RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université de Mohamed El-Bachir El-Ibrahimi- Bordj Bou Arreridj

Faculté *des Sciences et de la technologie*

Département d'électronique

Mémoire

Présenté pour obtenir

LE DIPLÔME DE MASTER

FILIÈRE : Électronique

Spécialité : Électronique des Systèmes Embarqués

Par

- **BESSA Youcef**
- **KOUAR Rezki**

Intitulé

Générateur de légendes d'image utilisant CNN et LSTM

Soutenu le : 04/072023

Devant le Jury composé de :

Nom & Prénom	*Grade*	*Qualité*	*Établissement*
M. N.Diffellah	*MCA*	*Président*	*Univ-BBA*
M. Dj.E.Boudechiche	*MCB*	*Encadreur*	*Univ-BBA*
M. F.Hamadache	*MAA*	*Examinateur*	*Univ-BBA*

Année universitaire : 2022/2023

Dédicace

Qui est-ce que je préfère à moi-même et pourquoi pas ? Tu t'es sacrifié pour moi tu n'épargnes toujours aucun effort pour me rendre heureux (chère mère).

Nous marchons sur les chemins de la vie, et celui qui contrôle nos esprits reste sur chaque chemin que nous empruntons. Le propriétaire d'un bon visage et de bonnes actions. Il ne m'a pas gardé toute sa vie (mon cher père).

À ceux qui m'ont quittée, pas un seul instant, et à ceux qui étaient à mes côtés dans mes moments les plus difficiles et qui m'ont mis la pression.
Ma présence a été mon plus grand soutien et la plus grande incitation à continuer.

À mon frère, à mes sœurs et à mes neveux.

À ma grande famille et à tous mes proches.

À ma ville natale (municipalité de Ksour)

Dans mon quartier où j'ai grandi et tous mes voisins (1008 logements)

À mon école primaire (École des martyrs Bilhoul Al-Arbi BBA)

À mon collège (mi-11 décembre 1960)

À mon lycée (Sheikh Mohammed Al-Mekrani)

À mon université (Université Sheikh Mohamed El-Bachir El-Ibrahimi)

À ma Faculté (Faculté des sciences et de la technologie)

À mon Département (Département d'électronique)

À mes collègues dans spécialiste électronique des systèmes embarques M2/2023

Vous présentes cette recherche, et j'espère qu'elle vous satisfera….

« Youcef Bessa »

Tout d'abord, je tiens à remercier DIEU
De m'avoir donné la force et le courage de mener
A bien ce modeste travail.

La recherche a traversé de nombreux obstacles, mais j'ai essayé de les surmonter avec persévérance, louange à Dieu et de sa part.
Pour mes parents, mes frères et sœurs et mes amis, ils ont été comme un soutien et un soutien pour mener à bien la recherche.
Je ne dois pas oublier mes professeurs qui ont eu le plus grand rôle en me soutenant et en me donnant de précieuses informations….
Je te donne mon message…
Nous demandons à Dieu Tout-Puissant de prolonger votre vie et de vous accorder le bien.

« KouarRezki»

Remerciements

Nous remercions Dieu Tout-Puissant qui nous a permis de mener à bien cette recherche scientifique et qui l'a inspirée avec santé, bien-être et détermination. Dieu merci, merci beaucoup. Nous adressons nos sincères remerciements et notre reconnaissance au professeur superviseur Dr Boudechiche Djamel pour tous les précieux conseils et informations qu'il nous a fournis et qui ont contribué à compléter le sujet de notre étude dans ses différents aspects.

Je veux exprimer par ces quelques lignes de remerciement notre gratitude envers tous ceux en qui par leur présence, leur soutien, leur disponibilité et leurs conseils, nous avons eu courage d'accomplir ce travail

Je remercierai les membres du jury de j'avoir fait l'honneur de faire partie de mon jury de ce travail.

C'est une iniquité et un déni de beauté si j'allais juste dire merci à cette personne qui avait mon lien et ma mission et qui ne m'a pas quitté dès le début de mon travail dans mes mémoires et jusqu'à la fin. Le meilleur d'Aoun et le meilleur de l'ami et bon de lien dans ces jours et moments qui semblaient difficiles, mais en sa présence j'ai pu surmonter tout cela et aujourd'hui je vous offre mon travail et mille grâce à lui.

Je remercierai également toutes les personnes qui, de près ou de loin, m'aident à l'élaboration de ce mémoire.

Merci à tous les amies et les membres de la famille pour leurs aides leur soutien et leurs encouragements dans les moments difficiles

Résumé

Le générateur de légende d'image est un modèle de réseau de neurones qui peut générer des légendes descriptives pour les images. Le modèle utilise un réseau CNN (*convolutional neurone network*) pour extraire les caractéristiques visuelles de l'image, qui sont ensuite alimentées dans un réseau LSTM (*long short term memory*) pour générer la légende.

Le CNN est utilisépour extraire des caractéristiques de haut niveau à partir de l'image, telles que la forme et la couleur, tandis que le LSTM est utilisé pour générer une séquence de mots qui décrivent l'image. Le modèle est entraîné sur un grand ensemble de données d'images avec des légendes correspondantes, de sorte qu'il puisse apprendre à associer des descriptions textuelles aux caractéristiques visuelles.

Le générateur de légende d'image est utile pour une variété d'applications, telles que la création de descriptions pour les images dans les bases de données d'images et les réseaux sociaux, la création de légendes pour les vidéos et les films, et même l'assistance pour les personnes ayant une déficience visuelle.

Abstract

The image caption generatoris a neural network model that can generate descriptive captions for images. The model uses a CNN (convolutional neural network) to extract the visualcharacteristics of the image, which are thenfedinto a LSTM (long short term memory) to generate the legend.

The CNN is used to extract high-level characteristics from the image, such as shape and color, while the LSTM is used to generate a sequence of wordsthatdescribe the image. The model istrained on a large set of image data with corresponding captions, so that it can learn to associate textual descriptions with visual characteristics.

The image caption generatorisuseful for a variety of applications, such as creating descriptions for images in image databases and social networks, creating captions for videos and movies, and even assisting people with visual impairments.

ملخص :

مولد أسطورة الصور باستخدام هو نموذج شبكة عصبية يمكنه إنشاء تعليقات وصفية للصور. يستخدم النموذج شبكة عصبية ملتفة (CNN) لاستخراج الخصائص البصرية للصورة، والتي يتم إدخالها بعد ذلك في شبكة من الخلايا العصبية المتكررة (LSTM) لتوليد الأسطورة.

يتم استخدام CNN لاستخراج ميزات عالية المستوى من الصورة، مثل الشكل واللون، بينما يتم استخدام LSTM لإنشاء سلسلة من الكلمات التي تصف الصورة. يتم تدريب النموذج على مجموعة كبيرة من بيانات الصورة مع التسميات التوضيحية المقابلة، بحيث يمكنه تعلم ربط الأوصاف النصية بالميزات المرئية.

مولد أسطورة الصور مفيد لمجموعة متنوعة من التطبيقات، مثل إنشاء أوصاف للصور في قواعد بيانات الصور والشبكات الاجتماعية، وإنشاء تسميات توضيحية لمقاطع الفيديو والأفلام، وحتى مساعدة ضعاف البصر.

Sommaire

Liste des Figures

Listes des tableaux

Liste des Abréviations

BM : Bitmap

RGB: Rouge Vert Bleu

GIF: Graphic Interchange Format

JPEG: Joint Photo Expert Group

FPX: Flash pix

PCD: Photo CD

PNG Portable Network Graphic

PSD: PhotoShop Document

PSP: Paint Shop Pro

TIF: Tagged Image File Format

ANN: Artificial Neural Network

NN : Neural Network

CNN : Réseau de neurones à convolution

RNN : Réseaux neuronaux récurrents

LSTM : Réseaux Long Short-Term Memory

IA: Intelligence Artificial

DL: Deep Learning

ML: Machine Learning

ResNet: Residual Network

OS: Operating System

NLTK; abréviation de Natural Language Toolkit

NumPy: *Numerical Python*

PIL: *Python Imaging Library*

Introduction générale

L'avènement des réseaux de neurones convolutionnels (CNN) et des réseaux de neurones récurrents avec une mémoire à court terme (LSTM) a révolutionné de nombreux domaines de l'apprentissage automatique, y compris celui de la génération de légendes d'images. Le générateur de légende d'image utilisant CNN et LSTM est un modèle puissant qui combine les capacités d'extraction de caractéristiques visuelles du CNN avec les capacités de génération de séquences du LSTM pour créer des descriptions textuelles détaillées et précises des images.

Dans le contexte de l'analyse d'images, les CNN ont prouvé leur efficacité en tant qu'outils de traitement d'images. Ils sont capables d'apprendre des représentations visuelles de haut niveau à partir des images, en capturant des informations telles que les contours, les textures et les relations spatiales entre les objets présents. Cependant, ces caractéristiques visuelles ne sont souvent pas suffisantes pour décrire complètement une image.

C'est là que les réseaux de neurones récurrents (RNN), et plus particulièrement les LSTM, entrent en jeu. Les LSTM sont des modèles de réseau de neurones capables de traiter des séquences de données et de capturer les dépendances à long terme entre les éléments d'une séquence. En utilisant un LSTM, il est possible de générer une séquence de mots qui décrit de manière cohérente et contextuelle les caractéristiques visuelles extraites par le CNN.

Le générateur de légende d'image utilisant CNN et LSTM est généralement entraîné sur de vastes ensembles de données d'images associées à leurs légendes correspondantes. Grâce à cette quantité importante de données, le modèle apprend à établir des correspondances entre les caractéristiques visuelles extraites par le CNN et les mots qui composent les légendes. Il est ainsi capable de générer des légendes précises et cohérentes pour des images qu'il n'a jamais vues auparavant.

Ce type de modèle présente de nombreuses applications pratiques, que ce soit dans le domaine de l'indexation et de la recherche d'images, où les légendes générées peuvent faciliter l'identification et la classification des images, ou dans les réseaux sociaux, où les utilisateurs peuvent bénéficier de légendes automatiques pour accompagner leurs publications. De plus, le générateur de légende d'image peut également être utilisé pour améliorer l'accessibilité des contenus visuels pour les personnes malvoyantes ou non voyantes, en leur fournissant des descriptions textuelles des images.

En résumé, le générateur de légende d'image utilisant CNN et LSTM représente une avancée significative dans le domaine de la génération de légendes d'images. En exploitant les capacités d'extraction de caractéristiques visuelles des CNN et les capacités de génération de séquences des LSTM, ce modèle est capable de générer des légendes descriptives et précises qui ajoutent une dimension textuelle à la compréhension des images.

Ce manuscrit commence par une introduction générale. Par la suite, il est réparti en quatre chapitres :

- Dans le premier chapitre, nous avons présenté les concepts de base liés à la représentation des images et leurs types et montré les méthodes de filtrage.

- Cependant, le deuxième chapitre est consacré à l'intelligence artificielle, l'apprentissage en profondeur. Fondamentalement, à travers le CNN et le LSTM.
- Dans le dernier chapitre, nous présentons le modèle Resnet-152 et expliquons le programme ainsi que le CNN et le LSTM principalement par les réseaux de neurones et l'apprentissage profond supervisé. Ce qu'il faut faire, c'est donner une étiquette courte et claire.Aussi, dans le dernier chapitre, nous discutons du programme sur lequel nous avons travaillé pour obtenir les résultats de notre étude, qui a été représenté, et dans ce contexte, nous avons utilisé des bibliothèques Python différentes en utilisant Google Colab comme environnement de travail.
- Finalement, le mémoire se termine par une conclusion générale et quelques perspectives.

Chapitre 1 : Initialisation de traitement d'image

Résumé :

Les images numériques sont présentées sous forme de matrices pour effectuer des opérations sur celles-ci, en particulier de calcul, afin de traiter les pixels de l'image, de les analyser et d'extraire leurs caractéristiques.

Par conséquent, dans ce chapitre, nous présenterons les concepts généraux de l'image et de son traitement numérique.

1 .1Introduction :

La discipline du traitement et de l'analyse d'images numériques se concentre sur l'étude des images numériques et de leurs transformations afin d'améliorer leur qualité ou d'en extraire des informations. Elle fait partie du domaine plus large du traitement et de l'analyse de signaux, mais est spécifiquement dédiée aux images et aux données qui en découlent, comme les vidéos. Contrairement aux techniques analogiques telles que la photographie traditionnelle ou la télévision, le traitement d'images opère dans le domaine numérique.

