

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



Mémoire de Fin d'études Master

Filière : Informatique

Option : Systèmes Informatiques

Thème :

**L'apprentissage profond appliqué à la
reconnaissance des anomalies mammaires.**

Encadré Par :
Dr FERKOUS Chokri

Présenté par :
MAKHLOUF Lazhar

JUIN 2022

Résumé

Le cancer du sein est l'une des tumeurs malignes les plus courantes chez les femmes, qui affectent gravement la santé physique et mentale des femmes et menacent même leur vie.

Actuellement, la mammographie est un outil important pour les médecins afin de diagnostiquer le cancer du sein. Cependant, en raison de la structure complexe des images mammographiques, il est relativement difficile pour les médecins d'identifier les caractéristiques du cancer du sein.

Actuellement, l'apprentissage en profondeur est l'algorithme de classification d'images le plus répandu. Pour cela, nous présentons l'architecture de notre travail qui nous permettra de choisir les hyperparamètres pour optimiser notre réseau de neurone à convolution ; les expérimentations de cette application seront tester sur des régions d'intérêts de la base de données mammographiques DDSM (The Digital Database for Screening Mammography). Le modèle élu après une série d'expérimentations nous permettra de classier les images selon leurs types (Normale / Anormale), ce modèle représente un système d'aide au diagnostic pour la classification des tissus mammaires. L'approche proposée a été évaluée sur 6688 régions d'intérêt.

Mots-clefs : Deep learning, mammographie, hyperparamètres, classification.

ملخص

يعد سرطان الثدي من أكثر الأورام الخبيثة شيوعًا عند النساء، وهو يؤثر بشكل خطير على الصحة البدنية والعقلية للمرأة بل ويهدد حياتها. في الوقت الحاضر، يعتبر التصوير الشعاعي للثدي معيارًا مهمًا للأطباء لتشخيص سرطان الثدي. ومع ذلك، نظرًا للتركيب المعقد لصور الماموجرام، يصعب نسبيًا على الأطباء تحديد سمات سرطان الثدي.

يعتبر التعلم العميق في الآونة الأخيرة كأكثر خوارزمية لتصنيف الصور شيوعًا. في هذا العمل نقدم مخطط سيسمح لنا باختيار الخصائص العامة للشبكة العصبية التلافيفية CNN بهدف التفريق بين الصور التي تحتوي على مرض خبيث والصور العادية؛ سيتم اختبار تجارب هذا التطبيق على قاعدة بيانات الصور الشعاعية للثدي DDSM (قاعدة البيانات الرقمية لفحص التصوير الشعاعي للثدي). يمكن اعتبار النموذج الأمثل لتصنيف الأنسجة الشعاعية للثدي وفقًا لأنواعها (طبيعي أو غير طبيعي) نظامًا مساعدًا في التشخيص يتم تطبيقه على الصور الشعاعية للثدي. تم تقييم المخطط المقترح على 6688 منطقة مستخرجة من صور الأنسجة الشعاعية للثدي.

الكلمات المفتاحية: التعلم العميق ، التصوير الشعاعي للثدي ، الخصائص العامة، التصنيف..

Abstract

Breast cancer is one of the most common malignant tumors in women, which seriously affect women's physical and mental health and even threat to life. At present, mammography is an important criterion for doctors to diagnose breast cancer. However, due to the complex structure of mammogram images, it is relatively difficult for doctors to identify breast cancer features. At present, deep learning is the most mainstream image classification algorithm.

We present the architecture of our work which will allow us to choose the hyperparameters of a convolutional neural network; the experiments of this application will be tested on the mammographic image database DDSM (The Digital Database for Screening Mammography). An optimized model for the classification of mammographic tissues according to their types (Normal / Abnormal) represents a diagnostic aid system applied to mammographic images. The proposed approach was evaluated on 6688 regions of interest extracted from mammographic tissues.

Keywords : Deep learning, mammography, hyperparameters, classification.

REMERCIEMENTS

Je tiens à remercier toutes les enseignants, mes collègues et mes amis qui ont contribué mon succès durant les années académiques du master.

Je tiens à remercier spécialement mon encadreur Dr.FERKOUS Chokri qui m'a honoré en acceptant mon encadrement et pour les efforts, les directives et les conseils qu'il m'a donnés pour rédiger le mémoire de fin d'études.

MAKHLOUF Lazhar

DEDICACE

Je dédie ce travail

A ma mère, que Dieu lui fasse miséricorde.

A mon père, mes frères, mes sœurs,

Ma petite famille : mes enfants Sadjed, Ferdous et ma femme qui m'a soutenu

et a été patient avec moi pendant mes études,

Et ceux qui ont partagé avec moi le respect et appréciation lors de mon cursus.

Ils m'ont chaleureusement supporté et encouragé tout au long de mon

parcours.

A tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de

succès.

MAKHLOUF Lazhar

Table des matières

RESUME	I
ملخص	II
ABSTRACT	III
REMERCIEMENTS	IV
DEDICACE	V
TABLE DES MATIERES	1
LISTE DES FIGURES	5
LISTE DES TABLEAUX	7
ABREVIATIONS ET ACRONYMES	8
INTRODUCTION GENERALE	10
CHAPITRE I: INTELLIGENCE & VISION ARTIFICIELLE	12
I.1. INTRODUCTION	13
I.2. L'INTELLIGENCE ARTIFICIELLE	13
I.3. L'APPRENTISSAGE AUTOMATIQUE.....	15
<i>I.3.1. L'apprentissage supervisé</i>	<i>16</i>
I.3.1.1. Les machines à vecteur de support SVM.....	17
I.3.1.1. Les réseaux de neurones:	18
I.3.1.1. L'apprentissage profond (Deep Learning)	18
<i>I.3.2. Apprentissage non supervisé.....</i>	<i>18</i>
I.3.2.1. La régression linéaire	19
I.3.2.2. L'algorithme des k plus proches voisins.....	21
I.3.2.3. Le clustering K-moyenne	22
<i>I.3.3. Apprentissage semi-supervisé</i>	<i>23</i>
I.3.3.1. Apprentissage par renforcement	23
I.4. DOMAINES D'UTILISATION DE L'APPRENTISSAGE AUTOMATIQUE	24
I.5. CONCLUSION	25
CHAPITRE II: LES RESEAUX DE NEURONES ARTIFICIELS ET A CONVOLUTION.....	26
II.1. INTRODUCTION.....	27
II.2. NEURONE BIOLOGIQUE	27
II.3. NEURONE ARTIFICIEL	28
<i>II.3.1. Structure des réseaux de neurones artificiels.....</i>	<i>28</i>

II.3.2.	<i>Types des réseaux de neurones artificiels</i>	30
II.3.3.	<i>L'initialisation du poids</i>	31
II.3.3.1.	Initialisation tous zéros et initialisation constante	31
II.3.3.2.	Initialisation aléatoire	31
II.3.3.3.	Initialisation LeCun (normaliser la variance)	31
II.3.3.4.	Initialisation de Xavier (initialisation de Glorot).....	31
II.3.3.5.	Initialisation HE (initialisation de Kaiming).....	32
II.3.4.	<i>Les fonctions d'activations</i>	33
II.3.4.1.	La fonction d'activation sigmoïde ou logit	33
II.3.4.2.	La fonction Softmax	33
II.3.4.3.	La fonction Tanh	34
II.3.4.4.	La fonction de l'Unité Linéaire Rectifiée (ReLU)	34
II.3.4.5.	La fonction LeakyReLU.....	34
II.3.4.6.	La fonction ReLU paramétrique (PReLU).....	35
II.3.4.7.	Fonction d'activation SWISH.....	36
II.3.5.	<i>Les fonctions de pertes (erreurs)</i>	36
II.3.5.1.	Erreur Quadratique Moyenne	37
II.3.5.2.	Erreur Absolue Moyenne	37
II.3.5.3.	Entropie croisée binaire.....	38
II.3.5.4.	Entropie croisée catégorielle	38
II.3.6.	<i>Les Optimiseurs</i>	38
II.3.6.1.	La descente de gradient (GD).....	38
II.3.6.2.	Descente de gradient par lots (BGD).....	39
II.3.6.3.	Descente de gradient stochastique (SGD)	40
II.3.6.4.	Descente de gradient en mini-lot.....	40
II.3.6.5.	Momentum	42
II.3.6.6.	Nesterov Momentum.....	42
II.3.6.7.	Descente de gradient adaptative (AdaGrad)	43
II.3.6.8.	Delta adaptatif (AdaDelta)	43
II.3.6.9.	Propagation racine carrée moyenne (RMSProp)	43
II.3.6.10.	Estimation adaptative du moment (Adam)	43
II.3.6.11.	Estimation du moment adaptatif maximal (AdaMax).....	44
II.3.6.12.	Estimation du moment adaptatif accélérée par Nesterov (Nadam)	44
II.4.	APPRENTISSAGE PROFOND (DEEP LEARNING OU DL)	44
II.4.1.	<i>Les réseaux de neurones convolutifs (CNN)</i>	44
II.4.1.1.	Les avantages de CNN	46
II.4.1.2.	Les inconvénients de CNN	46
II.4.1.3.	L'opération de convolution	47
II.4.1.4.	Mise en commun (Pooling)	48
II.4.1.5.	Aplanissement (Flattening)	48
II.5.	QUELQUES ARCHITECTURE BASEES CNN	49

II.5.1.	VGG16	49
II.5.2.	Inception Net.....	49
II.5.3.	ResNet.....	50
II.6.	CONCLUSION.....	51
CHAPITRE III: REVUES DE LA LITTERATURE		52
III.1.	INTRODUCTION.....	53
III.2.	CLASSIFICATION DES PATHOLOGIES MAMMAIRES	53
III.2.1.	<i>Classification basée sur l'apprentissage profond des images mammographiques</i>	<i>53</i>
III.2.2.	<i>Classification des mammographies basée sur l'apprentissage profond pour le cancer du sein</i>	<i>54</i>
III.2.3.	<i>Performances incohérentes des modèles d'apprentissage en profondeur sur la classification des mammographies</i>	<i>55</i>
III.2.4.	<i>Prédiction de l'efficacité de la chimiothérapie appliquée au cancer du sein par le traitement d'images et le Deep Learning.</i>	<i>55</i>
III.3.	CLASSIFICATION BI-RADS DES ANOMALIES MAMMAIRES	56
III.3.1.	<i>Un modèle de réseau de neurones profonds à haute performance pour une classification BI-RADS de la mammographie de dépistage</i>	<i>56</i>
III.3.2.	<i>Modèle RN-CNN d'apprentissage profond pour la classification BI-RADS du cancer du sein...</i>	<i>58</i>
III.3.3.	<i>Classification BI-RADS des images de mammographie à l'aide d'OHPLall</i>	<i>59</i>
III.3.4.	<i>L'Apprentissage en profondeur en utilisant une petite base de données de petite taille pour la classification de densité BI-RADS des images mammographiques.....</i>	<i>60</i>
III.4.	CLASSIFICATION SELON LA TYPOLOGIE.....	61
III.4.1.	<i>Classification multi-classes des anomalies du cancer du sein à l'aide du réseau de neurones à convolution (CNN).....</i>	<i>61</i>
III.4.2.	<i>Apprentissage en profondeur pour une détection précoce du cancer du sein.....</i>	<i>62</i>
III.4.3.	<i>Capacités d'apprentissage en profondeur pour la catégorisation des micro-calcifications</i>	<i>63</i>
III.4.4.	<i>Détection de la micro-calcification dans les mammographies à l'aide d'un réseau de neurone convolutif séparable en profondeur entièrement connecté</i>	<i>64</i>
III.5.	CONCLUSION.....	65
CHAPITRE IV: OPTIMISATION DES HYPERPARAMETRES ET DEPLOIEMENT		66
IV.1.	INTRODUCTION.....	67
IV.2.	ENVIRONNEMENT DE DEVELOPPEMENT DE SYSTEME.....	67
IV.2.1.	<i>Environnement matériel</i>	<i>67</i>
IV.2.2.	<i>Environnement logiciel et langages de programmation</i>	<i>67</i>
IV.2.2.1.	<i>Langage JAVA</i>	<i>67</i>
IV.2.2.2.	<i>Langage Python.....</i>	<i>68</i>
IV.2.2.3.	<i>Keras</i>	<i>68</i>
IV.2.2.1.	<i>Le Framework Flask.....</i>	<i>68</i>

IV.2.3.	<i>La base de données utilisée</i>	69
IV.3.	ARCHITECTURES GENERALE DE TRAVAIL.....	72
IV.3.1.	<i>Préparation des données</i>	73
IV.3.1.1.	Recadrage et redimensionnement des régions d'intérêts	73
IV.3.1.2.	Prétraitement des régions d'intérêts (ROIs).....	73
IV.3.2.	<i>Optimisation des hyperparamètres</i>	74
IV.3.2.1.	Optimisation avec Keras Tuner	75
IV.3.2.2.	L'hypermodèle	75
IV.3.2.3.	Instancier le réglage et effectuer l'hypertuning	76
IV.3.2.4.	Sélection & réapprentissage du meilleur modèle élu	78
IV.3.3.	<i>Évaluation de la performance</i>	80
IV.3.4.	<i>Déploiement de l'application</i>	82
IV.4.	CONCLUSION.....	83
	CONCLUSION GENERALE	85
	REFERENCES BIBLIOGRAPHIQUES	87

Liste des Figures

FIGURE I-1: LES RELATIONS ENTRE L'INTELLIGENCE ARTIFICIELLE, L'APPRENTISSAGE AUTOMATIQUE ET L'APPRENTISSAGE PROFOND. .	14
FIGURE I-2: TYPES D'APPRENTISSAGE EN APPRENTISSAGE PAR MACHINE.	15
FIGURE I-3: ARBRE REPRESENTE QUELQUES ALGORITHMES DE L'APPRENTISSAGE AUTOMATIQUE.	16
FIGURE I-4: SCHEMA D'APPRENTISSAGE SUPERVISE "SUPERVISED LEARNING" [RUS20].	17
FIGURE I-5: EXEMPLE SIMPLE REPRESENTE LA CLASSIFICATION SVM.....	17
FIGURE I-6: SCHEMA GENERALE D'UN RESEAU DE NEURONE.	18
FIGURE I-7: SCHEMA D'APPRENTISSAGE NON SUPERVISE "UNSUPERVISED LEARNING".....	19
FIGURE I-8: LA REGRESSION LINEAIRE.	20
FIGURE I-9: EXEMPLE DE K-PLUS PROCHE VOISINS (KPPV).	22
FIGURE I-10: SCHEMA D'APPRENTISSAGE SEMI SUPERVISE "SEMI SUPERVISED LEARNING".....	23
FIGURE I-11: SCHEMA D'APPRENTISSAGE PAR RENFORCEMENT "REINFORCEMENT LEARNING" [MAZ15].	24
FIGURE II-1: NEURONE BIOLOGIQUE.....	27
FIGURE II-2: NEURONE ARTIFICIEL.....	28
FIGURE II-3: SCHEMA SIMPLIFIE D'UN PERCEPTRON MULTICOUCHE COMPORTANT DEUX COUCHES CACHEES ET EFFECTUANT UNE CLASSIFICATION BINAIRE [VIA21].	29
FIGURE II-4: LES TYPES DE RESEAUX DE NEURONES LES PLUS CONNUS.....	30
FIGURE II-5: SCHEMA REPRESENTE LE PROCESSUS D'INITIALISATION DU POIDS [BOU22].	31
FIGURE II-6: SCHEMA REPRESENTE LA FONCTION D'ACTIVATION SIGMOID.....	33
FIGURE II-7: SCHEMA REPRESENTE LA FONCTION D'ACTIVATION SOFTMAX.....	33
FIGURE II-8: SCHEMA REPRESENTE LA FONCTION D'ACTIVATION TANH.	34
FIGURE II-9: SCHEMA REPRESENTE LA FONCTION D'ACTIVATION RELU.	34
FIGURE II-10: SCHEMA REPRESENTE LA FONCTION D'ACTIVATION LEAKYRELU.....	35
FIGURE II-11: SCHEMA REPRESENTE LA FONCTION D'ACTIVATION PRELU.....	35
FIGURE II-12: SCHEMA REPRESENTE LA FONCTION D'ACTIVATION SWICH.....	36
FIGURE II-13: ALGORITHME DE DESCENTE DE GRADIENT [HAJ21].	39
FIGURE II-14: DESCENTE DE GRADIENT STOCHASTIQUE [RUD16].	40
FIGURE II-15: MINIMA GLOBAUX Vs. MINIMA LOCAUX [HAJ21].	41
FIGURE II-16: LA COMPARAISON DES ALGORITHMES SGD AVEC ET SANS QUANTITE DE MOUVEMENT [HAJ21].	42
FIGURE II-17: STRUCTURE DU RESEAU NEURONAL A CONVOLUTION [SRI13].....	45
FIGURE II-18: SCHEMA REPRESENTE L'OPERATION DE CONVOLUTION.	47
FIGURE II-19: SCHEMA REPRESENTE LA COUCHE POOLING.....	48
FIGURE II-20: SCHEMA D'UN RESEAU A CONVOLUTION AVEC LA COUCHE FLATTENING.	49
FIGURE II-21: SCHEMA REPRESENTE L'ARCHITECTURE D'INCEPTION DE GOOGLE.....	50
FIGURE II-22: SCHEMA REPRESENTE L'ARCHITECTURE RESNET.....	50
FIGURE III-1: LA STRUCTURE DU MODELE DE RESEAU DE NEURONE DENSENET-II.	54

FIGURE III-2: L'ARCHITECTURE PROPOSEE DES RESEAUX DE NEURONES CONVOLUTIFS COUCHE PAR COUCHE.	54
FIGURE III-3: ARCHITECTURE DU MODELE D'APPRENTISSAGE PROFOND PROPOSEE PAR WANG ET AL. [WAN20]	55
FIGURE III-4: ARCHITECTURE SEGNET ADAPTEE DANS CE TRAVAIL.	56
FIGURE III-5: ORGANIGRAMME DU MODELE DE CLASSIFICATION BI-RADS PRESENTE PAR TSAI ET AL. [TSA22].	57
FIGURE III-6: ARCHITECTURE DU MODELE RN-BCNN PROPOSEE PAR SIDDEEQ ET AL. [SID21].	58
FIGURE III-7: MODELE CNN UTILISE DANS L'ETUDE EXPERIMENTALE DE VANDERHEYDEN ET AL. [VAN20].....	59
FIGURE III-8: RESEAU DE NEURONES EN PROFONDEUR AVEC TROIS COUCHES DE CONVOLUTION PROPOSE PAR SHI ET AL. [SHI19]... ..	60
FIGURE III-9: ARCHITECTURE DU MODELE PROPOSE PAR HEENAYE ET AL. [HEE21].	61
FIGURE III-10: L'ARCHITECTURE PROPOSEE PAR SHE ET AL. [SHE19], POUR LA DETECTION DES ANOMALIES MAMMAIRES.....	62
FIGURE III-11: L'ARCHITECTURE DU RESEAU DE NEURONES CONVOLUTIONELLE PROPOSEE PAR KUMAR ET AL. [KUM22].	64
FIGURE III-12: L'ARCHITECTURE DU SYSTEME CAO PROPOSE PAR REHMAN ET AL. [REH21].....	65
FIGURE IV-1: DES MASSES ET DES CALCIFICATIONS REPARTIS SUR DIFFERENTES CLASSES BI-RADS 0, 2, 3, 4 ET 5.....	70
FIGURE IV-2: DEUX REGIONS D'INTERETS RETIRES D'UNE MAMMOGRAPHIE CLASSEES BI-RADS 1	71
FIGURE IV-3: L'ARCHITECTURE GENERALE DE NOTRE TRAVAIL.	72
FIGURE IV-4: L'ARCHITECTURE D'OPTIMISATION EN UTILISANT KERAS-TUNER.	75
FIGURE IV-5: ÉVOLUTION DE L'APPRENTISSAGE SUR 50 ITERATIONS	79
FIGURE IV-6: ÉVOLUTION DE L'ERREUR SUR 50 ITERATIONS.	79
FIGURE IV-7: INTERFACE GRAPHIQUE (LA SELECTION DES REGIONS D'INTERETS).	82
FIGURE IV-8: INTERFACE GRAPHIQUE (RESULTAT DE CLASSIFICATION DE LA REGION D'INTERET SELECTIONNEE).....	83

Liste des tableaux

TABLEAU II-1: COMPARAISON ENTRE LES TECHNIQUES D'INITIALISATION DU POIDS.	32
TABLEAU IV-1: REPARTITION DES REGIONS D'INTERETS DE LA BASE DE DONNEES UTILISEE.	71
TABLEAU IV-2: LES RESULTATS DE L'EXPERIMENTATION EFFECTUEE SUR UNE SEULE COUCHE DE CONVOLUTION.	76
TABLEAU IV-3: LES RESULTATS DE L'EXPERIMENTATION EFFECTUEE SUR DEUX COUCHES DE CONVOLUTION.	77
TABLEAU IV-4: LES RESULTATS DE L'EXPERIMENTATION EFFECTUEE SUR TROIS COUCHES DE CONVOLUTION.	77
TABLEAU IV-5: LES RESULTATS OBTENUS APRES LA PHASE DE LA REAPPRENTISSAGE DU MODELE ELU.	78
TABLEAU IV-6: RELATION ENTRE VRAI POSITIVE, VRAI NEGATIVE, FAUX POSITIVE ET FAUX NEGATIVE DE LA MATRICE DE CONFUSION	80
TABLEAU IV-7: LES MESURES DE PERFORMANCE DE LA CLASSIFICATION DES TISSUES (NORMALE / ANORMALE).....	81

Abréviations et Acronymes

AdaDelta	Adaptive Delta
AdaGrad	Adaptive Gradient
Adam	Adaptive Moment Estimation
AENN	Auto Encoder Neural Network
BGD	Batch Gradient Descent
BI-RADS	Breast Imaging Reporting and Data System
BSD	Berkeley Software Distribution License
CC	CranioCaudal
CNN	Convolutional Neural Networks
ConvNet	Convolutional Networks
CPU	Central Processor Unit
DDSM	The Digital Database for Screening Mammography
DenseNet	Dense Convolutional Network
DFFN	Deep Feed Forward Neural Network
DL	Deep Learning
DNN	Deep Neural Network
FC	Fully Connected Layer
FN	Faux Négative
FP	Faux Positive
GD	Gradient Descent
HN	Hopfield Neural
HNN	Hamming Neural Network
KPPV	K Plus Proches Voisins
KSOMNN	Kohonen Self Organizing Maps Neural Network
LeakyReLU	Leaky Rectified Linear Unit
LR	Learning Rate (Taux d'apprentissage)
MAPE	Mean Absolute Percentage Error
MCN	Markov Chain Network
ML	Machine Learning
MLO	MedioLateral Oblique
MLPNN	Multi Layer Perceptron Neural Network
MSLE	Mean Square Logarithmic Error
NNLSTM	Neural Network Long Short Term Memory
OHPLall	Ordinal HyperPlaneLoss - all centroids

ONEIROS	Open-ended Neuro-Electronic Intelligent Robot Operating System
PCA	Principal Component Analysis
PReLU	Parametric Rectified Linear Unit.
PRM	Precision Runway Monitor Training
RAM	Read Acces Memory
RBFNN	Radial Basis Function Neural Network
RCNN	Regions with Convolutional Neural Networks
ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
RMSProp	Root Mean Square Propagation
RN-BCNN	ResNet-Based Customized Neural Network
RNN	Recurrent Neural Network
ROI	Region Of Interest
RVB	Rouge Vert Bleu
SGD	Stochastic Gradient Descent
SSL	Semi Supervised Learning
SVM	Support Vecteur Machine
TDNN	Time Delay Neural Network
VN	Vrai Négative
VP	Vrai Positive

INTRODUCTION GENERALE

Avec l'augmentation de la puissance des machines dans les dernières décennies a permis le développement des techniques d'apprentissage automatique, ces techniques qui nécessite un très grand nombre d'opérations mathématiques. Les puissances des processeurs graphiques actuels permettent d'effectuer du traitement complexe en temps réel. Les systèmes intelligents qui nous entourent (robotiques, systèmes de surveillance) jusque dans nos smartphones sont la preuve de cette révolution. Dans le domaine imagerie, on distingue deux grands axes principaux (la synthèse d'images et l'analyse d'images), ces deux axes sont étroitement liés.

L'analyse des images numériques est le domaine qui nous préoccupe particulièrement dans ce travail, elle présente un intérêt dans les domaines de classification automatique.

Le domaine qui nous intéresse dans ce projet est celui de l'analyse des images médicales qui représente aujourd'hui un outil efficace d'aide au diagnostic avec les modalités d'acquisition des images médicales qui sont nombreuses et la quantité d'images manipulées qui s'accroît dans le domaine médical. La radiographie en utilisant les rayons X constitue la plus grande partie dans le domaine médical, son principe consiste à impressionner sur un cliché radiographique les densités reflété d'un organe. Le cliché radiographique sera plus ou moins noirci selon l'organe. La mammographie numérique est un examen radiographique, effectué pour le dépistage du cancer du sein. Toutefois, la mammographie représente la modalité principale pour le diagnostic de ce type de cancer.

L'analyse des mammographies numériques à travers les techniques de l'intelligence artificielle permet de fournir aux médecins des informations quantitatives et fiables, ces informations complètent l'aspect visuel et qualitatif de l'examen. Le but de ce projet de fin d'étude est de proposer un système d'aide au diagnostic pour différencier entre les tissus anormales et les tissus normales dans les mammographies numériques.

Ce mémoire de fin d'étude est organisée de manière méthodique, en quatre chapitres comme suit :

Dans le 1^{er} chapitre, un état de l'art sur l'intelligence artificiel et ses axes de recherches est présenté, avec un accent particulier sur les techniques d'apprentissage automatique, à savoir, les

approches supervisées et non-supervisées ; ainsi que les avantages et les inconvénients de chaque méthode.

Dans le 2^{ème} chapitre, nous présentons les concepts de base des réseaux de neurones artificiels et à convolution, ce chapitre nous permettra de mieux comprendre les dernières technologies utilisées dans le domaine de la classification automatique.

Dans le 3^{ème} chapitre, nous présentons une revue de littérature des systèmes de détection et de diagnostic automatique des mammographies numériques, ce qui nous permettra d'étudier les approches et les outils utilisées pour la classification des anomalies dans les mammographies numériques.

Dans le quatrième et dernier chapitre nous présentons l'architecture de notre travail qui nous permettra de choisir les hyperparamètres d'un réseau de neurone à convolution ; les expérimentations de cette application seront tester sur la base d'image mammographique DDSM (The Digital Database for Screening Mammography). Un modèle optimisé pour la classification des tissus mammographiques selon leurs types (Normale / Anormale) représente un système d'aide au diagnostic appliquée sur les images mammographiques. L'approche proposée a été évaluée sur 6688 régions d'intérêt des tissus mammographiques extraites de la base de données numérique pour le dépistage (DDSM).

Enfin, la conclusion générale présente une synthèse de ce travail et les perspectives qui en découlent.

Chapitre

1

**INTELLIGENCE &
VISION ARTIFICIELLE**

I.1. Introduction

Durant la dernière décennie, l'intelligence artificielle (IA), l'apprentissage automatique (machine learning) et l'apprentissage profond (deep learning) ont fait l'objet d'un buzz médiatique intense. Ces technologies apparaissent dans d'innombrables articles et ouvrages scientifiques. A travers ces techniques plusieurs programmes sont développés pour effectuer des tâches spécifiques et qui sont utilisées pour un large éventail d'activités. Grâce à l'apparition de ces technologies on peut simplifier les tâches dans divers domaines y compris le contrôle des robots des voitures autonomes et d'assistants virtuels, la télédétection, le commerce électronique et le diagnostic médical... ect. Les tâches les plus laborieuses sont désormais effectuées par des programmes et des robots sophistiqués. Où les emplois humains seront rares puisque la plupart des activités économiques seront gérées par des robots ou par des agents IA (AI agents). Tous les secteurs sont alors touchés par ce changement : la finance, la santé, l'éducation, le transport ou encore la médecine.

Dans ce chapitre nous présentons les concepts essentiels pour comprendre les techniques de l'intelligence artificielle de l'apprentissage automatique.

I.2. L'intelligence artificielle

Le démarrage de l'intelligence artificielle à commencer à partir des années 50, quand une poignée de pionniers de l'informatique naissante ont commencé à se demander si les ordinateurs pouvaient être conçus pour « penser » ; une question dont nous continuons de la poser aujourd'hui d'explorer les ramifications. Une définition simple de l'intelligence artificielle selon Chollet [CHO21] serait la suivante : « c'est l'effort d'automatisation des tâches intellectuelles normalement effectuées par des humains ». L'IA est donc un domaine très vaste qui englobe les approches de l'apprentissage automatique et l'apprentissage profond, mais qui comprend également de nombreuses autres techniques qui n'impliquent aucun apprentissage. Par exemple, les premiers programmes d'échecs ne mettaient en œuvre que des règles codées en dur (hardcoded) élaborées par des programmeurs, et qui ne pouvaient pas être qualifiées d'apprentissage automatique. Pendant très longtemps, de nombreux experts ont cru qu'une intelligence artificielle équivalente à celle de l'homme pouvait être atteinte en faisant en sorte que les programmeurs fabriquent un ensemble suffisamment large de règles explicites pour manipuler les connaissances. Cette approche est connue sous le nom d'intelligence artificielle symbolique (symbolic AI), et ce fut le paradigme [CHO21].

La Figure I-1 présente clairement la relation entre l'intelligence artificielle (IA), l'apprentissage automatique et l'apprentissage profond et les liens qui existent entre eux.

