



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ DES MATHÉMATIQUES ET DE L'INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Génie Informatique

Par :

BENNADJI Mohamed
MAATOUG Taha Mokhtar

Sur le thème

Détection précoce de la rétinopathie diabétique à partir d'images de fond d'œil en utilisant le Deep Learning

Soutenu publiquement le /06/2025 à Tiaret devant le jury composé de :

Mr. BAGHDADI Mohamed	Grade Université	MCA	Encadrant
Mr. CHENINE Abdelkader	Grade Université	MAA	Examineur
Mr. MEZZOUG Karim	Grade Université	MAA	Examineur

2024 - 2025

Remerciements :

Nous tenons à remercier en premier lieu Dieu le Tout-Puissant de nous avoir aidés à réaliser ce modeste travail.

C'est avec joie et gratitude que nous remercions toutes les personnes ayant initié, coopéré ou contribué de près ou de loin à la réalisation de ce projet.

Nous remercions cordialement nos parents et tous les membres de nos familles, qui nous ont donné la volonté et le courage d'accomplir ce parcours.

Un grand merci à notre encadrant, Mr. BAGHDADI MOHAMED, pour son aide précieuse.

Nous remercions tout particulièrement les professeurs de l'université, notamment ceux du département d'informatique.

Dédicace :

Je dédie ce travail.

À mes chers parents,

À toute la famille BENNADJI,

À tous mes cher(e)s ami(e)s,

À toutes celles ou tous ceux qui m'ont aidé de près ou de loin.

BENNADJI MOHAMED

Dédicace :

Je dédie ce travail.

À mes chers parents,

À mes chères sœurs et à mon cher frère,

À toute la famille MAATOUG,

À tous mes cher(e)s ami(e)s.

À toutes celles ou tous ceux qui m'ont aidé de près ou de loin.

MAATOUG TAHA MOKHTAR

Résumé

La rétinopathie diabétique est une complication grave du diabète, pouvant entraîner une perte de vision irréversible si elle n'est pas détectée à temps. Face aux limites des méthodes traditionnelles de dépistage, ce travail vise à développer un système intelligent capable d'identifier automatiquement les signes de la rétinopathie diabétique à partir d'images de la rétine. Pour cela, nous avons utilisé des techniques d'apprentissage profond, en particulier les réseaux de neurones convolutifs (CNN), réputés pour leur efficacité dans le domaine de l'imagerie médicale.

Nous avons conçu un modèle personnalisé ainsi que plusieurs modèles basés sur le transfert learning en utilisant des architectures pré-entraînées telles que VGG19, EfficientNet, InceptionV3 et ResNet152. Les performances de chaque modèle ont été analysées en termes de précision et de fonction de perte. Malgré certaines limitations liées au dataset, nos résultats montrent que ces approches peuvent aider efficacement au dépistage de la maladie et constituent une base solide pour des améliorations futures.

Mots-clés : Deep Learning, CNN, Transfert Learning, Rétinopathie Diabétique, Imagerie Médicale, TensorFlow, Keras.

Abstract

Diabetic retinopathy is a serious complication of diabetes that can lead to irreversible vision loss if not detected early. Due to the limitations of traditional screening methods, this work aims to develop an intelligent system capable of automatically identifying signs of diabetic retinopathy from retinal images. To achieve this, we used deep learning techniques, particularly convolutional neural networks (CNNs), which are known for their effectiveness in medical imaging.

We designed a custom model as well as several models based on transfer learning, using pre-trained architectures such as VGG19, EfficientNet, InceptionV3, and ResNet152. The performance of each model was analyzed in terms of accuracy and loss function. Despite certain limitations related to the dataset, our results show that these approaches can effectively assist in disease screening and provide a solid foundation for future improvements.

Keywords: Deep Learning, CNN, Transfer Learning, Diabetic Retinopathy, Medical Imaging, TensorFlow, Keras.