Dans le contexte de la vision artificielle, le traitement d'images intervient après les étapes d'acquisition et de numérisation. Son objectif est de garantir que les transformations et les calculs effectués sur les images permettent une interprétation précise des données traitées. De plus en plus, cette phase d'interprétation est intégrée au processus de traitement d'images, en utilisant notamment l'intelligence artificielle pour exploiter les connaissances disponibles sur le contenu des images traitées (connaissance du domaine). Cela permet d'améliorer la précision et l'efficacité de l'analyse des images, ouvrant la voie à des applications avancées dans des domaines tels que la reconnaissance d'objets, la détection de motifs ou la compréhension du contexte visuel.[1]

1.2L'image numérique

Une image est une représentation bidimensionnelle d'une fonction *f(x, y),* où *x* et *y* sont des coordonnées spatiales et les amplitudes aux différents points *(x,y)* correspondent à des niveaux d'intensité ou de gris. Lorsque ces points et amplitudes sont discrétisés, nous obtenons une image numérique, où la fonction f est symbolisée par une lettre et les points *(x,y)* sont remplacés par des paires de nombres *(i,j)*. Une image numérique est composée d'un ensemble de points appelés pixels (abréviation de "*élément d'image*"). Les pixels sont généralement de forme rectangulaire ou parfois carrée. Leur taille peut être ajustée en modifiant les paramètres de l'écran ou de la carte graphique, de sorte que les pixels représentent les plus petites unités constitutives de l'image numérique. Tous ces pixels sont organisés dans un tableau à deux dimensions qui forme l'image dans son ensemble.[2]

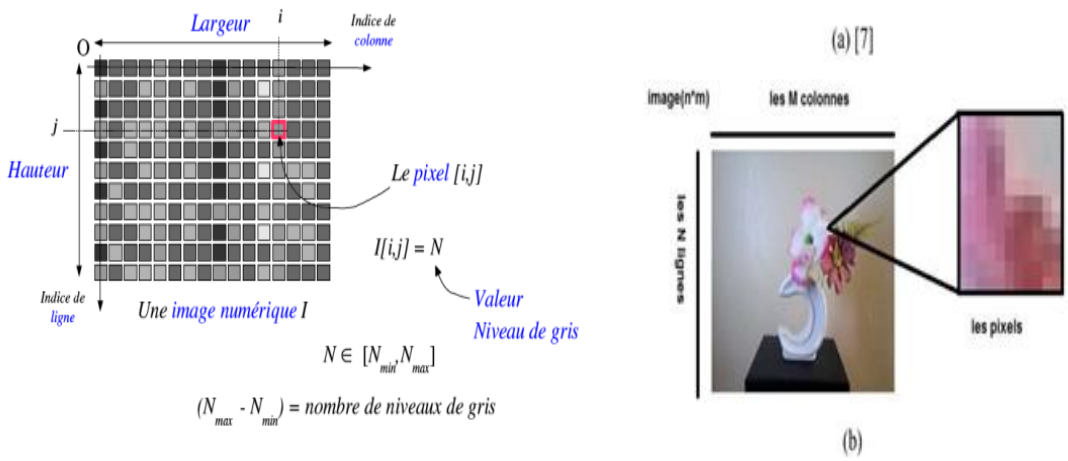


Figure 1. 1 : Représentation des notions image et pixel (c) Représentation matricielle (d) Image réelle [2]

1.3 Types d'images

Nous distinguons trois types d'images :

1.3.1 Image binaire

Les images numériques sont des reproductions électroniques de scènes visuelles ou des numérisations de documents tels que des photographies, des manuscrits, des textes imprimés et des œuvres d'art. Elles sont échantillonnées et représentées sous forme d'une grille de points ou d'éléments d'image appelés pixels. Chaque pixel est associé à une valeur de couleur (noir, blanc, niveaux de gris ou couleur), qui est représentée par un code binaire composé de zéro et des uns. Ces codes binaires, ou "*bits*", correspondant à chaque pixel, sont enregistrés dans un ordre séquentiel par l'ordinateur, et peuvent souvent être compressés mathématiquement pour économiser de l'espace de stockage. Lorsqu'ils sont interprétés et lus par un ordinateur, ces bits sont convertis en une version analogique de l'image pour être affichés ou imprimés.[2]

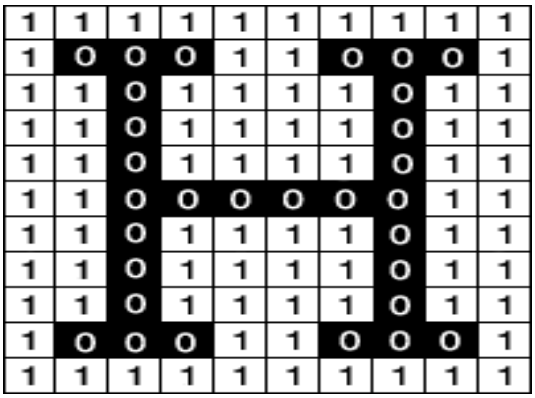


Figure 1.2 : Image binaire [8]

Comme indiqué dans l'image binaire, chaque pixel se voit assigner deux valeurs, dans ce cas 0 pour le noir et 1 pour le blanc.

1.3.2 Image en niveaux de gris

Chaque pixel est un niveau de gris, allant de 0 (noir) à 255 (blanc). Cet intervalle de valeurs signifie que chaque pixel est codé sur huit bits (un octet). 256 niveaux de gris sont généralement suffisants pour la reconnaissance de la plupart des objets d'une scène.[2]

Figure 1.3 : Image en niveaux de gris

1.3.3 Image couleur (ou RGB)

Chaque pixel possède une couleur décrite par la quantité de rouge (*R*), vert (*G*) et bleu (*B*). Chacune de ces trois composantes est codée sur l'intervalle [0, 255], ce qui donne $255^3 = 16\ 777\ 216$ couleurs possibles. Il faut donc 24 bits pour coder un pixel.[2]

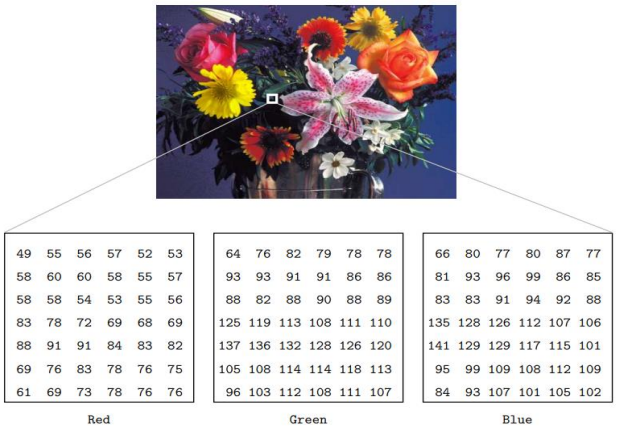


Figure 1. 4 : Image couleur réelle [6]

En utilisant l'image que nous avons pris et que nous allons exploiter par la suite, nous avons décomposé cette image en ses trois composantes : rouge, verte, bleu. Le résultat obtenu est donné sur la figure 1.5 ci-dessous.[2]

Figure 1.5 : Image couleur (Rouge, vert, bleu) [27]

1.4Formatsd'images

Les formats les plus utilisés sont :

1.4.1 BMP (bitmap)

On appelle *bitmap* le tableau contenant les couleurs de chaque pixel d'une image. Un fichier au format BMP contient une image non compressée. Il contient un entête de 54octets (*les paramétrages*) puis les composantes RGB (*Red-Green-Blue*) de chaque pixel.

Exemple : Ainsi un fichier BMP pour une image 800 × 600 possède une taille de :1 440 054 octets.

Pour économiser de la place, la plupart des images sont compressées sous les formats ces dessous.[2]

1.4.2 GIF (Graphic Interchange Format)

Il s'agit d'un format de compression sans perte. Ce format assure une division environ par 5 de la taille du fichier initial.[2]

1.4.3 JPEG (Joint Photo Expert Group)

Ce format permet une compression avec perte.Il diviser par 20 la taille du fichier initial. La compression des images au format JPEG supprime certaines informations qui ne peuvent être récupérées au moment de la décompression. La qualité de l'image peut donc être altérée.[2]

En plus de ces trois formats, on cite:

- FPX (Flash pix), PCD (Photo CD),
- PNG (Portable Network Graphic),
- PSD (PhotoShop Document),
- PSP(Paint Shop Pro),
- TIF (Tagged Image File Format).

1.5 Systèmes de traitement d'images

Le traitement de l'image est un domaine très vaste qui a évolué pendant des décennies, surtout au cours des dernières décennies. Le traitement numérique de l'image nous amène aux techniques et méthodes utilisées pour modifier l'image dans le but de l'améliorer et d'en extraire l'information, comme le montre la figure 1.6.[3]

Un système de traitement numérique d'images est composé de :

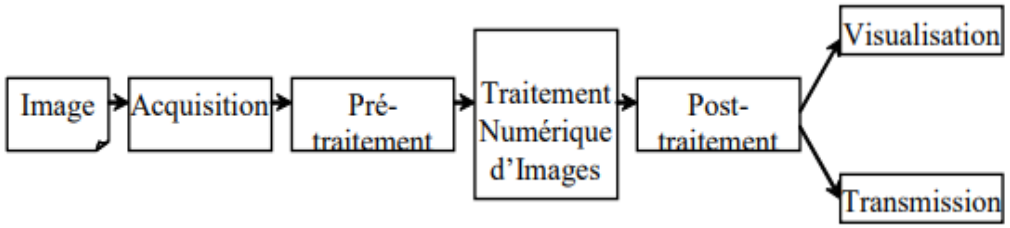


Figure 1. 6 : Schéma d'un système de traitement d'images [3]

1.6 Histogramme d'une imagemonochrome

Considérons une image monochrome dans laquelle *f (i, j)* représente la fonction intensité du pixel de coordonnées *(i, j)*. L'histogramme est la représentation graphique de la fréquence d'apparition ***h(f)*** de chaque niveau *f* dans l'image.[2]

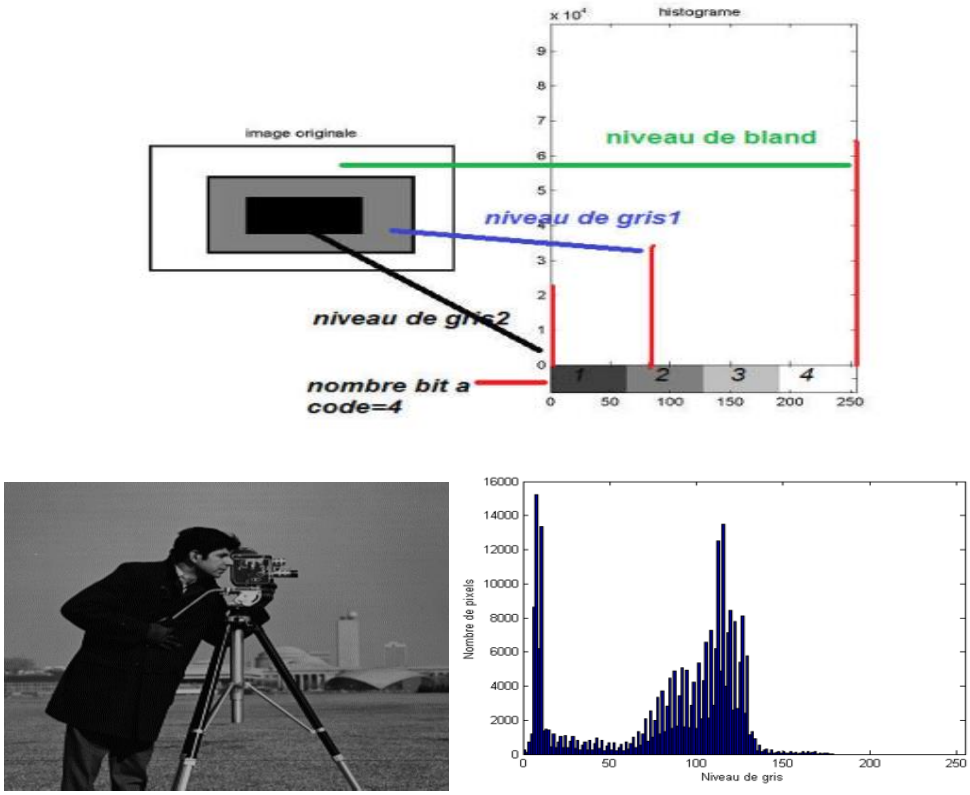


Figure 1.7 : L'histogramme d'une image [29]

L'abscisse d'un histogramme représente les niveaux d'intensité au plus claire à droit

1.7Filtrage des images

Les images brutes permettent rarement de parvenir à une extraction directe des objets à analyser :

- Soit parce que l'éclairement de l'objet n'est pas uniforme.
- Soit parce que l'objet est perçu à travers un bruit assez important : les images contiennent donc un signal et du bruit (dont on veut éliminer la plus grande partie possible).
- Soit encore parce que le contraste n'est pas suffisant.