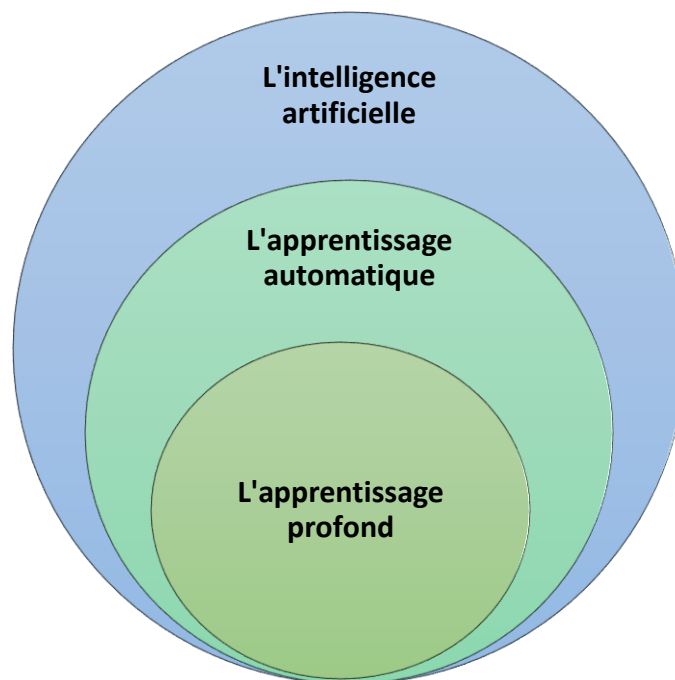


Figure I-1: Les relations entre l'intelligence artificielle, l'apprentissage automatique et l'apprentissage profond.

L'intelligence artificielle a atteint son apogée pendant le boom des systèmes experts (expert systems) pendant les années 80.

Bien que l'intelligence artificielle symbolique se soit révélée capable à résoudre des problèmes logiques bien définis, tels que jouer aux échecs, elle s'est avérée inapte de définir des règles explicites pour résoudre des problèmes plus complexes et flous, tels que la classification et la reconnaissance d'images, la reconnaissance de la parole et la traduction automatique. Une nouvelle approche est apparue, et elle a pris la place de l'intelligence artificielle symbolique : c'est l'apprentissage automatique (machine learning).

Les outils de programmation, les langages et les algorithmes sont aussi nombreux que les domaines d'application, et l'intelligence artificielle n'est pas réservée seulement à la robotique. En effet, on peut la retrouver dans quasiment tous les domaines informatisés, tels que la traduction automatique, la sécurité, reconnaissance de de parole, la reconnaissance de l'écriture et la médecine...ect. Elle nous entoure, sans même que l'on s'en rende compte, et peut améliorer quasiment tout type de logiciel. Il s'agit d'un domaine en plein essor et les capacités grandissantes des ordinateurs ont permis de mettre au point des algorithmes jusqu'alors impossibles. Sans aucun doute, l'intelligence artificielle va faire partie de notre futur et il est

donc important que tout développeur ou plus généralement informaticien comprenne les mécanismes sous-jacents pour pouvoir les appliquer. [MAT17]

I.3. L'apprentissage automatique

L'apprentissage automatique peut être assimilé à l'apprentissage d'un jeune enfant dès la naissance pour reconnaître progressivement les objets et les sons en s'entraînant et en corrigeant ses informations et en répétant ce processus.

L'apprentissage automatique est considéré le sous-domaine le plus important de l'IA. Son but principal est de permettre aux ordinateurs à apprendre par eux-mêmes sans programmer les règles de décisions pour chaque problème. L'algorithme d'apprentissage d'une machine lui permet d'identifier des modèles dans des données observées, construire des modèles qui expliquent le monde et prédire les résultats sans avoir règles et modèles préprogrammés explicites. [BIL16]

Un enfant apprend soit par la supervision directe de ses parents, soit par l'exposition directe à un événement, où il acquiert de l'expérience. [VIN03]

Les systèmes d'apprentissage automatique sont divisés en quatre catégories (Supervisé, Non supervisé, Semi-supervisé et l'apprentissage par renforcement)

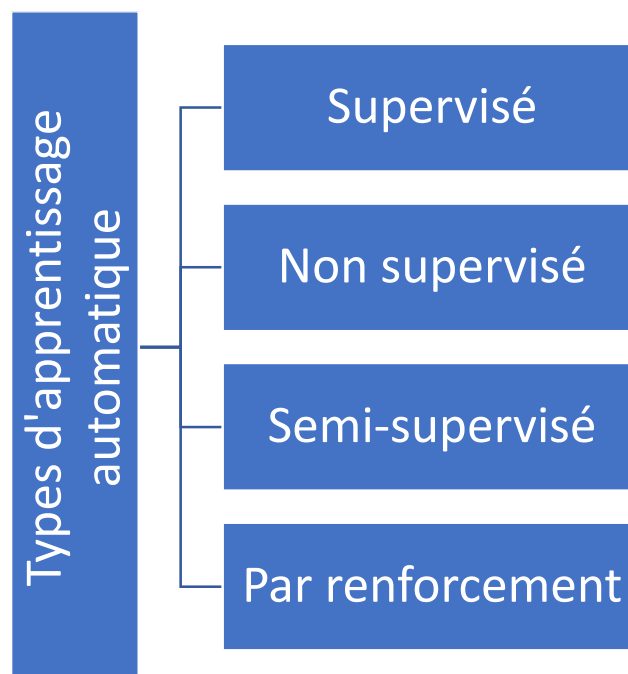


Figure I-2: Types d'apprentissage en apprentissage par machine.

Les systèmes de reconnaissance automatique sont des programmes développés en général et avec des règles pour traiter les données saisies sous toutes les formes et trouver des relations et des modèles dans les données en appliquant des équations statistiques et mathématiques - où chaque algorithme possède des caractéristiques et des sorties spécifiques , afin qu'il puisse représenter les données de différentes manières ou prédire de nouvelles sorties de données en fonction des relations et des modèles déduits des données saisies. Il existe de nombreux algorithmes d'apprentissage automatique, et nous allons passer certains d'entre eux.

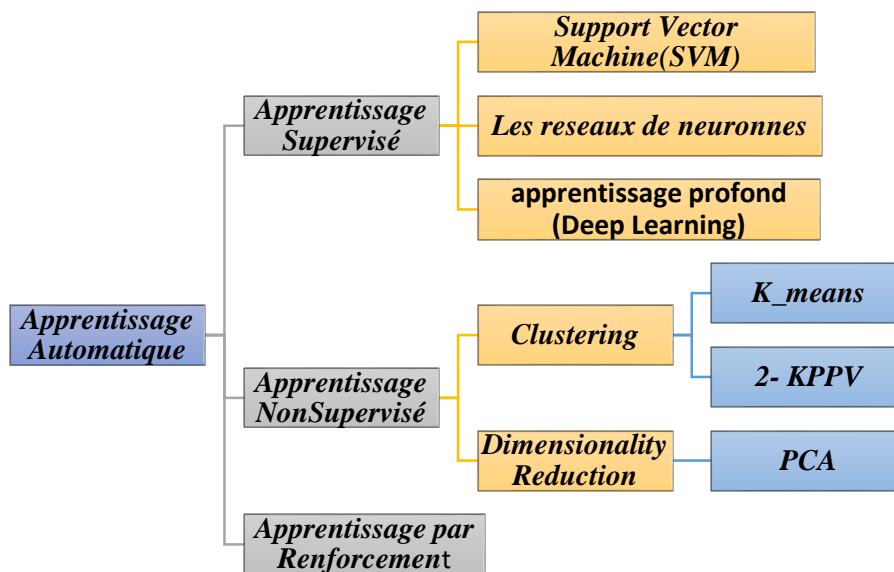


Figure I-3: Arbre représente quelques algorithmes de l'apprentissage Automatique.

I.3.1. L'apprentissage supervisé

L'apprentissage supervisé consiste à inférer la fonction d'association entre des données et leur étiquette ou valeur associée [COR11]. Cette étiquette ou valeur associée peut être numérique, dans le cas des problèmes de régression, ou discrète dans le cas des problèmes de classification. Elle consiste à apprendre comment associer les données à leur étiquette à partir d'un ensemble d'apprentissage étiqueté servant d'exemple. Cet ensemble de données d'apprentissage doit avoir été étiqueté par un expert, qui supervise l'apprentissage. L'objectif du système est ensuite de prédire l'étiquette qui doit être associée aux exemples inconnus. [BOU16].

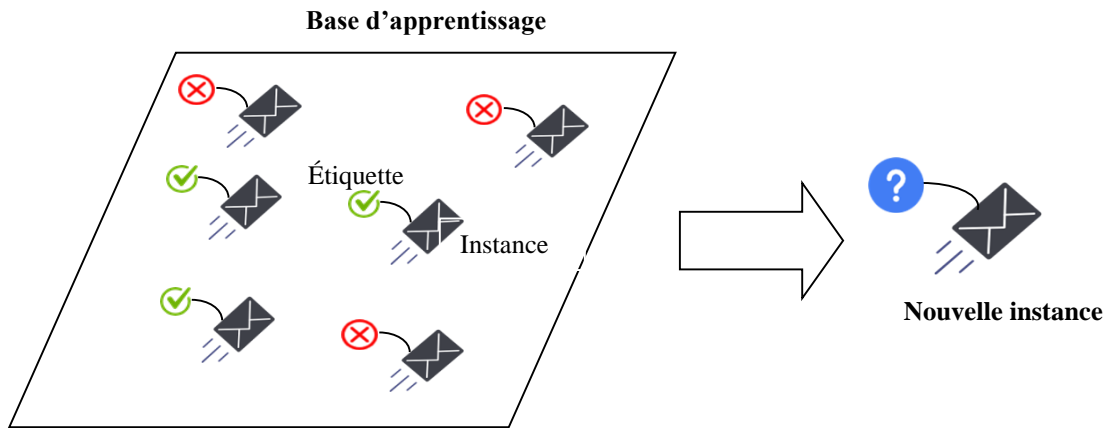


Figure I-4: Schéma d'apprentissage supervisé "supervised learning" [RUS20].

I.3.1.1. Les machines à vecteur de support SVM

La machine à vecteurs de support est un algorithme d'apprentissage automatique supervisé, les Support Vector Machine sont devenus des outils indispensables dans le domaine du Machine Learning en raison de leurs propriétés de généralisation et de leur capacité de régularisation. La machine à vecteurs de support peut être divisée en modèles linéaires et non linéaires. Il est appelé machine à vecteurs de support linéaire si le domaine de données peut être divisé linéairement (par exemple, ligne droite ou hyperplan) pour séparer les classes dans le domaine d'origine. Si le domaine de données ne peut pas être divisé linéairement, et s'il peut être transformé en un espace appelé espace de fonctionnalités où le domaine de données peut être divisé linéairement pour séparer les classes, alors il est appelé support non linéaire [KRA16].

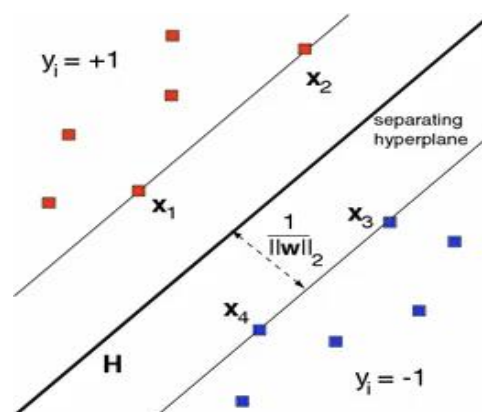


Figure I-5: exemple simple représente la classification SVM.

I.3.1.1. Les réseaux de neurones:

Ce sont des techniques de calcul conçues pour simuler la façon dont le cerveau humain exécute une tâche spécifique, grâce à un traitement massif distribué en parallèle, et composé d'unités de traitement simples. Ces unités ne sont rien d'autre que des éléments de calcul appelés neurones ou nœuds neurones, qui ont une propriété neuronale. En termes de processus, il stocke des informations connues et expérimentales pour les mettre à la disposition de l'utilisateur en ajustant les poids.

L'algorithme s'entraîne en ajustant les poids dans les nœuds et en les transmettant à d'autres couches pour l'apprentissage et la sortie des résultats.

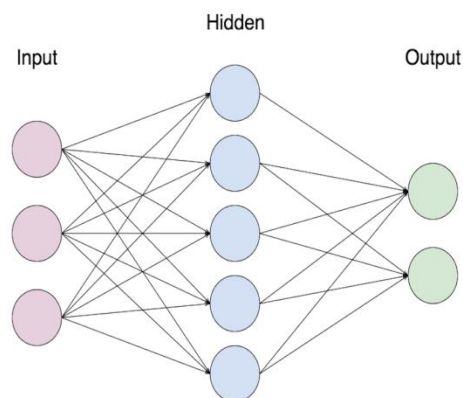


Figure I-6: schéma générale d'un réseau de neurone.

I.3.1.1. L'apprentissage profond (Deep Learning)

L'apprentissage profond est un domaine d'apprentissage automatique (ML) de pointe doté de puissantes capacités d'analyse, utilisé pour des applications telles que la vision par ordinateur, la bioinformatique et le traitement du langage naturel. Il est considéré comme démontrant des performances significatives améliorées par rapport à certains algorithmes ML et a récemment été utilisé dans différentes applications. Ces applications nécessitent des volumes substantiels de données à transférer sur le réseau, et DL offre de meilleures performances avec des données que ML. En outre, DL peut fonctionner avec de nouvelles fonctionnalités qui peuvent être utilisées pour résoudre les problèmes sans interactions humaines [OTO22].

I.3.2. Apprentissage non supervisé

L'apprentissage non-supervisé consiste à chercher une structure dans des données non étiquetées [COR11]. Comme les données ne sont pas étiquetées, elles ne sont pas rangées dans

des classes prédéfinies. Les algorithmes d'apprentissage non supervisés cherchent alors à modéliser la distribution des données. La tâche d'apprentissage non-supervisé la plus connue est le regroupement (clustering). [BOU16].

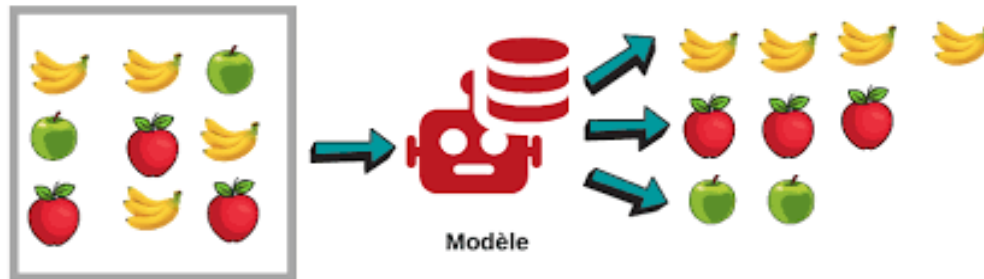


Figure I-7: Schéma d'apprentissage non supervisé "unsupervised learning".

Un médecin pourrait par exemple dans un ensemble assez large de patients de cancers du foie tenter de faire émerger des hypothèses explicatives, l'ordinateur pourrait différencier différents groupes, qu'on pourrait ensuite associer par exemple à leur provenance géographique, génétique, à l'alcoolisme ou à l'exposition à un métal lourd ou à une toxine telle que l'aflatoxine.

Dans ce type de problèmes, un algorithme va apprendre les caractéristiques des données d'entraînement (Training Set) pour produire un modèle prédictif. On retrouve deux classes de prédictions : la régression, et la classification. Un algorithme de régression pourra, par exemple, apprendre à prédire le prix d'une maison en fonction de ses caractéristiques. Par ailleurs, un algorithme de classification nous détectera si une transaction bancaire est frauduleuse ou non. La régression tout comme la classification fait partie de l'apprentissage supervisé (Supervised Learning). La régression permettra de prédire des valeurs continues (prix d'une maison, poids d'une personne...etc.). Quant à la classification, elle permet de déduire une valeur discrète (une classe) comme Spam/non Spam, tumeur maligne/bénigne etc... [TAM21]

I.3.2.1. La régression linéaire

La régression est le terme le plus fréquemment utilisé en analytique et en science des données. Il capture la relation entre la fonction cible et les fonctions indépendantes. Il est utilisé lorsque nous voulons prédire une valeur continue. La **Figure I-8** montre comment fonctionne la régression linéaire [KUL22].

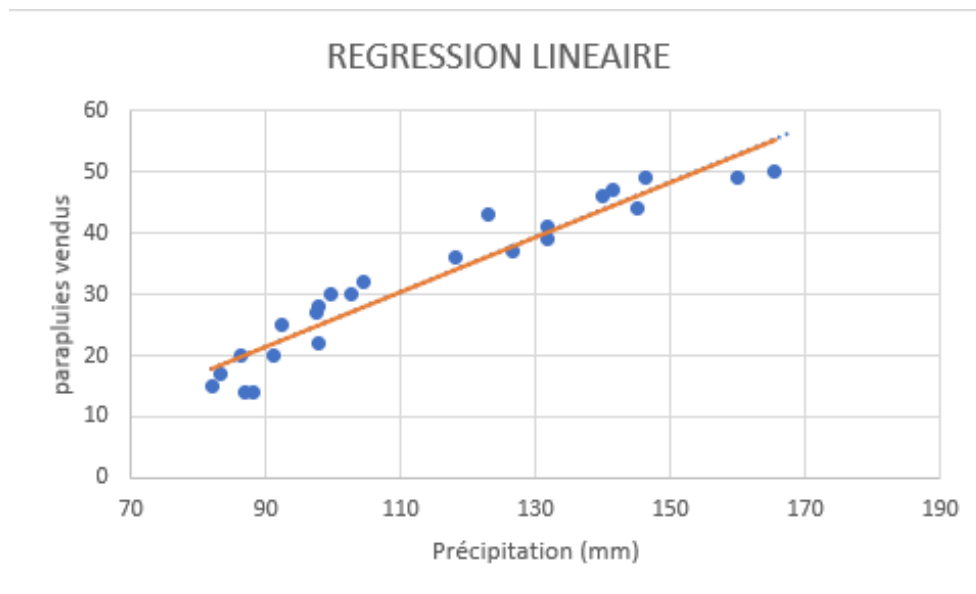


Figure I-8: la régression Linéaire.

Parmi les d'hypothèses dans la régression linéaire :

- ✓ Il devrait toujours y avoir une relation linéaire entre les variables dépendantes et les variables indépendantes.
- ✓ Les données doivent être distribuées normalement.
- ✓ Il ne doit pas y avoir de colinéarité entre les variables indépendantes.
- ✓ La multi colinéarité fait référence à une forte relation linéaire entre des variables indépendantes. Ces variables corrélées agissent comme redondantes et doivent être traitées.
- ✓ Il y a homoscédasticité, ce qui signifie que la variance des erreurs doit être constante dans le temps.
- ✓ La variance ne doit pas être supérieure pour les valeurs de sortie supérieures et inférieure pour les valeurs de sortie inférieures.
- ✓ Les termes d'erreur ne doivent avoir aucune corrélation entre eux.
- ✓ Une fois que vous avez créé le modèle et vous souhaitez connaître ses performances. Pour ce faire, vous pouvez utiliser des métriques. Le R au carré (R^2) est la mesure la plus largement utilisée pour calculer la précision de la régression linéaire. La valeur R^2 montre comment la variance des variables dépendantes est expliquée par des variables indépendantes. Le R au carré varie de 0 à 1.
- ✓ L'erreur quadratique moyenne (RMSE) indique la mesure de la dispersion des valeurs prédites par rapport aux valeurs réelles.

Les avantages :

- ✓ L'un des algorithmes les plus connus en statistiques et en apprentissage automatique.
- ✓ Technologie simple et rapide et bon algorithme initial.
- ✓ Facile à utiliser car il nécessite un réglage minimal.

Les inconvénients :

- ✓ Limité aux relations linéaires.
- ✓ Sensible aux valeurs aberrantes.
- ✓ Parce qu'il est simple, parfois, son incapacité à susciter des relations dans certains cas complexes.
- ✓ Les données doivent être indépendantes.
- ✓ Capable de prédire la sortie numérique.

I.3.2.2. L'algorithme des k plus proches voisins

L'algorithme des k plus proches voisins (KPPV) est une méthode d'apprentissage à base d'exemples. Il ne comporte pas de phase d'apprentissage en tant que telle. Les données faisant partie de l'ensemble d'entraînement sont seulement emmagasinés. Lorsqu'un nouvel exemple à classer arrive, il est comparé aux exemples (prototypes) d'entraînement à l'aide d'une mesure de similarité. Bien que la phase d'apprentissage est inexistante puisque les données sont stockées telles quelles en mémoire, la classification d'un nouveau cas est par contre coûteuse puisqu'il faut comparer ce cas à tous les exemples déjà classés. En plus un KPPV de base utilise toutes les variables descriptives d'un exemple pour calculer la similarité avec un nouvel exemple à classer. Hors dans des bases de données à grandes dimensions ce qui est le cas en data mining, les variables discriminantes sont considérées au même titre que les autres (les moins discriminantes). Pour remédier à ce problème les variables sont pondérées suivant leur pertinence. Afin de choisir ces poids, la méthode de validation croisée peut être utilisée [MAT10]. Les performances de la méthode dépendent du nombre de voisins, du choix de la distance, et du mode de combinaison des réponses des voisins [ZER15].

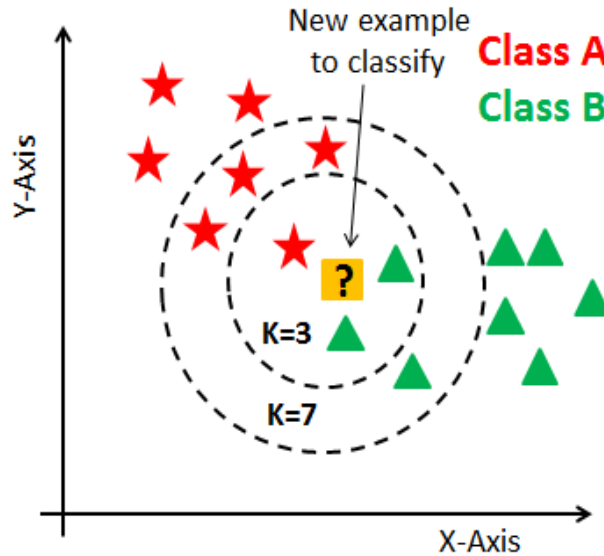


Figure I-9: Exemple de K-plus proche voisins (KPPV).

Cette approche est très simple et directe. Elle ne nécessite pas d'apprentissage mais simplement le stockage des données d'apprentissage. Son principe est le suivant. Une donnée de classe inconnue est comparée à toutes les données stockées. On choisit pour la nouvelle donnée la classe majoritaire parmi ses K plus proches voisins (Elle peut donc être lourde pour des grandes bases de données) au sens d'une distance choisie. [CHA13]. Afin de trouver les K plus proches d'une donnée à classer, on peut choisir la distance euclidienne. Soient deux données représentées par deux vecteurs x_i et x_j , la distance entre ces deux données est donnée par :

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2}$$

I.3.2.3. Le clustering K-moyenne

L'approche K-moyenne proposé par MacQueen en 1967 est la méthode non supervisée la plus couramment utilisée. Algorithme d'apprentissage automatique pour partitionner un ensemble de données donné en un ensemble de k groupes (c'est-à-dire k clusters), où k représente le nombre de groupes prédéfinis par l'analyste. Cette approche classe les objets dans plusieurs groupes (c'est-à-dire des clusters), de sorte que les objets d'un même cluster sont aussi similaires que possible (c'est-à-dire une forte similarité intra-classe), alors que les objets provenant de grappes différentes sont aussi dissemblables que possible (c'est-à-dire, faible similarité interclasse).

Dans le clustering k-moyenne, chaque cluster est représenté par son centre (c'est-à-dire le centroïde) qui correspond à la moyenne des points attribués au cluster.

I.3.3. Apprentissage semi-supervisé

L'apprentissage semi-supervisé (SSL) se situe entre l'apprentissage non supervisé et l'apprentissage supervisé et combine une petite quantité de données étiquetées avec une plus grande quantité de données non étiquetées pendant l'entraînement [OUA20]. Plus formellement, l'objectif de SSL est d'exploiter les données non étiquetées (Unlabeled Data) pour produire une fonction de prédiction $f(\theta)$ avec des paramètres entraînaables (θ), qui est plus précis que ce qui aurait été obtenu en utilisant uniquement les données étiquetées (Labeled Data). Par exemple, Du peut fournir des informations supplémentaires sur la structure de la distribution des données ($p(x)$) pour mieux estimer la frontière de décision entre les différentes classes pour améliorer le modèle de connaissance du classifieur [CHE20] [SCH22].

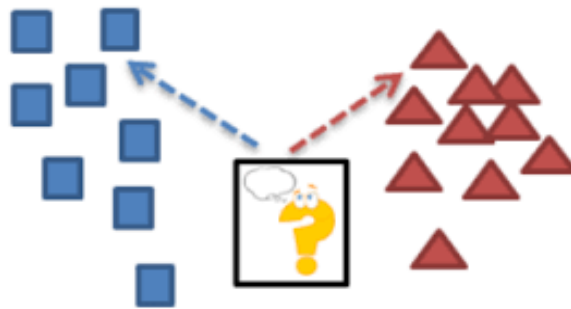


Figure I-10: Schéma d'apprentissage semi supervisé "semi supervised learning"

I.3.3.1. Apprentissage par renforcement

L'apprentissage par renforcement est un autre type de système d'apprentissage automatique. Un agent dans un "système d'IA" observera l'environnement, effectuera des actions données, puis recevoir des récompenses en retour. Avec ce type, l'agent doit apprendre par lui-même. Liens appelé une politique. Nous pouvons trouver ce type d'apprentissage dans des nombreuses applications robotiques qui apprennent comment faire des réactions dans son environnement [RUS20].

L'apprentissage par renforcement est le problème auquel est confronté un agent qui doit apprendre son comportement par tâtonnement c'est-à-dire interactions par essais et erreurs (trial-and-error) dans un environnement dynamique. L'agent est connecté à l'environnement par la perception et l'action. A chaque pas de l'interaction, l'agent reçoit en entrée une

perception « p » de l'état courant, et génère une action « a » comme sortie qui impacte l'environnement. La valeur de cette transition est communiquée à l'agent par un signal de renforcement ou récompense « r », comme illustré par la Figure I-11. Le comportement de l'agent (ou politique) doit permettre de choisir les actions qui vont maximiser la somme des valeurs des récompenses sur un long terme. La politique fait correspondre les états aux actions. Formellement le modèle est représenté par le triplet : (S, A, R) avec S un ensemble fini d'états, A un ensemble fini d'actions et R une fonction de renforcement ; Où l'environnement est non déterministe : pour une action exécutée deux fois dans un même état, l'état résultant ou la récompense peuvent être différents. [MAZ15].

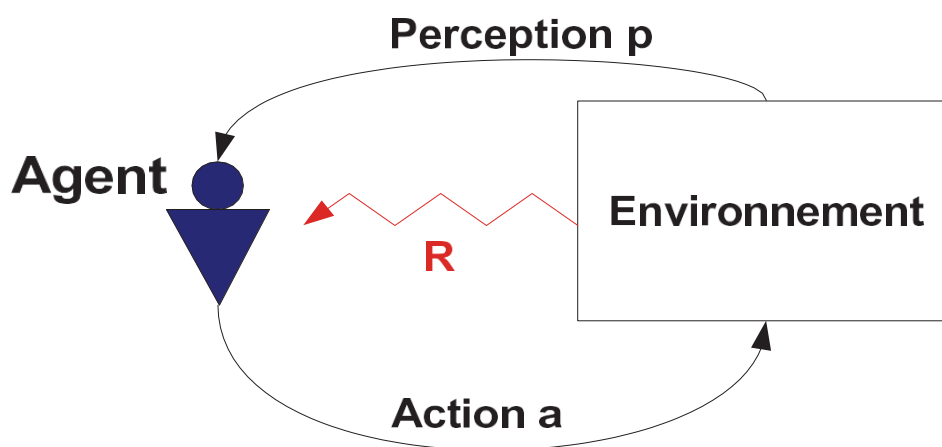


Figure I-11: Schéma d'apprentissage par renforcement "reinforcement learning" [MAZ15].

I.4. Domaines d'utilisation de l'apprentissage automatique

L'apprentissage automatique dans nos jours s'utilise dans plusieurs domaines parmi eux [AHU20] :

- Distinguer et reconnaître des objets (visages, images, lettres, mots, textes avec images, sons, musique).
- Aider à trouver des médicaments qui aident à guérir les maladies incurables.
- Moteurs de recherche, sécurité de l'information et marketing.
- Répondre aux questions (via une banque de données).
- Prendre des décisions et des recommandations en simulant le réseau de neurones du cerveau humain.
- Contribuer au développement d'un robot pour effectuer divers travaux, notamment des travaux difficiles ou délicats, dans des conditions difficiles pour l'homme ...ect.

I.5. Conclusion

L'intelligence Artificielle un domaine de recherche actif s'évoluant de plus en plus, où la classification des images constitue l'outil principal de la vision par machine.

Dans le domaine de l'intelligence artificielle, plus la description effectuée est proche de l'interprétation humaine, plus elle est considérée comme robuste et fidèle à la scène initiale. Les méthodes de classification dépendent de l'objectif visé (description globale ou locale) et du type de l'objet à analyser (caractère, son, visage, image médicale ...etc.). Dans ce chapitre nous avons les principes de l'intelligence artificielle et l'apprentissage automatique. Dans le chapitre suivant, nous présentons en détail les approches basées sur les réseaux de neurones artificiels (ANN) et les réseaux de neurones à convolution CNN en anglais (convolutional neural network).

Chapitre

2

Les réseaux de neurones artificiels et à convolution

II.1. Introduction

Les réseaux de neurones artificiels et à convolution sont des méthodes d'apprentissage automatique supervisé. Le neurone formel est à l'origine créé pour modéliser les neurones biologiques et imiter par analogie leurs capacités. Néanmoins, ces premiers modèles ne sont même pas capables d'apprendre et il faudra attendre l'apparition du perceptron pour cela. De plus, il faudra attendre plusieurs innovations telles que la rétro propagation du gradient et les modèles multicouches, qui sont couplés avec le développement des capacités des outils informatiques ; cette révolution informatique a permis la réalisation des premières utilisations pratiques des réseaux de neurones artificiel.