TABLE DES MATIERES

Introduction générale.....	1
CHAPITRE 01 (Généralités sur la rétinopathie diabétique et le traitement d'images)	
I.1.....	Introduction 4
I.2 diabétique.....	Rétinopathie 4

I.2.1	Rétinopathie.....				
...	4				
I.2.2	diabétique.....			Rétinopathie	5
I.2.3	Classification de la			rétinopathie	
diabétique.....					6
I.2.3.1	Rétinopathie		diabétique	non	
proliférante.....					6
A)	Rétinopathie		diabétique	non	proliférante
minime.....					6
B)	Rétinopathie		diabétique	non	proliférante
modérée.....					7
C)	Rétinopathie		diabétique	non	proliférante
sévère.....					7
I.2.3.2			Rétinopathie		diabétique
proliférante.....					7
A)					
Débutante.....					7
B)					
Modérée.....					8
C)					
Sévère.....					8
I.2.4				Fond	
d'œil.....					8
I.2.4.1	Examen		du	fond	
d'œil.....					8
I.2.5			Fond	d'œil	et
diabète.....					9
I.2.5.1			Diabète	de	type
1.....					9
I.2.5.2			Diabète	de	type
2.....					10
I.2.6	Diagnostic de la		rétinopathie	diabétique	sur le plan
clinique.....					10
I.2.6.1					
Microanévrismes.....					1
0					
I.2.6.2			Hémorragies		rétiniennes
punctiformes.....					10
I.2.6.3					Nodules
cotonneux.....					11
I.2.6.4			Hémorragies		en
flammèche.....					11
I.2.6.5			Hémorragies		en
tâche.....					11

I.2.6.6	Anomalies de la microcirculation intra-rétinienne (AMIR).....	12
I.2.6.7	Anomalies veineuses.....	12
I.2.6.8	Néo-vaisseaux.....	13
I.2.6.9	Œdème maculaire.....	13
I.2.6.10	Exsudats.....	13
I.3	Traitement d'images.....	14
I.3.1	Notions de base.....	14
I.3.1.1	Images.....	14
I.3.1.2	Image numérique.....	15
A)	Définition de l'image numérique.....	15
B)	Représentation des images numériques.....	15
B.1)	Images bitmap (ou matricielles).....	15
B.2)	Images vectorielles.....	16
C)	Caractéristiques des images numériques.....	16
C.1)	Pixel.....	16
C.2)	Voisinage.....	17
C.3)	Contraste.....	17
C.4)	Luminance.....	18
C.5)	Résolution d'une image.....	18
C.6)	Bruit.....	18
C.7)	Textures.....	18

C.8)	Niveaux	de
gris ...		19
C.9)		
Couleurs...		19
I.3.2	Traitement	
d'images.....		19
I.3.2.1 Définition du traitement		
d'images.....		19
I.3.2.2 Exemples de traitement		
d'images.....		21
I.3.2.3 Système de traitement		
d'images.....		21
A) Acquisition et		
numérisation.....		21
B) Prétraitement des		
images.....		22
B.1) Filtre		
gaussien.....		23
B.2) Filtre		
anisotrope.....		24
B.3) Filtre		
médian.....		24
C) Segmentation des		
images.....		25
C.1) Objectifs de la		
segmentation.....		25
C.2) Méthodes de		
segmentation.....		25
D) Interprétation des		
images.....		26
I.4		
Conclusion.....		26

CHAPITRE 02 (Deep learning)

II.1 Introduction	28
II.2 Définition du Deep Learning..	29
II.3 Pourquoi le Deep Learning?...	29
II.4 Domaines d'application du Deep Learning	30
II.5 Principe et fonctionnement du Deep Learning...	31
II.6 Réseaux de neurones classiques..	31

II.6.1 Définition d'un réseau de neurones.....	33
II.6.2 Neurone biologique.....	33
II.6.2.1 Cellule nerveuse.....	33
II.6.2.2 Corps cellulaire....	34
II.6.2.3 Dendrites....	3
4	
II.6.2.4 Axone....	3
4	
II.6.2.5 Transmission synaptique.....	34
II.6.3 Réseau de neurones artificiels.....	35
II.6.3.1 Réseaux « Feed-Forward ».....	36
II.6.3.2 Réseaux « Feed-Back »..	37
II.6.4 Apprentissage.....	38
II.6.4.1 Apprentissage supervisé.....	38
II.6.4.2 Apprentissage par renforcement.	38
II.6.4.3 Apprentissage non-supervisé (ou auto-organisationnel).....	38
II.6.4.4 Apprentissage hybride.....	38
II.6.5 Neurone formel.....	38
II.6.5.1 Interprétation mathématique.....	39
II.6.5.2 Fonctions d'activation.....	40
II.6.6 Réseau de neurones multicouches....	40
II.7 Réseaux de neurones convolutifs.....	42
II.7.1 Architecture des CNNs.....	43
II.7.1.1 Couche d'entrée.....	44
II.7.1.2 Couche de convolution.....	45