Avant d'extraire les objets et d'analyser une image, il est donc souvent nécessaire d'améliorer l'image. Il existe un grand nombre de filtres possibles, et à quelques exceptions près, on peut les classer en 2 grandes catégories : les filtres linéaires et les filtres non linéaires. Dans les filtres linéaires, la méthode de filtrage par moyenne est la plus utilisée à cause de la simplicité de son algorithme et la qualité de résultats qu'elle donne par rapport à d'autres filtres. Le cas des filtres non linéaires, c'est le filtrage médian.[4]

1.7.1 Filtrage par la moyenne

Cette méthode vise à atténuer les variations de niveaux de gris entre les pixels adjacents, créant ainsi un effet de lissage sur les images. Elle est utilisée pour réduire le bruit indésirable. Le filtrage par la moyenne consiste à remplacer chaque pixel par la valeur moyenne de ses voisins, y compris lui-même. Cette approche permet d'harmoniser les niveaux de gris trop différents de ceux de leurs voisins, ce qui peut être interprété comme une réduction du bruit ou des niveaux de gris "*anormaux*". La taille du filtre choisie (*3x3, 5x5,* etc.) déterminera l'intensité du lissage souhaité, mais il est important de comprendre quecela entraînera une perte de netteté des contours de l'image d'origine. La figure 1.8illustrent le processus de filtrage par la moyenne et la figure 1.9 illustrent exemple sur cette type de filtrage.[4]

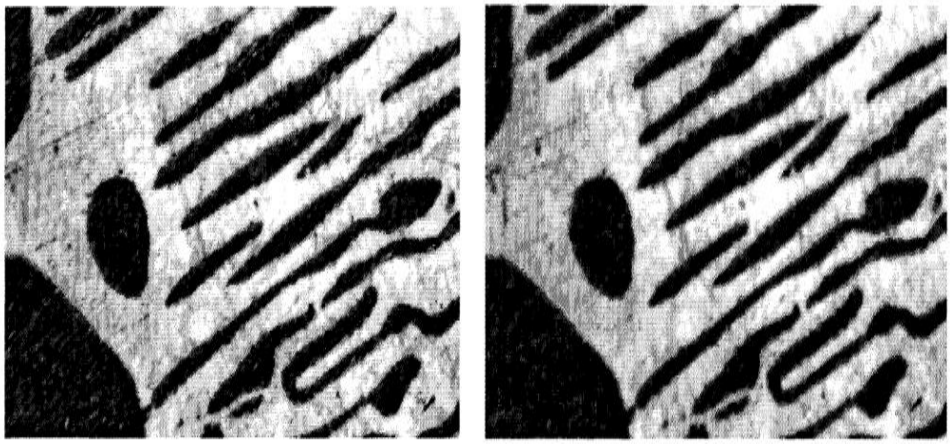
Filtre 3*3

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Filtre 5*5

1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25

Figure 1.8: Filtrage par la moyenne [4]



(a) (b)

Figure 1. 9 : (a) Image avant le filtrage et(b) Image après le filtrage moyenne 3x3 [4]

Les inconvénients évidents de ce filtre de moyenne sont les suivants :

- Un pixel isolé avec un niveau de gris « *anormal* » pour son voisinage va perturber les valeurs moyennes des pixels de son voisinage.
- Sur une frontière de régions, le filtre va estomper le contour et le rendre flou, ce qui est gênant en visualisation bien sûr, mais éventuellement aussi pour un traitement ultérieur qui nécessiterait des frontières nettes.

Il est possible de moduler ces effets néfastes en réalisant en chaque pixel une convolution « *conditionnelle* ».

1.7.2 Filtrage médian

Le filtre médian réalise un lissage de l'image un peu plus performant que le filtre moyen en ce qui concerne les détails dans l'image.

- **Méthode**

Chaque pixel est traité en considérant ses voisins sur un voisinage donné. Le pixel lui-même et ses voisins forment alors un ensemble dont on calcule la « *médiane* ». Le pixel sera alors remplacé par cette valeur médiane. Comme montre la figure xx le filtre médian donne : 97.

Voisinage 3*3

12	14	5	4	2
4	99	23	56	4
5	126	121	120	5
4	97	12	9	4
2	4	5	4	2

Figure 1.10 : Filtrage médian [4]

- **Intérêt du filtre médian :**

- Un pixel non représentatif dans le voisinage affectera peu la valeur médiane.
- La valeur médiane choisie étant le niveau de gris d'un des pixels considérés, on ne crée pas alors de nouveaux niveaux de gris dans l'image.

Ainsi lorsque le filtre passe sur un contour très marqué il le préservera mieux.[4]

1.8Conclusion

Le traitement d'image englobe un ensemble de techniques visant à modifier une image afin de l'améliorer ou d'en extraire des informations. Ces techniques, regroupées sous le terme de prétraitement, sont utilisées pour améliorer la qualité ou les caractéristiques de l'image. Le processus de prétraitement consiste à appliquer différentes opérations sur l'image d'origine afin d'obtenir une version modifiée et améliorée.

Il existe une variété de techniques de prétraitement disponibles et utilisées en fonction des objectifs spécifiques. Ces techniques de base sont essentielles pour comprendre et extraire des informations pertinentes de l'image. En effet, toute analyse ultérieure de l'image nécessiterait l'utilisation de ces techniques de traitement d'image pour préparer les données avant de les soumettre à des algorithmes ou des méthodes d'analyse plus avancés.

Chapitre 2 : LeDeep Learning

Résumé

Dans cette partie, nous serons guidés par des discussions et des recherches sur un sujet très important et intéressant, qui est le sujet de l'apprentissage profond ou le ***D**eep**L**earning*(*DL*). On commence de parler sur le réseau de neurones et puis sur le DL. Dans ce chapitre, on discute sur le fonctionnement de DL et les algorithmes d'optimisation comme les optimisateurs ADAM et RSMprop. Et enfin, on présente deux modèles de DL comme le CNN et RNN.

2.1 Introduction

L'apprentissage profond ou le ***D**eep**L**earning* est un domaine de l'apprentissage automatique qui exploite le réseau de neurones synthétiques pour résoudre des problèmes complexes. Ce réseau est composé de couches de nœuds interconnectés qui apprennent des modèles et des relations dans les données en se basant sur de vastes ensembles de données. L'apprentissage profond a connu des avancées significatives dans divers domaines tels que la vision par ordinateur, le traitement du langage naturel, la reconnaissance vocale et la robotique. Parmi les applications les plus populaires de DL, citons la reconnaissance d'images, la reconnaissance d'objets et la traduction de langues.

Cependant, le DL requiert une quantité importante de données et de ressources informatiques pour former les modèles. Néanmoins, les progrès technologiques tant matériels que logiciels ont rendu cette approche plus accessible et largement utilisée dans des secteurs variés tels que la santé, la finance et le marketing. Dans l'ensemble, l'apprentissage profond s'est révélé prometteur pour résoudre des problèmes complexes et enrichir notre compréhension de l'intelligence artificielle.

En résumé, le DL englobe diverses techniques et modèles. Par exemple, leCNNest couramment utilisé pour l'analyse d'images, tandis que le réseau de neurones récurrents (*Recurrent Neural Networks - RNN*), notamment le LSTM (*Long Short-Term Memory - LSTM*), est essentiel dans les domaines qui nécessitent le stockage, la déduction et la réutilisation d'informations antérieures, tel que les prévisions météorologiques. Dans les pages suivantes, nous aborderons ces sujets plus en détail dans le cadre de notre étude.

2.2Réseau de neurones

Le réseau de neurones artificiels (*Artificial Neural Network, ANN*), plus communément appelés réseaux de neurones (*Neural Network, NN*), est des systèmes informatiques inspirés de réseau de neurones biologiques qui constituent le cerveau humain. Nous trouvons un exemple dans la figure2.1.[5]

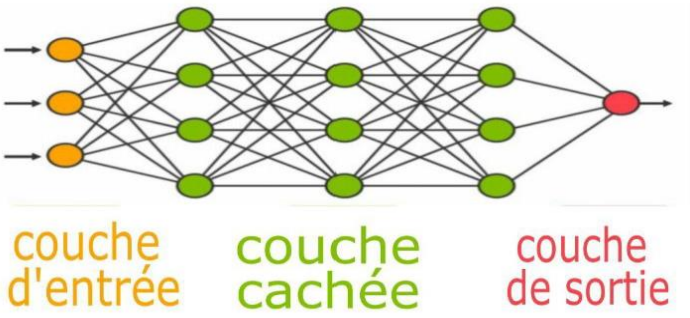


Figure 2. 1 : Réseau de neurones [5]

2.2.1 Un bref historique :

Bien que le réseau de neurones puisse sembler être un sujet nouveau et intéressant, il est en fait une longue histoire. Dès 1958, un psychologue américain du nom de *Frank Rosenblatt* a imaginé une machine capable d'imiter l'esprit humain, qu'il a baptisée "Perceptron". Le réseau de neurones fonctionne en apprenant à partir d'exemples, d'une manière similaire au fonctionnement des cerveaux biologiques. Il reçoit et traite des données externes de la même manière que le cerveau humain.[5]

2.2.2 La structure en couches des réseaux neuronaux :

Nous savons que différentes sections du cerveau humain sont câblées pour traiter différents types d'informations. Ces parties du cerveau sont organisées hiérarchiquement en niveaux. Au fur et à mesure que l'information pénètre dans le cerveau, chaque couche ou niveau de neurones fait son travail particulier de traitement de l'information entrante, en tire des conclusions et les transmet à la couche suivante, plus élevée**.**

C'est ainsi que le cerveau fonctionne par étapes. Le réseau de neurones artificiels fonctionne de manière similaire. Le réseau de neurones tente de simuler cette approche multicouche du traitement de diverses entrées d'informations et de la prise de décisions en fonction de celles-ci.

Au niveau cellulaire ou au niveau des neurones individuels, les fonctions sont réglées avec précision. Les neurones sont les cellules nerveuses du cerveau. Les cellules nerveuses possèdent de fines extensions appelées dendrites. Elles reçoivent des signaux et les transmettent ensuite au corps cellulaire. Le corps cellulaire traite les stimuli et prend la décision de déclencher des signaux vers d'autres neurones du réseau. Si la cellule décide de le

faire, le prolongement du corps cellulaire, appelé axone, conduira le signal à d'autres cellules par transmission chimique. Le fonctionnement des réseaux neuronaux s'inspire de la fonction des neurones de notre cerveau, bien que le mécanisme d'action technologique soit différent du mécanisme biologique.[5]

2.2.3 Le fonctionnement d'un réseau de neurones :

Dans sa forme la plus élémentaire, un réseau de neurones artificiels comporte trois couches de neurones. Les informations circulent de l'une à l'autre, comme dans le cerveau humain :

- La couche d'entrée : le point d'entrée des données dans le système
- La couche cachée : où l'information est traitée
- La couche de sortie : où le système décide de la marche à suivre en fonction des données.

Le réseau de neurones artificiels plus complexes comportera plusieurs couches, dont certaines seront cachées.

Le réseau de neurones fonctionne via une collection de nœuds ou d'unités connectées, tout comme les neurones artificiels. Ces nœuds modèlent grossièrement le réseau de neurones du cerveau animal. Tout comme son homologue biologique, un neurone artificiel reçoit un signal sous la forme d'un stimulus, le traite et envoie un signal aux autres neurones qui lui sont connectés. Mais les similitudes s'arrêtent là.[5]

2.2.4 Avantages d'un réseau de neurones :

Dans le cas de problèmes comportant des relations dynamiques ou non linéaires, le réseau de neurones est la capacité de se former de manière efficace, notamment lorsque les modèles de données internes sont solides. Cette capacité est d'autant plus renforcée selon le contexte d'application.