II.2. Neurone biologique

Un réseau de neurone artificiel est un ensemble interconnecté d'éléments de traitement simple dont la fonctionnalité s'inspire du neurone biologique. Le cerveau humain est constitué d'environ 100 milliards de cellules nerveuses ou neurones (Figure II-1). Les neurones communiquent par des signaux électriques et les connections entre elles sont assurées par des jonctions électrochimiques appelées synapse, située sur des branches de la cellule appelée dendrites. Chaque neurone reçoit généralement plusieurs milliers de connexions avec d'autres neurones et reçoit ainsi en permanence une multitude de signaux entrants qui finissent par atteindre le corps cellulaire. Si le signal résultant dépasse un certain seuil, le neurone se déclenche ou génère une impulsion de tension en réponse. Celle-ci est ensuite transmise à d'autres neurones par une fibre ramifiée appelée axone [GUR18].

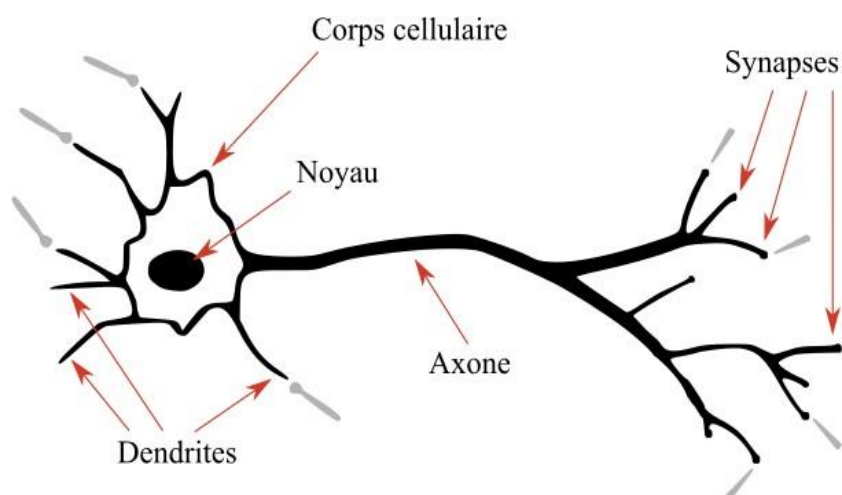


Figure II-1: Neurone biologique.

II.3. Neurone Artificiel

Un neurone artificiel est un système ayant plusieurs entrées et une seule sortie. Chaque neurone possède deux modes (le mode d'entraînement et le mode de fonctionnement). Dans le mode d'entraînement, le neurone apprend à être déclenché contre des signaux d'entrée spécifiques ou à se déclencher à un terme. En mode de fonctionnement, une sortie correspondante est fournie lorsque les informations fournis à l'entrée sont reconnus. Si l'entrée ne fait pas partie des entrées pré-identifiées, les règles fixés après la phase d'apprentissage décident de la déclencher ou non. En omettant certaines des propriétés critiques des neurones et leurs interconnexions [GUO20]. Un modèle élémentaire du neurone artificiel typique peut être simulé par un ordinateur.

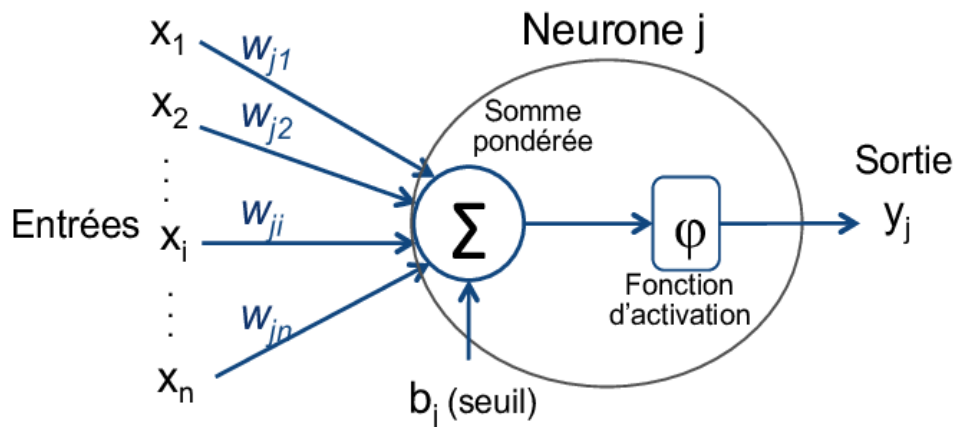


Figure II-2: Neurone artificiel.

Un neurone est composé d'une couche d'entrée acceptant des variables d'entrées $x_i \in R$, d'un biais b_i , de poids w_i , d'une fonction d'activation σ , et d'une ou plusieurs valeurs de sortie y_i de sortie. On a ainsi la formule suivante :

$$y = \sigma \left(w_0 + \sum_{j=0}^p w_j x_j \right)$$

II.3.1. Structure des réseaux de neurones artificiels

Comme le montre la Figure II-3, un réseau neurone artificiel est constitué des composants de couches et de poids de connexion. Le comportement du réseau dépend également de la communication entre les membres. Il existe généralement trois types de couches neuronales:

1. **La couche d'entrée** : cette couche reçoit les données brutes introduites dans le réseau.

2. **Les Couches cachées** : La performance de ces couches est déterminée par les entrées et les poids de la connexion entre elles et les couches cachées. Les poids entre les neurones d'entrée et cachées détermine quand un neurone de cette couche doit être activée.

3. **La couche de sortie** : la sortie de l'unité de sortie dépend de l'activité de la couche cachée et des poids de la connexion entre l'unité cachée et la sortie.

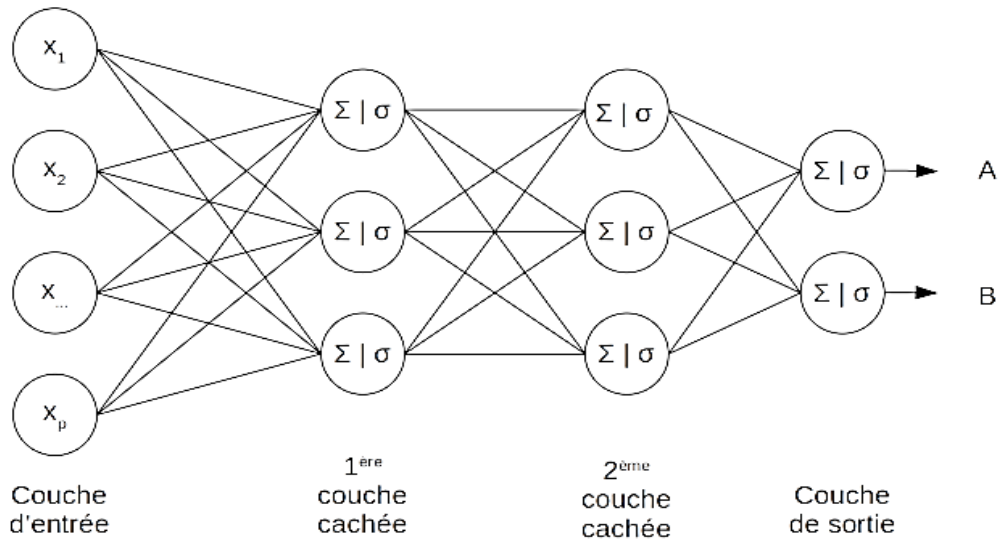


Figure II-3: Schéma simplifié d'un perceptron multicouche comportant deux couches cachées et effectuant une classification binaire [VIA21].

Il existe également des réseaux monocouches et multicouches ; dans l'organisation monocouche toutes les neurones sont connectées à une couche, ce type de réseaux neuronaux est le plus utilisé et possède plus de potentiel de calcul que les organisations multicouches. Dans les réseaux multicouches, les neurones sont numérotés par couches (au lieu de suivre la numérotation globale). Les deux couches d'un réseau sont reliées par des poids et en fait des connexions [JOU18]. Dans les réseaux de neurones, il existe plusieurs types de connectivité ou gain de poids :

- Anticipation: la plupart des connexions sont des types dans lesquelles les signaux se déplacent dans une seule direction. De l'entrée à la sortie, dans ce cas il n'existe pas de retour (boucle). La sortie de chaque couche n'affecte pas la même couche.
- En arrière: les données sont renvoyées des nœuds de la couche supérieure vers la couche inférieure nœuds de couche.
- Côté: les sorties des nœuds de chaque couche sont utilisées comme entrées pour les nœuds de la même couche.

II.3.2. Types des réseaux de neurones artificiels

Voici les types de réseaux de neurones les plus connus (Figure II-4) :

1. Réseau de neurones Perceptron multicouche (MLPNN).
2. Réseaux de neurones à fonction de base radiale (RBFNN).
3. Réseau de neurones Hopfield (HN).
1. Réseau de neurones de Hamming (HNN).
2. Réseau neuronal cartographique auto-organisé de Kohonen (KSOMNN).
3. Réseau de neurones à retardement (TDNN).
4. Réseaux de neurones à alimentation profonde (DFNN).
5. Réseaux de neurones récurrents (RNN).
6. Réseaux de neurones à mémoire à long et à court terme (NNLSTM).
7. Réseau neuronal des encodeurs automatiques (AENN).
8. Réseaux de chaînes de Markov (MCN).

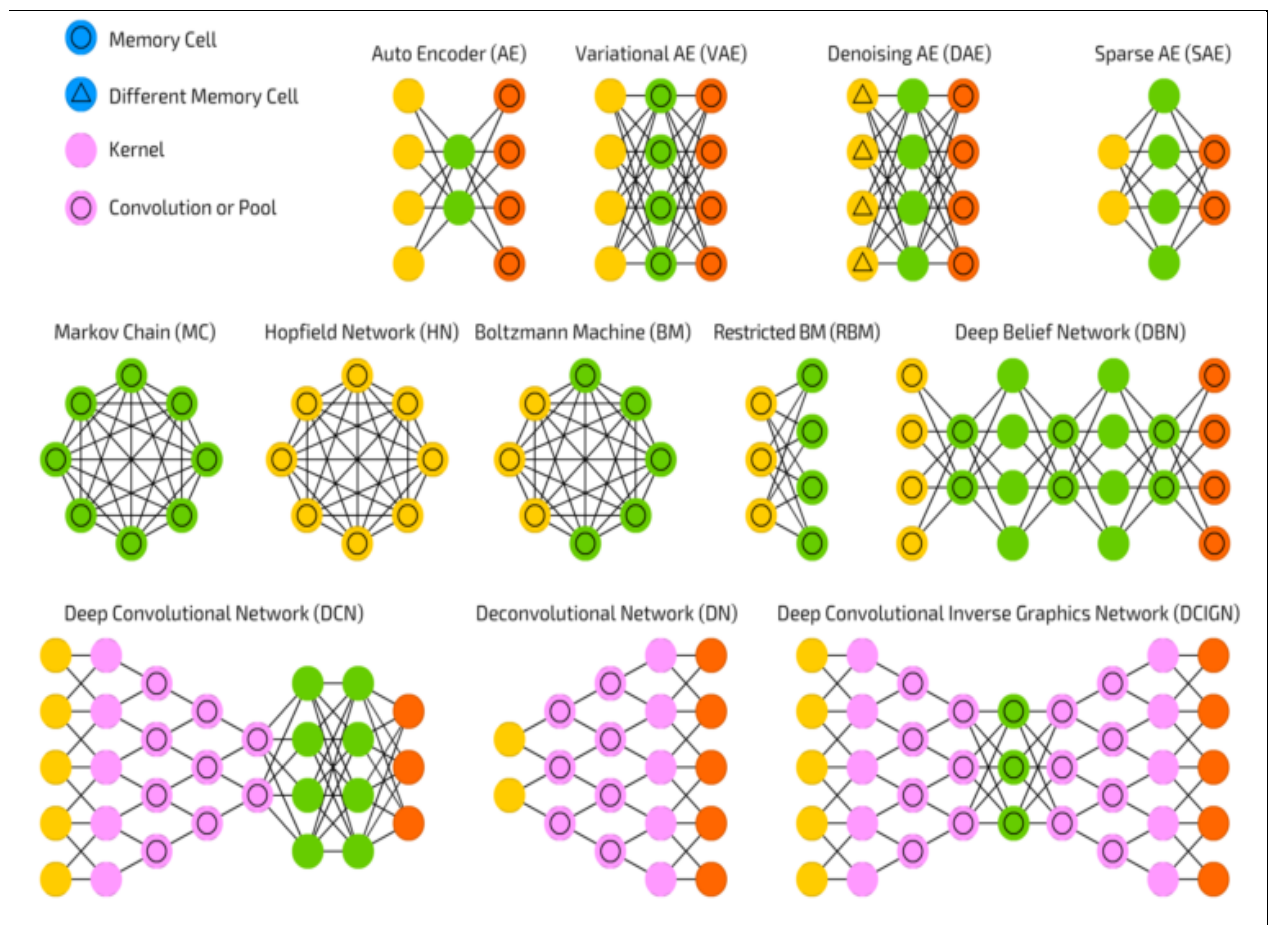


Figure II-4: Les types de réseaux de neurones les plus connus.

II.3.3. L'initialisation des poids

L'objectif de l'initialisation des poids dans les réseaux de neurones artificiels est d'empêcher les sorties d'activation de couche d'exploser ou disparaître pendant la phase d'apprentissage. L'apprentissage du réseau sans une initialisation de poids peut conduire à une convergence très lente ou à une incapacité à converger [DEN20].

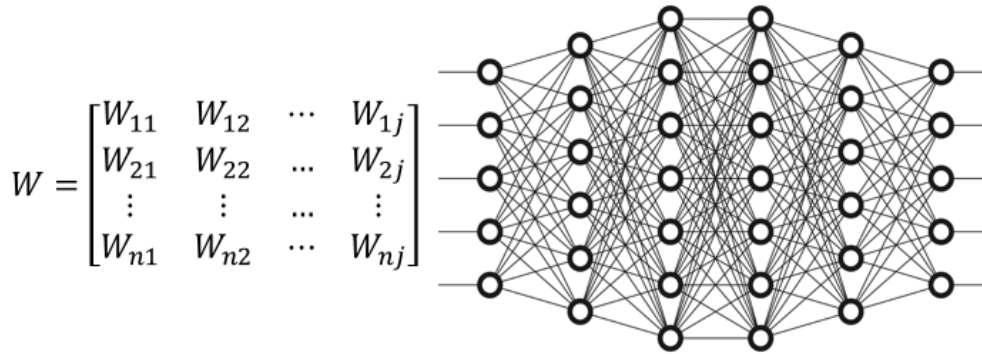


Figure II-5: Schéma représente le processus d'initialisation des poids [BOU22].

Les techniques d'initialisation des poids [LIH20] les plus utilisées sont les suivantes:

II.3.3.1. Initialisation tous zéros et initialisation constante

Cette méthode définit tous les poids sur zéros (respectivement à constante). De plus, toutes les activations dans tous les neurones sont les mêmes, et donc tous les calculs sont les mêmes, ce qui fait que ce type d'initialisation concerne les modèles linéaires.

II.3.3.2. Initialisation aléatoire

Cette technique améliore le processus de brisure de symétrie et offre une plus grande précision. Les poids sont initialisés très près de zéro et aléatoirement. Cette méthode empêche l'apprentissage de la même fonctionnalité pour l'entrée paramètres.

II.3.3.3. Initialisation LeCun (normaliser la variance)

L'initialisation LeCun vise à empêcher la disparition ou l'explosion des gradients lors de la rétro-propagation en résolvant la variance croissante avec le nombre d'entrées et en fixant une variance constante.

II.3.3.4. Initialisation de Xavier (initialisation de Glorot)

Xavier a proposé une méthode plus simple, où les poids tels que la variance des activations sont le même sur chaque couche. Cela empêchera le dégradé d'exploser ou disparaître.

II.3.3.5. Initialisation HE (initialisation de Kaiming)

Cette initialisation préserve non linéarité des fonctions d'activation telles que les activations ReLU. L'initialisation des poids avec la méthode **HE** peut réduire ou amplifier les amplitudes des entrées de manière exponentielle.

Dans le Tableau II-1, une comparaison entre les techniques d'initialisation de poids est présentée pour décrire les principales techniques d'initialisation de poids en fonction de leurs avantages et inconvénients.

Méthode d'initialisation	Avantages	Inconvénients
Tout-zéros/constante	✓ Simplicité	❖ Problème de symétrie amenant les neurones à apprendre la même chose
Aléatoire (Random)	✓ Améliore le processus de rupture de symétrie	❖ Une saturation peut se produire conduisant à un gradient de fuite. ❖ La pente ou le gradient est faible, ce qui peut ralentir la descente du gradient.
LeCun	✓ Résout la variance croissante et problèmes de gradient	❖ Pas utile en largeur constante réseaux. ❖ Tient compte de la propagation vers l'avant du signal d'entrée. ❖ Cette méthode n'est pas utile lorsque la fonction d'activation est non différentiable.
Xavier	✓ Diminue la probabilité du problème de disparition/explosion du gradient.	❖ Cette méthode n'est pas utile lorsque la fonction d'activation est non différentiable. ❖ Problème de neurone mourant pendant l'entraînement.
HE	✓ Résout les problèmes de neurones mourants.	❖ Cette méthode n'est pas utile pour les couches avec les fonctions d'activations différentiable telles que ReLU ou LeakyReLU.

Tableau II-1: Comparaison entre les techniques d'initialisation du poids.

II.3.4. Les fonctions d'activations

Les fonctions peuvent être de différents types en fonction du problème ou des données. Il existe deux types de fonctions d'activations (linéaire et non-linéaire), dans le mode d'activation linéaire un neurone linéaire prend une combinaison linéaire des entrées pondérées ; et la sortie peut prendre n'importe quelle valeur entre infini et infini. Dans le mode d'activation non-linéaire est le mode d'activation le plus utilisé, où la sortie se limite entre une certaine gamme, les fonctions d'activation les plus connues sont les suivantes:

II.3.4.1. La fonction d'activation sigmoïde ou logit

La fonction d'activation sigmoïde permet de réduire la sortie entre 0 et 1 en appliquant la fonction *log*, qui rend les problèmes de classification plus facile [MON12].

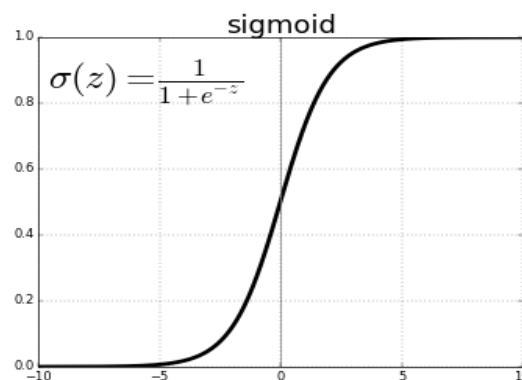


Figure II-6: Schéma représente la fonction d'activation Sigmoid.

II.3.4.2. La fonction Softmax

La fonction Softmax est presque similaire à sigmoïde, mais il calcule les probabilités des événements sur 'n' classes différentes, ce qui sera utile pour déterminer la cible dans la classification multi-classes.

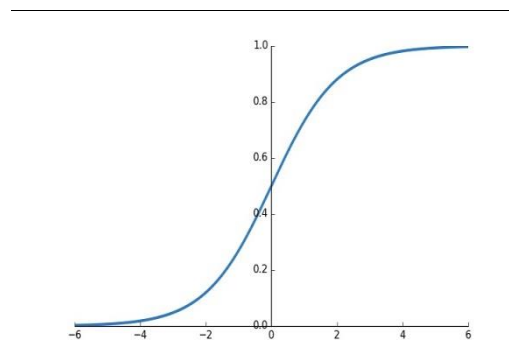


Figure II-7: Schéma représente la fonction d'activation Softmax.

II.3.4.3. La fonction Tanh

La plage de la fonction d'activation tanh est de (-1 à 1) [MON12] comme le montre la **Figure II-8**.

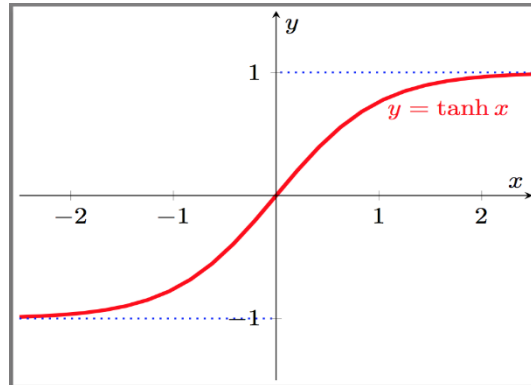


Figure II-8: Schéma représente la fonction d'activation Tanh.

II.3.4.4. La fonction de l'Unité Linéaire Rectifiée (ReLU)

La fonction de l'unité linéaire rectifiée (ReLU) [NAI10] est l'une des fonctions d'activation non saturées les plus remarquables. Elle est définie comme suit : $a_{i,j,k} = \max(z_{i,j,k}, 0)$.

ReLU convertit tout ce qui est inférieur à zéro en zéro. Alors, la plage devient de 0 à l'infini.

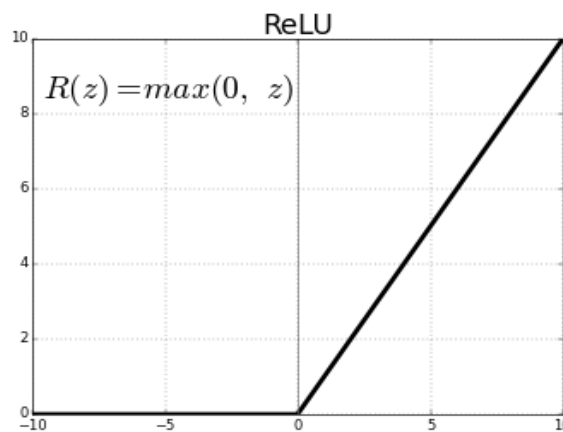


Figure II-9: Schéma représente la fonction d'activation Relu.

II.3.4.5. La fonction LeakyReLU

Un inconvénient potentiel de l'unité ReLU est qu'elle a un gradient nul chaque fois que le neurone n'est pas activé. Cela peut faire en sorte que les neurones qui ne sont pas activés initialement ne soient jamais actifs car l'optimisation basée sur le gradient n'ajustera pas leurs poids. En outre, cela peut ralentir le processus d'apprentissage en raison des gradients zéro

constants. Pour pallier ce problème, Mass et al. [MAA13]. Introduire LeakyReLU (LReLU) qui est défini comme :

$$a_{i,j,k} = \max(z_{i,j,k}, 0) + \lambda \min(z_{i,j,k}, 0)$$

Où λ est un paramètre prédéfini dans la plage (0, 1). Par rapport à ReLU, Leaky ReLU comprime la partie négative plutôt que de la mapper sur un zéro constant, ce qui permet un petit gradient non nul lorsque le neurone n'est pas activé.

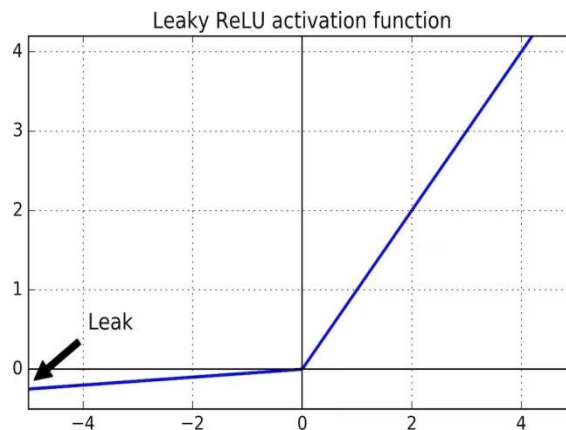


Figure II-10: Schéma représente la fonction d'activation LeakyRelu

II.3.4.6. La fonction ReLU paramétrique (PReLU)

Plutôt que d'utiliser un paramètre prédéfini λ dans LeakyReLU, par exemple. La fonction d'activation de l'unité linéaire rectifiée paramétrique (PReLU) apprend de manière adaptative les paramètres des redresseurs afin d'améliorer la précision [AGG22]. Mathématiquement, la fonction PReLU est définie comme :

$$a_{i,j,k} = \max(z_{i,j,k}, 0) + \lambda_k \min(z_{i,j,k}, 0)$$

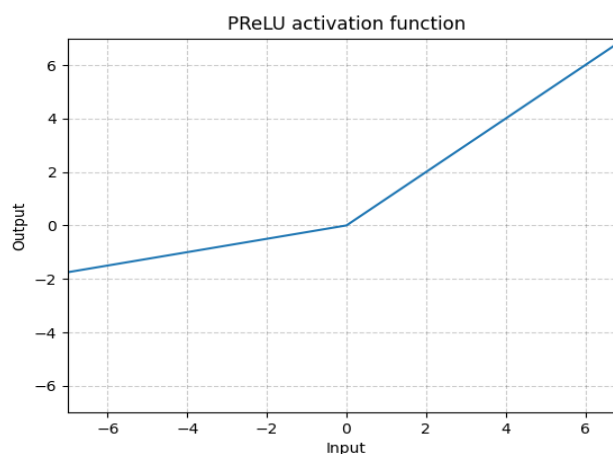


Figure II-11: Schéma représente la fonction d'activation PRelu.

Où λ_k est le paramètre appris pour le k-ième canal. Comme PReLU n'introduit qu'un très petit nombre de paramètres supplémentaires, par exemple, le nombre de paramètres supplémentaires est le même que le nombre de canaux du réseau, il n'y a pas de risque supplémentaire de sur-apprentissage et le coût de calcul supplémentaire est négligeable. Il peut également être formé simultanément avec d'autres paramètres par rétro-propagation.

II.3.4.7. Fonction d'activation SWISH

La fonction d'activation automatique (SWISH) [WAN20] est en fait une version de la fonction sigmoïde, ça fonction et est définie comme suit:

$$f(x) = x * \sigma(x)$$

Où $\sigma(x)$ est le sigmoïde de x donné par : $\sigma(x) = \frac{1}{1+e^{-x}}$

La fonction d'activation SWISH est une fonction non monotone (Figure II-12).

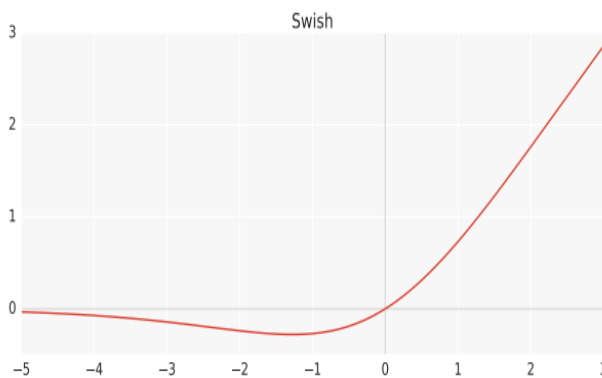


Figure II-12: Schéma représente la fonction d'activation SWICH.

II.3.5. Les fonctions de pertes (erreurs)

La fonction de perte est la métrique qui aide un réseau à comprendre s'il apprend dans la bonne direction. Pour encadrer la fonction de perte en termes simples, considérez-la comme le score que vous obtenez lors d'un examen. Supposons qu'un étudiant veut participer à plusieurs tests sur le même sujet : quelle métrique utilisera l'étudiant pour comprendre ses performances à chaque test ? De toute évidence, le score du test. Supposons qu'il a obtenu 11, 12, 15, 17 et 19 sur 20 dans cinq tests de langue consécutifs.

Il est clairement que l'amélioration des résultats des tests est une indication de ses performances. Si les résultats des tests avaient diminué, le verdict serait que ses performances diminuent et qu'il aura besoin de changer ses méthodes d'étude ou son matériel pour qu'il s'améliore.

De même, comment un réseau comprend-il s'il s'améliore son processus d'apprentissage à chaque itération?, Il utilise la fonction de perte, qui est analogue au score du test. La fonction de perte mesure essentiellement la perte de la cible. Supposons que nous développons un modèle pour prédire si un élève réussira ou échouera et que la probabilité de réussite ou d'échec est définie par la probabilité. Perte d'entropie croisée. Ainsi, 1 indiquerait qu'il réussira avec une certitude de 100 % et 0 indiquerait qu'il échouera définitivement.

Le modèle apprend des données et prédit un score de 0,87 pour l'étudiant à passer. Ainsi, la perte réelle ici serait de $1,00 - 0,87 = 0,13$. S'il répète l'exercice avec quelques mises à jour de paramètres afin de s'améliorer et obtient maintenant une perte de 0,40, il comprendrait que les changements qu'il a apportés n'aident pas le réseau à apprendre correctement. Alternativement, une nouvelle perte de 0,05 indiquerait que les mises à jour ou les changements de l'apprentissage vont dans la bonne direction.

En fonction du type de résultat de données, nous avons plusieurs pertes standard fonctions définies dans l'apprentissage par ordinateur. Pour les cas d'utilisation de régression (c'est-à-dire, où la prédiction finale serait un nombre continu comme les notes obtenues par un étudiant, le nombre d'unités de produits vendues par un magasin, le nombre d'appels reçus de clients dans un centre de contact, etc.), voici quelques fonctions de perte populaires disponibles [MOO19].

II.3.5.1. Erreur Quadratique Moyenne

Cette erreur est définie par la différence quadratique moyenne entre la valeur réelle et prédite. La différence au carré permet de pénaliser plus facilement le modèle pour une différence plus élevée. Ainsi, une différence de 3 entraînerait une perte de 9, mais une différence de 9 entraînerait une perte de 81. L'équivalent mathématique est défini par la fonction suivante:

$$\sum_{n=1}^k \frac{(\text{Actuelle} - \text{prédite})^2}{K}$$

II.3.5.2. Erreur Absolue Moyenne

L'erreur absolue moyenne entre le réel et le prévu. L'équivalent mathématique serait :

$$\sum_{n=1}^k |\text{Actuelle} - \text{prédite}|$$

De même, peu d'autres variantes sont :

- MAPE : Erreur absolue moyenne en pourcentage (**M**ean **A**bsolute **P**ercentage **E**rror).
- MSLE : Erreur logarithmique quadratique moyenne (**M**ean **S**quare **L**ogarithmic **E**rror).

Pour les résultats catégoriels, la prédiction serait pour une classe, comme si un étudiant réussira (1) ou échouera (0), si le client réussira un achat ou non, que le client soit en défaut de paiement ou non, etc.

