèle a pu atteindre une précision globale de **98.93%**.Voici un résultat de test en utilisant un exemple d'image.



Figure 48 : Résultat de test en utilisant un exemple d'image de test.

4.5. Conclusion

Dans ce chapitre nous avons donné les outils nécessaires pour la réalisation de notre travail. Nous avons présenté aussi l'environnement de développement.

Aussi, nous avons présenté notre modèle en donnant quelques captures d'écran qui expliquent le déroulement et le fonctionnement de notre approche, ainsi que les résultats obtenus . On a donné une exemple de reconnaissance d'images par notre approche.

A la fin, nous avons montré l'impact de la méthode choisi (basée CNN) sur la qualité les résultats de la reconnaissance (par classification).

Conclusion générale

Par leur succès bien avéré dans la classification des images, et la reconnaissance d'objets, nous avons opté pour les réseaux de neurones profonds et plus particulièrement des réseaux de neurones de convolution (CNN : Convolutional Neural Networks). Actuellement, et vue la littérature du domaine, il s'agit du modèle le plus performant pour la classification et la recherche d'image.

Nous avons présenté les outils, les plateformes et les bibliothèques que nous avons considéré dans notre mise en œuvre. Ensuite, nous avons exposé les détails de notre modèle, en commençant par le data set qui nous a servis pour l'apprentissage du modèle et pour les tests. Les résultats de tests que nous avons obtenus, montrent l'intérêt de l'utilisation des modèles à base des CNN.

Pour le futurs, nous comptons faire collaborer le modèle CNN que nous avons construit avec une application (en Java) qui permet l'interface entre l'utilisateur (médecin, ingénieur, ...etc.) afin de la faire profiter de notre travail.

Bibliographie

- [1] Platon : République, trad. par E. Chambry, Les belles lettres, Paris, 1961.
- [2] M. ANDRE, Introduction aux techniques de traitement d'images, Eyrolles, 1987.
- [3] <https://www.bang-marketing.com/publications/ebang/comprendre-enfin-les-differents-types-de-fichiers-graphiques/>
- [4] <https://www.imedias.pro/cours-en-ligne/graphisme-design/definition-resolution-taille-image/les-images-vectorielles-matricielles/>
- [5] <https://www.chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://edu1d.ac-toulouse.fr/politique-educative-31/site-ressources31/files/04-photofiltre7-resolution-definition-simplifie.pdf>
- [6] <https://www.web-creatif.net/des-photos-de-qualite-definition-et-resolution-methode-facile-iv/>
- [7] ASSAS Ourda, Classification floue des images, Université de Batna ,2013.
- [8] http://fr.wikipedia.org/wiki/R%C3%A9solution_spatiale_des_images_matricielles
- [9] <https://perso.esiee.fr/~perretb/15FM/TAI/histogramme/index.html>
- [10] http://bnazarian.free.fr/MyUploads/IN_GBM_06_BINAIRES.PDF
- [11] Gonzalez, R. C, &Woods, R. E, Traitement d'images numériques, De Boeck Supérieur ,université de Tennessee, 2008.
- [12] https://www.fr.m.wikibooks.org/wiki/Fichier:Bits_et_niveaux_de_gris.svg
- [13] <https://www.nicolasroger.fr/2015/01/definition-et-resolution-dune-image-numerique/>
- [14] Pratt, W. K, Traitement d'images numériques : à l'intérieur de PIKS (4e éd.),Wiley, 2007.
- [15] <https://www.waebo.com/webp-compression-sans-perte-d-images-et-transparence.html>
- [16] <https://www6.inrae.fr/pfl-cepia/Axe-images/Tutoriel/L-image-numerique>
- [17] <http://www.map.toulouse.archi.fr/works/panoformation/imagenum/imagenum.htm>
- [18] <https://www.bang-marketing.com/publications/ebang/comprendre-enfin-les-differents-types-de-fichiers-graphiques/>
- [19] https://www.fr.wikipedia.org/wiki/Traitement_d%27images[Consulté le 29/mars/2023]
- [20] https://www.researchgate.net/figure/Binarisation-dune-portion-dimage-ERS1-183-x-159-pixels_fig27_273017562
- [21] B. GOSSELIN, " Application des réseaux de neurones artificielles aux reconnaissances automatiques de caractères manuscrits ", Thèse de Doctorat, Faculté Polytechnique de Mons, 1996 .
- [22] https://www.researchgate.net/figure/Figure10-un-exemple-de-squelettisation-dun-objet-une-image-La-squelettisation_fig16_324530779
- [23] http://www.fr.wikipedia.org/wiki/R%C3%A9solution_spatiale_des_images_matricielles
- [24] <http://humbert-lorent.tp-developpez.com/algorithmique/traitement/traitement.pdf>