Le réseau de neurones offre une approche analytique alternative aux techniques standard qui peuvent être limitées par des hypothèses strictes de linéarité, de normalité et d'indépendance des variables. Grâce à leur capacité à explorer diverses relations, les réseaux neuronaux permettent aux utilisateurs de modéliser rapidement des phénomènes qui auraient été autrement difficiles, voire impossibles, à appréhender.[5]

2.2.5 Les types de réseau de neurones :

Les réseaux de neurones artificiels peuvent être classés en fonction de la façon dont les données circulent des nœuds d'entrée aux nœuds de sortie. Voici quelques exemples :[6]

2.2.5.1 Réseau de neurones à action directe :

Le réseau de neurones *feedforward* fonctionne en traitant les données de manière unidirectionnelle, du début à la fin. Chaque nœud d'une couche est connecté à tous les nœuds de la couche suivante. Ce réseau utilise également un mécanisme de rétroaction pour améliorer progressivement les prédictions au fil du temps.[6]

2.2.5.2 Algorithme de rétropropagation :

Les réseaux de neurones artificiels apprennent en permanence à améliorer leur analyse prédictive en utilisant des boucles de rétroaction correctives. En termes simples, vous pouvez imaginer que les données circulent des nœuds d'entrée vers les nœuds de sortie via plusieurs chemins différents dans un réseau de neurones. Un chemin unique est le chemin correct reliant le nœud d'entrée au nœud de sortie correct. Pour trouver ce chemin, le réseau de neurones utilise une boucle de rétroaction, qui fonctionne comme suit :

a. Chaque nœud fait une supposition sur le prochain nœud du chemin.

b. Il vérifie si la supposition était correcte. Les nœuds attribuent des valeurs de poids plus élevées aux chemins qui mènent à un plus grand nombre de suppositions correctes et des valeurs de poids plus faibles aux chemins de nœuds qui mènent à des suppositions incorrectes.

c. Pour le point de données suivant, les nœuds effectuent une nouvelle prédiction en utilisant les chemins de poids plus élevé, puis répètent l'étape 1.[6]

2.3 Le Deep Learning

2.3.1 Définition :

Le Deep Learning, également connu sous le nom d'apprentissage profond, est une branche de l'intelligence artificielle (*IA*) qui repose sur des techniques de Machine Learning (*ML*), permettant aux machines d'apprendre par elles-mêmes plutôt que de suivre des instructions préprogrammées. Cette approche repose sur des modèles mathématiques pour la modélisation des données. L'histoire de l'intelligence artificielle remonte à 1950, lorsque *Alan Turing* s'est penché sur la notion de machines pensantes, ce qui a conduit au développement de la machine Learning, une approche où les machines peuvent communiquer et agir en fonction des informations stockées. L'apprentissage profond est une avancée dans ce domaine, s'inspirant du fonctionnement du cerveau humain. Il repose sur de vastes réseaux de neurones artificiels interconnectés, capables de traiter et de mémoriser des informations, de comparer des problèmes ou des situations à des expériences passées similaires, d'analyser les solutions et de résoudre les problèmes de la manière la plus optimale.[7]

2.3.2 Historique :

Le *DL* a ses origines dans les années 1940, lorsque la recherche initiale sur les modèles mathématiques des neurones a commencé. Cependant, il a fallu attendre les années 1980 pour que la recherche sur l'apprentissage profond prenne son envol, avec l'émergence de nouveaux concepts et le développement des premiers réseaux de neurones artificiels multicouches. Un chercheur clé dans ce domaine est *Yann Le Cun*, considéré comme l'un des pionniers de (*DL*). Cependant, les réseaux de neurones étaient limités à l'époque en raison de contraintes de puissance de calcul et de disponibilité de données. Ce n'est qu'avec l'avènement du « big data » dans les années 2000 que le DeepLearnig a connu une véritable révolution.

En particulier, en 2012, lors du concours ImageNet de reconnaissance et de classification d'images organisé par l'Université de Stanford, une équipe a remporté le défi en utilisant le *DL*. Cela a attiré l'attention sur les performances impressionnantes de cette approche. Depuis lors, le *DL* connu une progression significative, permettant aux ordinateurs d'apprendre, de prédire et de réagir de manière autonome à partir de vastes ensembles de données.[8]

2.3.3Applications du Deep Learning :

Le *DL* est utilisé dans de nombreux domaines :

- Reconnaissance d'image,
- Traduction automatique,
- Voiture autonome,
- Compression des données
- Diagnostic médical
- Recommandations personnalisées,
- Modération automatique des réseaux sociaux,
- Prédiction financière et trading automatisé,
- Identification de pièces défectueuses,
- Détection de malwares ou de fraudes,
- *Chabots* (agents conversationnels),
- Exploration spatiale,
- Robots intelligents.

C'est aussi grâce au DL que l'IA de Google « Alpha Go » a réussi à battre les meilleurs champions de « Go » en 2016. Le moteur de recherche du géant américain est lui-même de plus en plus basé sur l'apprentissage par *Deep Learning* plutôt que sur des règles écrites.[8]

2.3.4 L'objectif du Deep Learning :

Le *DL* trouve de nombreuses applications dans le domaine des technologies de l'information et de la communication. Il est utilisé dans les smart phones pour la reconnaissance faciale et vocale, ainsi que dans la robotique pour permettre aux équipements intelligents de réagir de manière appropriée dans différentes situations. Par exemple, un réfrigérateur intelligent peut détecter une porte restée ouverte ou une température anormale et émettre une alarme. Lorsque vous vous demandez comment Facebook reconnaît vos amis sur les photos, la réponse réside dans le (*DL*). Les chercheurs utilisent également cette approche pour leurs travaux, y compris dans le domaine de l'étude et de la manipulation de l'ADN. Les applications du (*DL*) s'étendent également à la traduction automatique, aux véhicules autonomes, à la médecine pour le diagnostic basé sur l'imagerie médicale telle que les radios, IRM et scanners, à la recherche de particules en physique, ainsi qu'à la reproduction d'œuvres artistiques.[8]

2.3.5 Fonctionnement du Deep Learning :

Le *DL* repose sur un réseau de neurones artificiels qui imite le fonctionnement du cerveau humain. Ce réseau est constitué de multiples couches de neurones, pouvant aller de quelques dizaines à plusieurs centaines, où chaque couche reçoit et interprète les informations provenant de la couche précédente. Par exemple, le système apprendra d'abord à reconnaître les lettres, puis à analyser les mots dans un texte, ou à déterminer la présence d'un visage sur une photo avant d'identifier la personne en question. Comme la figure2.2.

À chaque étape, les « mauvaises » réponses sont éliminées et renvoyées vers les niveaux en amont pour ajuster le modèle mathématique. Au fur et à mesure, le programme réorganise les informations en blocs plus complexes. Lorsque ce modèle est par la suite appliqué à d'autres cas, il est normalement capable de reconnaître un chat sans que personne ne lui ait jamais indiqué qu'il n'a jamais appris le concept de chat. Les données de départ sont essentielles : plus le système accumule d'expériences différentes, plus il sera performant.[8]

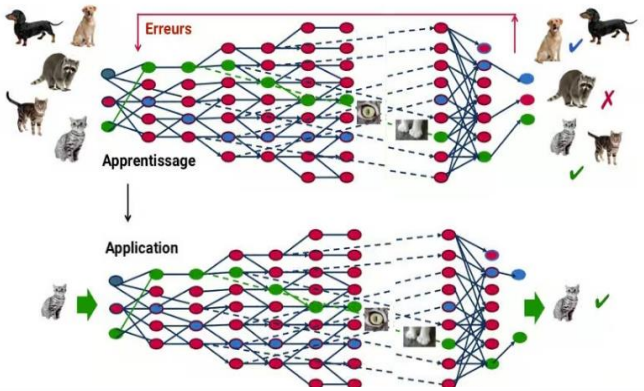


Figure 2. 2 : Exemple de fonctionnement de DL [8]

2.4Algorithmes d'optimisation

2.4.1 Optimiseur RMSprop :

RMSprop, une méthode d'optimisation basée sur le gradient, est largement utilisée lors de l'entraînement de réseaux neuronaux. Elle a été introduite par *Geoffrey Hinton*, considéré comme le pionnier de la rétropropagation. Lorsqu'il s'agit de fonctions complexes comme les réseaux neuronaux, les gradients ont tendance à diminuer ou à exploser lors de la propagation des données à travers la fonction (problème des gradients disparus). RMSprop résout ce problème en utilisant une moyenne mobile des gradients au carré pour normaliser le gradient. Cette normalisation équilibre la taille du pas (Momentum) en diminuant le pas pour les grands gradients afin d'éviter les explosions, et en augmentant le pas pour les petits gradients afin d'éviter leur disparition. En d'autres termes, RMSprop adapte le taux d'apprentissage au lieu de le traiter comme un hyperparamètre fixe, ce qui signifie que le taux d'apprentissage change avec le temps.[9]

$$\upsilon_{dw} = \boldsymbol{\beta} \cdot \upsilon_{dw} + (\mathbf{1} - \boldsymbol{\beta}) \cdot \boldsymbol{dw^2} \quad (1)$$

$$\upsilon_{db} = \boldsymbol{\beta} \cdot \upsilon_{dw} + (\mathbf{1} - \boldsymbol{\beta}) \cdot \boldsymbol{db^2} \quad (2)$$

$$\boldsymbol{w} = \boldsymbol{w} - \propto \cdot \frac{\boldsymbol{dw}}{\sqrt{\upsilon_{dw}} + \boldsymbol{\epsilon}} \quad (3)$$

$$\boldsymbol{b} = \boldsymbol{b} - \propto \cdot \frac{\boldsymbol{db}}{\sqrt{\upsilon_{db}} + \boldsymbol{\epsilon}} \quad (4)$$

2.4.2 Optimiseur ADAM :

Adam est un algorithme avancé d'optimisation utilisé dans l'apprentissage automatique pour traiter les fonctions objectives stochastiques à l'aide des gradients de premier ordre. Il se distingue par ses estimations adaptatives des moments d'ordre inférieur. Adam est largement adopté par de nombreux professionnels de l'apprentissage automatique. La mise à jour des paramètres est effectuée en utilisant le rapport entre le premier moment normalisé et le deuxième moment, ce qui permet d'obtenir une direction de mise à jour pertinente.[9]

2.4.3 Le meilleur algorithme d'optimisation pour le (DL) :

Enfin, nous pouvons discuter de la question de savoir quel algorithme est le meilleur pour entraîner un réseau de neurones. En général, un algorithme de descente de gradient normal est plus que suffisant pour des tâches plus simples. Si vous n'êtes pas satisfait de la précision de votre modèle, vous pouvez essayer RMSprop ou ajouter un terme d'impulsion à vos algorithmes de descente de gradient.

Cependant, d'après mon expérience, ADAM est le meilleur algorithme d'optimisation de réseau neuronal disponible aujourd'hui. Cet algorithme d'optimisation est excellent pour presque tous les problèmes d'apprentissage en profondeur que vous rencontrerez dans la pratique. Surtout si vous définissez les hyperparamètres d'ADAM sur les valeurs suivantes.[10]

2.5 Modèles du Deep Learning

On va présenter dans cette section les modèles de DeepLearning utilisés dans notre mémoire (à savoir les CNN et les LSTM).