Certains cas d'utilisation peuvent avoir plusieurs classes comme résultat, comme classer les types de maladies (type A, B ou C), classer les images en tant que chats, chiens, voitures, chevaux, paysages, etc. Dans de tels cas, les pertes définies ci-dessus ne peuvent être utilisées en raison à des raisons évidentes. Nous aurions besoin de quantifier le résultat de la classe comme probabilité et définir les pertes sur la base des estimations de probabilité comme prédictions.

II.3.5.3. Entropie croisée binaire

Définit la perte lorsque résultats catégoriels est une variable binaire, c'est-à-dire avec deux classes possibles : (Réussite / Échec) ou (Oui / Non), La forme mathématique de la fonction de l'entropie croisée binaire est la suivante :

$$Perte = - [y * \log(p) + (1-y) * \log(1-p)]$$

II.3.5.4. Entropie croisée catégorielle

Définit la perte lorsque les résultats catégoriques sont non binaires, c'est-à-dire le nombre de classes possibles est supérieur à 2, exemple : classes (Oui/Non/Peut-être) ou (Type 1/ Type 2/...Type n), la forme mathématique de l'entropie croisée catégorielle serait :

$$Perte = \sum_i y'_i \log_2(y_i).$$

II.3.6. Les Optimiseurs

Pour trouver un minimum local d'une équation différentielle, il est nécessaire d'utiliser un algorithme d'optimisation itératif. La théorie consiste à effectuer des pas répétés dans la direction opposée du gradient au stade actuel (ou un gradient approximatif) de la fonction car cela est considéré comme la direction de descente vers l'erreur minimale [CHA19]. D'un autre côté, un pas dans la direction du gradient se traduira par un maximum local de cette caractéristique. Ensuite, l'approche est appelée ascension de gradient [ZEE21].

II.3.6.1. La descente de gradient (GD)

Il existe trois variantes de descente de gradient, qui varient dans la quantité de données que nous utilisons pour évaluer le gradient de la fonction objectif [HAR16]. En fonction

de la quantité de données [RUD16], nous avons un compromis entre l'exactitude du paramètre de mise à jour et le temps d'exécution d'une mise à jour.

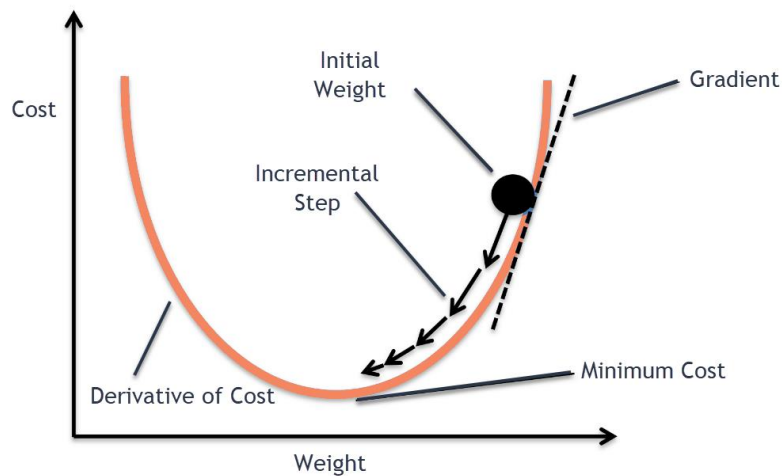


Figure II-13: Algorithme de descente de gradient [HAJ21].

II.3.6.2. Descente de gradient par lots (BGD)

La descente de gradient par lots, parfois nommé vanilla GD, détecte l'erreur pour chaque individu de la base d'apprentissage. Mais les poids du modèle sont révisés après que tous les individus de cette base ont été évalués. Tout ce processus est comme une boucle, et c'est ce qu'on appelle une époque d'entraînement [DOG18]. Certains des avantages de l'optimisation avec descente de gradient par lots sont son calcul efficace, un gradient d'erreur équilibré et une convergence stable [SIZ19]. Certaines faiblesses sont que le gradient d'erreur constant peut souvent conduire à une condition de convergence qui n'est pas la meilleure que le modèle puisse accomplir. Cela exige également que les éléments de la base d'apprentissage soit en mémoire et disponible pour l'algorithme [YID19]. BGD offre les performances les plus précises mais nécessite de nombreuses analyses complètes coûteuses des données réelles [KAO17].

L'optimisation par la descente de gradient par lots calcule le gradient de la fonction d'erreur [SAE20] concernant les paramètres θ pour la base d'apprentissage, comme le montre l'équation suivante :

$$\theta_{\text{nouveau}} = \theta_{\text{ancien}} - \eta \cdot \Delta_{\theta} J(\theta)$$

Où:

θ_{nouveau} = position suivante

θ_{ancien} = position actuelle

$\eta = \text{taux d'apprentissage (taille du pas)}$

$\Delta \theta J(\theta) = \text{Direction de l'augmentation la plus rapide}$

II.3.6.3. Descente de gradient stochastique (SGD)

Les algorithmes SGD se sont avérés efficaces dans l'optimisation des modèles d'apprentissage à grande échelle [CUI18]. Le terme stochastique désigne un mécanisme ou une méthode liée à une possibilité aléatoire ; par conséquent, au lieu de passer toute la base d'apprentissage à chaque itération pour mettre à jour les poids du modèle, seulement quelques échantillons sont choisis [WAN20]. SGD vise à trouver le minimum global en modifiant la structure du réseau après chaque étape d'apprentissage [HAL17]. Cette approche réduit simplement l'erreur en se rapprochant du gradient pour un lot choisi au hasard au lieu de trouver le gradient de la base d'apprentissage. En réalité, l'échantillonnage aléatoire est effectué en mélangeant au hasard la base d'apprentissage et en se déplaçant pas à pas dans les lots de la fonction objective [ZHA19], comme le montre la Figure II-14.

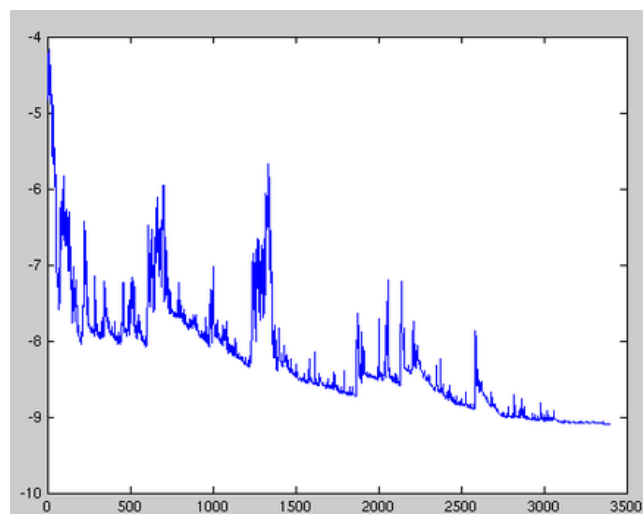


Figure II-14: Descente de gradient stochastique [RUD16].

II.3.6.4. Descente de gradient en mini-lot

Une autre variante des principes de SGD et BGD est le mini-batch GD. Il divise la base d'apprentissage en lots plus petits et effectue une mise à jour pour chacun de ces lots. Cela fournit un équilibre entre la robustesse du SGD et l'efficacité du GD par lots [YAN19].

Pour plusieurs problèmes d'optimisation, ces trois algorithmes GD (Batch GD, SGD et Mini-batch GD) fonctionnent bien et peuvent tous converger vers un optimum prometteur (local

ou global), mais ils souffrent toujours de nombreux problèmes tels que ceux mentionnés ci-dessous :

Le taux d'apprentissage η peut avoir un effet majeur sur la convergence des algorithmes GD [LIU19]. Il existe un compromis entre le taux de convergence et le dépassement lors de la définition d'un taux d'apprentissage. Si le taux d'apprentissage est trop élevé, nous pourrions dépasser le minimum et commencer à rebondir sans atteindre le minimum. En revanche, si le taux d'apprentissage est trop faible, la formation peut être trop longue [ZHA19].

Dans la plupart des cas, pour les algorithmes GD, un taux d'apprentissage fixe ne fonctionne pas bien dans le processus de mise à jour des poids [LID18]. L'algorithme nécessitera un taux d'apprentissage plus élevé dans les étapes initiales pour obtenir un optimum rapide réussi (local ou global). Dans les dernières étapes cependant, l'algorithme devra modifier le taux d'apprentissage [ZHA19].

Pour différentes variables, au lieu du processus de mise à jour, leur mise à niveau peut simplement impliquer un taux d'apprentissage différent. Par conséquent, il est nécessaire et approprié d'utiliser un taux d'apprentissage individuel pour diverses variables [ZHA19].

Un autre défi majeur est d'éviter d'être coincé dans leurs multiples minima locaux sous-optimaux en minimisant les fonctions d'erreur extrêmement non convexes communes aux réseaux de neurones [RUD16]. Les minima locaux sont le plus grand défi pour analyser la convergence [MOH18] comme la Figure II-15.

Les algorithmes d'optimisation constituent la base sur laquelle une machine peut bénéficier de sa pratique [ZEE20]. Ils mesurent des gradients et tentent de minimiser la fonction des coûts. L'apprentissage peut être mis en œuvre de plusieurs manières avec différents types de méthodes d'optimisation [OZT20].

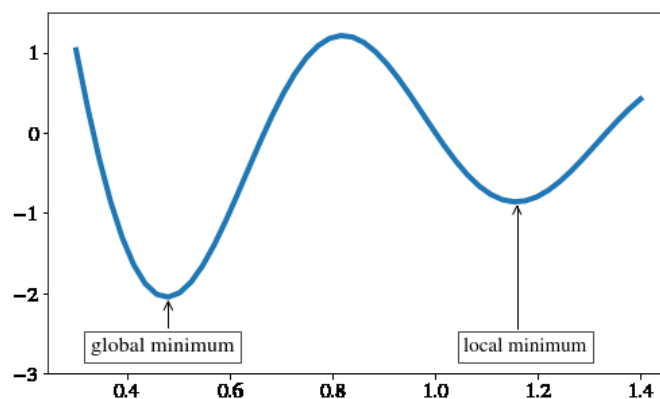


Figure II-15: Minima globaux Vs. Minima locaux [HAJ21].

II.3.6.5. Momentum

Une technique d'optimisation courante est SGD, mais lors de l'apprentissage du modèle, le temps d'exécution est relativement élevé. Momentum est conçu pour apprendre rapidement, en particulier face à de larges courbures, des gradients petits mais bruyants ou des gradients stables. L'utilisation d'un terme de quantité de mouvement est une autre approche qui peut aider le réseau à se débarrasser des minima locaux [SAD19]. C'est peut-être l'extension la plus courante de l'algorithme de rétropropagation. D'autres cas dans lesquels cette approche n'est pas utilisée sont difficiles à trouver. Le gradient ne pointe pas vers le minimum à des points spécifiques de la surface, et les pas successifs de GD oscilleront d'un côté à l'autre, n'avançant que très lentement vers le minimum [ZEE19]. La Figure II-16 illustre comment l'incorporation de la quantité de mouvement, en amortissant ces oscillations, tend à conduire à la convergence vers le minimum.

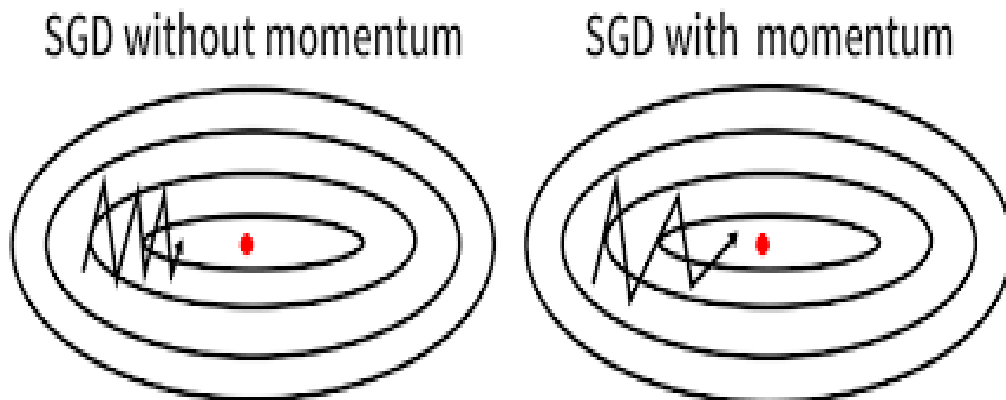


Figure II-16: La comparaison des algorithmes SGD avec et sans quantité de mouvement [HAJ21].

II.3.6.6. Nesterov Momentum

Sur la base de paramètres qui estiment des positions potentielles plutôt que des paramètres actuels, le gradient est calculé. Nesterov Momentum est une amélioration par rapport à l'élan et ne décide pas de l'emplacement futur du paramètre [BAR20]. Le Nesterov est une version de l'algorithme de momentum inspiré de la méthode du gradient accéléré de Nesterov. La distinction entre cette approche et la méthode de l'impulsion est que la vitesse est déjà ajoutée aux paramètres lors du calcul du gradient dans la méthode Nesterov. Cela peut être considéré comme une tentative d'incorporer un facteur correctif à la forme de quantité de mouvement conventionnelle [GYL17].

II.3.6.7. Descente de gradient adaptative (AdaGrad)

C'est une méthode qui sélectionne le rythme d'apprentissage en fonction de la situation [SOL20]. Comme le taux réel est calculé à partir de paramètres, les taux d'apprentissage ont tendance à s'adapter. Un gradient de paramètre plus élevé aura un taux d'apprentissage réduit et vice versa [ZHA18]. La théorie d'AdaGrad est similaire à l'algorithme AdaDelta en ce sens qu'elle mesure différents taux d'apprentissage pour d'autres éléments de paramètres. Pourtant, il utilise l'agrégation des carrés de gradient : contrairement à AdaDelta, il utilise la moyenne mobile des carrés de gradient [BAI19].

II.3.6.8. Delta adaptatif (AdaDelta)

AdaDelta est l'extension AdaGrad. AdaDelta fonctionne en utilisant plusieurs fenêtres de taille fixe au lieu d'accumuler les dégradés. Il ne surveillera que les gradients disponibles à l'intérieur de la fenêtre [NWA20]. Étant donné que l'algorithme SGD nécessite une sélection manuelle du taux d'apprentissage, le taux d'apprentissage inapproprié choisi conduirait à une faible précision de prédiction. Cependant, en tant qu'optimisation de l'algorithme SGD, l'Adadelta, connu sous le nom d'algorithme de taux d'apprentissage adaptatif (LR adaptatif), peut ajuster automatiquement le taux d'apprentissage et augmenter la précision de la prédiction [QUZ19].

II.3.6.9. Propagation racine carrée moyenne (RMSProp)

L'un des algorithmes stochastiques adaptatifs les plus courants pour l'apprentissage profond (Deep Neural Network ou DNN) est RMSProp. [ZOU19]. RMSProp modifie Adagrad de manière à accumuler le gradient. Les dégradés s'agrègent en une moyenne pondérée de manière exponentielle. RMSProp rejette le passé et ne préserve que les connaissances actuelles sur le gradient [ZEB20].

II.3.6.10. Estimation adaptative du moment (Adam)

Adam est une méthode d'optimisation SGD qui mesure des taux d'apprentissage adaptables pour chaque paramètre. Adam est l'une des stratégies d'optimisation la plus courante dans le domaine des réseaux de neurones. Le nom a été tiré d'Adaptive Moments. C'est un mélange de RMSProp et de Momentum. L'opération de mise à niveau considère la variante de gradient lisse et fournit un mécanisme de correction de biais. Adam réduit les coûts de calcul, nécessite moins de mémoire d'exécution et est invariant au redimensionnement diagonal du gradient [YAQ20]. RMSprop est un optimiseur basé sur le gradient qui utilise un taux d'apprentissage adaptatif

(LR adaptatif) qui varie dans le temps au lieu de traiter le taux d'apprentissage comme un hyperparamètre [FAT20].

II.3.6.11. Estimation du moment adaptatif maximal (AdaMax)

AdaMax est une forme de SGD avec taux d'apprentissage adaptatif, et une version Adam basée sur la norme de l'infini. AdaMax présente l'avantage majeur d'être beaucoup moins sensible à l'option des hyper-paramètres par rapport au SGD [VAN19]. L'équation AdaMax utilise la valeur complète de la deuxième composante de quantité de mouvement de la méthode d'estimation ADAM. Cela offre une solution plus stable [YID20].

II.3.6.12. Estimation du moment adaptatif accélérée par Nesterov (Nadam)

Cette technique est une combinaison d'approches d'optimisation Adam et RMSProp. Elle a été conçue de manière similaire à l'approche d'optimisation d'Adam. Cependant, elle a été remplacé par l'élan de Nesterov. Cela se traduit par une augmentation marquée de la surperformance du momentum [KEM19].

II.4. Apprentissage profond (Deep learning ou DL)

L'apprentissage en profondeur (DL) est l'une des plus prometteuses techniques d'apprentissage qui ont été appliquées dans plusieurs domaines. Les architectures basées sur DL sont caractérisées par de nombreuses couches cachées de neurones. Cependant, la principale limite de ces architectures est le long temps nécessaire à l'apprentissage. Obtenir une excellente précision et un temps d'apprentissage raisonnable est un objectif ambitieux pour la communauté de recherche DL.

II.4.1. Les réseaux de neurones convolutifs (CNN)

Le réseau de neurones à convolution également connu sous les noms (ConvNet, CNN ou apprentissage en profondeur), c'est technique qui se compose de plusieurs couches. Chaque couche de convolution est connue sous le nom de carte de caractéristiques.

En imagerie la carte des caractéristiques de la couche d'entrée est une matrice 3D d'intensités de pixels pour différents canaux de couleur (par exemple RVB). La carte des caractéristiques de n'importe quelle couche interne est un multicanal induit l'image d'entrée, dont le « pixel » peut être considéré comme une caractéristique spécifique. Tous les neurones sont connectés à une petite partie des neurones adjacents de la couche précédente. Différents

types de transformations peuvent être effectuées sur les cartes, telles que le filtrage (Convolution) et la mise en commun (pooling) [BOT14].

Dans la phase de Filtrage (convolution) l'opération convolue une matrice de filtre (poids appris) avec les valeurs d'un champ récepteur de neurones et prend une fonction non linéaire (telle que sigmoïde [WAD52], ReLU ...) pour obtenir les réponses.

L'Opération de mise en commun (pooling), telle que la mise en commun maximale (max pooling), la moyenne pooling, L2-pooling et normalisation locale du contraste [KAV09], résume les réponses d'un champ récepteur en une seule valeur pour produire des descriptions de fonctionnalités plus robustes. Avec un entrelacement entre la convolution et la mise en commun.

Une hiérarchie initiale des caractéristiques est construite, qui peut être affinée de manière supervisée en ajoutant une couche de neurones entièrement connectés (fully connected layer FC) pour s'adapter aux différentes tâches visuelles.

Selon les tâches impliquées, la couche finale avec différentes fonctions d'activation est ajoutée pour obtenir une probabilité conditionnelle spécifique pour chaque neurone de sortie, et le réseau peut être optimisé sur une fonction objective (par exemple erreur quadratique moyenne ou entropie croisée) via la méthode de descente de gradient stochastique (SGD).

Par exemple le VGG16 typique a au total 13 couches convolutionnelles, 3 couches entièrement connectées, 3 couches max-pooling et un softmax couche finale (couche de classification). Les filtres de convolutions sont produites par fenêtres de filtre 3x3 alambiquées et résolutions de cartes sont réduits avec 2 couches max-pooling. Pour tester le modèle, une image de la même taille que les échantillons d'apprentissage peut être traitée avec ce modèle formé. Dans ce cas une opération de remise à l'échelle ou de recadrage peut être nécessaire si différentes tailles sont fournies [KRI12].

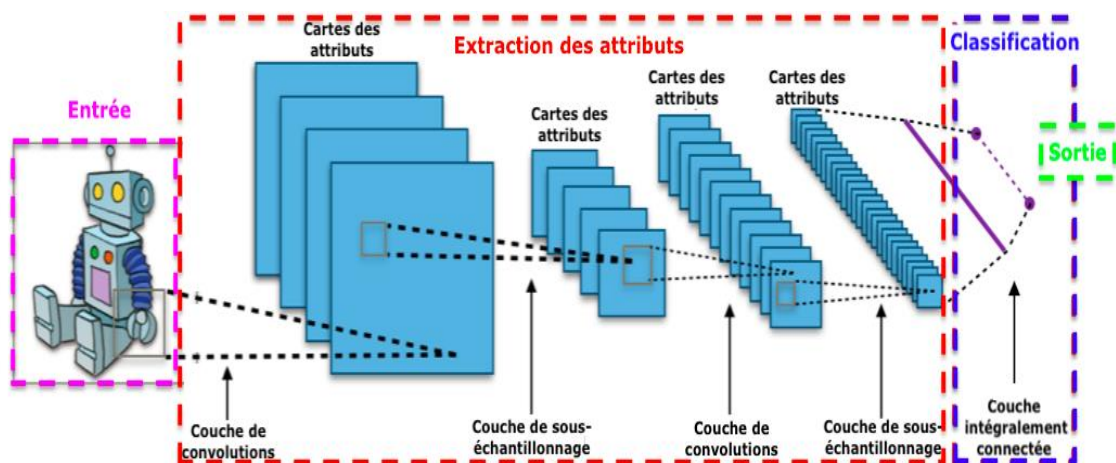


Figure II-17: Structure du réseau neuronal à convolution [SRI13].

II.4.1.1. Les avantages de CNN

Les avantages des réseaux à convolution par rapport aux méthodes de classifications traditionnelles peuvent être résumés comme suit :

Représentation hiérarchique des caractéristiques, qui correspond aux représentations multi niveaux du pixel aux caractéristiques sémantiques de haut niveau apprises par une structure hiérarchique à plusieurs étapes [JIA14].

L'architecture des réseaux CNN offre la possibilité d'optimiser conjointement plusieurs tâches connexes (par exemple, Fast RCNN combine la classification et la régression de la boîte englobant dans un mode d'apprentissage multitâches).

Bénéficiant de la grande capacité d'apprentissage des CNN, certains défis classiques de la vision par ordinateur peuvent être reformulé en tant que problèmes de transformation de données de grande dimension et résolu d'un point de vue différent.

En raison de ces avantages, les réseaux à convolution ont été largement appliqués dans de nombreux domaines de recherche, tels que la super-résolution d'images, reconstruction, classification d'images, récupération d'images, reconnaissance faciale, détection et analyse vidéo [JIA18].

II.4.1.2. Les inconvénients de CNN

Malgré ses progrès par rapport aux méthodes traditionnelles et l'importance des réseaux à convolution dans la détection pratique d'objets, il y a encore quelques inconvénients :

En raison de l'existence de couches entièrement connecté (fully connected), les réseaux de neurones à convolution nécessite une image d'entrée de taille fixe (par exemple, 227×227), qui mène directement au recalcul du modèle pour chaque région, prenant beaucoup de temps dans la période de test.

L'apprentissage du R-CNN est un pipeline en plusieurs étapes. En premier, le réseau convolutif (ConvNet) sur les propositions d'objets est peaufiné. Ensuite, le classifieur softmax appris par réglage fin est remplacé par des SVM pour s'adapter aux fonctionnalités de ConvNet. Enfin, les régresseurs de la boîte englobante sont entraînés. L'apprentissage est coûteux en espace et en temps. Les caractéristiques sont extraites de différentes propositions de région et stockés sur le disque. Il faudra beaucoup de temps pour traiter un problème avec une petite base de données avec les techniques des réseaux à convolution, tels que VGG16. Au en même temps,

la mémoire de stockage requise par ces caractéristiques devrait également être un sujet de préoccupation.

Bien que la recherche sélective puisse générer des propositions de région avec des rappels relativement élevés, les propositions de régions obtenues sont toujours redondantes et cette procédure prend beaucoup de temps.

II.4.1.3. L'opération de convolution

L'opération de convolution est très similaire aux filtres de convolution dans le domaine du traitement d'images tels que le filtre de Sobel, robert et le filtre gaussien. Le noyau du filtre se glisse sur une image et se multiplie les poids avec chaque pixel aligné, élément par élément à travers ce filtre. Ensuite le biais valeur est ajouté pour le résultat [TEO22].

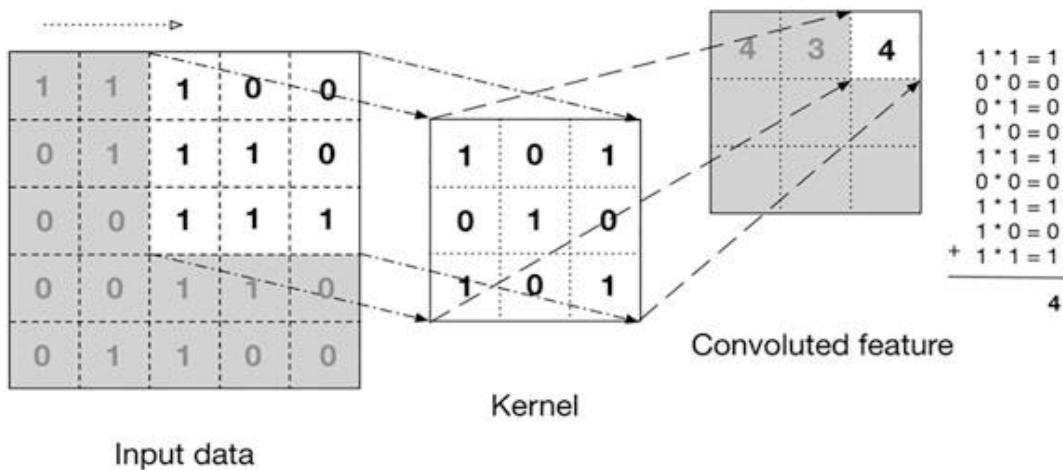


Figure II-18: schéma représente l'opération de convolution.

Dans la couche de convolution les trois hyperparamètres qui produisent la carte des caractéristiques sont:

Un pas (Stride S): représente le pas à chaque fois que nous faisons glisser le filtre de convolution. Lorsque le pas $S = 1$, nous déplaçons les filtres d'un pixel à chaque fois. Lorsque le pas $S = 2$ (ou 3 ou plus), alors les filtres sautent 2 pixels à un temps comme nous glissons [KON17].

Rembourrage (Padding P): Les images en entrées seront remplies avec une bordure de taille fixée selon la valeur spécifiée. Plus communément, le rembourrage zéro (zéro-padding) est utilisé pour tampon. Dans le framework de réseaux de neurones (caffe, TensorFlow, PyTorch, MXNet), la taille du padding « P » est un hyperparamètre. La taille du padding peut également être utilisée pour contrôler la taille spatiale des volumes de sorties [MIT21].

Profondeur (Depth D): la profondeur du volume de sortie est également un hyperparamètre; c'est correspond pour le nombre de filtres que nous utilisons pour une couche de convolution [GRI20].

II.4.1.4. Mise en commun (Pooling)

De nos jours, Les réseaux de neurones à convolution exploitent toujours un partage de poids étendu pour réduire les degrés de la liberté des modèles [WAN20]. Une couche de mutualisation permet de réduire le temps de calcul et construire progressivement une invariance spatiale et configurative. Pour la compréhension des images, la couche de mise en commun (pooling) aide à extraire plus de sens sémantique. Par exemple la couche de pooling maximum (Max-pooling) simplement renvoie la valeur maximale sur les valeurs sur lesquelles l'opération du noyau est appliquée. L'exemple ci-dessous illustre les sorties d'une mise en commun maximale (max_pooling) et d'une mise en commun moyenne opération (average pooling), respectivement, donné sur un noyau de taille 2 et un pas = 2.

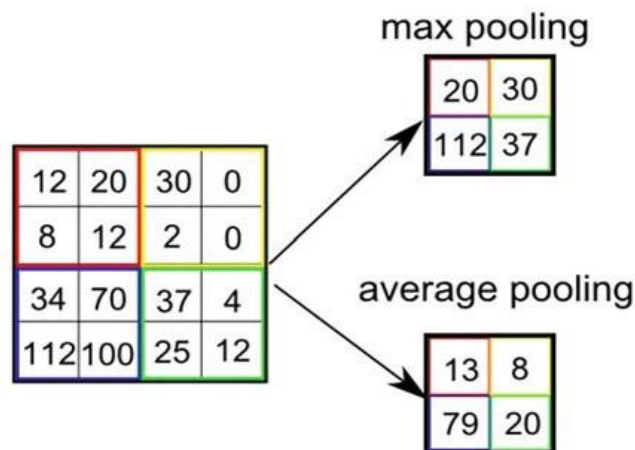


Figure II-19: schéma représente la couche pooling.

II.4.1.5. Aplanissement (Flattening)

L'ajout d'une couche entièrement connectée (Fully connected Layer) est une couche nécessaire vue que le non linéarité des combinaisons des caractéristiques de haut niveau telles qu'ils sont représentées par la sortie de la couche de convolution [COS17].

En aplatissant les images dans un vecteur (colonne), nous permettra de convertir les images de la couche de convolution sous forme appropriée pour notre Perceptron multi-couches. La sortie aplatie est alimentée à un réseau de neurones feed-forward et rétropropagation sera appliquées à chaque itération du processus d'apprentissage ; après une série d'itérations, le modèle est capable de faire la distinction entre dominantes et certaines caractéristiques de bas

niveau dans les images et les classer à l'aide de la technique de classification des réseaux de neurones.