II.7.1.3 Couche de pooling...	46
II.7.1.4 Couche de normalisation (Batch normalisation).....	47
II.7.1.5 Couche d'aplatissement (Flatten).....	47
II.7.1.5 Couche entièrement connectée	48
II.7.1.6 Couche de sortie	49
II.7.2 Autres composantes et techniques utilisées dans les CNNs.....	50
II.7.2.1 Fonction de perte (ou fonction d'erreur).....	50
II.7.2.2 Abandon aléatoire (Dropout)....	50
II.7.2.3 Normalisation par lots (Batch Normalization)	51
II.7.2.4 Arrêt anticipé (Early Stopping).....	51
II.7.2.5 Augmentation des données (Data Augmentation).....	52
II.7.2.6 Apprentissage par transfert (Transfer Learning).....	52
II.7.3 Optimisation dans les CNN.....	52
II.7.3.1 Descente de gradient stochastique... ..	52
II.7.3.2 Optimiseur RMSprop.....	53
II.7.3.3 Optimiseur Adam... ..	53
II.7.3 Avantages et inconvénients des CNN.....	53
II.7.3.1 Avantages des CNN.....	53
II.7.3.2 Inconvénients des CNN.....	54
II.8 Conclusion.....	54

CHAPITRE 03 (Expériences et résultats)

III.1 Introduction	56
III.2 Environnement de travail.....	56
III.2.1 Environnement matériel	56

III.2.2 Langage de programmation	
Python.....	56
III.2.3 Google	
Colab.....	57
III.2.4 Navigateur	
Anaconda.....	57
III.2.5 Jupyter Notebook	
.....	58
III.2.6	
Keras.....	58
III.2.7	
TensorFlow....	59
III.2.8	
OpenCV....	59
III.2.9	
Numpy...	60
III.2.10	
Pandas....	60
III.3 Dataset	
utilisé.....	61
III.3.1 Simplification de la classification ICDR dans notre	
étude.....	62
III.4 Augmentation des	
données.....	63
III.5 Modèles de détection de la	
rétinopathie.....	64
III.5.1 Architecture de notre modèle 1 (modèle	
personnalisé).....	66
III.5.2 Modèles utilisant le transfert	
learning.....	67
III.5.2.1 Transfer learning avec VGG19 (Modèle	
2).....	68
III.5.2.2 Transfer learning avec EfficientNet (Modèle	
3).....	69
III.5.2.3 Transfert learning avec InceptionV3 (Modèle	
4).....	70
III.5.2.4 Transfert learning avec ResNet152 (Modèle	
5).....	71
III.6 Résultats obtenus avec nos	
modèles.....	73
III.6.1 Résultats du modèle 1 (modèle	
personnalisé).....	73
III.6.2 Résultats obtenus avec nos modèles de transfert	
learning.....	74

III.7 Comparaison des différents résultats.....	76
III.8 Conclusion.....	
.77	

Conclusion générale.....	78
Références bibliographiques.....	80

LISTE DES FIGURES

Figure I.1 : Image d'une rétine normale observée lors d'un examen de fond d'œil (absence de rétinopathie diabétique)	
[2].....	5
Figure I.2 : Rétinopathie diabétique non proliférante minimale	
[2].....	7
Figure I.3 : Rétinopathie diabétique proliférante sévère	
[2].....	8
Figure I.4 : Examen de fond d'œil	
[8].....	9
Figure I.5 : Microanévrismes	
[11].....	10
Figure I.6 : Hémorragies rétinienne punctiformes	
[11].....	11
Figure I.7 : Nodules cotonneux	
[11].....	11
Figure I.8 : Hémorragies en taches et anomalies de la microcirculation intra-rétinienne (AMIR)	
[11].12	
Figure I.9 : Anomalies veineuses	
[11].....	12
Figure I.10 : Néo vaisseaux prépapillaires (a), et néo-vaisseaux pré-rétiniens (b)	
[11].....	13
Figure I.11 : Œdème maculaire	
[11].....	13
Figure I.12 : Les exsudats	
[11]	14
Figure I.13 : Représentation d'une image.	
.....	15
Figure I.14 : Exemple d'image en niveaux de gris, représentée par des entiers compris entre 0 (noir) et 255 (blanc)...	
..15	
Figure I.15 : Pixel d'une image.....	16
Figure I.16 : Représentation des pixels dans une image.....	16
Figure I.17 : Deux images possédant un contraste différent.....	17
Figure I.18 : Image bruitée.....	18
Figure I.19 : Différents niveaux de gris... ..	19
Figure I.20 : Chaîne de traitement d'images (exemple d'une image médicale).....	20
Figure I.21 : Exemple d'un diagramme de prétraitement des images de rétinopathie.....	22
Figure I.22 : (a) Image saine ; (a1) image saine après application du filtre gaussien; (b) image malade ; (b1) image malade après application du filtre gaussien.....	23