2.5.1 Réseau de neurones à convolution (CNN) :

Le terme "réseau neuronal convolutif" indique l'utilisation de l'opération mathématique de convolution dans ce type spécifique de réseau neuronal. Les CNN sont des réseaux neuronaux spécialement conçus qui utilisent des convolutions à la place des multiplications matricielles générales dans au moins une de leurs couches. Ces algorithmes d'apprentissage sont extrêmement efficaces pour effectuer des convolutions, ce qui permet d'extraire des caractéristiques pertinentes à partir de données localement corrélées. Les sorties des convolutions sont ensuite soumises à une unité de traitement non linéaire, également appelée fonction d'activation. Cela favorise l'apprentissage de l'abstraction et introduit la non-linéarité dans l'espace des caractéristiques. Cette non-linéarité permet d'obtenir différentes activations

pour différentes réponses, facilitant ainsi l'apprentissage des différences sémantiques dans les images. La topologie des CNN est composée de plusieurs étapes d'apprentissage qui combinent des couches convolutives, des unités de traitement non linéaires et des couches de sous-échantillonnage. La Figure 2.3 présente la structure générale d'un réseau CNN.[11]

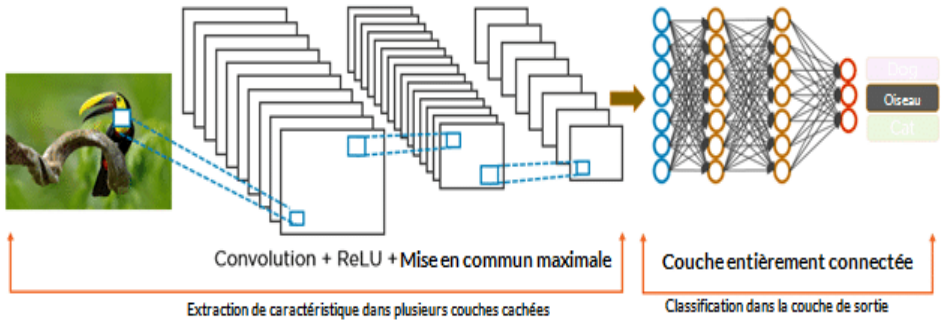


Figure 2. 3 : Réseau de neurones avec de nombreuses couches convolutives [33]

- **Couche de convolution**: La couche de convolution est responsable de l'extraction des caractéristiques d'une image d'entrée. Elle maintient la relation entre les pixels en apprenant les motifs de l'image à partir de petites régions de données d'entrée appelées filtres ou noyaux. La convolution est une opération mathématique qui combine la matrice de l'image avec le filtre. La Figure 2.4 illustre une étape simple de convolution où un filtre est appliqué à l'image.[12]

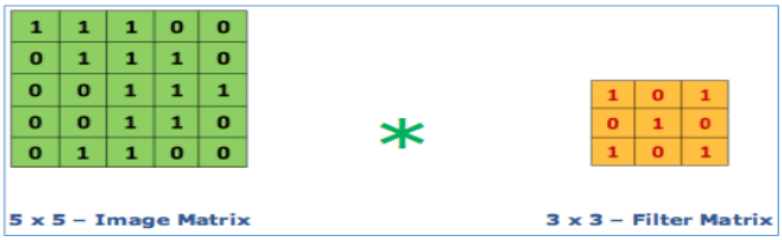
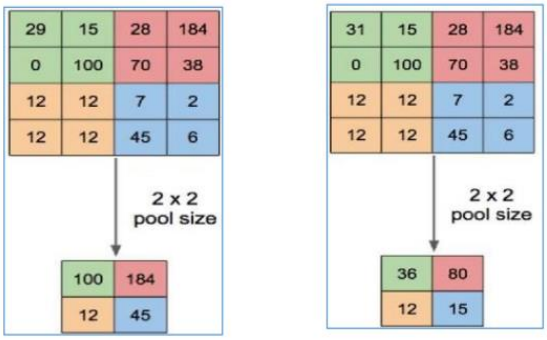


Figure 2. 4 : Exemple explicative sur l'opération de convolution [14]

- **Couche de pooling**: La couche de pooling est généralement située entre deux couches de convolution. Le pooling est un processus de réduction basé sur l'échantillonnage qui vise à simplifier une représentation (comme une image ou une matrice de sortie de couche cachée) en réduisant sa dimensionnalité tout en préservant les caractéristiques importantes des sous-régions groupées. Il existe différents types de pooling, Figure 2.5, notamment le pooling moyen qui calcule la moyenne des pixels dans la région sélectionnée et le pooling maximal qui sélectionne le pixel ayant la valeur maximale parmi tous les pixels de la région sélectionnée.[12]



(a) **(b)**

Figure2. 5 : (a) pooling moyne, (b) pooling maximal [14]

- **Couche fully-connected:**La couche entièrement connectée dans un réseau convolutif fonctionne de manière similaire aux réseaux entièrement connectés des modèles conventionnels. Les sorties de la phase précédente (qui comprend la convolution et le Pooling répétitifs) sont introduites dans la couche entièrement connectée, où le produit scalaire entre le vecteur de poids et le vecteur d'entrée est calculé pour obtenir la sortie finale. Comme la montre dans la figure2.6 suivante.[12]

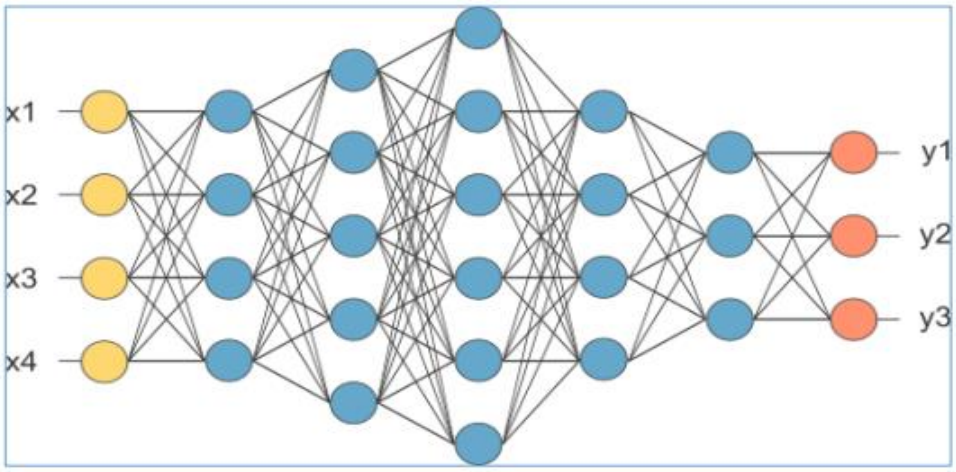


Figure 2. 6 : Couche fullyconnected [14]

2.5.2 Réseaux neuronaux récurrents (RNN) :

Les réseaux de neurones récurrents (RNN) sont une variante essentielle des réseaux de neurones, largement utilisée dans le traitement du langage naturel. Ils sont appelés "récurrents", car ils effectuent la même opération sur chaque élément d'une séquence, et la sortie dépend des calculs précédents. On peut également les considérer comme ayant une "mémoire" qui capture les informations des étapes précédentes. En théorie, les RNN peuvent utiliser des informations provenant de séquences de longueur arbitraire, mais en pratique, ils sont généralement limités à un nombre restreint d'étapes précédentes. Les RNN sont une catégorie de réseaux de neurones qui permettent d'utiliser les prédictions antérieures comme entrée en utilisant des états cachés. Leur forme est représentée sur la figure2.7.[12]

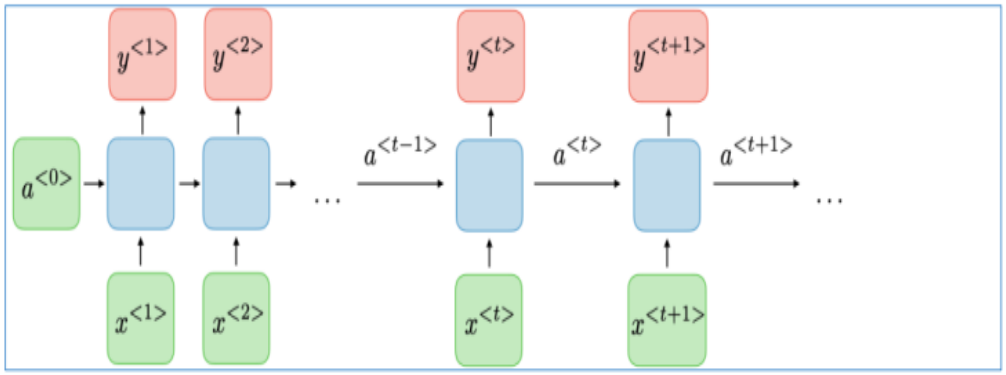


Figure 2. 7 : Architecture de RNN [14]

2.5.2.1 Historique

Le concept de RNN a été introduit en 1986, et l'architecture célèbre LSTM a été inventée en 1997. Contrairement aux CNN, le nombre d'architectures de RNN ayant acquis une renommée est relativement limité. Comme dit le dicton, "une image vaut mille mots", les images contiennent une quantité d'informations et de richesse plus importante. Il n'est donc pas étonnant que l'histoire évolutive des RNN soit moins complexe. Je vais diviser cette histoire en trois phases : la mémoire robuste à un seul port du RNN classique, la mémoire à multiples portes du LSTM, et l'attention dans l'architecture encodeur-décodeur du RNN. En les présentant, vous réaliserez que le dénominateur commun de l'évolution des RNN est la lutte contre l'amnésie. En d'autres termes, chaque génération de RNN tente de se souvenir d'un maximum d'informations importantes afin de prédire l'avenir avec plus de précision.[12]

2.5.2.2Réseaux Long Short-Term Memory (LSTM)

Ces réseaux sont une variante spéciale de RNN qui peut apprendre des dépendances à long terme. Ils ont été introduits par et ont été améliorés et popularisés par plusieurs chercheurs dans leurs travaux ultérieurs. Les LSTM excellent dans une large gamme de problèmes et sont maintenant largement utilisés. Les LSTM sont spécifiquement conçus pour résoudre le problème des dépendances à long terme. Ils sont naturellement capables de conserver des informations sur de longues périodes, ce qui en fait leur comportement par défaut, plutôt qu'une difficulté d'apprentissage ! Tous les RNN ont une structure sous forme de chaîne de

modules récurrents de réseau neuronal. Dans les RNN standards, ce module récurrent a une structure très simple, généralement une seule couche de tanh. Voir la figure2.8.[13]

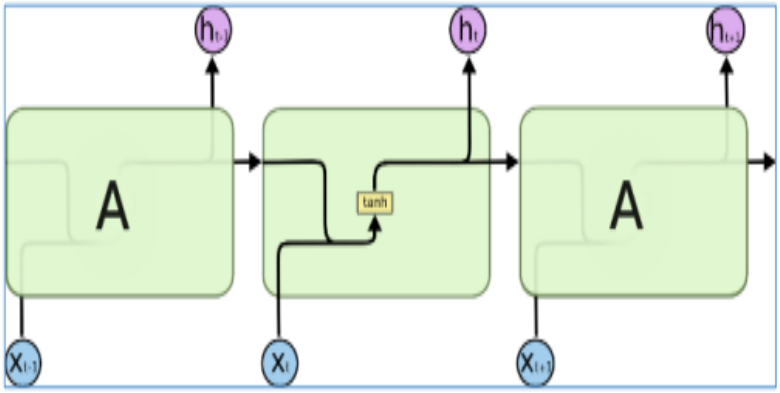


Figure 2. 8 : Architecture de RNN [14]

Les LSTM ont également une structure en chaîne, mais le module répétitif à une structure différente. Au lieu d'avoir une seule couche de réseau neuronal, il y en a quatre, interagissant d'une manière très spéciale montrée dans la figure2.9.

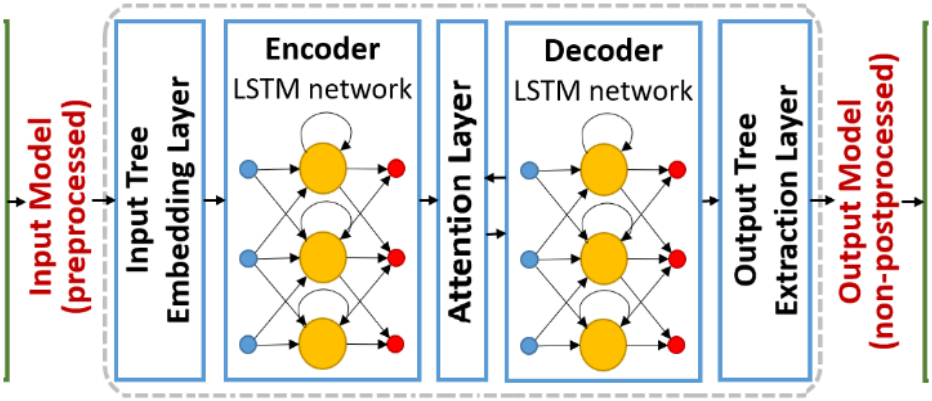


Figure 2. 9 : Le module répétitif dans un LSTM contient quatre couches [34]

- Un réseau LSTM classique est constitué de blocs de mémoire appelés cellules. Ces cellules transmettent deux états à la cellule suivante : l'état de la cellule et l'état caché. L'état cellulaire est la voie principale du flux de données, permettant aux données de circuler pratiquement inchangées vers l'avant. Cependant, des transformations linéaires peuvent se produire. Les données peuvent être ajoutées ou supprimées de l'état cellulaire grâce à l'utilisation de portes sigmoïdes. Une porte est similaire à une couche ou à une séquence d'opérations matricielles, avec des poids individuels différents. Les LSTM sont conçus pour

résoudre le problème des dépendances à long terme en utilisant des portes pour contrôler le processus de mémorisation.