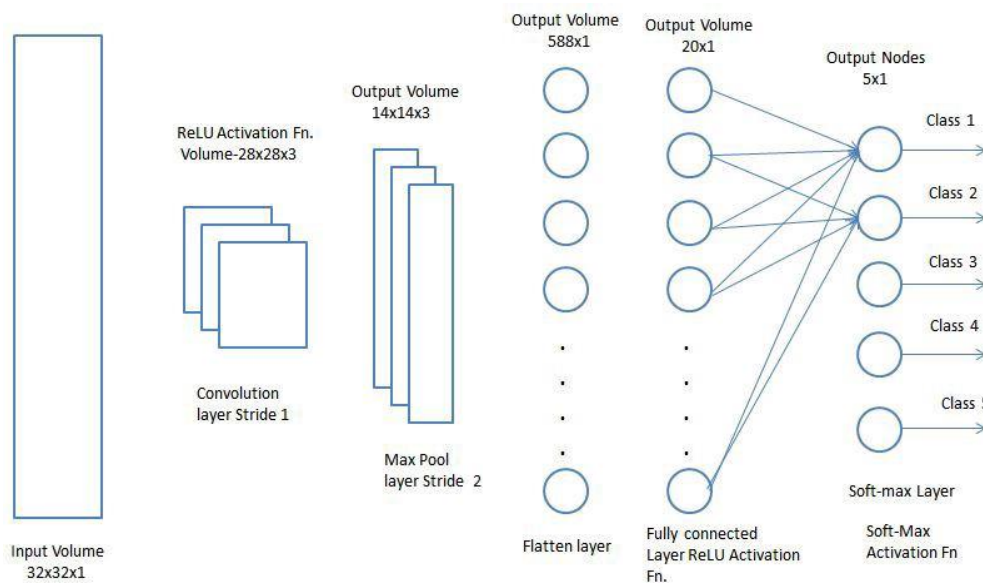


Figure II-20: schéma d'un réseau à convolution avec la couche Flattening.

II.5. Quelques Architecture basées CNN

Différentes architectures de réseau sont utilisées pour les tâches de classification d'images. VGG16, Inception Net (GoogLeNet) et Resnet sont parmi les plus notables.

II.5.1. VGG16

L'architecture VGG16 a suscité beaucoup d'attention en 2014. Elle apporte l'amélioration par rapport à son prédécesseur, AlexNet, en remplaçant les gros filtres de la taille d'un noyau (11 et 5 dans la première et la deuxième couche convolutive, respectivement) avec multiples filtres de la taille 3×3 [GOE21].

II.5.2. Inception Net

Avant les couches denses (qui sont placées au début du réseau), chaque fois que nous ajoutons une nouvelle couche, nous sommes confrontés à deux décisions principales :

1. Décider si nous voulons opter pour une opération de pooling ou de convolution.
2. Décider de la taille et du nombre de filtres à passer par la sortie de la couche précédente.

Les chercheurs de Google ont développé le module Inception qui nous permet d'appliquer toutes les options en une seule couche. L'idée principale du module Inception est d'exécuter plusieurs opérations (pooling, convolution) avec plusieurs tailles de filtre (3×3 , 5×5 ...) en parallèle [CHO17].

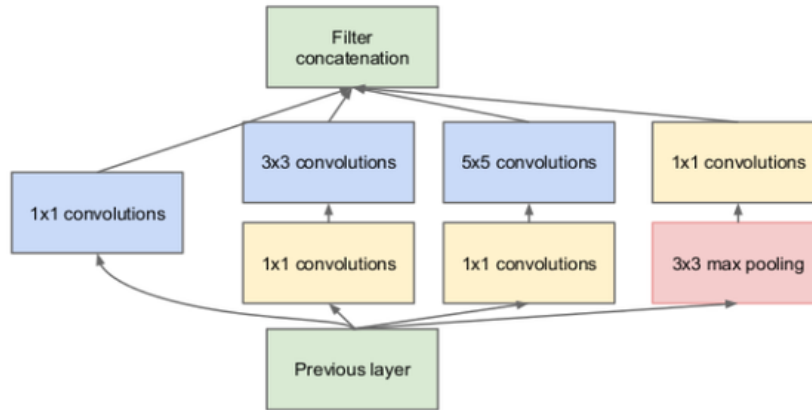


Figure II-21: Schéma représente l'architecture d'Inception de google.

II.5.3. ResNet

Les chercheurs pensaient que l'augmentation du nombre de couches améliorerait la précision des modèles. Mais il y a deux problèmes qui y sont associés :

1. Problème de fuite du gradient résolu avec une régularisation comme un lot normalisé. Les gradients deviennent de plus en plus petits à mesure que le réseau devient plus profond, ce qui rend plus difficile l'apprentissage des réseaux profonds.
2. Les chercheurs ont observé que l'ajout de couches supplémentaires n'améliorait pas la précision. [DEE20].

L'intuition de base des connexions résiduelles est que, à chaque couche de convolution, le réseau apprend certaines caractéristiques des données $F(x)$ et transmet les erreurs restantes plus loin dans le réseau. Nous pouvons donc dire que l'erreur de sortie de la couche de convolution est :

$$H(x) = F(x) - x.$$

Cette solution a également contribué à atténuer le problème de fuite du gradient car les gradients peuvent s'écouler par les connexions résiduelles.

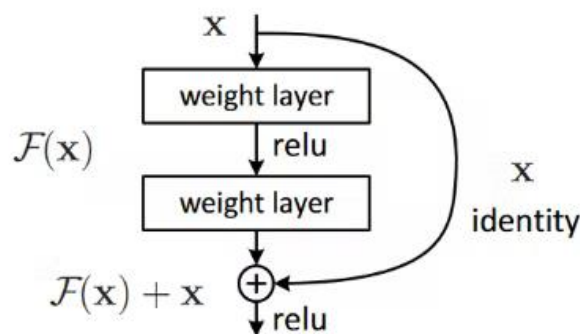


Figure II-22: Schéma représente l'architecture ResNet

II.6. Conclusion

Dans ce chapitre nous avons présenté quelques concepts fondamentales des réseaux de neurones et l'apprentissage profond, en présentant un neurone artificielle et sa relation avec un neurone biologique, l'initialisation des poids, les différents types des fonctions d'activation (Sigmoid/ tanh/Relu/leakyrelu,... etc), ainsi que les différentes types des fonctions d'erreurs, les optimiseurs (Descente de gradient /La descente de gradient élan stochastique/ ADAGARD/ADADELTA/ADAM). Ensuite, nous avons présenté les réseaux de neurones convolutifs CNN et leurs avantages et inconvénients, le principe de l'opération de convolution, les concepts (padding/ pooling / Flattening) et quelques applications pratiques comme VGG16, Inception et ResNet.

Chapitre

3

REVUES DE LA LITTERATURE

III.1. Introduction

L'apprentissage automatique fait partie des principales techniques de l'IA. Il permet de réaliser des prédictions en se basant sur une base de données en vue d'automatiser des tâches. L'un des avantages des modèles de l'apprentissage automatique et de deep learning, est qu'ils sont facilement adaptables à tous les problèmes, cette flexibilité les rend très populaires. La contrepartie est qu'ils sont assez difficiles à paramétrer, tant le nombre de degrés de liberté est élevé pour certains modèles. Et même si un premier entraînement avec des paramètres arbitraires nous offre de bons résultats, nous pourrions améliorer encore ces performances en optimisant les paramètres de notre modèle.

III.2. Classification des pathologies mammaires

III.2.1. Classification basée sur l'apprentissage profond des images mammographiques

Li et al. [LIH19] proposent un modèle de réseau de neurone DenseNet amélioré, également connu sous le nom de modèle de réseau de neurone DenseNet-II, pour la classification efficace et précise des images de mammographie bénignes et malignes.

Premièrement, les images de mammographie sont prétraitées. La normalisation d'image évite les interférences de la lumière, tandis que l'adoption de l'amélioration des données empêche le sur-apprentissage causé par une petite taille de base de données.

Deuxièmement, le modèle DenseNet est amélioré et un nouveau modèle de réseau de neurone DenseNet-II est inventé, qui doit remplacer la première couche convolutive du modèle DenseNet par la structure Inception.

Enfin, les bases de données de mammo-gramme prétraités sont entrés dans AlexNet, VGGNet, GoogLeNet, le modèle de réseau DenseNet, le modèle de réseau DenseNet-II, et les résultats expérimentaux sont analysés et comparés. Selon la méthode de validation croisée 10 fois, les résultats montrent que le modèle de réseau de neurone DenseNet-II a de meilleures performances de classification que les autres modèles de réseau. La précision moyenne du modèle atteint 94,55%, ce qui améliore la précision de la classification bénigne et maligne des images de mammographie.

A la fois, cela prouve également que le modèle a une bonne généralisation et une bonne robustesse.

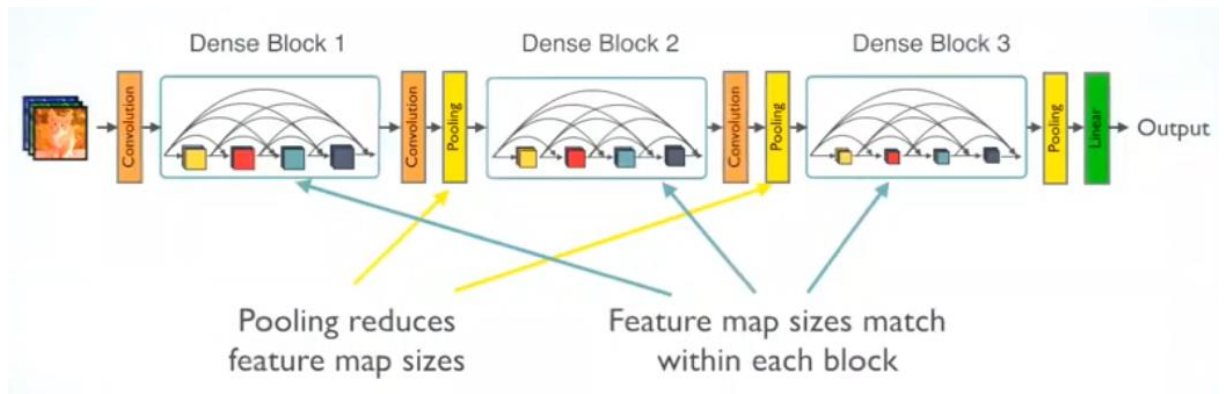


Figure III-1: La structure du modèle de réseau de neurone DenseNet-II.

III.2.2. Classification des mammographies basée sur l'apprentissage profond pour le cancer du sein

Altan et al. [ALT20] ont proposée l'architecture de réseau de neurones convolutionnelle (CNN) sur l'apprentissage simplifié des fonctionnalités et le modèle de classificateur affiné pour séparer les cas de cancer normaux sur les mammographies.

Le cancer du sein est une maladie répandue et mortelle résultant de la mutation du tissu normal en pathologie tumorale. Les mammographies sont des outils communs et efficaces pour le diagnostic du cancer du sein. Les systèmes assistés par ordinateur basés sur DL ont la capacité d'analyser en détail même les petites pathologies qui peuvent mener à la progression de la guérison pour une évaluation complète. Le modèle proposé basé sur DL visait à évaluer l'applicabilité de divers modèles d'apprentissage de caractéristiques et à améliorer la capacité d'apprentissage des modèles DL pour un diagnostic opératoire du cancer du sein à l'aide de CNN. Les mammographies ont été introduites dans le DL pour évaluer les performances de classification conformément aux différentes architectures CNN. Le modèle Profond proposé a atteint des taux de performance de classification élevés de 92,84 %, 95,30 % et 96,72 % pour l'exactitude, la sensibilité, la spécificité et la précision, respectivement.

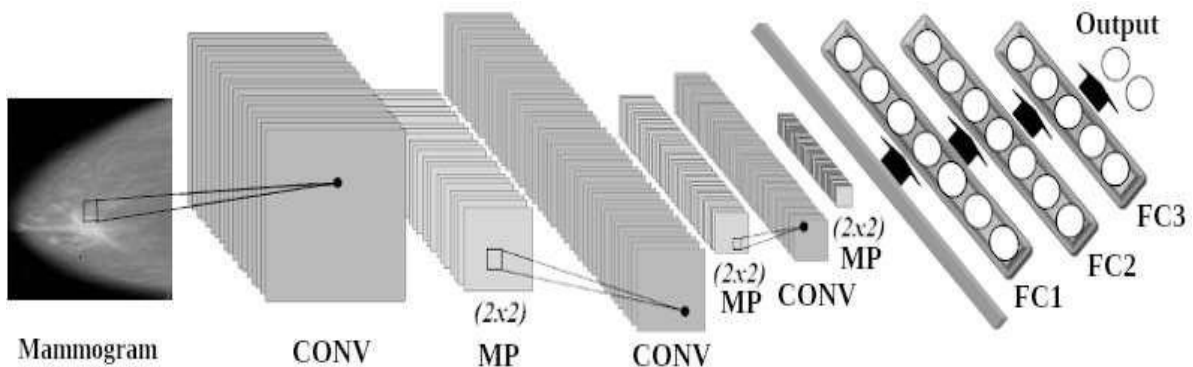


Figure III-2: L'architecture proposée des réseaux de neurones convolutifs couche par couche.

III.2.3. Performances incohérentes des modèles d'apprentissage en profondeur sur la classification des mammographies

Wang et al. [WAN20] ont proposé une étude sur la cohérence des performances de six modèles d'apprentissage profond dans des bases de données de quatre mammographies. Les mammographies contenant des calcifications, des masses ou les deux ont été classées comme bénignes ou malignes. Les modèles sont deux bout à bout, un frcnn_cad et trois nouveaux modèles qui a conçus.

Premièrement, l'entrainement des modèles sur la base de données numérique pour la mammographie de dépistage (DDSM), car il s'agit de la plus grande base de données de mammographie accessible au public. Ensuite, la validation de la précision de chaque modèle et enregistré les résultats sur DDSM. Finalement, le teste de chaque modèle sur trois bases de données différentes et comparé ces résultats avec les résultats de validation pour évaluer la cohérence des performances.

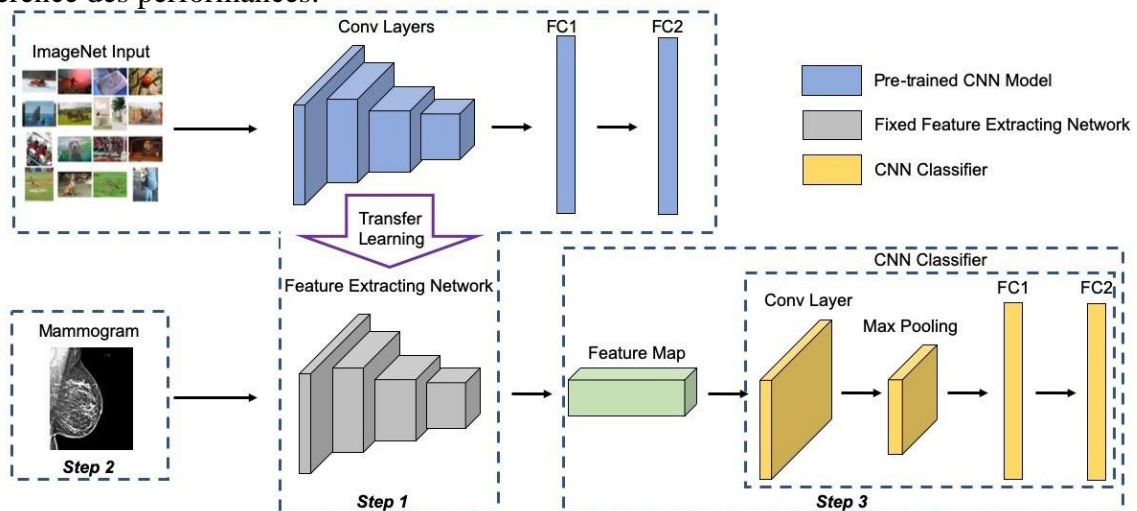


Figure III-3: Architecture du modèle d'apprentissage profond proposée par Wang et al. [WAN20]

III.2.4. Prédiction de l'efficacité de la chimiothérapie appliquée au cancer du sein par le traitement d'images et le Deep Learning.

Dans cette recherche de El Adoui et al. [ELA21], les techniques d'apprentissage profond par réseaux de neurones profonds (Deep Learning) sont proposées pour la segmentation du volume tumoral et la prédiction de la réponse d'un cancer du sein à la chimiothérapie. Afin d'automatiser ce processus, deux bases de données fournies par plusieurs instituts et hôpitaux

internationaux ont été utilisées. Les résultats prometteurs de cette étude montrent une valeur de précision de 89% en utilisant la méthode PRM, et une moyenne de précision de 93% pour l'apprentissage profond.

De plus, l'explication visuelle des résultats obtenus par le Deep Learning a montré une grande cohérence avec les résultats trouvés par les oncologues. Ces résultats se situent au-dessus de ce qui est présenté dans la littérature. La référence standard utilisée pour valider toutes les méthodes proposées est la réponse pathologique complète obtenue pour chaque patiente.

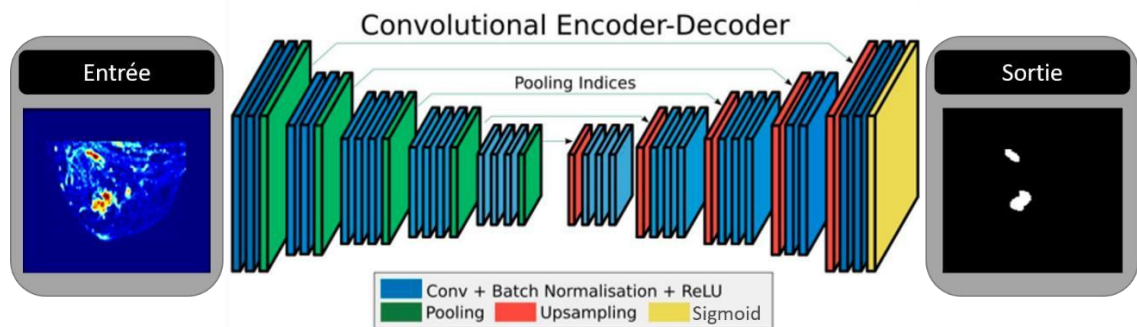


Figure III-4: Architecture SegNet adaptée dans ce travail.

III.3. Classification BI-RADS des anomalies mammaires

III.3.1. Un modèle de réseau de neurones profonds à haute performance pour une classification BI-RADS de la mammographie de dépistage

Tsai et al. [TSA22] ont présenté un modèle basé sur DNN pour localiser et classifier de manière efficace et fiable les lésions mammaires à partir de mammographies. Des images basées sur des blocs, segmentées à partir de mammographies collectées, ont été utilisées pour former adéquatement le modèle, grâce auquel la charge de travail des radiologues peut être considérablement allégée, en particulier lors de l'interprétation des mammographies dans un programme de dépistage du cancer du sein à grande échelle. Pour la première fois dans la littérature, les lésions mammaires peuvent être complètement classées dans les catégories BI-RADS 0, 1, 2, 3, 4A, 4B, 4C et 5. La surperformance de ce modèle a été indiquée par une précision globale de 94,22 %, une sensibilité moyenne de 95,31 %, une spécificité moyenne de 99,15 % et une ASC moyenne de 0,9723. Lorsqu'il est appliqué au dépistage du cancer du sein chez les femmes asiatiques, qui sont plus susceptibles d'avoir des seins denses, ce modèle devrait donner une plus grande précision que d'autres dans la littérature, car il a été formé à l'aide de mammographies prélevées sur des femmes taiwanaises. Il convient de mentionner que

ce travail peut offrir trois avantages aux industries de la santé. Premièrement, l'outil développé peut aider les radiologues à interpréter les mammographies dans les travaux cliniques et peut également améliorer l'efficacité de l'interprétation des mammographies. Deuxièmement, la charge de travail des radiologues peut être considérablement réduite. Troisièmement, l'outil peut aider les médecins généralistes à interpréter les mammographies en raison d'une pénurie de radiologues ou de chirurgiens du sein dans la plupart des régions éloignées. Dans la prochaine étape, l'objectif est vise à augmenter la taille de la base de données collectées afin de mieux entraîné le modèle et améliore également le renforcement des capacités. Pour cela, les efforts sont continus pour améliorer les performances du modèle, en particulier dans le pire des cas BI-RADS de catégorie 1. Enfin, le teste de généralisé ce modèle en tant que projet inter-hospitalier.

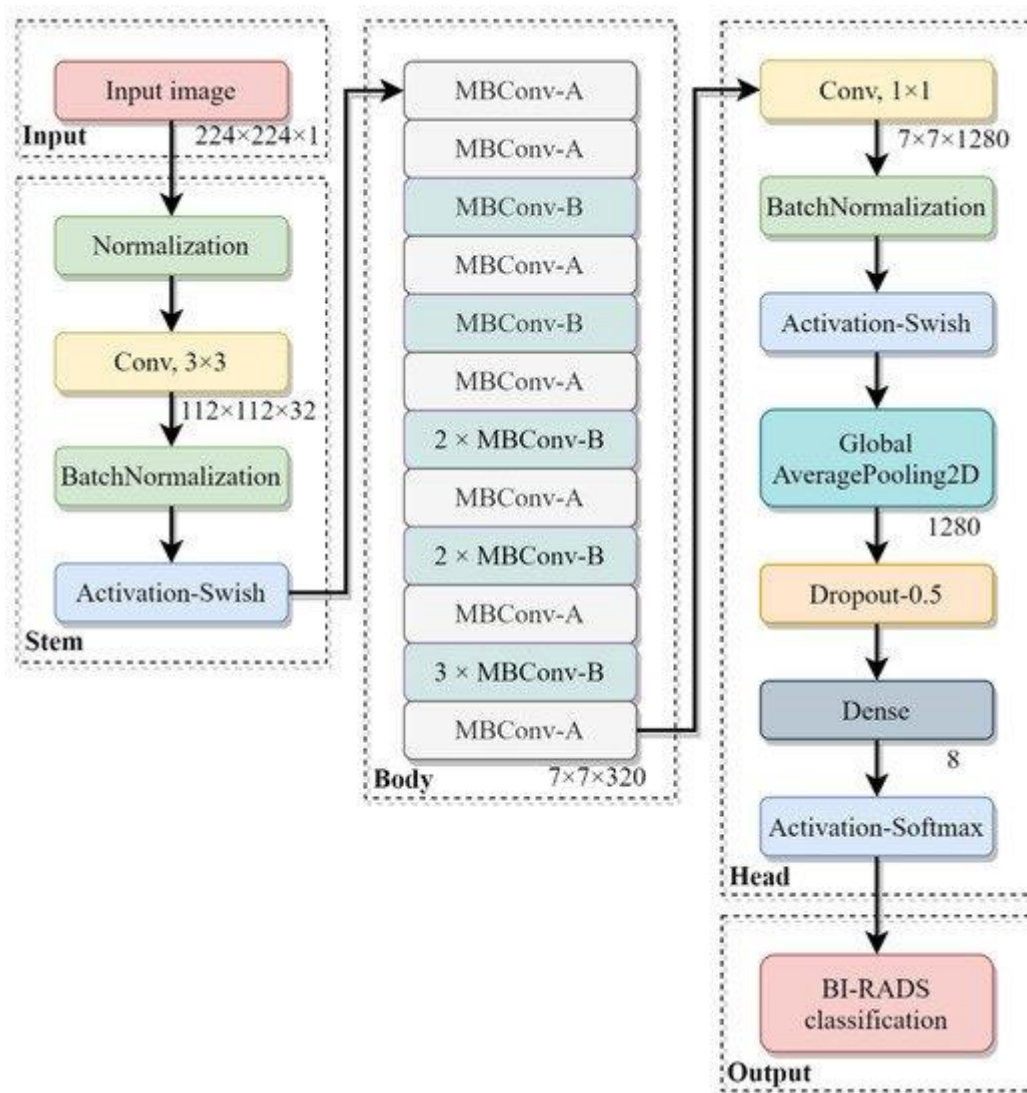


Figure III-5: Organigramme du modèle de classification BI-RADS présenté par Tsai et al. [TSA22].

III.3.2. Modèle RN-CNN d'apprentissage profond pour la classification BI-RADS du cancer du sein

Siddeeq et al. [SID21] ont utilisé des neurones personnalisés basés sur ResNet (RN-BCNN) pour la classification BI-RADS du cancer du sein. Dans la méthodologie proposée, déformation élastique avec rotation aléatoire et la technique de zoom pour l'augmentation des données est utilisée par rapport à transformation affine et réseau neuronal personnalisé basé sur ResNet (RN-BCNN) a été développé. Cette méthodologie améliore les résultats avec deux choses. Le premier est un entraînement déséquilibré de la base de données par rapport à la base d'entraînement équilibré et la deuxième avec une réduction de la taille d'image qui prend moins de temps tandis que les performances du modèle par rapport à la grande taille de l'image. Donc, résumé les résultats améliorés de 85,9% avec la base d'entraînement déséquilibré. Pour les travaux futurs, sera étendu le rapport d'imagerie mammaire et système de données (BI-RADS) de six à huit catégories avec comprendre les évaluations des sous-niveaux de la catégorie 4 : 4a, faible suspicion de malignité (2-9 %) ; 4b, suspicion modérée de malignité (10-49 %) ; 4c, une forte suspicion de malignité (50-94%) avec ou sans combinaison de multi-vues (CC & MLO) et aussi une manière des travaux futurs sera conservé ces six catégories d'évaluation avec combinaison de deux vues CC & MLO.

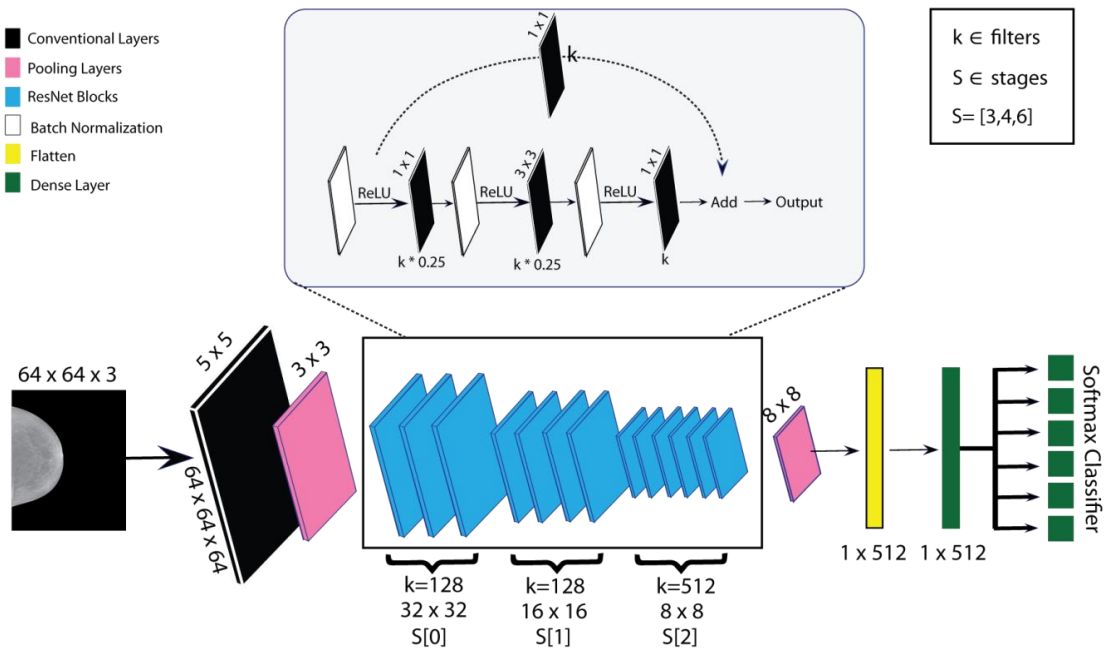


Figure III-6: Architecture du modèle RN-BCNN proposée par Siddeeq et al. [SID21].

III.3.3. Classification BI-RADS des images de mammographie à l'aide d'OHPLall

L'analyse et la classification d'images médicales, utilisant l'apprentissage automatique, en particulier les réseaux de neurones convolutifs, ont connu un grand succès. La recherche sur la classification des images mammographies avait tendance à se concentrer soit sur le résultat binaire (malignité ou bénigne) soit sur la classification nominale (non ordonnée) pour les étiquettes multi-classes. La métrique standard de l'industrie pour la classification des images de mammographie par les radiologues est une échelle d'évaluation appelée BI-RADS (Breast Imaging Reporting and Data System), où les valeurs 1 à 5 sont une progression distincte de l'évaluation destinée à indiquer un risque plus élevé de malignité, basé sur les caractéristiques des anomalies au sein d'une image. Le développement d'un classificateur qui prédit BI-RADS 1-5, fournirait aux radiologues une deuxième opinion objective sur les anomalies d'image. Vanderheyden et al. [VAN20] ont appliqué une nouvelle méthode d'apprentissage en profondeur appelée OHPLall (Ordinal HyperplaneLoss - all centroids), spécialement conçue pour les données avec des classes ordinales, aux prédictions des échelles BI-RADS sur les images de mammographie. Leur étude expérimentale a démontré des résultats prometteurs générés par OHPLall et un grand potentiel d'utilisation des modèles OHPLall comme outil de diagnostic supplémentaire.

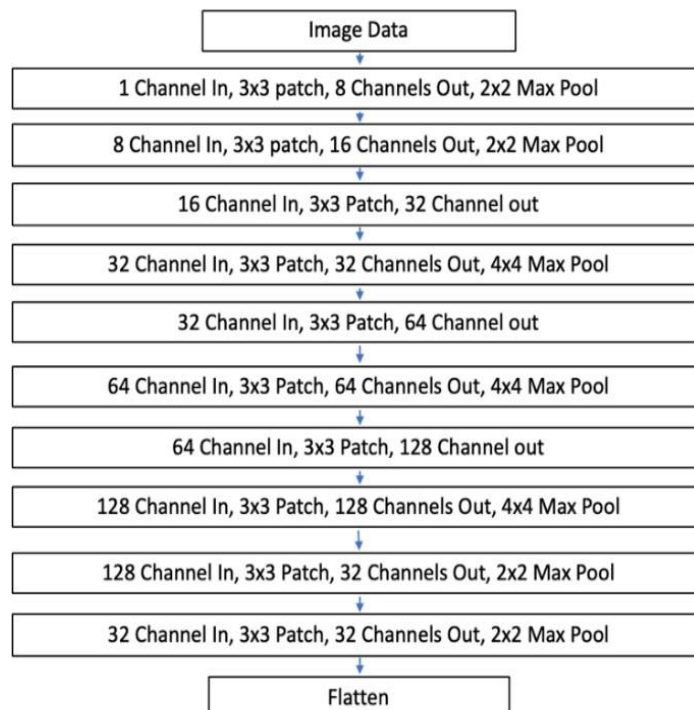


Figure III-7: Modèle CNN utilisé dans l'étude expérimentale de Vanderheyden et al. [VAN20].