Figure I.23 : (a) Image RVB ; (b) Image en niveaux de gris ; (c) Image après application du filtre médian.....	24
Figure II.1 Hiérarchie entre l'intelligence artificielle, l'apprentissage automatique (ou machine learning) et le deep learning [26].....	28
Figure II.2 : Architecture d'un modèle de Deep Learning [27].....	31
Figure II.3 : Connexions inter-neuronales chez l'être humain.....	32
Figure II.4 : (a) Neurone biologique et (b) Neurone artificiel.....	32
Figure II.5 : Représentation d'une cellule nerveuse.....	34
Figure II.6 : Transmission synaptique.....	35
Figure II.7 : Neurone artificiel.....	36
Figure II.8 : Neurone formel.....	39
Figure II.9 : Représentation mathématique du neurone formel.....	39
Figure II.10 : Les fonctions d'activation : à seuil (a), linéaire par morceaux (b), sigmoïde (c) et gaussienne (d).....	40
Figure II.11 : Architecture d'un réseau de neurones multicouches avec une seule couche cachée.....	41
Figure II.12 : Représentation d'un nœud unique dans un MLP.....	41
Figure II.13 : MLP avec une seule couche cachée.....	42
Figure II.14: Architecture d'un CNN [27].	43
Figure II.15 : Différentes couches d'un CNN..	44
Figure II.16 : Trois dimensions d'une image.....	45
Figure II.17 : Exemple de convolution d'une image d'entrée I avec un filtre K.....	46
Figure II.18 : Deux principaux types de pooling (max et moyen).....	46
Figure II.19 : Opération d'aplatissement (flattening).....	48
Figure II.20 : Architecture de la couche entièrement connectée.....	48
Figure II.21 : Fonction softmax [40]	49
Figure II.22 : Fonction sigmoïde[40].....	49

Figure II.23 : Utilisation de l'arrêt anticipé durant la phase d'entraînement d'un modèle.....	51
Figure III.1 : Logo de Python.....	57
Figure III.2 : Interface de Google Colab.....	57
Figure III.3 : Navigateur Anaconda.....	58
Figure III.4 : Jupyter notebook.....	58
Figure III.5: Logo de Keras.....	59
Figure III.6: Logo de TensorFlow.....	59
Figure III.7: Exemple d'images du dataset Messidor-2.....	63
Figure III.8: Opérations d'augmentation des données.....	65
Figure III.9 : Structure de base d'un CNN.....	66
Figure III.10: Architecture du modèle 1 (modèle personnalisé).....	67
Figure III.11: Architecture du modèle de transfer learning avec VGG19.....	68
Figure III.12: Différents composants et paramètres du modèle de transfer learning avec VGG19.....	69
Figure III.13: Différents composants et paramètres du modèle de transfer learning avec InceptionV3.....	71
Figure III.14: Différents composants et paramètres du modèle de transfer learning avec ResNet152.....	72
Figure III.15: Évolution de l'accuracy et de la perte pendant l'entraînement et la validation du modèle 1.....	73
Figure III.16: Valeurs d'accuracy et de perte au cours de l'entraînement et de la validation pour le modèle 2 (transfer learning avec VGG19).....	74
Figure III.17: Valeurs d'accuracy et de perte au cours de l'entraînement et de la validation pour le modèle 3 (transfer learning avec EfficientNet).....	75
Figure III.18: Valeurs d'accuracy et de perte au cours de l'entraînement et de validation pour le modèle 4 (transfer learning avec InceptionV3).....	75
Figure III.19: Valeurs d'accuracy et de perte dans les phases d'entraînement et de validation pour le modèle 5 (transfer learning avec ResNet152).....	76