- La première étape de la construction d'un réseau LSTM implique l'identification des informations non nécessaires qui doivent être exclues de la cellule à cette étape. Cette identification et exclusion des données sont déterminées par la fonction sigmoïde, qui utilise la sortie de l'unité LSTM précédente (h_{t-1}) au temps (t-1) et l'entrée actuelle X_t au temps t. La fonction sigmoïde détermine également quelle partie de la sortie précédente doit être oubliée. Cette opération est effectuée par une porte appelée "la porte de l'oubli" (ou f_t), où f_t est un vecteur avec des valeurs comprises entre 0 et 1., correspondant à chaque nombre dans l'état de cellule, $Ct - 1$.

$$\boldsymbol{f_t} = \boldsymbol{\sigma}\,(\boldsymbol{W_f}\,[\boldsymbol{h_t}-1,] + \boldsymbol{b_f}) \qquad (5)$$

Ici, σ est la fonction sigmoïde, et Wf et bf sont les matrices de poids et le biais, respectivement, la porte du oubliez.

- Ensuite, il y a l'étape de décision et de stockage des informations de la nouvelle entrée X_{t-1} dans l'état de la cellule, ainsi que la mise à jour de l'état de la cellule. Cette étape se compose de deux parties : la couche sigmoïde et la couche tanh. Tout d'abord, la couche sigmoïde détermine si les nouvelles informations doivent être mises à jour ou ignorées (0 ou 1), puis la fonction tanh attribue des poids aux valeurs qui sont transmises, en déterminant leur niveau d'importance (-1 à 1). Les deux valeurs sont ensuite multipliées pour mettre à jour le nouvel état de la cellule. Cette nouvelle mémoire est ensuite ajoutée à l'ancienne mémoire $Ct - 1$ résultant en Ct.[14]

$$\boldsymbol{i_t} = \boldsymbol{\sigma}(\boldsymbol{W_i}[\mathbf{h_t - 1}, \mathbf{X_t}] + \boldsymbol{b_i}) \qquad (6)$$

$$\boldsymbol{N_t} = \boldsymbol{tanh}(\boldsymbol{W_n}[\boldsymbol{h_t} - \mathbf{1}\,, \boldsymbol{Xt_t}] + \boldsymbol{b_n}) \qquad (7)$$

$$\boldsymbol{C_t} = \mathbf{C_{t-1}}\,\boldsymbol{f_t} + \boldsymbol{N_t i_t} \qquad (8)$$

Ici, $Ct-1$ et Ct sont les états de cellule au temps $t - 1$ et t, tandis que W et b sont les matrices de poids et le biais, respectivement, de l'état de la cellule.

- Dans la dernière étape, les valeurs de sortie h_t est dérivée de l'état de la cellule de sortie O_t, mais elles sont filtrées. Tout d'abord, une couche sigmoïde détermine quelles parties de l'état de la cellule contribuent à la sortie. Ensuite, la sortie de la porte sigmoïde O_t est multipliée par les nouvelles valeurs générées par la couche tanh à partir de l'état de la cellule, avec des valeurs comprises entre -1 et 1.[14]

$$\boldsymbol{O_t} = \boldsymbol{\sigma}(\boldsymbol{W_o}[\mathbf{h_t - 1}, \mathbf{X_t}] + \boldsymbol{b_o}), \qquad (9)$$

$$\boldsymbol{h_t} = \boldsymbol{O_t tanh}(\mathbf{C_t}) \qquad (10)$$

Ici, Wo et bo sont les matrices de poids et le biais, respectivement, de la porte de sortie.

2.6 Conclusion

En conclusion, le Deep Learning et les réseaux de neurones ont révolutionné le domaine de l'intelligence artificielle en permettant aux machines d'apprendre à partir des données de manière autonome. Ces approches ont éliminé la dépendance à l'égard d'une ingénierie manuelle de caractéristiques, en permettant aux algorithmes d'extraire automatiquement des représentations pertinentes des données brutes.

Grâce à l'augmentation de la profondeur des réseaux de neurones, le Deep Learning a permis d'obtenir des performances impressionnantes dans des domaines tels que la vision par ordinateur, le traitement du langage naturel, la reconnaissance vocale et bien d'autres. Au début, nous avons rapidement passé en revue certaines définitions et clarifications des réseaux neuronaux, puis nous avons abordé l'apprentissage profond avec un peu de détails, pour montrer quelle structure nous avions choisie pour notre application.

Chapitre 03 : Implémentation et Résultats

Résumé

Nous reprenons les pages suivantes de notre travail, où nous considérons la dernière partie de notre étude sur le sujet " Générateur de légendes d'image utilisant CNN et LSTM ". Dans ce chapitre, nous parlons du système sur lequel nous avons travaillé, comment obtenir les résultats souhaités, et les outils que nous avons utilisés pour y parvenir à partir d'un environnement de travail et de diverses bibliothèques de logiciels pertinentes.

3.1 Introduction

Beaucoup de gens utilisent des images dans leur vie quotidienne. En particulier le domaine scientifique. Certains d'entre eux ont du mal à reconnaître ce que portent les images, ou ils peuvent sembler imprécis et avoir du mal à les comprendre. Parlons d'abord du Resnet152, qui est basé sur la structure de notre système. Dans un contexte connexe, nous consacrerons une partie de ce chapitre à nous familiariser avec l'environnement de travail que nous avons utilisé, c'est-à-dire « *Google Colab* », ainsi qu'avec nos bibliothèques de langage de programmation, qui étaient fondées sur *Pytorch*. Puis nous avons dédié une partie qui explique les étapes de travail sur le système pour obtenir la légende des photos et puis il y a une partie dédiée aux résultats

3.2Architecture générale du système

Le but de notre travail est d'extraire ou de donner des légendes (titres) d'images via CNN et LSTM, car notre travail se concentre brièvement sur trois points principaux, nous mettons d'abord l'image dans le système, puis la transmettons à CNN pour une chose importante, qui est d'extraire ses caractéristiques, et après la fin de cette étape, nous passons à LSTM, Ce qui nous permet d'utiliser la mémoire pour prédire et reconnaître le légende de l'image que nous avons et finalement nous obtenons ce légende et la figure 3.1 montres tout ce que nous avons mentionné.

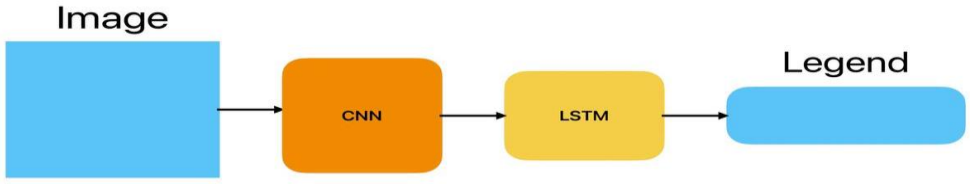


Figure 3. 1 : Architecture générale du système

3.3 ResNet-152

ResNet-152 est un réseau de neurones convolutifs profond qui fait partie de la famille des architectures ResNet (*Residual Network*). Il a été introduit par les chercheurs Kaiming He, Xiangyu Zhang, ShaoqingRen et Jian Sun en 2015 dans leur article intitulé "DeepResidual Learning for Image Recognition".[15]

Le principal objectif de ResNet-152 est de résoudre le problème de la dégradation de la performance des réseaux de neurones profonds à mesure que leur profondeur augmente. Ils ont observé que l'ajout de couches supplémentaires à un réseau de neurones pouvait entraîner une dégradation de la performance, ce qui était contre-intuitif. Pour résoudre ce problème, ils ont introduit des "connexions résiduelles" qui permettent de "sauter" certaines couches et de transmettre les informations directement à des couches plus profondes du réseau.[15]

ResNet-152 est l'une des variantes les plus profondes de l'architecture ResNet. Il comporte 152 couches, y compris des couches de convolution, de normalisation, de regroupement et de classification. Les connexions résiduelles sont utilisées pour faciliter le flux d'informations à travers les différentes couches du réseau, ce qui permet de préserver et de propager plus efficacement les gradients pendant l'apprentissage.[16]

Grâce à sa profondeur accrue, ResNet-152 est capable d'apprendre des représentations plus complexes et de capturer des détails fins dans les images. Il a été pré-entraîné sur de vastesensembles de données, tels que **Image-Net**, et a atteint des performances de pointe dans des tâches de vision par ordinateur, telles que la classification d'images.[16]

ResNet-152 a été largement adopté et -*/utilisé comme une base pour de nombreuses tâches de vision par ordinateur, y compris la détection d'objets, la segmentation sémantique, la reconnaissance faciale, etc. Son architecture profonde et son efficacité en termes de performances en ont fait un choix populaire parmi les chercheurs et les praticiens de l'apprentissage automatique.[16]

3.3.1 Architecture de Resnet152 :

Voici quelques caractéristiques clés de l'architecture ResNet-152 :

1) Blocs résiduels :

L'architecture ResNet-152 repose sur l'utilisation de blocs résiduels, également connus sous le nom de "skip connexions".Ces blocs résiduels permettent de créer des connexions directes entre les couches d'un bloc et les couches d'un bloc ultérieur, sautant ainsi plusieurs couches.Les connexions résiduelles aident à atténuer le problème de disparition du gradient et permettent un entraînement plus profond et plus stable.

2) Profondeur :

ResNet-152 est une version extrêmement profonde de l'architecture ResNet.Elle comporte 152 couches, ce qui signifie qu'elle a une capacité à capturer des caractéristiques complexes et à apprendre des représentations hiérarchiques à partir des données.

3) Couches de convolution :

ResNet-152 utilise des couches de convolution pour extraire les caractéristiques de l'image en entrée.Il utilise des convolutions avec des tailles de noyau différentes et des paramètres d'espacement pour capturer des motifs à différentes échelles.

4) Couches de regroupement (Pooling) :

Des couches de regroupement, généralement de type "Max Pooling", sont utilisées pour réduire progressivement la résolution spatiale des caractéristiques extraites.Cela permet de réduire la quantité de calcul nécessaire et d'introduire une certaine invariance aux déformations spatiales mineures.

5) Couches entièrement connectées :

Après les couches de convolution et de regroupement, ResNet-152 utilise des couches entièrement connectées pour effectuer la classification ou d'autres tâches spécifiques.Ces couches permettent de transformer les caractéristiques extraites en une sortie finale, généralement un vecteur de probabilités représentant les différentes classes d'objets possibles.[17]

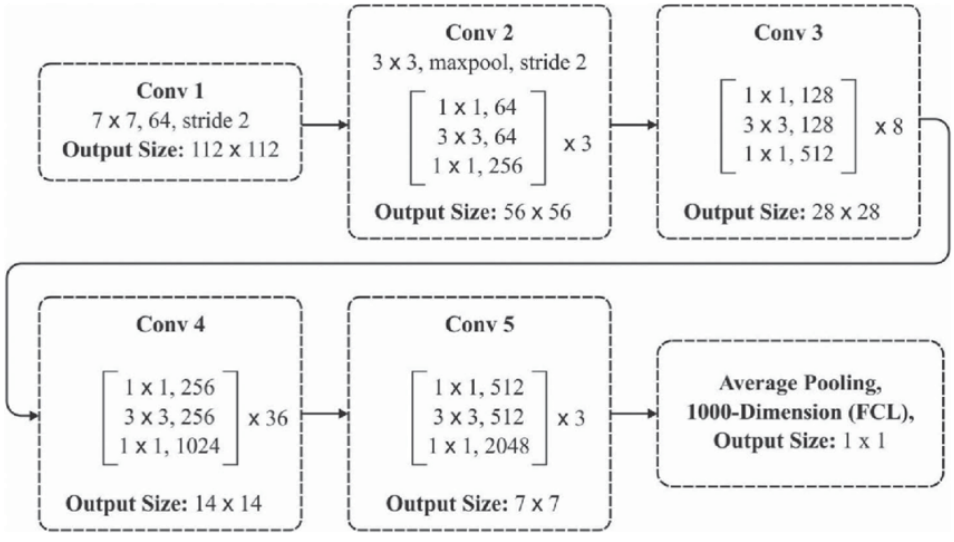


Figure 3. 2 : Architecture de Resnet152 [35]

3.4 Implémentation

Cette partie est réservée aux détails de l'environnement de développement et le langage de programmation utilisés pour la réalisation de notre système. Nous avons présenté aussi la base d'apprentissage et de test utilisés, les détails de l'implémentation de l'architecture proposée et l'interface graphique de l'application.