III.3.4. L'Apprentissage en profondeur en utilisant une petite base de données de petite taille pour la classification de densité BI-RADS des images mammographiques

Shi et al. [SHI19], ont présente une étude d'apprentissage profond de la classification en densité BI-RADS à l'aide de la base de données MIAS (Mammographic Image Analysis Society), basée sur une architecture légère de réseaux neuronaux convolutifs (CNN). Il s'agit d'un problème car MIAS est une mini base de données, n'a que 322 images avec une vérité terrain, pour cela, il faut procéder au prétraitement et l'augmentation d'image pour résoudre le problème.

La validation croisée en cinq méthodes est utilisée pour évaluer l'approche proposée et a atteint une précision de test de 83,6 % en moyenne. Cela suggère que l'apprentissage profond a le potentiel de résoudre le problème de données réduites en mammographie, qui nous rencontre dans de nombreuses tâches d'analyse d'images médicales.

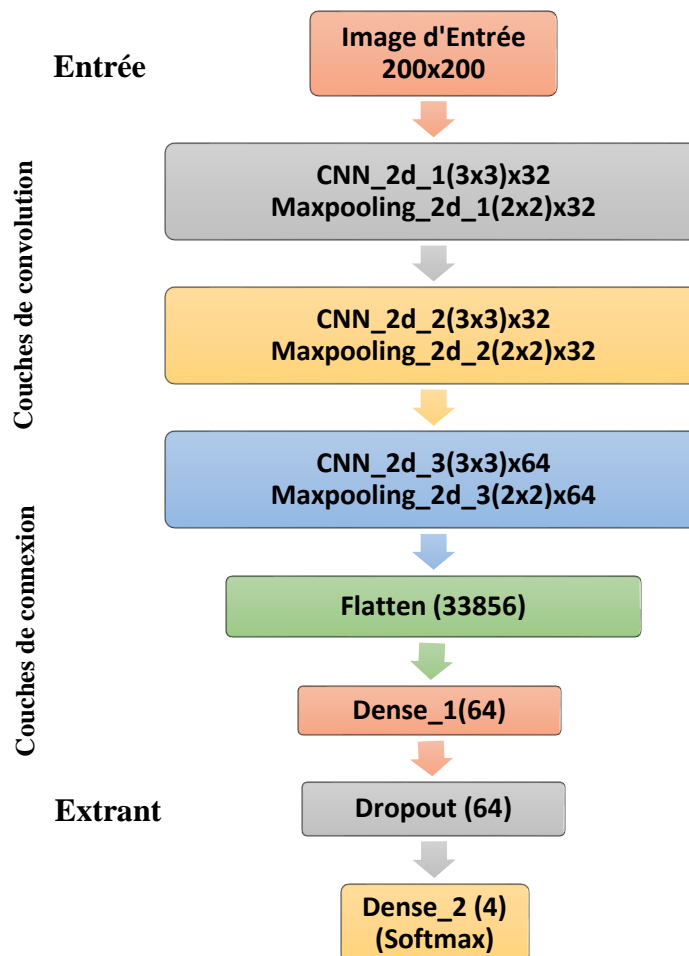


Figure III-8: Réseau de neurones en profondeur avec trois couches de convolution proposé par Shi et al. [SHI19].

III.4. Classification selon la typologie

III.4.1. Classification multi-classes des anomalies du cancer du sein à l'aide du réseau de neurones à convolution (CNN)

Selon Heenaye-Mamode et al. [HEE21], Les progrès récents dans le traitement d'images et l'apprentissage profond créent des espoirs dans la conception d'applications plus avancées pouvant être utilisées pour la détection précoce du cancer du sein. Dans ce travail, ils ont développé un réseau de neurones à convolution profonde (CNN) pour segmenter et classer les différents types d'anomalies mammaires, telles que les calcifications, les masses, l'asymétrie et les carcinomes, contrairement aux travaux de recherche existants, qui classaient principalement la maladie en cancer bénin et malin, conduisant à une meilleure gestion de la maladie. Premièrement, un apprentissage par transfert a été effectué sur la base de données par l'utilisation du modèle pré-entraîné ResNet50. Dans le même ordre d'idées, ils ont aussi développé un modèle d'apprentissage profond amélioré, dans lequel le taux d'apprentissage est considéré comme l'un des attributs les plus importants lors de l'entraînement du réseau de neurones. Le taux d'apprentissage est fixé de manière adaptative dans le modèle proposé en fonction des changements dans les courbes d'erreur au cours du processus d'apprentissage impliqué. Le modèle d'apprentissage profond proposé a atteint une performance de 88% dans la classification de ces quatre types d'anomalies du cancer du sein telles que les masses, les calcifications, les carcinomes et les mammographies d'asymétrie.

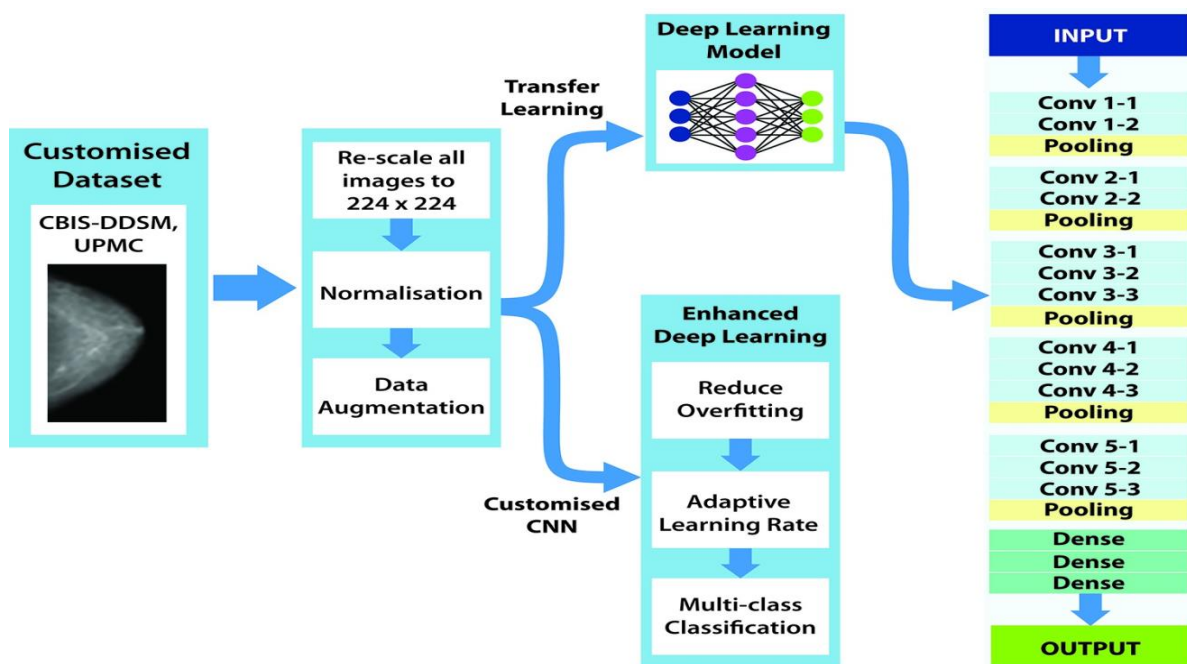


Figure III-9: Architecture du modèle proposé par Heenaye et al. [HEE21].

III.4.2. Apprentissage en profondeur pour une détection précoce du cancer du sein

She et al. [SHE19], ont développé un algorithme d'apprentissage profond qui peut détecter avec précision le cancer du sein sur les mammographies de dépistage par l'utilisation d'un modèle d'entraînement "de bout en bout" qui exploite efficacement les bases de données avec soit une annotation clinique complète, soit uniquement le statut du cancer (étiquette) de l'image entière. Dans cette approche, les annotations de lésion ne sont requises que dans la phase de d'entraînement initiale, et les étapes suivantes ne nécessitent que des étiquettes au niveau de l'image, éliminant ainsi la dépendance à des annotations de lésion rarement disponibles. Cette méthode de réseau entièrement convolutif pour classer les mammographies de dépistage a atteint d'excellentes performances par rapport aux méthodes précédentes. Sur un ensemble de tests indépendants de mammographies sur film numérisées de la base de données numérique pour le dépistage de la mammographie (DDSM), le meilleur modèle unique a atteint une AUC par image de 0,88, et la moyenne de quatre modèles a amélioré l'AUC à 0,91 (sensibilité : 86,1 %, spécificité : 80,1 %). Sur une base de validation d'images de mammographie numérique plein champ (FFDM) de la base de données INbreast, le meilleur modèle unique a atteint une AUC par image de 0,95, et la moyenne de quatre modèles a amélioré l'AUC à 0,98 (sensibilité : 86,7 %, spécificité : 96,1 %).

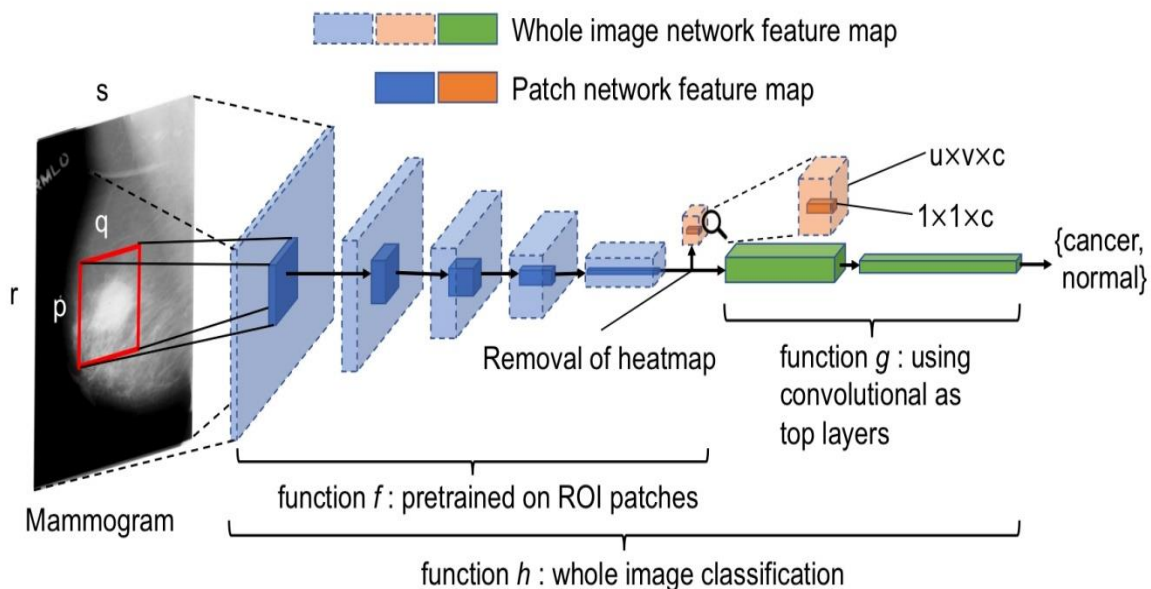


Figure III-10: L'architecture proposée par She et al. [SHE19], pour la détection des anomalies mammaires.

She et al. [SHE19], ont démontré également qu'un classificateur d'images complet formé à l'aide de l'approche de bout en bout sur les mammographies numérisées DDSM peut être transféré aux images INbreast FFDM par l'utilisation uniquement une mini base de données INbreast pour un réglage fin et sans dépendre d'avantage de la disponibilité de annotations de lésions. Ces résultats montrent que les méthodes automatiques d'apprentissage profond peuvent être facilement entraînées pour atteindre une grande précision sur des plates-formes de mammographie hétérogènes, et sont extrêmement prometteuses pour améliorer les outils cliniques afin de réduire les résultats de mammographie de dépistage faux positifs et faux négatifs.

Conversion d'un classificateur de patch en un classificateur d'image entière pouvant être formé de bout en bout à l'aide d'une conception entièrement convolutive. La fonction f a d'abord été entraînée sur des patches puis affinée sur des images entières. She et al. [SHE19] ont évalué si la suppression de la carte thermique améliorerait le flux d'informations des couches inférieures du classificateur de patch vers les couches convolutionnelles supérieures dans le classifieur d'images. La loupe affiche une version agrandie de la carte thermique. Ce chiffre est mieux vu en couleur.

III.4.3. Capacités d'apprentissage en profondeur pour la catégorisation des micro-calcifications

Kumar et al. [KUM22] ont proposé un pipeline automatique pour la détection de différentes catégories de microcalcification. Ils ont effectué un apprentissage en profondeur à l'aide de réseaux de neurones à convolution (CNN) pour la détection et la classification automatiques des trois catégories de microcalcification. CNN a été appliqué à l'aide de quatre optimiseurs différents (ADAM, ADAGrad, ADADelta et RMSProp). Les images d'entrée d'une taille de $299 \times 299 \times 3$, avec des fonctions d'activation de sorte RELU et SoftMax entièrement connectées, ont été utilisées dans cette étude. La carte des fonctionnalités a été obtenue à l'aide du modèle InceptionResNetV2 pré-entraîné.

L'évaluation des performances de schéma de classification a été testée sur un sous-ensemble d'imagerie mammaire organisé de base de données de mammographie DDSM (CBIS-DDSM), et les résultats ont été exprimés en termes de sensibilité, de spécificité, de précision et d'aire sous la courbe (AUC). Le schéma de classification proposé surpasse la capacité des approches d'apprentissage profond précédemment utilisées et des schémas d'apprentissage automatique classiques.

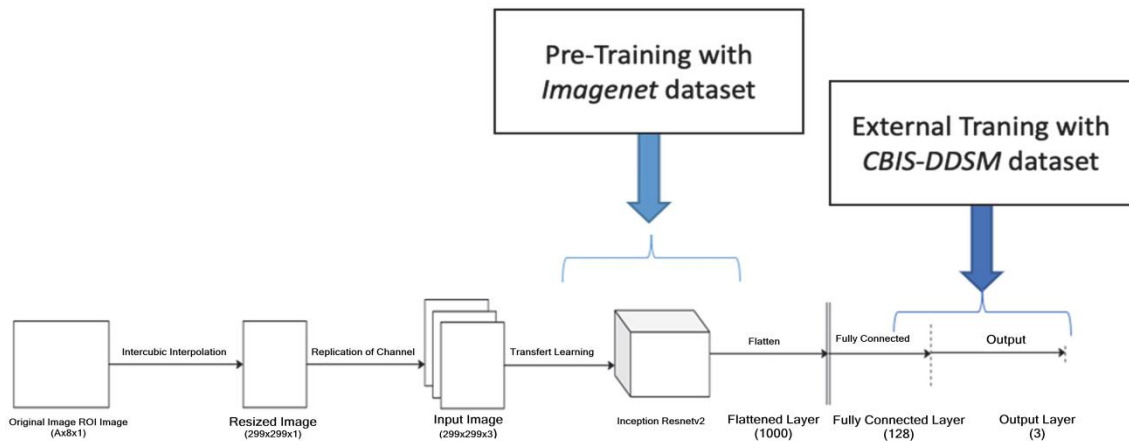


Figure III-11: L'architecture du réseau de neurones convolutionnelle proposée par Kumar et al. [KUM22].

III.4.4. Détection de la micro-calcification dans les mammographies à l'aide d'un réseau de neurone convolutif séparable en profondeur entièrement connecté

Rehman et al. [REH21], ont proposé un système de CAO FC-DSCNN basé sur la vision par ordinateur pour la détection des grappes de micro-calcification à partir des mammographies et la classification en classes malignes et bénignes. La méthode de vision par ordinateur contrôle automatiquement le bruit et le contraste des couleurs d'arrière-plan et détecte directement l'objet MC à partir des mammographies, ce qui augmente les performances de classification du réseau neuronal. Le cadre de classification du cancer du sein comporte quatre étapes : prétraitement et augmentation de l'image, transformation du canal RVB en niveaux de gris, segmentation de la région de micro-calcification et classification MC ROI à l'aide de FC-DSCNN pour prédire les cas malins et bénins. La méthode proposée a été évaluée sur 3568 images de mammographie DDSM et 2885 images de mammographie PINUM avec extraction automatique des caractéristiques, obtenant un score de 0,97 avec un rapport de vrais positifs de 2,35 et 0,99 avec 2,45 faux positifs par image, respectivement. Les résultats expérimentaux ont démontré que les performances de la méthode proposée restent supérieures aux approches traditionnelles et précédentes.

L'architecture de classification du cancer du sein comporte quatre étapes : prétraitement et augmentation de l'image, segmentation du retour sur investissement de la micro-calcification, conversion des canaux RVB en niveaux de gris et classification des grappes de micro-calcification pour prédire les cas malins et bénins.

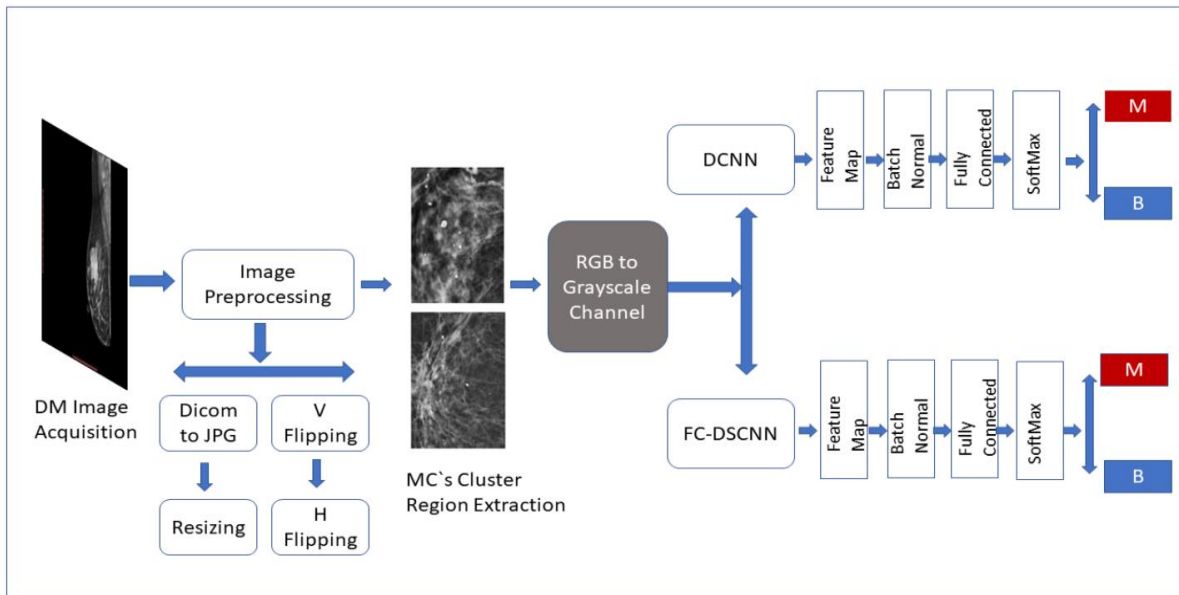


Figure III-12: L'architecture du système CAO proposé par Rehman et al. [REH21].

III.5. Conclusion

Nous pouvons conclure que l'application des méthodes de détection et de classification des images tissues mammographiques est encore reste un sujet de recherche très vaste. Ce chapitre, regroupe les différentes techniques basées sur l'apprentissage profond utilisées dernièrement pour la classification d'anomalies mammaires, leur objectif et certain méthodes consacré à la détection des tissus anormales dans les images mammographiques. Dans le prochain chapitre nous présentons notre travail qui ne permettra de définir un modèle pour classification des tissus anormales à partir d'un hypermodèle en utilisant la bibliothèque Keras-tuner.

Chapitre

4

OPTIMISATION DES HYPERPARAMETRES ET DEPLOIEMENT

IV.1. Introduction

Au fur et à mesure que les technologies d'apprentissage profond (deep learning) se développent avec des architectures plus complexes. Et avec la difficulté de réaliser des modèles en se basant sur les bases de données disponibles dans le domaine médical en raison de leur texture complexe, de leur taille limitée, de leur variabilité. Il existe un fort besoin de concevoir des architectures plus simples mais puissantes capables de gérer avec succès le problème de la classification des tissus mammaires. Pour atteindre cet objectif, nous présentons dans ce chapitre un modèle d'apprentissage profond basé sur les réseaux CNN. Ce modèle permet de réaliser des prédictions en se basant sur la base de données mammographique DDSM afin d'automatiser la tâche de classification des tissus mammographiques.

IV.2. Environnement de développement de système

Ce projet a été développé dans les conditions suivantes

IV.2.1. Environnement matériel

Notre travail est réalisé sur la plateforme de Google colab et sur un laptop dont la configuration est :

- ❖ CPU : Intel Core i7-10750H
- ❖ RAM : 16GB DDR4
- ❖ Carte graphique : NVIDIA GeForce RTX 2070 8GB.

IV.2.2. Environnement logiciel et langages de programmation

Le système d'exploitation installé est windows 11 64 bits, et les langages et les frameworks utilisés sont les suivants :

IV.2.2.1. Langage JAVA

Java est un langage de programmation et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995, et acheté par oracle par la suite, Le langage Java a la particularité principale que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux avec peu ou pas de modifications. Afin de faciliter l'exploitation de la base des images DDSM, nous avons opté pour la mettre à jour d'une application en JAVA de la promotion précédente, cette application nous permettra d'extraire les informations stockés dans des fichiers ASCII (.ics et .OVERLAY) et les organisent dans une base de données relationnelle, de cette façon nous offre

une carte détaillé et bien organisé de la base mammographique DDSM. Cette carte ainsi obtenu, nous avons développé une extension de cette application pour extraire à partir chaque mammographie classé normal une sous-image qui représente un tissu normale, cette ROI est placé dans un sous répertoire qui porte le nom de la classe Normal.

IV.2.2.2. Langage Python

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl. Le langage Python est placé sous une licence libre proche de la licence BSD et fonctionne sur la plupart des plates-formes informatiques, des smartphones aux ordinateurs centraux, de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et peut aussi être traduit en Java ou .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser [LUT13].

IV.2.2.3. Keras

Keras est une bibliothèque qui nous permettra d'interagir avec les algorithmes de réseaux de neurones profonds et d'apprentissage automatique, notamment Tensorflow, Theano, Microsoft Cognitive Toolkit ou PlaidML. Cette bibliothèque est conçue pour permettre une expérimentation rapide avec les réseaux de neurones profonds, elle se concentre sur son ergonomie, sa modularité et ses capacités d'extension. Elle a été développée dans le cadre du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System).

IV.2.2.1. Le Framework Flask

Flask [ARM22] est considéré comme un framework open-source de développement web en langage Python. Il a pour objectif de garder un noyau simple mais extensible. Il n'intègre pas de système d'authentification, pas de couche d'abstraction de base de données, ni d'outil de validation de formulaires. Cependant, de nombreuses extensions permettent d'ajouter facilement des fonctionnalités.

Flask a été créé par Armin Ronacher distribué sous licence BSD. Le souhait de Ronacher était de réaliser un framework web contenu dans un seul fichier Python mais pouvant maintenir des applications très demandées. Il est classé comme microframework car il est très léger.

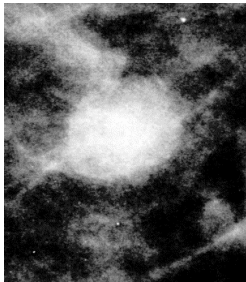
IV.2.3. La base de données utilisée

La base de données DDSM [HEA98] [HEA00] utilisé dans ce travail est une base de données open source accessible au public, cette base a été effectuée par « American college of Radiology » de l'université de la Floride du Sud en se basant sur le lexique de BI-RADS (Breast Imaging Reporting and Data System) [LEV05]. La base correspondante contient 2620 dossiers de patientes classés en trois groupes : normaux, bénignes et malignes.

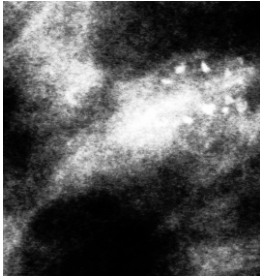
Dans la base de données DDSM, les mammographies ont été scannées par une résolution entre 0,042 et 0,05 mm par pixel et avec une taille moyenne de (3000 pixels × 5000 pixels). Le nombre d'anomalies total de cette base est 4638, dont 1929 cas malignes et 2709 bénignes. Chaque anomalie a été étiquetée selon (le type de lésion, l'évaluation BI-RADS, la pathologie et les limites de la région d'intérêt « ROI » de l'anomalie suspectée). Toutes les anomalies de cette base de données sont utilisées dans cette étude quel que soit leur type (masse/calcification), leur degré de densité (faible, moyenne ou élevée), leur forme ou leur type de contour ; et répartis également sur les quatre classes BI-RADS (2, 3, 4 et 5).

Dans la catégorie BI-RADS 0 l'évaluation est incomplète et nécessite des compléments d'imagerie. Cette catégorie est presque toujours utilisée en situation de dépistage mais rarement en situation diagnostique. Les mammographies classées BI-RADS 1 l'examen est considéré négatif (mammographie normale). La classe BI-RADS 2 à une constatation bénigne : Cela correspond pratiquement à une mammographie négative mais le radiologue désire décrire une anomalie dont les caractères radiologiques sont caractéristiques et qui peuvent être étiquetées avec beaucoup de confiance dans l'interprétation. Dans la catégorie BI-RADS 3 une anomalie probablement bénigne (> 98 %). Proposition d'une surveillance à court terme. Les anomalies classées BI-RADS 4 sont suspectes, dans ce cas une biopsie doit être envisagée. Cette probabilité est de 10 à 50 % [ORE99]. Une vérification histologique est nécessaire. Les anomalies classées BI-RADS 5 ont une haute probabilité de malignité. Ces lésions ont une très forte probabilité de correspondre à un cancer du sein (> 95 %).

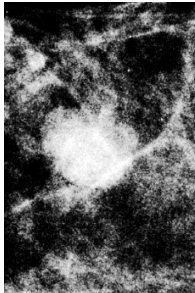
La figure IV.1 et IV.2 illustrent des exemples de régions d'intérêts (ROIs mammographiques) selon leurs étiquettes, sans tumeur montrée dans une catégorie avec l'étiquette «BI-RADS 1» et avec tumeur ont été montrées dans l'autre avec les étiquettes BI-RADS (0, 2, 3, 4 ou 5).



Masses bénigne (BI-RADS 0)



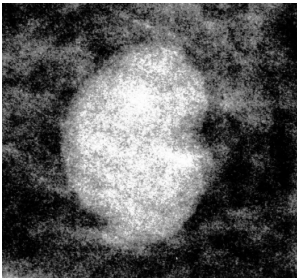
Calcifications maligne (BI-RADS 0)



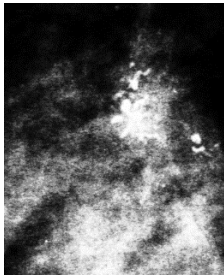
Masses bénigne (BI-RADS 2)



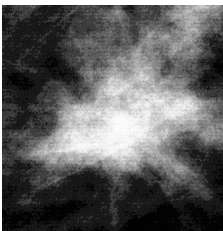
Calcifications bénignes (BI-RADS 2)



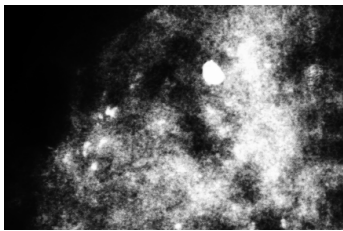
Masses bénigne (BI-RADS 3)



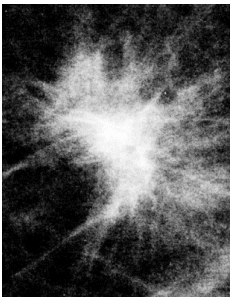
Calcifications maligne (BI-RADS 3)



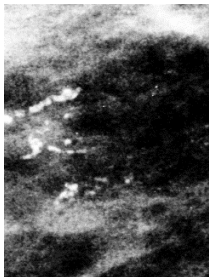
Masses maligne (BI-RADS 4)



Calcifications (BI-RADS 4)



Masses (BI-RADS 5)



Calcifications (BI-RADS 5)

Figure IV-1: Des masses et des calcifications répartis sur différentes classes BI-RADS 0, 2, 3, 4 et 5.

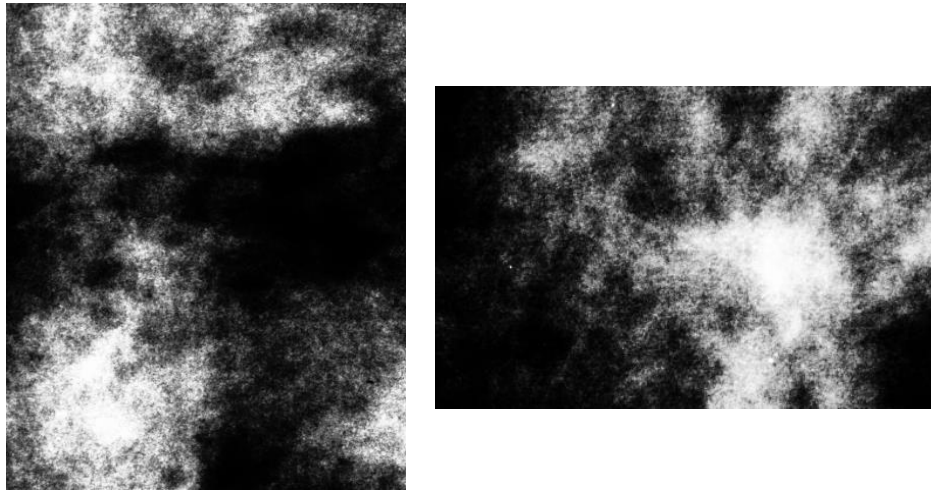


Figure IV-2: deux régions d'intérêts retirés d'une mammographie classées BI-RADS 1

Les expérimentations ont été faites sur un ensemble de 6688 régions d'intérêts choisies à partir de la base de données mammographique (DDSM). Les tissus anormales de cette expérimentation englobent toutes les anomalies de la base DDSM sans aucune exception (4638 Anomalies), et les tissus normales sont retirées aléatoirement à partir des mammographies des patients classées normales (Les mammographies classées BI-RADS 1) avec des dimensions aléatoirement choisies entre (240 pixels et 500 pixels de hauteur /largeur). Le tableau suivant présente la répartition des régions d'intérêt (ROI Normale ou ROI Anormales) sur les classes BI-RADS.