- [25] http://fr.wikipedia.org/wiki/R%C3%A9solution_spatiale_des_images_matricielles
- [26] C. LURETTE, " Développement d'une technique neuronale auto-adaptative pour la classification dynamique de données évolutives, Application à la supervision d'une presse hydraulique ", Thèse de Doctorat, Université des sciences et technologies de LILLE, 2003.
- [27] J. Bezdek, L. Hall, L. Clarke, " Review of MR Image Segmentation Techniques using Pattern Recognition ", Medical Physics, 20:1033-1048, 1993.
- [28] Bishop, C. M, Pattern Recognition and Machine Learning (Traduction française), Dunod, 2006.
- [29] Patil, S. & Kulkarni, U. Accuracy Prediction for Distributed Decision Tree using Machine Learning approach, 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019.
- [30] KORTI Djazila Souhila, mémoire Classification des obstacles par la technique d'apprentissage profond, Université Aboubakr Belkaid – Tlemcen –, 2019-2020
- [31] Goodfellow, I., Bengio, Y., & Courville, A. Deep Learning. MIT Press, Cambridge, 2016.
- [32] <https://metalblog.ctif.com/2021/04/12/les-reseaux-de-neurones-et-la-data-augmentation/> [Consulté le 18/mars/2023]
- [33] Haykin, S. , Neural Networks and Learning Machines (3rd Edition), Pearson, 2009.
- [34] E.M, Olasagasti, « Application des réseaux de neurones à l'identification d'un axe de machine-outil, » These de doctorat, Institut National Polytechnique de Grenoble, 2002.
- [35] X. Liu, S. Li, M. Kan, J. Zhang, S. Wu, W. Liu, H. Han, S. Shan, X. Chen, Aenet: Deeply learned regressor and classifier for robust apparent age estimation. in: ICCV Workshops: 16-24, 2015
- [36] https://www.researchgate.net/figure/Modele-dun-neurone-artificiel_fig30_324929383
- [37] <http://informatique.coursgratuits.net/methodes-numeriques/reseaux-de-neurones-formels.php>
- [38] https://www.researchgate.net/figure/Comparaison-entre-neurone-biologique-et-neurone-artificiel_fig6_281015872
- [39] Wikipedia, perceptron multicouche, < https://www.fr.wikipedia.org/wiki/perceptron_multicouche >.
- [40] Nadia, A. I. Une approche IA pour la reconnaissance des expressions faciales, Algérie/Bouira, (2019-2020)..
- [41] https://www.researchgate.net/figure/Exemple-dune-operation-de-convolution_fig3_337635267
- [42] <https://www.quantmetry.com/blog/mobilenet-optimisation-de-la-convolution-pour-les-reseaux-de-neurones-embarques/>
- [43] LILIA, D. R, La Détection de la colère chez le conducteur en utilisant le Deep Learning. Algérie/BISKRA, 2020.
- [44] https://www.researchgate.net/figure/An-example-of-pooling-with-a-2-2-filter-and-a-stride-of-2_fig4_324165524
- [45] Foued, N, Reconnaissance d'expression faciale à partir d'un visage réel, Algérie/Guelma, (2019).

- [46] M. NOURESSADAT, «Etude des performances des réseaux de neurones dynamiques à représenter des systèmes réels :une approche dans l'espace d'état,» Thèse de magister, Université de Sétif, 2009.
- [47] <https://www.blog.hubspot.fr/website/differents-types-fichiers-image>
- [48] BEGGARI Samiha,KHAMRAKhaoula, Système de reconnaissance de visage par un réseau de neurone convolutionnel(CNN),2017
- [49] Mage waran Architecture générale d'un CNN ,2019.
- [50] Dj. Y. Moualek «Deep Learning pour la classification des images» Mémoire de master, Université Abou Bakr Belkaid, Tlemcen, 2017
- [51] [Enligne].<https://www.medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5>.
- [52] F. Nacer «Reconnaissance d'expression faciale à partir d'un visage réel» Mémoire de master, Université de 8/Mai/19450, Guelma , 2019
- [53] Machinelearning.<https://colab.research.google.com/notebooks/intro.ipynb#scrollTo=OwuxHmxllTwN>
- [54] 4 Reasons Why You Should Use Google Colab for Your Next Project. Disponible à l'adresse <https://towardsdatascience.com/4-reasons-why-you-should-use-google-colab-for-your-next-projectb0c4aaad39ed>
- [55] 4 Reasons Why You Should Use Google Colab for Your Next Project. Disponible à l'adresse <https://towardsdatascience.com/4-reasons-why-you-should-use-google-colab-for-your-next-projectb0c4aaad39ed>
- [56] What is Python? Executive Summary
<https://www.python.org/doc/essays/blurb/>
- [57] Why TensorFlow is So Popular – Tensorflow Features Disponible à l'adresse <https://www.geeksforgeeks.org/why-tensorflow-is-so-popular-tensorflow-features/>
- [58] About. Disponible à l'adresse <https://opencv.org/about/>
- [59] Qu'est-ce que la convolution. <https://www.natural-solutions.eu/blog/lareconnaissance-dimage-avec-les-rseaux-de-neurones-convolutifs>.
- [60] Lardinois, Frederic; Mannes, John; Lynley, Matthew (March 8, 2017). "Google is acquiring data science community Kaggle". Techcrunch. Archived from the original on March 9, 2017.Retrieved March 9, 2017. Sources tell us that Google is acquiring Kaggle [...] the official announcement could come as early as tomorrow. "Google buys Kaggle and its gaggle of AI geeks". CNET. 2017-03-08. Retrieved 2018-06-01