3.4.1 Environnement de développement

Google Colab, également connu sous le nom de Colaboratory ou Colab, est une plateforme offerte par Google qui permet aux utilisateurs d'écrire et d'exécuter du code Python de leur choix directement depuis leur navigateur. Basé sur Jupyter Notebook, Colab est spécialement conçu pour la formation et la recherche en apprentissage automatique. Il offre la possibilité d'entraîner des modèles de Machine Learning dans le Cloud. Les fonctionnalités de ***Colab*** comprennent :

- L'amélioration des compétences en programmation Python.
- Le développement d'applications en Deep Learning en utilisant des bibliothèques populaires telles que ***Keras, TensorFlow, PyTorch et OpenCV***.
- L'utilisation d'un environnement de développement (***Jupyter Notebook***) qui ne nécessite aucune configuration préalable.
- Cependant, ce qui distingue ***Colab*** des autres services est l'accès gratuit à un processeur graphique (***GPU***).[18]
-

3.4.2Langage de programmation et bibliothèques

3.4.2.1 Python

Au cours des dernières années, Python est devenu le langage de programmation préféré des informaticiens. Il occupe désormais une position de premier plan dans la gestion de l'infrastructure, l'analyse de données et le développement de logiciels. Ce langage permet aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes formelles qui étaient présentes dans les langages plus anciens. En conséquence, le développement de code avec Python est plus rapide que dans d'autres langages.

3.4.2.2Bibliothèques utilisées :

- **OS :** Le module **os** (*Operating System*) est une bibliothèque Python qui offre des fonctionnalités permettant d'interagir avec le système d'exploitation sur lequel le programme est exécuté. Il fournit des moyens de travailler avec des fonctionnalités spécifiques au système d'exploitation, telles que la gestion des fichiers et des répertoires, l'accès aux variables d'environnement, la manipulation des chemins de fichiers, et .Le module **os** offre de nombreuses autres fonctionnalités pour interagir avec le système d'exploitation. Il permet de rendre les programmes plus portables en gérant les différences entre les systèmes d'exploitation et en offrant des moyens de manipuler les fichiers, les répertoires et les variables d'environnement de manière efficace.[19]
- **NLTK** : abréviation de *Natural LanguageToolkit*, est une bibliothèque Python très populaire utilisée pour le traitement du langage naturel (NLP). Elle offre une variété d'outils et de ressources pour la préparation, l'analyse et la manipulation de textes écrits dans un langage naturel. Grâce à ses nombreuses fonctionnalités, NLTK est

largement utilisé dans des domaines tels que l'analyse de sentiment, la classification de texte, la génération de texte, la recherche d'informations, l'extraction d'entités nommées, et bien plus encore.[20]

- **pickle :** Le module pickle en Python fournit des fonctionnalités pour la sérialisation (pickling) et ledé-sérialisation (unpickling) d'objets Python. La sérialisation est le processus de conversion d'un objet Python en une représentation binaire, tandis que ledé-sérialisation est le processus inverse de conversion d'une représentation binaire en un objet Python. Le module pickle implémente des protocoles binaires de sérialisation et dé-sérialisation d'objets Python. La sérialisation est le procédé par lequel une hiérarchie d'objets Python est convertie en flux d'octets.Cependant, il est important de noter que la sérialisation avec pickle est spécifique à Python, ce qui signifie que les objets sérialisés avec pickle peuvent ne pas être compatibles avec d'autres langages de programmation. De plus, lors dudé-sérialisation, il est important de faire confiance à la source des données sérialisées, car des objets malveillants peuvent potentiellement être injectés et exécutés lors dudé-sérialisation.[21]
- **NumPy :** (*Numerical Python*) est une bibliothèque fondamentale en Python pour effectuer des calculs numériques et des opérations sur des tableaux multidimensionnels. Elle fournit des structures de données performantes pour manipuler et traiter des données numériques, ainsi qu'une vaste collection de fonctions mathématiques.Ci-dessous vous trouverez certaines des principales caractéristiques offertes par NumPy.[22]
- **PIL**: (*Python Imaging Library*) est une bibliothèque populaire en Python pour le traitement d'images. Elle offre une gamme étendue de fonctionnalités pour manipuler, modifier et traiter des images de manière efficace.Les principales fonctionnalités de cette bibliothèque incluent Chargement et enregistrement d'images Manipulation d'images Traitement d'images avancé et Transformation et composition d'images Intégration avec d'autres bibliothèques.[23]
- **COCO** : La bibliothèque COCO, également connue sous le nom de pycocotools, est un ensemble d'outils logiciels permettant de travailler avec le jeu de données COCO de manière pratique et efficace. Elle fournit des fonctionnalités pour charger, traiter et évaluer les données du jeu de données COCO. La bibliothèque COCO est largement utilisée dans la recherche en vision par ordinateur, en particulier dans des domaines tels que la détection d'objets, la segmentation sémantique, l'évaluation des performances des modèles et le développement de nouveaux algorithmes. Elle fournit une interface pratique pour travailler avec les données du jeu de données COCO, facilitant ainsi le développement et l'évaluation de modèles de vision par ordinateur.[24]
- **Matplotlib** est une bibliothèque populaire de visualisation de données en Python. Elle offre un large éventail de fonctionnalités pour créer des graphiques, des diagrammes et des visualisations interactives à partir de données.Matplotlib est largement utilisé dans les domaines de la science des données, de la recherche, de la visualisation de données, de l'apprentissage automatique et de nombreux autres domaines où la représentation visuelle des données est importante.[25]

- **PyTorch :** est une bibliothèque d'intelligence artificielle développée par Meta (anciennement Facebook). Elle est écrite en Python et est largement utilisée pour le deeplearning et la création de réseaux de neurones artificiels. PyTorch permet de réaliser des calculs de gradients et de manipuler des tableaux multidimensionnels grâce à des tenseurs.PyTorch est un projet open source sous licence BSD modifiée, disponible depuis 2016. En 2018, Meta a fusionné PyTorch avec Caffe2, une autre infrastructure de deeplearning, afin de bénéficier de ses capacités de déploiement et de traitement d'algorithmes d'apprentissage complexes avec de nombreux paramètres.[26]

3.5 Les étapes de travail et exécution :

1- Premier de chose, il faut utilise la bibliothèque ***google.colab*** pour monter le lecteur ***Google Drive*** dans l'environnement de développement ***Google Colab***.

2- Et puis nous utilisons le code ***wget***, qui permet de télécharger des fichiers compressés des données contenant des images de l'ensemble de données ***COCO***, et puis on extrait des images de fichiers compressés et on supprime les fichiers compressés téléchargés pour libérer l'espace mémoire.

3- Nous écrivons ensuite le code qui nous permet d'importer les bibliothèques et les modules nécessaires pour effectuer les différents processus de traitement des données, de modélisation des réseaux neuronaux.

4- «***nltk.download('punkt')***» Ce code utilise la bibliothèque **nltk** (*Natural LanguageToolkit*) pour télécharger les données supplémentaires nécessaires à l'utilisation de la fonction ***punkt***.La fonction ***punkt*** de ***nltk*** est utilisée pour la tokenisation, c'est-à-dire la division d'un texte en une liste de mots ou de phrases. Cependant, la première fois que vous utilisez cette fonctionnalité, vous devez télécharger les données supplémentaires requises.La ligne de code ***nltk.download('punkt')*** déclenche le téléchargement des données nécessaires à la fonction ***punkt*** depuis les serveurs de ***nltk***. Ces données comprennent notamment des modèles de langage pré-entrainés qui sont utilisés pour la tokenisation.Après l'exécution de cette ligne de code, les données nécessaires à la tokenisation sont téléchargées et prêtent à être utilisées avec la fonction ***punkt*** de ***nltk***.

5- ***« class Vocab(object) »***ce code définit une classe Vocab (vocabulaire) pour créer et sauvegarder un vocabulaire à partir d'annotations de légendes dans le jeu de données ***COCO***.La classe Vocab est utilisée pour stocker et gérer le vocabulaire. En résumé, ce code construit un vocabulaire à partir des légendes du jeu de données ***COCO*** et la sauvegarde dans un fichier ***pickle***. Le vocabulaire est utilisé pour mapper les mots aux indices et vice versa, ce qui sera utile pour la création de modèles de traitement du langage naturel.

6- « ***defreshape_image(image, shape)***»est une fonction pour redimensionner une image à la forme spécifiée. La fonction utilise la méthode redimensionnée de l'objet Image de la bibliothèque ***PIL*** (*Python Imaging Library*) pour effectuer le redimensionnement. Elle retourne l'image redimensionnée.

7- Ensuite, nous allons au code qui définit une classe ***CNNModel***qui représente un modèle de CNN. Le modèle est basé sur ***ResNet-152***, un réseau pré-entraîné avec 152 couches de convolution.
8- Pour stabiliser l'apprentissage en ajustant les valeurs des fonctionnalités dans un intervalle spécifique. Une couche de normalisation a été ajoutée pour le modèle Resnet-152. En résumé, cette classe CNNModel représente un modèle CNN basé sur ResNet-152. Il permet d'extraire des fonctionnalités à partir d'images en utilisant ResNet-152 pré-entraîné, puis de les réduire en dimensionnalité à l'aide d'une couche linéaire. Ce modèle est utilisé généralement comme extracteur de caractéristiques pour les tâches de vision par ordinateur telles que la classification d'images ou la génération de légendes.
9- Les modèles CNN et LSTM sont configurés et déplacés sur le périphérique spécifié. Les poids pré-entraînés peuvent être chargés si nécessaire. Ensuite, une image est chargée et prétraitée, puis transférée sur le périphérique spécifié. Le modèle CNN est utilisé pour extraire les fonctionnalités de l'image, qui sont ensuite utilisées par le modèle LSTM pour générer une légende en utilisant la méthode de recherche gloutonne. Les mots prédits sont convertis en mots réels en utilisant le vocabulaire jusqu'à ce que la fin de la légende soit atteinte. Enfin, la légende générée est imprimée et l'image originale est affichée à l'aide de la bibliothèque ***Matplotlib***.

3.6 Configuration expérimentale

Pour avoir les meilleurs paramètres qui conduisent aux bons résultats, nous avons relancé l'apprentissage plusieurs fois. Les paramètres principaux ont fixé pour l'apprentissage sont : epoch, steps-per-epoch, fonction d'activation, optimiseurs :

- **Epoch** : est le nombre total d'itérations d'apprentissage, il est défini comme un critère d'arrêt que ce soit les résultats. Nous l'avons fixé à 3 epochs.
- **Steps-per-epoch** : est le nombre total d'étapes à produire du générateur avant de déclarer une époque terminée et de commencer la prochaine époque. Il doit généralement être égal à (nombre d'échantillons total / nombre de batch-size). Nous utilisons les appellations step-per-epoch pour la partie apprentissage et validation-steps pour la partie test.
- **Fonction d'activation :** dans un réseau neuronal définit comment la somme pondérée de l'entrée est transformée en sortie à partir d'un nœud ou de nœuds dans une couche du réseau. Nous avons utilisé la fonction Tanh pour la couche cachée de Conv1d après la fonction.
- **Optimiseur** : Les optimiseurs sont des algorithmes ou des méthodes utilisées pour modifier les attributs de votre réseau de neurones tels que les poids et le taux d'apprentissage afin de réduire les pertes

Nous avons évalué et validé l'efficacité du modèle hybride CNN-LSTM proposé en utilisant la base de données ***COCO***. Nous avons formé notre modèle sur Google Colab avec un moteur de calcul Google Python 3 plusieurs expérimentations ont été menées pour trouver les meilleures valeurs des paramètres d'apprentissage

- **Loss_criterion = nn.CrossEntropyLoss()** : La variable ***loss_criterion*** est définie comme une instance de la *classe CrossEntropyLoss* de PyTorch. Cette fonction de perte est couramment utilisée pour les problèmes de classification multi-classe. Elle calcule la perte en comparant les sorties du modèle avec les étiquettes cibles (dans ce cas, les légendes) en utilisant la perte de l'entropie croisée.
- **Optimizer = torch.optim.Adam(parameters, lr=0.001) :** L'optimiseur est défini comme une instance de la classe Adam de PyTorch. L'optimiseur Adam est une méthode d'optimisation couramment utilisée qui ajuste les taux d'apprentissage individuels pour chaque paramètre. Il prend en entrée les paramètres à optimiser et le taux d'apprentissage (*lr*) fixé à **0.001** dans ce cas. L'optimiseur sera utilisé pour mettre à jour les paramètres lors de la rétropropagation du gradient pendant l'entraînement.