Type de ROI Classe BI-RADS	ROI Anormale		ROI Normale
	4638		2050
	malignes	bénignes	
	1929	2709	
BI-RADS 0	262		0
BI-RADS 1	0		2050
BI-RADS 2	723		0
BI-RADS 3	641		0
BI-RADS 4	2291		0
BI-RADS 5	721		0

Tableau IV-1: répartition des régions d'intérêts de la base de données utilisée.

Les régions d'intérêts de cette base seront redimensionnées pour préparer trois bases de données de tailles (150 × 150, 200 × 200 et 240 × 240), chacune s'utilise pour l'entraînement la recherche des hyperparamètres optimales d'un réseau de convolution.

IV.3. Architectures générale de travail

L'architecture générale de notre travail englobe trois phases principales ; dans la première étape nous préparons les images d'entraînement et de validation des expérimentations de ce travail. Dans la deuxième étape nous utilisons la bibliothèque keras-tuner pour la sélection des valeurs des hyperparamètres des réseaux de convolution de chaque expérimentation ; le meilleur modèle après cette étape sera sélectionné pour une phase de réapprentissage en utilisant la bibliothèque keras afin d'améliorer les résultats de classification. Dans la troisième et la dernière étape nous développons une application web à travers le micro-Framework Flask ; l'interface graphique de cette application nous permettra de sélectionner une région d'intérêt (ROI) qui représente un tissu d'une mammographie et de prédire s'il s'agit d'une anomalie ou non.

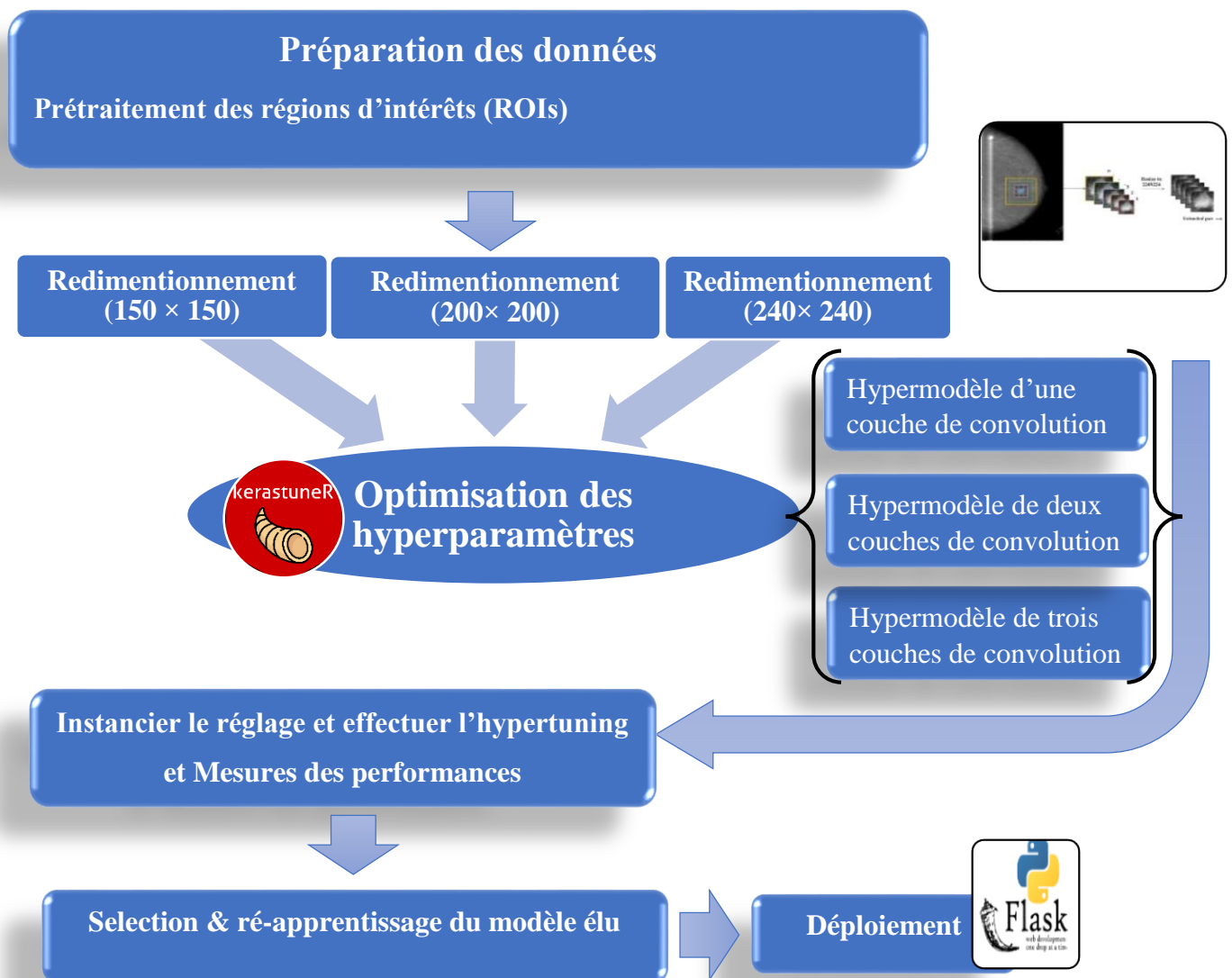


Figure IV-3: L'architecture générale de notre travail.

IV.3.1. Préparation des données

Dans un système de classification automatique, la préparation de données prépare les images d'apprentissage en vue de leur ingestion par le modèle de reconnaissance. Il s'agit d'une étape préliminaire qui regroupe un ensemble de phases préparatoires, de la collecte jusqu'aux processus de validation des données. L'une des étapes principale du processus consiste à formater les données selon leurs densités ou leurs tailles et rectifier les éventuelles erreurs qu'elles peuvent comporter.

La préparation de données en apprentissage automatique améliore la qualité des images avant leur traitement. Cette préparation consiste à corriger les biais qui pourraient avoir un impact négatif sur les résultats de classification.

IV.3.1.1. Recadrage et redimensionnement des régions d'intérêts

Étant donné que les régions d'intérêts des 4638 anomalies de la base de données DDSM utilisées dans cette étude sont hétérogènes, car elles sont de tailles, de largeurs et de hauteurs différents, donc il est nécessaire de recadrer les régions d'intérêts dans une taille (240×240 pixels) si les ROIs des anomalies sont inférieur de cette taille cette taille est considéré comme une taille de référence minimale pour toutes les région d'intérêts utilisées pendant cette expérimentation .

Dans les systèmes d'apprentissage en profondeur il est recommandé de redimensionner les ROIs à une hauteur et une largeur égales pour obtenir des résultats optimaux. Il est également souhaitable de redimensionner les ROIs de manière à ce qu'elles ne perdent pas leurs informations importantes ou ne dégradent pas leur qualité. Dans ce projet, les régions d'intérêts sont redimensionnées pour préparer trois bases de données de tailles (150×150 , 200×200 et 240×240), chacune s'utilise pour l'entraînement la recherche des hyperparamètres optimales d'un réseau de neurone à convolution.

IV.3.1.2. Prétraitement des régions d'intérêts (ROIs)

Le prétraitement des ROIs consiste à convertir des données brutes en données préparées. Par ailleurs, l'étape de classification automatique traite ensuite ces données préparées de manière à créer des attributs utiles pour l'apprentissage automatique. Les données brutes sont des données sous leur forme source non traitées pour l'apprentissage automatique. Le prétraitement des ROIs peut comprendre diverses opérations. Chaque opération vise à aider le système d'apprentissage automatique à élaborer de meilleurs résultats de classification et

prédiction. Étant donné que les mammographies de la base de données DDSM ont été scannées en utilisant des scanners de différentes marques, avec différentes conditions de contraste et de résolutions, donc à l'intérieur de cette étape nous avons opté pour l'égalisation d'histogramme qui permet de mieux répartir les intensités sur l'ensemble des valeurs possibles des niveaux de gris.

IV.3.2. Optimisation des hyperparamètres

Lors de la construction d'un modèle d'apprentissage automatique, l'on fait face aux choix des hyperparamètres. Sachant qu'un hyperparamètre est une caractéristique dont la valeur est définie avant le début du processus d'apprentissage, par contre les valeurs des autres paramètres sont obtenues pendant l'entraînement.

L'apprentissage profond est la réalisation d'un modèle d'apprentissage qui se base sur un réseau de neurones artificiels. Ce type de modèle contient un nombre considérable d'hyperparamètres, et leur réglage devient plus complexe. Une partie de la configuration d'un modèle basé sur l'apprentissage profond consiste à régler les hyperparamètres qui se regroupent en 3 types :

- **Les hyperparamètres des couches** : taille du masque de convolution, dropout, méthode d'activation des couches cachés, méthode d'activation de la couche final.
- **Les hyperparamètres de compilation** du modèle: optimizer utilisé, taux d'erreur, taux d'apprentissage ...
- **Les hyperparamètres d'exécution** du modèle : batch size, nombre d'epochs, ...

Ces valeurs de configuration qu'il faut définir manuellement font partie de l'hyperparameter tuning d'un modèle basé sur l'apprentissage profond.

Ces mesures doivent être réglées pour que le modèle puisse résoudre de manière optimale le problème d'apprentissage automatique. L'optimisation des hyperparamètres trouve un tuple d'hyperparamètres qui produit un modèle optimal qui minimise une fonction de perte prédéfinie sur des données indépendantes.

Afin d'obtenir les meilleurs performances, les datascientistes passent des heures à régler ces hyperparamètres manuellement, car l'implémentation d'un modèle semble être un casse-tête pour eux.

Il est donc clair que le choix des hyperparamètres est crucial pour les performances des modèles d'apprentissage profond, de leurs vitesses et leurs qualités. Il faut donc trouver un outil de les régler d'une façon systématique. Pour le faire, nous avons besoin d'une méthode automatique.

Heureusement, on a plusieurs méthodes qui peuvent nous aider à faire ce travail telles que la recherche en grille (Gridsearch) [KAU20], la recherche aleatoire (Random Search) [VIL21]. Egalement des techniques plus sophistiquées telles que l'optimisation bayésienne, Hyperbande [LIL17].

Dans ce travail, nous essayons de trouver le meilleur modèle de classification d'images des tissus mammaires retirés de la base de données mammographique DDSM en se basant sur l'hyperbande et l'optimisation baésienne à travers le framework Keras Tuner [OMA19].

IV.3.2.1. Optimisation avec Keras Tuner

Keras Tuner est une librairie d'optimisation d'hyperparamètres developpée par O'Malley et al. [OMA19] facile à utiliser et distribuable, elle résout les problèmes liés à l'exécution d'une recherche d'hyperparamètres. Keras Tuner est livré avec des algorithmes d'optimisation bayésienne, d'hyperbande et de recherche aléatoire intégrée, et est également conçu pour être facile à étendre pour les chercheurs afin d'expérimenter de nouveaux algorithmes de recherche [YAN22].

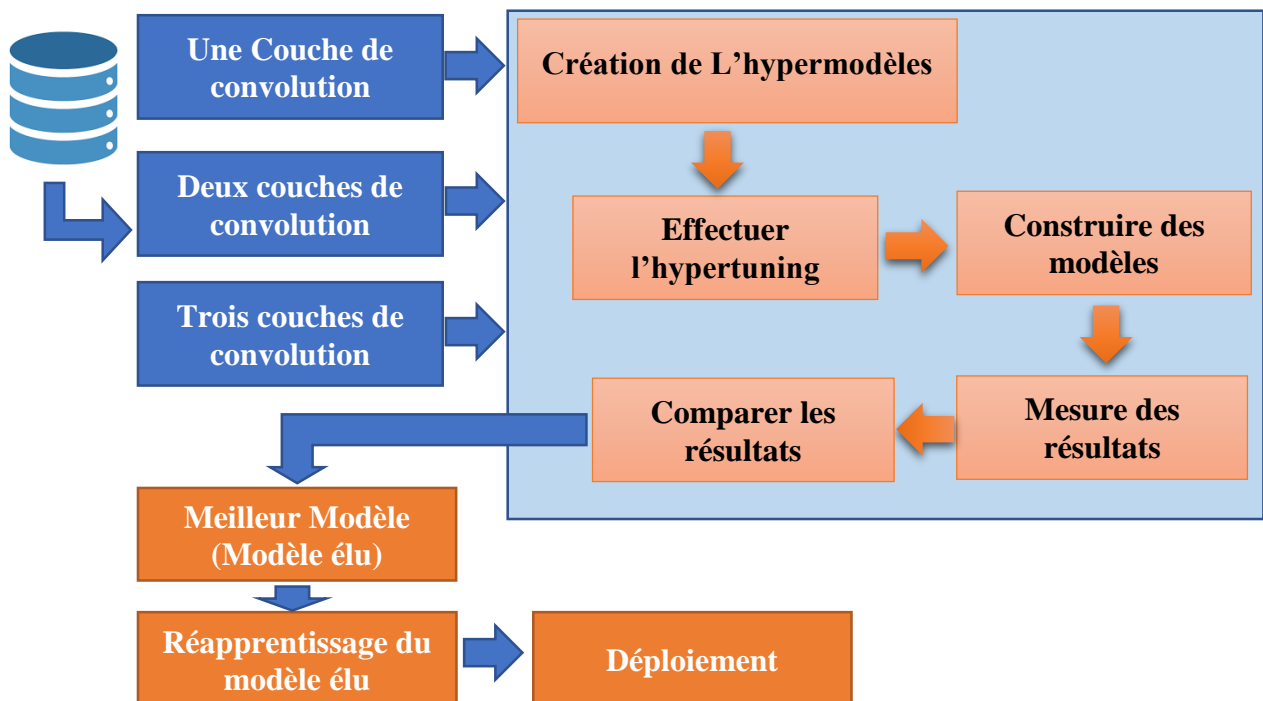


Figure IV-4: L'architecture d'optimisation en utilisant keras-tuner.

IV.3.2.2. L'hypermodèle

Cette première étape consiste à créer un hypermodèle dans lequel nous définissons les différents hyperparamètres pour le réglage automatique à travers la bibliothèque keras-tuner.

Dans ce hypermodèle, nous définissons les méthodes de choix des hyperparamètres. Chacune à un nom et une liste de valeurs à choisir. Ces valeurs doivent être du même type. Dans la plateforme Keras Tuner, le choix des hyperparamètres se fait en utilisant les méthodes (Float, Int, Boolean et Choice), ces méthodes seront paramétrées par un nom unique et la liste des valeurs pour les choix possibles dans le cas de la méthode « choice » à travers l'intervalle de définition de l'hyperparamètre et dans le cas des autres méthodes à travers une valeur minimale, maximale, une valeur de pas et une valeur par défaut.

L'hypermodèle que nous avons adopté est paramétré comme suit:

- ❖ Le nombre de filtres pour les couches de convolutions :
(Valeur min = 128 ; Valeur max = 256 ; pas = 16)
- ❖ La taille du masque de convolution : filtres de tailles [(3×3) ou (5×5)]
- ❖ Le nombre d'unités pour la couche dense :
(Valeur min= 128 ; Valeur max = 512 ; pas = 16)
- ❖ Le taux d'apprentissage pour l'optimizer Adam : un choix parmi 0.01, 0.001 ou bien 0.0001.

Sachant que cette expérimentation a été faite en utilisant une couche, deux couches et trois couches de convolution, sur les tailles de la couche d'entrée (150×150), (200×200) et (240×240).

IV.3.2.3. Instancier le réglage et effectuer l'hypertuning

Pour instancier le mécanisme de recherche à travers la méthode RandomSearch de keras-tuner, nous devons spécifier l'hypermodèle, l'objectif à optimiser (objective) et le nombre d'essais à travers le paramètre (max_trials).

	Hyperparamètre	Taille de la région d'intérêt (ROI)		
		(150 × 150)	(200 × 200)	(240 × 240)
Une couche de convolution	Nombre de filtres de convolution N°1	112	48	112
	Taille du masque de convolution N°1	5	5	3
	Nombre de neurones de la couche dense	80	48	48
	Taux d'apprentissage	0.0001	0.0001	0.0001
	Taux de reconnaissance (Apprentissage)	78.36%	82.67%	69.71%
	Taux de reconnaissance (validation)	77.48%	74.79%	68.51%
	Erreur d'apprentissage	0.4280	0.3684	0.6767
	Erreur de validation	0.4509	0.5171	0.6778
	Temps d'exécution	11m 57s	24m 27s	39m 15s

Tableau IV-2: Les résultats de l'expérimentation effectuée sur une seule couche de convolution.

Après l'exécution du processus de recherche des hyperparamètres, les meilleurs résultats obtenus pour les expérimentations sont présentés sur les tableaux (Tableau IV-2, IV-3 et IV-4).

Après la première expérimentation, il a été remarqué que la base d'images ayant les tailles (150 × 150) a donné la plus petite erreur de validation (0.4509), elle a donnée également le meilleur taux de reconnaissance pendant la validation (77.48% des régions d'intérêts bien classées)

	Hyperparamètre	Taille de la région d'intérêt (ROI)		
		(150 × 150)	(200 × 200)	(240 × 240)
2 couches de convolution	Nombre de filtres de convolution N°1	224	160	192
	Taille du masque de convolution N°1	3	3	5
	Nombre de filtres de convolution N°2	208	208	224
	Taille du masque de convolution N°2	5	3	3
	Nombre de neurones de la couche dense	96	64	64
	Taux d'apprentissage	0.0001	0.0001	0.0001
	Taux de reconnaissance (Apprentissage)	83.59%	90.26%	87.27%
	Taux de reconnaissance (validation)	79.22%	76.88%	78.23%
	Erreur d'apprentissage	0.3400	0.2565	0.3011
	Erreur de validation	0.4218	0.4544	0.4415
	Temps d'exécution	47m 17s	1H 18m 13s	2H 38m 6s

Tableau IV-3: Les résultats de l'expérimentation effectuée sur deux couches de convolution.

	Hyperparamètre	Taille de la région d'intérêt (ROI)		
		(150 × 150)	(200 × 200)	(240 × 240)
3 couches de convolution	Nombre de filtres de convolution N°1	192	240	208
	Taille du masque de convolution N°1	3	5	5
	Nombre de filtres de convolution N°2	144	224	160
	Taille du masque de convolution N°2	3	3	3
	Nombre de filtres de convolution N°2	96	80	80
	Taille du masque de convolution N°3	3	3	5
	Nombre de neurones de la couche dense	32	128	128
	Taux d'apprentissage	0.0001	0.0001	0.0001
	Taux de reconnaissance (Apprentissage)	83.85%	89.23%	78.62%
	Taux de reconnaissance (validation)	78.48%	77.23%	78.18%
	Erreur d'apprentissage	0.3232	0.2715	0.4017
	Erreur de validation	0,4196	0.4493	0.4259
	Temps d'exécution	50m 47s	1H 21m 13s	2H 01m 9s

Tableau IV-4: Les résultats de l'expérimentation effectuée sur trois couches de convolution.

Après la deuxième expérimentation à travers deux couches de convolutions, il a été remarqué que la base d'images ayant les tailles (150 × 150) a toujours donné la plus petite erreur de validation (0.4218), elle a donnée également le meilleur taux de reconnaissance pendant la validation (79.22% des régions d'intérêts bien classées).

Dans la troisième expérimentation à travers trois couches de convolutions, il a été remarqué que la base d'images ayant les tailles (150 × 150) a toujours donné la plus petite erreur de validation (0,4196), mais seulement 78.48% des régions d'intérêts bien classées pendant le processus de validation.

Dans toutes les expérimentations le nombre total d'essaies (max_trials) a été fixé par 30 essais, l'objectif fixé de notre tuner est le taux d'erreur sur la base de validation. Le nombre d'itérations du mécanisme de recherche a été fixé par 7.

IV.3.2.4. Sélection & réapprentissage du meilleur modèle élu

Les expérimentations effectuées dans ce travail ont été terminés après un temps total = 10 heures 12 minutes et 24 secondes sur un Laptop ayant les caractéristiques (Système d'exploitation : windows 11, CPU : Intel Core i7-10750H, RAM : 16GB DDR4 et une carte graphique : NVIDIA GeForce RTX 2070). Seulement l'expérimentation de la base de données des régions d'intérêts ayant la taille (240 × 240) a été effectuée sur la plateforme de Google colab, il a fallu 2 heures 38 minutes 6 secondes pour finaliser l'optimisation.

Nombre de filtres de convolution N°1 :	192
Taille du masque de convolution N°1 :	3
Nombre de filtres de convolution N°2 :	144
Taille du masque de convolution N°2 :	3
Nombre de filtres de convolution N°3 :	96
Taille du masque de convolution N°3 :	3
Nombre de neurones de la couche dense :	32
Taux d'apprentissage	0.00001
Taux de reconnaissance (Apprentissage) :	90.69 %
Taux de reconnaissance (validation)	80.07 %
Erreur d'apprentissage	0.2447
Erreur de validation	0.4462

Tableau IV-5: Les résultats obtenus apres la phase de la réapprentissage du modèle élu.

Parmi les meilleurs modèles obtenus des neuf expérimentations effectuées nous avons choisi le modèle ayant la plus petite erreur de validation comme meilleur modèle élu. L'architecture de ce modèle et les résultats obtenus après la phase de réapprentissage sur 50 itérations sont présentés dans le **Tableau IV-5**.

Les résultats obtenus semblent être très intéressants avec une précision globale à 87.50 % sur une petite base par rapport aux exigences des réseaux de neurones à convolution, pour finaliser le processus de sélection le modèle élu sera ré-entraîné sur 50 itérations, cette expérimentation a été terminée dans 10 minutes et 55 secondes.

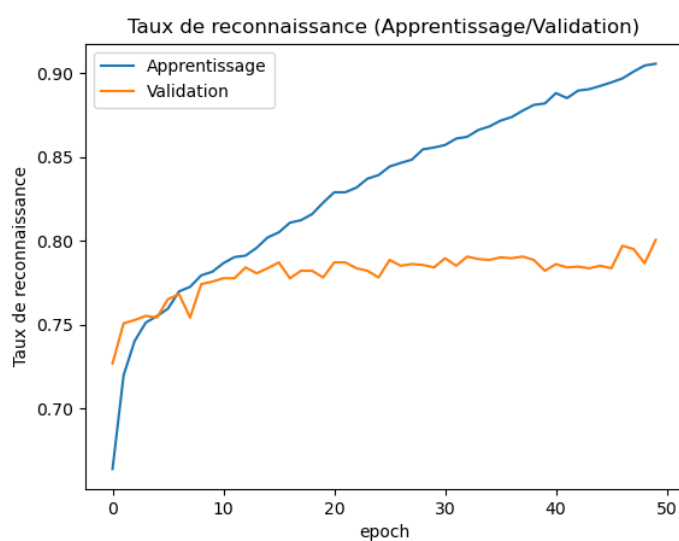


Figure IV-5: Évolution de l'apprentissage sur 50 itérations

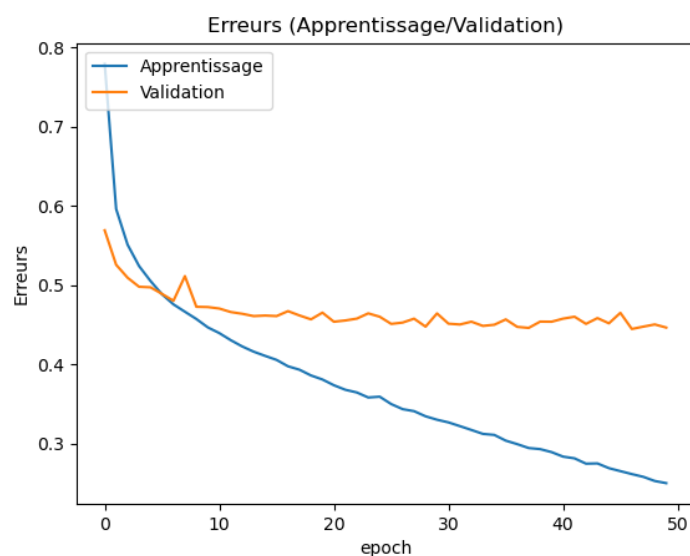


Figure IV-6: Évolution de l'erreur sur 50 itérations.

La Figure IV-5 et la Figure IV-6 représentent respectivement l'évolution du taux de reconnaissance et l'erreur sur la base d'apprentissage et de validation sur les 50 itérations.

IV.3.3. Évaluation de la performance

Afin d'évaluer les performances de notre système de classification des tissus mammographiques (Normale/ Anormale), la performance de la classification binaire doit être mesurée, pour cela nous avons calculé la matrice de confusion sur la totalité des régions d'intérêt de notre base de données. La matrice de confusion présentée dans le Tableau IV-6 montre la relation entre les différents indices de performance pour la classification des tissus mammographiques.

Matrice de confusion		Classé	
		Positive (Anormale)	Négative (Normale)
Les données réelles	Positive (Anormale) 4638 Anomalies	Vrai positive (VP) 4406	Faux positive (FP) 232
	Négative (Normale) 2050 Tissus normales	Faux négative (FN) 604	Vrai négative (VN) 1446

Tableau IV-6: Relation entre Vrai positive, Vrai négative, Faux positive et Faux négative de la matrice de confusion

La performance d'un classifieur binaire ne peut pas être décrite par une seule valeur, elle est habituellement mesurée par sa précision (le taux de classification) pendant la phase de test, c'est-à-dire la fraction de points mal-classés sur l'ensemble des cas de la base de test. La performance d'un classifieur binaire peut être mieux décrite en termes de sa sensibilité et de sa spécificité, en mesurant sa performance sur des exemples faux positifs (FP) et faux négatifs (FN) [VER01].

Pour la classification informatisée des tissus mammaires (Normale / Anormale) dans les mammographies de dépistage, les quatre indices de performance (Vrai positive, Vrai négative, Faux positive et Faux négative) du Tableau IV-7 sont calculés en comparant la production prédite du système d'apprentissage automatique avec les étiquettes réelles déterminées après une biopsie. En utilisant ces quatre mesures de performance, nous pouvons calculer des mesures relatives à la classification binaire.

La mesure de performance	Définition
Vrai positif (VP)	Tissu marquée comme Anomalie (maligne/bénigne) après une biopsie, qui est classée également comme Tissu Anomalie par la machine.
Faux positif (FP)	Tissu marquée comme Anomalie (maligne/bénigne) après une biopsie, qui est classée comme Tissu Normale par la machine.
Vrai négatif (VN)	Tissu marquée comme Normale, qui est également classée comme Normale par la machine d'apprentissage.
Faux négatif (FN)	Tissu marquée comme Normale, qui est classée comme Anormale par la machine d'apprentissage.

Tableau IV-7: Les mesures de performance de la classification des tissus (Normale / Anormale).

La sensibilité est définie comme le rapport du nombre des tissus anormale qui sont bien classées au nombre des tissus classées anormales par le système de classification, elle est donnée par :

$$\text{sensitivité} = \frac{\text{Nbr_de_positifs_bien_classés}}{\text{Nbr_Anomalies_Classés_Positifs}} = \frac{VP}{VP + FN} \quad \text{Équation IV-1}$$

$$\text{La sensibilité} = \frac{4406}{4406 + 604} = 87.94\%$$

La spécificité est définie comme le rapport des nombre des tissus normales bien classées, à toutes les tissus classées normale par le système de classification, elle est donnée par :

$$\text{spécificité} = \frac{\text{Nbr_de_négatifs_bien_classés}}{\text{Nbr_Anomalies_Classés_Négatifs}} = \frac{VN}{VN + FP} \quad \text{Équation IV-2}$$

$$\text{La spécificité} = \frac{1446}{1446 + 232} = 86.17\%$$

La précision globale est le rapport entre le nombre total des tissus correctement classés à la taille de la base de test, elle est donnée par :

$$précision = \frac{Nbr_de_cas_bien_classés}{Nbr_Total_des_Anomalies} = \frac{VP + VN}{VN + VP + FN + FP} \quad \text{Équation IV-3}$$

$$La\ précision = \frac{4406 + 232}{6688} = 87.50\%$$

Ces résultats montrent que ce modèle possède une très bonne valeur de précision et sensibilité aux tissus anormales, et cela par rapport à la taille de notre base de données et aux exigences des réseaux de neurones à convolution.

IV.3.4. Déploiement de l'application

Pour déployer notre application, nous avons opté pour le framework de développement web en Python Flask. Il en existe d'autres, le plus connu d'entre eux est Django. Ce framework nous permettra de créer un site web à usage principalement privé.

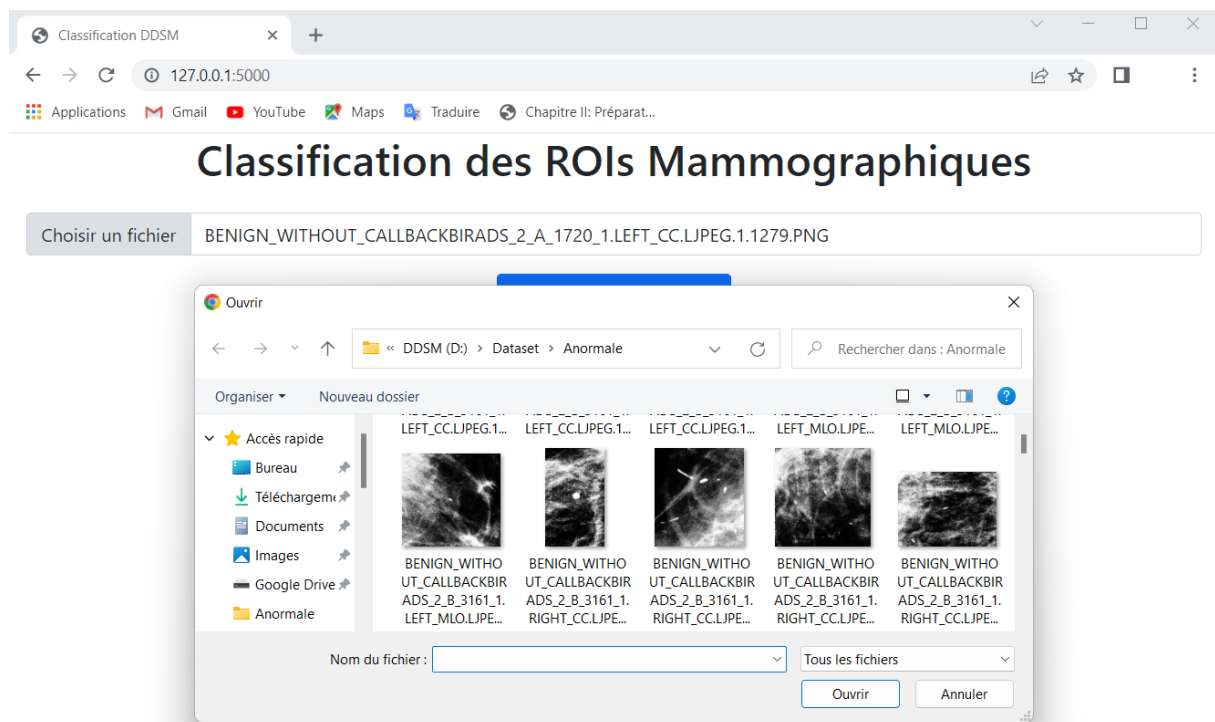


Figure IV-7: Interface graphique (la sélection des régions d'intérêts).