LISTE DES TABLEAUX

Table I.1 : Pourcentage de risque de développer une rétinopathie diabétique dans le monde.....	6
--	---

Table III.1 : Classification	
------------------------------	--

ICDR.....	62
-----------	----

LISTE DES ABREVIATIONS

Abréviation	Signification
RD	Rétinopathie diabétique
IA	Intelligence artificielle
CNN	Convolutional Neural Networks
AMIR	Anomalies de la microcirculation intra-rétinienne
OCT	Tomographie par cohérence optique
DL	Deep learning
RNA	Réseau de neurones artificiel
MLP	Multi layer perceptron
RBF	Radial Basis Function
BN	Batch Normalization
FC	Fully connected layer
SGD	Stochastic Gradient Descent
Adam	Adaptive Moment Estimation

Introduction Générale

La rétinopathie diabétique (RD) est une complication redoutable du diabète, affectant la rétine et pouvant entraîner une perte de vision irréversible, voire la cécité. Cette pathologie résulte d'une détérioration progressive des microvaisseaux rétiens sous l'effet de l'hyperglycémie chronique. Elle constitue l'une des principales causes de déficience visuelle dans le monde, en particulier chez les personnes en âge de travailler.

Le diagnostic précoce de la rétinopathie diabétique est crucial pour prévenir sa progression vers des stades avancés, souvent irréversibles. Cependant, la reconnaissance des différentes phases de la maladie, ainsi que l'identification des multiples signes cliniques sur les images du fond d'œil, requièrent une expertise médicale pointue et un temps d'analyse conséquent. De plus, la variabilité inter- et intra-observateur dans l'interprétation des images peut conduire à des erreurs diagnostiques.

Face à ces défis, l'intelligence artificielle (IA), et plus spécifiquement les techniques d'apprentissage profond (Deep Learning), offrent une solution prometteuse pour automatiser la détection et la classification des stades de la rétinopathie diabétique.

Les réseaux de neurones convolutifs (CNN) permettent d'extraire des caractéristiques complexes à partir des images rétiniennes, améliorant ainsi la précision et la rapidité du diagnostic.

Ce projet vise à développer un modèle d'apprentissage profond capable de classer automatiquement les images du fond d'œil selon la gravité de la rétinopathie, en s'appuyant sur une base de données standard telle que Messidor.

L'objectif est d'optimiser la détection précoce de la RD, notamment dans les zones où l'accès aux spécialistes est limité, afin d'améliorer la prise en charge des patients et de réduire le fardeau de cette maladie.

Ce mémoire comprend trois chapitres :

Chapitre 1 : Ce chapitre présente une introduction détaillée à la rétinopathie diabétique, en décrivant sa définition, ses stades cliniques, ainsi que son impact épidémiologique. Il aborde également les bases du traitement d'images médicales, en particulier les techniques de prétraitement des images du fond d'œil, la segmentation et l'extraction des caractéristiques essentielles à l'analyse automatisée. Ce cadre théorique est fondamental pour comprendre les enjeux du diagnostic assisté par ordinateur.

Chapitre 2 : Ce chapitre expose les principes fondamentaux de l'apprentissage profond, avec un focus sur les architectures les plus utilisées en imagerie médicale, notamment les CNNs. Il détaille le fonctionnement de ces réseaux, les méthodes d'entraînement, ainsi que les techniques d'optimisation et de régularisation permettant d'améliorer la performance des modèles sur des données complexes.

Chapitre 3 : Ce dernier chapitre détaille la mise en œuvre pratique du modèle de détection, en précisant les outils utilisés (Python, TensorFlow, Keras, OpenCV, etc.) et les étapes de développement. Il présente les résultats obtenus sur les bases de données test, analyse la précision et la robustesse du modèle, et compare ses performances à celles d'autres méthodes existantes. Enfin, il discute les implications cliniques, les limites du travail réalisé, et propose des pistes d'amélioration pour les recherches futures.