3.7Résultats

Après l'apprentissage du modèle, nous avons utilisé les données de test pour obtenir des données prédites Qui étaient 4 images que nous avons utilisées dans test ci-dessous nous passons en revue nos résultats après notre formation du modèle que nous avons travaillé à et nous avons pris les résultats de quatre formations différentes comme suit :

Les résultats, qui sont des légendes photographiques, sont en **anglais et** en **français**

Au début des résultats des exercices, ils n'étaient pas exacts, car les légendes des images que nous avons obtenues à l'époque ne contenaient pas la bonne adresse présentée dans l'image proposée sur le formulaire, comme le montrent les tableaux 3.1 et tableau 3.2

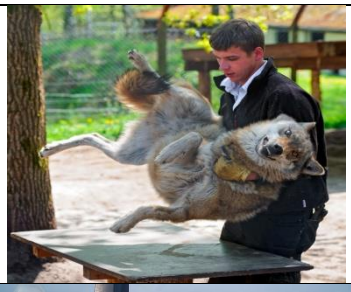
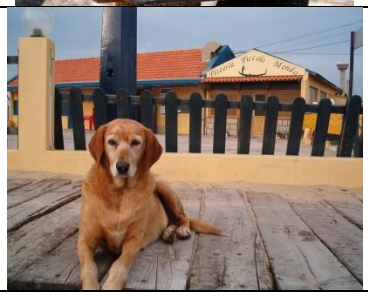
Image de test	Légende d'image (en Anglais)
	<start> a man is on a skateboard in the air . <end> un homme est sur une planche à roulettes dans les airs
	<start> a man iswalking on the beachwith a surfboard .<end> un homme marche sur la plage avec une planche de surf
	<start> a dog sitting on a bench in a park . <end> un chien assis sur un banc dans un parc .
	<start> a dog issitting on a bench in the water .<end> un chien est assis sur un banc dans l'eau

Tableau 0-1résultat de l'itération 1

Par exemple, dans la Légende de la première photo, il nous a été mentionné qu'un homme est sur une planche à roulettes en l'air, et il en va de même pour la deuxième photo, le titre n'était pas assez clair, tout ce qu'il disait était qu'un homme marche au bord de la mer

La même chose a été répétée avec la troisième image, où nous avons trouvé dans La légende de l'image qu'il est mentionné qu'un chien est assis sur un banc de parc

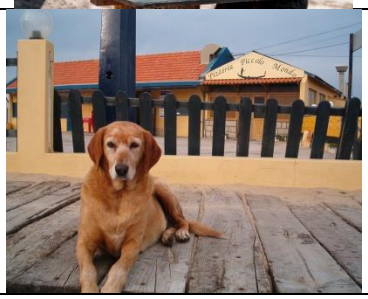
Image de test	Légende d'image (en Anglais)
	<start> a man is doing a trick on a skateboard . <end> un homme fait un tour sur un skateboard.
	<start> a man riding a surfboard on top of a wave .<end> un homme sur une planche de surf sur une vague .
	<start> a man sitting on a bench with a dog . <end> un homme assis sur un banc avec un chien.
	<start> a dog islooking out of a window. <end> un chien regarde par la fenêtre.

Tableau 0-2résultat de l'itération2

Mais avec les formations successives sur le modèle spécial, nous avons commencé à en tirer les fruits, à mesure que les mythes des images résultantes commençaient à s'améliorer progressivement, et à chaque fois le titre devenait plus clair, et c'est ce que nous touchons dans le tableau 3.2, par exemple, nous trouvons dans le mythe de la troisième image que nous avons obtenu

À ce stade des résultats, s'il est correct de l'appeler cette phrase, nous constatons que les résultats obtenus après cette formation n'étaient pas si clairs

C'est ce qui nous a incité à effectuer une autre formation pour notre modèle, afin d'obtenir des résultats plus précis et des légendes d'images plus claires qu'auparavant, donc après avoir terminé la formation, nous obtenons les résultats qui se trouvent dans le tableau suivant (tableau 3.3)

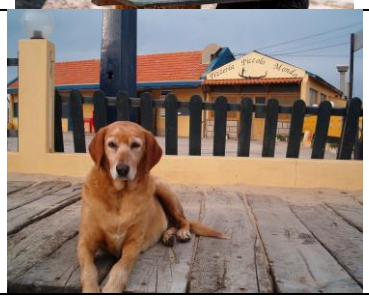
Image de test	**Légende d'image (en Anglais)**
	<start> a man is holding a tennis racket in his hand. <end> un homme tient une raquette de tennis à la main.
	<start> a man is walking on the beach with a surfboard. <end> un homme marche sur la plage avec une planche de surf
	<start> a man is sitting on a bench with a dog. <end> un homme est assis sur un banc avec un chien
	<start> a dog is standing on a sidewalk near a building. <end> > un chien est debout sur un trottoir près d'un immeuble

Tableau 3-3 résultat de l'itération 3

Comme nous l'avons déjà mentionné, nous avons suivi une autre formation pour obtenir de meilleurs résultats, puis nous mentionnons dans ce contexte ce que nous avons tiré des mythes des photos, dont les résultats étaient plus précis que les précédents, et nous trouvons la troisième image qu'il exprime un homme assis avec un chien sur un banc

La même chose que nous trouvons dans le mythe de la quatrième image que le chien est debout sur le trottoir près du bâtiment et c'était le résultat le plus précis que nous ayons obtenu.

La nature de ces modèles nécessite un grand nombre de formations pour obtenir les résultats les plus précis, tout comme élever un petit enfant, tout ce qui était enseigné de manière optimale et correcte était sa croissance et une plus grande générosité dans son avenir, et pour le goût du temps et notre retard notable dans l'achèvement de notre travail à temps, nous ne pouvions pas obtenir plus que cela, comme nous l'avons déjà mentionné, ces modèles nécessitent un grand nombre de temps de formation .

Et il y a aussi un point qu'il faut souligner, et que nous avons rencontré dans notre travail, c'est que les images choisies dans la formation ne doivent pas être complexes ou composées de plusieurs choses, cela rend difficile la tâche du modèle de donner des légendes d'images plus claires et précises, et c'est exactement ce que nous sommes tombés dans le piège car dans le premier nous avons choisi des images que le modèle ne pouvait pas nous fournir de résultats exacts, donc ces images ont été modifiées avec celles trouvées ici dans les exemples ci-dessus, et c'est l'une des raisons pour lesquelles nous n'avons pas fait plus de temps de formation .

3.8 Conclusion

Ce chapitre était divisé en deux parties : implémentation et résultats ; La premièrepartie, nous avons introduit l'environnement de travail, le langage de programmation, bases de l'apprentissage et des tests, détails d'implémentation de l'étape d'apprentissage, ainsi qu'une interface graphique. Dans la deuxième partie, nous avons présenté la composition expérimentale et résultat. Selon les résultats, le modèle CNN-LSTM proposé n'a pas donné les meilleurs résultats, car il nécessitait un plus grand nombre d'implémentations

Conclusion générale

En conclusion, l'utilisation de réseaux de neurones convolutionnels (CNN) et de réseaux de neurones à mémoire à long terme (LSTM) dans les générateurs de légendes d'images représente une approche avancée pour générer automatiquement des descriptions textuelles pour des images. Cette combinaison d'architectures de réseaux de neurones permet de tirer parti des caractéristiques visuelles extraites par le CNN et de les utiliser comme entrée pour générer des légendes cohérentes et pertinentes à l'aide du LSTM.

Les générateurs de légendes d'images utilisant CNN et LSTM ont le potentiel d'améliorer l'accessibilité des images pour les personnes aveugles ou malvoyantes en fournissant des descriptions textuelles détaillées. Ils peuvent également être utilisés dans divers domaines tels que la reconnaissance d'images, la classification automatique d'images, la création de légendes pour des médias sociaux, la recherche d'images et bien d'autres.

Cependant, il est important de noter que ces générateurs de légendes d'images ne sont pas exempts de limites. La génération automatique de légendes d'images peut encore présenter des erreurs ou des incohérences, en particulier lorsqu'il s'agit d'interpréter des images complexes ou ambigües. Par conséquent, la vérification humaine et l'évaluation critique des légendes générées restent essentielles pour garantir leur exactitude et leur pertinence.

En somme, les générateurs de légendes d'images utilisant CNN et LSTM sont une technologie prometteuse qui facilite la description et l'interprétation des images grâce à l'intelligence artificielle. Leur développement continu et leur amélioration contribuent à rendre les images plus accessibles et à enrichir notre expérience visuelle dans différents domaines d'application.

Bibliographie

[1] K. MEZZOUG, «traitement et analyse des images numeriques,» Université Ibn Khaldoun – Tiaret. Université Ibn Khaldoun – Tiaret, 2019/2020.

[2] Z. :. Boulgamh, «Traitement d'images monochromes Détection de contours, Filtrage (Spatial et fréquentiel)etSegmentation par Réseaux de Neurones,» 2016/2017.

[3] I. DJILI, «Système de reconnaissance d'iris par réseaux de neurones convolutionnels,» universite Kasdi MERBAH Ouargla, 2019/2020.

[4] L. LAOUAMER, «Approche exploratoire sur la classification,» MÉMOIRE PRÉSENTÉ À, 2006.

[5] https://www.tibco.com/fr/reference-center/what-is-a-neural-network.

[6] https://aws.amazon.com/fr/what-is/neural-network/.

[7] https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/.

[8] https://blog.hubspot.fr/marketing/deep-learning.

[9] https://medium.com/analytics-vidhya/a-complete-guide-to-adam-and-rmsprop-optimizer-75f4502d83be.

[10]] https://artemoppermann.com/optimization-in-deep-learning-adagrad-rmsprop-adam/.

[11] A. S. U. Z. A. Q. A Khan, «A survey of the recent architectures of deep convolutionnal neural networks,» *artificial intelligence review ,* n° %153, pp. 5455-5516, 2020.

[12] Indolia, «Conceptual understanding of convolutional neural network-a deep learning approach.,» *Procedia computer science,* n° %1132, p. 679–688., 2018.

[13] S. a. J. S. Hochreiter, «Long short-term memory.,» *Neural computation 9.8,* pp. 1735-1780, 1997.

[14] M. A. DJABALLAH, «Système de prédiction de la consommation d'énergie basé Deep Learning,» Université de 8 Mai 1945 – Guelma –, 2021.

[15] X. Z. S. R. a. J. S. K. He, «Deep Residual Learning for Image Recognition.,» *arXiv.org,* 2015.

[16] [En ligne]. Available: https://www.run.ai/guides/deep-learning-for-computer-vision/pytorch-resnet.

[17] L. D. e. a. Nguyen, «Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation.,» *2018 IEEE international symposium on circuits and*

systems (ISCAS), p. IEEE, 2018.

[18] [En ligne]. Available: https://research.google.com/colaboratory/faq.html?hl=fr .

[19] F. Lundh, «Python standard library.,» *O'Reilly Media, Inc.,* 2001.

[20] [En ligne]. Available: https://www.nltk.org/ .

[21] [En ligne]. Available: https://www.quennec.fr/trucs-astuces/langages/python/python-le-module-pickle.

[22] D. e. a. Ascher, «Numerical python.,» 2001.

[23] P. Umesh, «Image processing in python.,» *CSI Communications ,* 2012.

[24] P. N. e. a. Chowdhury, «FS-COCO: towards understanding of freehand sketches of common objects in context.,» *Computer Vision–ECCV ,* 2022.

[25] [En ligne]. Available: https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/ .

[26] [En ligne]. Available: https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501871-pytorch140922/.

[27] numerical-tours.com, 2023.

[28] A. McAndrew, «An Introduction to Digital Image,» www.audentia-gestion.fr, 2023.

[29] «Traitement et analyse d'images,» https://perso.esiee.fr/~perretb/I5FM/TAI/index.html.

[30] *2000-2003 Bibliothèque de l'Université Cornell/Département de Recherches,* 2000-2003.

[31] :. K. DALIA, «Classification non supervisée de pixels,» 2016/2017.

[32] k. e. al, «A survey of the recent architectures of deep convolutional neural networks. Artificial Intelligence Review, pages 1–62.,» 2020.

[33]] https://www.researchgate.net/figure/Architecture-generale-dun-CNN-Mageswaran-2019-Un-reseau-neuronal-convolutif-se_fig4_355652191.

[34] [En ligne]. Available: https://modeling-languages.com/lstm-neural-network-model-transformations/.

[35] [En ligne]. Available: https://github.com/rasbt/deeplearning-models/blob/master/pytorch_ipynb/cnn/cnn-resnet152-celeba.ipynb.