Le langage de programmation Python et un langage très riche en bibliothèques et Framework, on peut créer assez facilement des applications. Forcément, plus l'application que nous voulons de réaliser sera compliquée et plein de fonctionnalités, plus le temps qu'il nous

faudra pour la coder sera long. Mais avec le framework Flask, on peut obtenir des résultats, sans trop se perdre en profitant des ressources et les technologies de développement web.

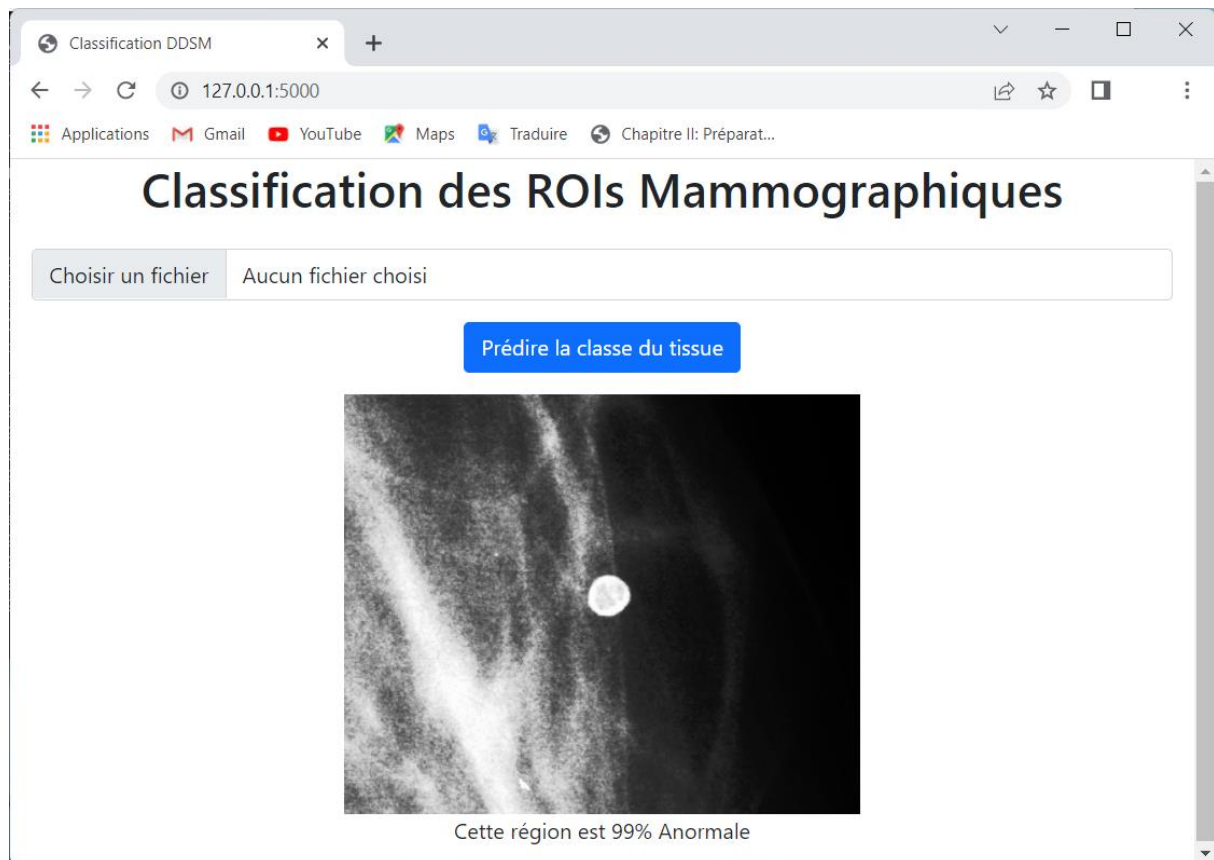


Figure IV-8: Interface graphique (Résultat de classification de la région d'intérêt sélectionnée).

Cette interface utilisateur permettra à l'utilisateur de sélectionner une région d'intérêt ROI, et de prédire s'il s'agit d'une anomalie ou non.

IV.4. Conclusion

Les techniques de diagnostic assisté par ordinateur sont des outils de calcul qui sont utilisés par les radiologues, qui vise à améliorer la détection et le diagnostic correct des anomalies mammaires. Les systèmes d'aide aux diagnostics sont des algorithmes computationnels qui caractérisent les tissus par une analyse automatique à travers les algorithmes de classification. Dans ce travail nous avons proposé un hypermodel pour régler (optimiser) les hyperparamètres d'un modèle de classification des tissus mammaires avec les techniques de l'apprentissage profonds. Cette optimisation se considère comme un problème sur le choix des hyperparamètres des réseaux à convolution.

Le réglage d'hyperparamètres nous permettra de construire un modèle d'apprentissage plus optimale, pour cela nous sommes revenus dans ce chapitre sur la définition et les différents types d'hyperamètres. Puis nous avons présenté une architecture qui nous permettra de choisir les hyperparamètres d'un réseau de neurone à convolution à travers la plateforme Keras-tuner; neuf expérimentations ont été effectuées et évaluées sur une dataset de 6688 régions d'intérêt des tissus mammographiques de la base de données numérique pour le dépistage (DDSM). Le modèle élu après cette phase d'expérimentations nous permettra de distinguer entre les tissus mammographiques, ce système qui représente un système d'aide au diagnostic qui peut être appliqué dans le domaine de détection des anomalies mammaires.

CONCLUSION GENERALE

Le cancer du sein est le cancer le plus fréquent chez les femmes. Il représente plus du tiers de l'ensemble des nouveaux cas de cancer, où la mammographie est considéré comme la modalité d'imagerie la plus efficace pour la détection et le diagnostic du cancer du sein. Des études rétrospectives montrent que, dans les dépistages actuels du cancer du sein, environ 30 % des cas de cancer du sein sont manqués par les radiologues. Avec les progrès des techniques de traitement et d'analyse d'images, il est envisagé que les radiologistes auront la possibilité de diminuer cette marge d'erreur et donc d'améliorer leur diagnostic.

L'apprentissage automatique fait partie des principales techniques de l'IA. Il permet de réaliser des prédictions en se basant sur une base de données en vue d'automatiser des tâches. L'un des avantages des modèles de l'apprentissage automatique et de l'apprentissage profond, est qu'ils sont facilement adaptables à tous les problèmes, cette flexibilité les rend très populaires. La contrepartie est qu'ils sont assez difficiles à paramétrer, tant le nombre de degrés de liberté est élevé pour certains modèles. Et même si un premier entraînement avec des paramètres arbitraires nous offre de bons résultats, nous pourrions améliorer encore ces performances en optimisant les paramètres de notre modèle. L'optimisation est le problème du choix d'hyperparamètres. Le réglage d'hyperparamètres nous permettra de réaliser un modèle d'apprentissage plus optimale. Un hyperparamètre est une caractéristique dont la valeur est utilisée pour contrôler le processus d'apprentissage.

Le but de ce travail est de proposer un système d'aide au diagnostic pour différencier entre les tissus normales et anormales dans les mammographies de dépistages. Dans ce projet de fin d'étude nous avons présenté une architecture qui nous permettra de choisir les hyperparamètres optimales d'un réseau de neurone à convolution ; les expérimentations effectuées dans ce projet ont été testé sur des régions d'intérêt (ROIs) de la base de données mammographique DDSM (The Digital Database for Screening Mammography). Le modèle élu après la phase des expérimentations nous permettra de distinguer entre les tissus mammographiques normale et anormale) avec une précision globale de 87.50 %, ce système qui représente un système d'aide au diagnostic qui peut être appliqué dans le domaine de détection des anomalies mammographiques. Les expérimentations effectuées ont été évaluées sur 6688 régions d'intérêt des tissus mammographiques extraites de la base de données numérique pour le dépistage (DDSM).

Cependant dans les travaux futurs, il peut être suggéré d'utiliser une très grande base de données et d'optimiser le modèle sur plusieurs couches de convolutions, sur plusieurs fonctions d'activations et sur plusieurs optimiseurs, ce qui nous permettra de traiter les tissus selon d'autres classes comme la classification (malignes/ bénignes) ou la classification BI-RADS.

Références bibliographiques

- [AGG22] Aggarwal, A., Bansal, M., Verma, T., & Sood, A. (2022). Residual in Residual Cascade Network for Single-Image Super Resolution. In *Advances in Information Communication Technology and Computing* (pp. 335-346). Springer, Singapore.
- [AHU20] Ahuja, R., Chug, A., Gupta, S., Ahuja, P., & Kohli, S. (2020). Classification and clustering algorithms of machine learning with their applications. In *Nature-inspired computation in data mining and machine learning* (pp. 225-248). Springer, Cham.
- [ALT20] Altan, G. (2020). Deep Learning-based Mammogram Classification for Breast Cancer. *International Journal of Intelligent Systems and Applications in Engineering*, 8(4), 171-176.
- [BAI19] Bai, M., Liu, H., Chen, H., Gu, S., & Zhang, Z. (2019, October). An improved algorithm for radar adaptive beamforming based on machine learning. In *Journal of Physics: Conference Series* (Vol. 1325, No. 1, p. 012114). IOP Publishing.
- [BAR20] Bargarai, F., Abdulazeez, A., Tiryaki, V., & Zeebaree, D. (2020). Management of wireless communication systems using artificial intelligence-based software defined radio.
- [BIL16] Billard, A. G., Calinon, S., & Dillmann, R. (2016). *Learning from humans*. Springer handbook of robotics, 1995-2014.
- [BOT14] Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1717-1724).
- [BOU16] Bouillon, M. (2016). *Apprentissage actif en-ligne d'un classifieur évolutif, application à la reconnaissance de commandes gestuelles* (Thèse de doctorat, INSA de Rennes).
- [BOU22] Boulila, W., Driss, M., Alshantqi, E., Al-Sarem, M., Saeed, F., & Krichen, M. (2022). Weight Initialization Techniques for Deep Learning Algorithms in Remote Sensing: Recent Trends and Future Perspectives. *Advances on Smart and Soft Computing*, 477-484.
- [CHA13] Chamroukhi, F. (2013). *Classification supervisée: Les K-plus proches voisins*. Mémoire de fin d'étude, Université du Sud Toulon-Var.
- [CHA19] Chandra, K., Meijer, E., Andow, S., Arroyo-Fang, E., Dea, I., George, J., ... & Yang, S. (2019). Gradient descent: The ultimate optimizer. *arXiv preprint arXiv:1909.13371*.
- [CHE20] Chen, P., Ma, T., Qin, X., Xu, W., & Zhou, S. (2020). Data-efficient semi-supervised learning by reliable edge mining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9192-9201).
- [CHO17] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251-1258).
- [CHO21] Chollet, F. (2021). *Deep learning with Python*. Simon and Schuster.
- [COR11] Cornuéjols, A., & Miclet, L. (2011). *Apprentissage artificielle : concepts et algorithmes*. Editions Eyrolles.

- [COS17] Coşkun, M., YILDIRIM, Ö., Ayşegül, U. Ç. A. R., & Demir, Y. (2017). An overview of popular deep learning methods. *European Journal of Technique (EJT)*, 7(2), 165-176.
- [CUI18] Cui, X., Zhang, W., Tüske, Z., & Picheny, M. (2018). Evolutionary stochastic gradient descent for optimization of deep neural networks. *Advances in neural information processing systems*, 31.
- [DEE20] Deeba, K., & Amutha, B. (2020). ResNet-deep neural network architecture for leaf disease classification. *Microprocessors and Microsystems*, 103364.
- [DEN20] Deng, Z., Cao, Y., Zhou, X., Yi, Y., Jiang, Y., & You, I. (2020). Toward efficient image recognition in sensor-based IoT: a weight initialization optimizing method for CNN Based on RGB influence proportion. *Sensors*, 20(10), 2866.
- [DOG18] Dogo, E. M., Afolabi, O. J., Nwulu, N. I., Twala, B., & Aigbavboa, C. O. (2018, December). A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks. In *2018 international conference on computational techniques, electronics and mechanical systems (CTEMS)* (pp. 92-99). IEEE.
- [ELA21] El Adoui, M. (2021). Prédiction de l'efficacité de la chimiothérapie appliquée au cancer du sein par le traitement d'images et le Deep Learning (Doctoral dissertation, Université de Mons (UMONS)).
- [FAT20] Fatima, N. (2020). Enhancing performance of a deep neural network: A comparative analysis of optimization algorithms. *Advances in Distributed Computing and Artificial Intelligence Journal*, 9(2), 79-90.
- [GOE21] Goel, R., Mehmood, I., & Ugail, H. (2021). A Study of Deep Learning-Based Face Recognition Models for Sibling Identification. *Sensors*, 21(15), 5068.
- [GRI20] Gridi, A., & Guerram, T. (2020). Un Outil de Deep Learning pour les données textuelles. *Mémoire de master, Université Oum El Bouaghi*.
- [GUO20] Guo, L., Li, M., Xu, S., & Yang, F. (2020, August). Application of Stochastic Gradient Descent Technique for Method of Moments. In *2020 IEEE International Conference on Computational Electromagnetics (ICCEM)* (pp. 97-98). IEEE.
- [GUR18] Gurney, K. (2018). *An introduction to neural networks*. CRC press.
- [GYL17] Gylberth, R., Adnan, R., Yazid, S., & Basaruddin, T. (2017, October). Differentially private optimization algorithms for deep neural networks. In *2017 International Conference on Advanced Computer Science and Information Systems (ICACISIS)* (pp. 387-394). IEEE.
- [HAJ21] Haji, S. H., & Abdulazeez, A. M. (2021). Comparison of optimization techniques based on gradient descent algorithm: A review. *PalArch's Journal of Archaeology of Egypt/Egyptology*, 18(4), 2715-2743.
- [HAL17] Hallen, R. (2017). *A study of gradient-based algorithms*. *Memoire de Master. Université de Lund (Suède)*
- [HAR16] Hardt, M., Recht, B., & Singer, Y. (2016, June). Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning* (pp. 1225-1234). PMLR.
- [HEA00] Heath, M., Bowyer, K., Kopans, D., Moore, R., & Kegelmeyer, W. P. (Juin 2000). The digital database for screening mammography. *Proceedings of the 5th international workshop on digital mammography*. Medical Physics Publishing. 212-218. Toronto (Canada).

- [HEA98] Heath, M., Bowyer, K., Kopans, D., Kegelmeyer Jr, P., Moore, R., Chang, K., & Munishkumaran, S. (1998). Current status of the digital database for screening mammography. *Computational Imaging and Vision*, 13 (1), 457-460.
- [HEE21] Heenaye-Mamode Khan, M., Boodoo-Jahangeer, N., Dullull, W., Nathire, S., Gao, X., Sinha, G. R., & Nagwanshi, K. K. (2021). Multi-class classification of breast cancer abnormalities using Deep Convolutional Neural Network (CNN). *Plos one*, 16(8), e0256500.
- [JIA14] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... & Darrell, T. (2014, November). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 675-678).
- [JIA18] Jiang, Y. G., Wu, Z., Tang, J., Li, Z., Xue, X., & Chang, S. F. (2018). Modeling multimodal clues in a hybrid deep learning framework for video classification. *IEEE Transactions on Multimedia*, 20(11), 3137-3147.
- [JOU18] Jouni, H. (2018). Cellules analogiques CMOS pour réseaux de neurones. Application à la classification des cellules cancéreuses dans le sein (Doctoral dissertation, COMUE Université Côte d'Azur (2015-2019); Lebanese international university).
- [KAO17] Kaoudi, Z., Quiané-Ruiz, J. A., Thirumuruganathan, S., Chawla, S., & Agrawal, D. (2017, May). A cost-based optimizer for gradient descent optimization. In *Proceedings of the 2017 ACM International Conference on Management of Data* (pp. 977-992).
- [KAU20] Kaur, S., Aggarwal, H., & Rani, R. (2020). Hyper-parameter optimization of deep learning model for prediction of Parkinson's disease. *Machine Vision and Applications*, 31(5), 1-15.
- [KAV09] Kavukcuoglu, K., Ranzato, M. A., Fergus, R., & LeCun, Y. (2009, June). Learning invariant features through topographic filter maps. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 1605-1612). IEEE.
- [KEM19] Kemal, A. D. E. M., & Kilicarslan, S. (2019, October). Performance analysis of optimization algorithms on stacked autoencoder. In *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)* (pp. 1-4). IEEE.
- [KON17] Kong, C., & Lucey, S. (2017). Take it in your stride: Do we need striding in CNNs?. *arXiv preprint arXiv:1712.02502*.
- [KRA16] Kramer, O. (2016). *Machine learning for evolution strategies* (Vol. 20). Heidelberg: Springer.
- [KRI12] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- [KUL22] Kulkarni, A., Shivananda, A., & Kulkarni, A. (2022). Natural Language Processing and Artificial Intelligence Overview. In *Natural Language Processing Projects* (pp. 1-37). Apress, Berkeley, CA.
- [KUM22] Kumar Singh, K., Kumar, S., Antonakakis, M., Moirogiorgou, K., Deep, A., Kashyap, K. L., ... & Zervakis, M. (2022). Deep learning capabilities for the categorization of microcalcification. *International Journal of Environmental Research and Public Health*, 19(4), 2159.
- [LEV05] Lévy, L., Suissa, M., Bokobsa, J., Tristant, H., Chiche, J. F., Martin, B., & Teman, G. (2005). Présentation de la traduction française du BI-RADS® (Breast Imaging Reporting System and Data System). *Gynécologie obstétrique & fertilité*, 33 (5), 338-347.

- [LID18] Li, D., Chen, C., Lv, Q., Gu, H., Lu, T., Shang, L., ... & Chu, S. M. (2018, April). AdaError: An adaptive learning rate method for matrix approximation-based collaborative filtering. In Proceedings of the 2018 World Wide Web Conference (pp. 741-751).
- [LIH19] Li, H., Zhuang, S., Li, D. A., Zhao, J., & Ma, Y. (2019). Benign and malignant classification of mammogram images based on deeplearning. *Biomedical Signal Processing and Control*, 51, 347-354.
- [LIH20] Li, H., Krček, M., & Perin, G. (2020, October). A comparison of weight initializers in deep learning-based side-channel analysis. In *International Conference on Applied Cryptography and Network Security* (pp. 126-143). Springer, Cham.
- [LIL17] Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1), 6765-6816.
- [LIU19] Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., & Han, J. (2019). On the variance of the adaptive learning rate and beyond. arXiv preprint arXiv:1908.03265.
- [LUT13] Lutz, M. (2013). *Learning python: Powerful object-oriented programming*. " O'Reilly Media, Inc."
- [MAA13] Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013, June). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml* (Vol. 30, No. 1, p. 3).
- [MAT10] Mathieu-Dupas, E. (2010). Algorithme des k plus proches voisins pondérés et application en diagnostic. In *42èmes Journées de Statistique*.
- [MAT17] Mathivet, V. (2017). *L'intelligence artificielle pour les développeurs: concepts et implémentations en C#*. Éditions ENI.
- [MAZ15] Mazac, S. (2015). *Approche décentralisée de l'apprentissage constructiviste et modélisation multi-agent du problème d'amorçage de l'apprentissage sensorimoteur en environnement continu: application à l'intelligence ambiante* (Doctoral dissertation, Université Claude Bernard-Lyon I).
- [MIT21] Mittal, S. (2021). A survey of accelerator architectures for 3D convolution neural networks. *Journal of Systems Architecture*, 115, 102041.
- [MOH18] Mohammed, N. N., Cawthorne, M., & Abdulazeez, A. M. (2018). Detection of Genes Patterns with an Enhanced Partitioning-Based DBSCAN Algorithm. *Journal of information and communication engineering*, 4(1), 188-195.
- [MON12] Montavon, G., Orr, G., & Müller, K. R. (Eds.). (2012). *Neural networks: tricks of the trade* (Vol. 7700). springer.
- [MOO19] Moolayil, J., Moolayil, J., & John, S. (2019). *Learn Keras for deep neural networks* (pp. 1-192). Birmingham: Apress.
- [NAI10] Nair, V., & Hinton, G. E. (2010, January). Rectified linear units improve restricted boltzmann machines. In *Icml*.
- [NWA20] Nwankpa, C. E. (2020). Advances in optimisation algorithms and techniques for deep learning. *Advances in Science, Technology and Engineering Systems Journal*, 5(5), 563-577.
- [ORE99] Orel, S. G., Kay, N., Reynolds, C., & Sullivan, D. C. (1999). BI-RADS categorization as a predictor of malignancy. *Radiology*, 211 (3), 845-850.

- [OTO22] Otoum, Y., Liu, D., & Nayak, A. (2022). DL-IDS: a deep learning–based intrusion detection framework for securing IoT. *Transactions on Emerging Telecommunications Technologies*, 33(3), e3803.
- [OUA20] Ouali, Y., Hudelot, C., & Tami, M. (2020). An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*.
- [OZT20] Öztürk, M. M., Cankaya, I. A., & Ipekci, D. (2020). Optimizing echo state network through a novel fisher maximization based stochastic gradient descent. *Neurocomputing*, 415, 215-224.
- [QUZ19] Qu, Z., Yuan, S., Chi, R., Chang, L., & Zhao, L. (2019). Genetic optimization method of pantograph and catenary comprehensive monitor status prediction model based on adadelta deep neural network. *IEEE Access*, 7, 23210-23221.
- [REH21] Rehman, K. U., Li, J., Pei, Y., Yasin, A., Ali, S., & Mahmood, T. (2021). Computer vision-based microcalcification detection in digital mammograms using fully connected depth wise separable convolutional neural network. *Sensors*, 21(14), 4854.
- [RUD16] Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.
- [RUS20] Russell, R. (2020). *Machine Learning: Step-By-Step Guide to Implement Machine Learning Algorithms with Python*. (Knxb).
- [SAD19] Sadeeq, H., & Abdulazeez, A. M. (2018, October). Hardware implementation of firefly optimization algorithm using FPGAs. In *2018 International Conference on Advanced Science and Engineering (ICOASE)* (pp. 30-35). IEEE.
- [SAE20] Saeed, F., Al-Hadhrami, T., Mohammed, F., & Mohammed, E. (2020). *Advances on Smart and Soft Computing*. Published: Apr, 12.
- [SCH22] Schubert, M., Sonderegger, W., Luković, M., & Kläusler, O. (2022). Semi-Supervised Learning For Quality Control Of High-Value Wood Products.
- [SHE19] Shen, L., Margolies, L. R., Rothstein, J. H., Fluder, E., McBride, R., & Sieh, W. (2019). Deep learning to improve breast cancer detection on screening mammography. *Scientific reports*, 9(1), 1-12.
- [SHI19] Shi, P., Wu, C., Zhong, J., & Wang, H. (2019, August). Deep learning from small dataset for BI-RADS density classification of mammography images. In *2019 10th International Conference on Information Technology in Medicine and Education (ITME)* (pp. 102-109). IEEE.
- [SID21] Siddeeq, S., Li, J., Bhatti, H. M. A., Manzoor, A., & Subhan Malhi, U. (2021, January). Deep Learning RN-BCNN Model for Breast Cancer BI-RADS Classification. In *2021 The 4th International Conference on Image and Graphics Processing* (pp. 219-225).
- [SIZ19] Si, Z., Wen, S., & Dong, B. (2019). NOMA codebook optimization by batch gradient descent. *IEEE Access*, 7, 117274-117281.
- [SOL20] Solanke, A. V., & Patnaik, G. K. (2020). Intrusion Detection using Deep Learning Approach with Different Optimization. *International Journal for Research in Applied Science and Engineering Technology*, 8(5), 128-134.
- [SRI13] Srivastava, N., & Salakhutdinov, R. R. (2013). Discriminative transfer learning with tree-based priors. *Advances in neural information processing systems*, 26.

- [TAM21] Tamenu, Y. (2021). Proptech: la Data Science appliquée à l'immobilier (Doctoral dissertation, Haute école de gestion de Genève).
- [TEO22] Teoh, T. T., & Rong, Z. (2022). Artificial Intelligence with Python. Springer Nature.
- [TSA22] Tsai, K. J., Chou, M. C., Li, H. M., Liu, S. T., Hsu, J. H., Yeh, W. C., ... & Hwang, S. H. (2022). A High-Performance Deep Neural Network Model for BI-RADS Classification of Screening Mammography. *Sensors*, 22(3), 1160.
- [VAN19] Vani, S., & Rao, T. M. (2019, April). An experimental approach towards the performance assessment of various optimizers on convolutional neural network. In 2019 3rd international conference on trends in electronics and informatics (ICOEI) (pp. 331-336). IEEE.
- [VAN20] Vanderheyden, R., & Xie, Y. (2020, August). Mammography image BI-RADS classification using ohplall. In 2020 IEEE Sixth International Conference on Big Data Computing Service and Applications (BigDataService) (pp. 120-127). IEEE.
- [VER01] Veropoulos, K. (2001). Machine learning approaches to medical decision making (Thèse pour obtenir le grade de Docteur), University de Bristol.
- [VIA21] Vialaneix, N., Brouard, C., & Vismara, P. (2021). Réseaux de neurones pour graphe pour la prédiction de phénotypes. Montpellier SupAgro.
- [VIL21] Villalobos-Arias, L., & Quesada-López, C. (2021, August). Comparative study of random search hyper-parameter tuning for software effort estimation. In Proceedings of the 17th International Conference on Predictive Models and Data Analytics in Software Engineering (pp. 21-29).
- [VIN03] Vincent, P. (2003). Modèles à noyaux à structure locale. (Thèse de doctorat, Université de montréal).
- [WAD52] Wadley, F. M. (1952). Probit analysis: a statistical treatment of the sigmoid response curve.
- [WAN20] Wang, X., Liang, G., Zhang, Y., Blanton, H., Bessinger, Z., & Jacobs, N. (2020). Inconsistent performance of deeplearning models on mammogram classification. *Journal of the American College of Radiology*, 17(6), 796-803.
- [YAN19] Yang, Z., Wang, C., Zhang, Z., & Li, J. (2019). Mini-batch algorithms with online step size. *Knowledge-Based Systems*, 165, 228-240.
- [YAQ20] Yaqub, M., Feng, J., Zia, M. S., Arshid, K., Jia, K., Rehman, Z. U., & Mehmood, A. (2020). State-of-the-art CNN optimizer for brain tumor segmentation in magnetic resonance images. *Brain Sciences*, 10(7), 427.
- [YID19] Yi, D., Ji, S. et Bu, S. (2019). Un schéma d'optimisation amélioré basé sur des méthodes de descente de gradient pour l'apprentissage automatique. *Symétrie*, 11 (7), 942.
- [YID20] Yi, D., Ahn, J., & Ji, S. (2020). An effective optimization method for machine learning based on ADAM. *Applied Sciences*, 10(3), 1073.
- [ZEB20] Zebari, D. A., Zeebaree, D. Q., Saeed, J. N., Zebari, N. A., & Adel, A. Z. (2020). Image steganography based on swarm intelligence algorithms: A survey. *people*, 7(8), 9.
- [ZEE19] Zeebaree, D. Q., Haron, H., Abdulazeez, A. M., & Zebari, D. A. (2019, April). Machine learning and region growing for breast cancer segmentation. In 2019 International Conference on Advanced Science and Engineering (ICOASE) (pp. 88-93). IEEE.

- [ZEE20] Zeebaree, D. Q., Abdulazeez, A. M., Hassan, O. M. S., Zebari, D. A., & Saeed, J. N. (2020). Hiding Image by Using Contourlet Transform. vol, 83, 16979-16990.
- [ZEE21] Zeebaree, D. Q. (2021). Robust watermarking scheme based LWT and SVD using artificial bee colony optimization. Indonesian Journal of Electrical Engineering and Computer Science, 21(2), 1218-1229.
- [ZER15] ZERHOUNI, M. N. (2015). Sélection de Données pour l'Apprentissage des Réseaux de Neurones, Arbres de Décision et les k-Plus Proches Voisins: Application en Diagnostic de Pannes (Doctoral dissertation, Université de Tlemcen).
- [ZHA18] Zhang, N., Lei, D., & Zhao, J. F. (2018). An improved Adagrad gradient descent optimization algorithm. In 2018 Chinese Automation Congress (CAC) (pp. 2359-2362). IEEE.
- [ZHA19] Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), 3212-3232.
- [ZOU19] Zou, F., Shen, L., Jie, Z., Zhang, W., & Liu, W. (2019). A sufficient condition for convergences of adam and rmsprop. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11127-11135).

Webographie

- [ARM22] Armin Ronacher <https://flask.palletsprojects.com> [Consulté le 13/05/2022]
- [OMA19] O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., & Invernizzi, L. (2019). Keras tuner. Disponible sur : github.com/keras-team/kerastuner. [Consulté le 10/01/2022]
- [YAN22] Yannick Serge <https://yannicksergeobam.medium.com> [Consulté le 05/05/2022]