Chapitre 1

Généralités sur la rétinopathie diabétique et le traitement d'images

I.1 Introduction

La rétinopathie diabétique est une complication fréquente du diabète, causée par une détérioration des vaisseaux sanguins de la rétine. Elle évolue progressivement et peut entraîner une perte partielle ou totale de la vision si elle n'est pas détectée et traitée à temps. Ce problème de santé publique est en constante augmentation, notamment en raison de la hausse du nombre de personnes atteintes de diabète dans le monde. Il est donc essentiel de comprendre les différents stades de la maladie ainsi que son impact sur la santé visuelle des patients.

Dans le domaine médical, les images du fond d'œil sont souvent utilisées pour diagnostiquer cette pathologie. Toutefois, ces images doivent être traitées et analysées avec précision afin d'en extraire les signes caractéristiques de la maladie. Le traitement d'image joue donc un rôle fondamental dans ce processus.

Ce chapitre présente les bases du traitement d'image appliqué à la rétinopathie diabétique : les techniques de prétraitement visant à améliorer la qualité des images, les méthodes de segmentation permettant d'isoler les zones importantes, et l'extraction de caractéristiques, étape clé pour identifier les lésions typiques de la maladie.

En combinant les connaissances médicales et les outils du traitement d'image, ce chapitre prépare le terrain à l'utilisation de techniques avancées telles que le Deep Learning dans le diagnostic automatisé de la rétinopathie diabétique.

I.2 Rétinopathie diabétique

I.2.1 Rétinopathie

La rétinopathie est une maladie de la rétine, la membrane transparente située au fond de l'œil, responsable de la vision. Elle peut entraîner une cécité totale ou partielle si elle n'est pas détectée à temps. Elle est principalement causée par le diabète ou l'hypertension. Ses conséquences vont d'une simple gêne visuelle à une perte complète de la vision.

Le traitement dépend de la cause : contrôle de la glycémie (taux de sucre dans le sang) ou de la pression artérielle. Il existe différents types de rétinopathies :

- La rétinopathie hypertensive

- La rétinopathie du prématuré
- La rétinopathie photique
- La rétinopathie due aux radiations
- La rétinopathie diabétique [1]

I.2.2 Rétinopathie diabétique

Avec le temps, le diabète a de nombreuses répercussions sur l'ensemble de l'organisme et peut endommager le cœur, les reins, les nerfs, les vaisseaux sanguins et les yeux. La rétinopathie diabétique (RD) entraîne une baisse de l'acuité visuelle pouvant aller jusqu'à la cécité dans les cas les plus graves. Il s'agit de l'une des principales causes de perte de vision et elle touche 25 à 30 % des personnes atteintes de diabète, soit environ 120 millions de personnes dans le monde. Heureusement, des traitements efficaces existent pour ralentir l'évolution de la maladie.

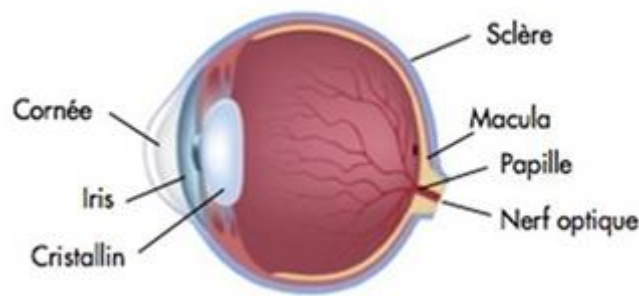


Figure I.1 : Image d'une rétine normale observée lors d'un examen de fond d'œil (absence de rétinopathie diabétique) [2].

La RD peut apparaître sans symptôme, d'où l'importance d'effectuer régulièrement des examens de dépistage afin de contrôler sa progression. Lors de ces examens, des images numériques du fond d'œil du patient sont prises à l'aide de caméras spécialisées. En général, il est recommandé aux techniciens de capturer trois images par œil : deux centrées sur la macula et une centrée sur la papille. Ces images sont ensuite examinées par un clinicien, qui établit un diagnostic sur l'état d'avancement de la RD.

Les signes et symptômes courants de la rétinopathie diabétique sont :

- Une vision floue
- Une vision double, apparue soudainement
- Une sécheresse oculaire
- Une difficulté à percevoir les couleurs
- La présence de corps flottants
- Une difficulté à voir dans la pénombre [2]

Durée du diabète	Diabète de type 1 sous insuline	Diabète de type 2
5 – 7 ans	0 % : absence de rétinopathie	20 % : parfois présente dès le diagnostic
> 20 ans	95 % : RD, dont 50 % avec Rétinopathie Diabétique Proliférante (RDP) ¹	85 % : RNP, 4 % : RDP

Tableau I.1 : Pourcentage de risque de développer une rétinopathie diabétique dans le monde.

I.2.3 Classification de la rétinopathie diabétique

L'évaluation des stades de gravité de la rétinopathie diabétique et de la maculopathie montre que la concordance entre les stades attribués aux photographies de dépistage et aux photographies de référence varie de 60,6 % à 75,8 % pour la rétinopathie diabétique, et de 93,2 % à 94,6 % pour l'œdème maculaire. Une surestimation est plus fréquemment observée, en raison notamment d'artefacts, de la localisation des lésions principalement au pôle postérieur, d'anomalies microvasculaires interprétées comme des néovaisseaux, ou encore d'altérations de l'épithélium mimant des exsudats secs.

Les cas de sous-estimation sont généralement dus à des hémorragies rétiniennes situées en dehors du champ exploré lors du dépistage. Malgré une bonne concordance globale, ces résultats soulignent la nécessité d'une formation approfondie et d'un apprentissage rigoureux à la lecture des photographies du fond d'œil pour tous les lecteurs [3].

¹ La rétinopathie diabétique proliférante est une forme grave de la rétinopathie diabétique (stade avancé de la maladie)

La classification simplifiée utilisée lors du dépistage, notamment pour les maculopathies, tient compte de la localisation des signes lésionnels par rapport au centre de la macula. Seules les formes menaçant la vision sont orientées vers un ophtalmologiste.

Enfin, l'angiographie rétinienne est fréquemment réalisée pour confirmer la sévérité de l'ischémie rétinienne.

I.2.3.1 Rétinopathie diabétique non proliférante

A) Rétinopathie diabétique non proliférante minime

On observe un petit nombre de micro-anévrysmes et d'hémorragies punctiformes lors des examens cliniques.

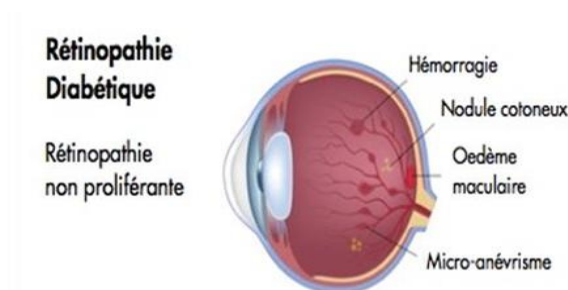


Figure I.2 : Rétinopathie diabétique non proliférante minime [2].

B) Rétinopathie diabétique non proliférante modérée

D'un point de vue clinique, le patient présente :

- Des micro-anévrysmes.
- Et/ou des hémorragies en « flammèche » ou punctiformes.
- Et/ou des nodules cotonneux.
- Et/ou des anomalies de la microcirculation intra-rétinienne.
- Et/ou des anomalies veineuses (dans moins de deux quadrants de la périphérie rétinienne).
- Et/ou des hémorragies rétinienne en « tache » dans moins de quatre quadrants de la périphérie rétinienne [4].

C) Rétinopathie diabétique non proliférante sévère

Ce stade présente un risque élevé d'évolution vers la néovascularisation. Sa définition correspond à l'un des trois critères suivants :

- Hémorragies sévères dans les quatre quadrants de la périphérie rétinienne.
- Et/ou anomalies veineuses moniliformes dans deux quadrants.
- Et/ou anomalies de la microcirculation intra-rétinienne (AMIR) nombreuses dans un quadrant de la périphérie rétinienne [5].

I.2.3.2 Rétinopathie diabétique proliférante

A) Débutante

Des néovaisseaux pré-rétiniens de petite taille sont présents dans un ou plusieurs quadrants de la périphérie rétinienne [6].

B) Modérée

On retrouve des néovaisseaux de plus grande taille dans un ou plusieurs quadrants de la périphérie rétinienne, pouvant être accompagnés de néovaisseaux prépapillaires de petite taille [6].

C) Sévère

Elle se caractérise par la présence de néovaisseaux prépapillaires de grande taille [6].

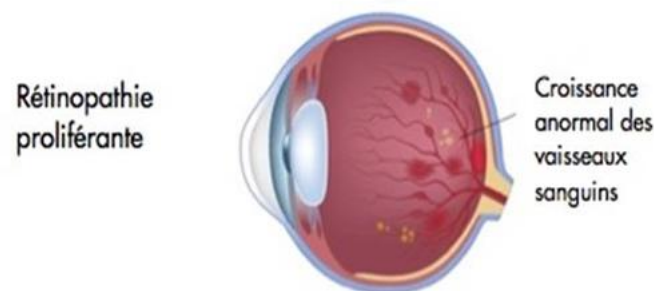


Figure I.3 : Rétinopathie diabétique proliférante sévère [2].

I.2.4 Fond d'œil

Le diagnostic de la rétinopathie diabétique repose sur l'examen du fond d'œil. La photographie couleur du fond d'œil permet de classifier la rétinopathie en différents grades. L'angiographie à la fluorescéine permet de déterminer l'étendue de la maladie, d'élaborer un schéma thérapeutique adapté et d'en surveiller les effets. La tomographie par cohérence optique (OCT) est également utile pour évaluer la gravité de l'œdème maculaire et la réponse au traitement.

Les principaux examens utilisés sont :

- La photographie couleur du fond d'œil
- L'angiographie à la fluorescéine
- La tomographie par cohérence optique (OCT) [7]

Il est recommandé aux personnes atteintes de diabète de type 2 de réaliser un examen du fond d'œil dès le diagnostic, puis tous les deux ans si le diabète est non insulino-dépendant et bien équilibré. Dans le cas contraire, la surveillance doit être annuelle.

I.2.4.1 Examen du fond d'œil

Le fond d'œil est un examen ophtalmologique qui permet d'observer les structures situées à l'arrière du cristallin, notamment le corps vitré, la rétine, la macula, les vaisseaux rétiens, ainsi que la tête du nerf optique, appelée papille. Cet examen permet de vérifier l'état de santé de la rétine, de la macula et des vaisseaux sanguins de l'œil.

La rétine, constituée de fibres nerveuses et riche en photorécepteurs, a pour rôle de capter la lumière et de transmettre au cerveau les informations visuelles reçues. Il s'agit d'une membrane sensorielle qui tapisse l'intérieur de l'œil, derrière l'iris et le cristallin.

La macula, située au centre de la rétine, est la zone responsable de la vision des détails [8].



Figure I.4 : Examen de fond d'œil [8].

I.2.5 Fond d'œil et diabète

Lorsqu'il est mal équilibré, le diabète peut entraîner des complications ophtalmologiques, dont la rétinopathie diabétique. Le fond d'œil constitue un des examens permettant de poser un diagnostic précoce, car la maladie s'installe souvent sans signes d'alerte. Il est donc possible d'être atteint de rétinopathie tout en conservant une bonne vision et sans présenter de symptôme, d'où l'importance d'un contrôle régulier par un ophtalmologiste et d'un dépistage précoce.

Le fait de laisser la maladie progresser peut entraîner des troubles de la vision graves et irréversibles. Le fond d'œil permet de détecter des hémorragies rétinienne, des anomalies vasculaires ou encore des lésions blanchâtres de la rétine appelées nodules cotonneux [9].

I.2.5.1 Diabète de type 1

Un examen du fond d'œil est recommandé dès la découverte du diabète. Chez les enfants, le premier examen doit être réalisé à l'âge de 10 ans. Un fond d'œil annuel est ensuite préconisé, avec une première angiographie dès l'apparition des premiers signes de rétinopathie diabétique (et éventuellement de manière systématique après cinq années de diabète).

I.2.5.2 Diabète de type 2

Un examen du fond d'œil est recommandé dès la découverte du diabète, ainsi qu'une angiographie dès les premiers signes de rétinopathie diabétique (et éventuellement de manière systématique dès la découverte du diabète).

I.2.6 Diagnostic de la rétinopathie diabétique sur le plan clinique

La rétinopathie diabétique doit être dépistée par un examen ophtalmologique systématique, réalisé lors de la découverte du diabète ou dans le cadre de la surveillance annuelle du patient diabétique.

Le diagnostic de la RD repose sur l'examen biomicroscopique du fond d'œil après dilatation pupillaire, complété par des photographies du fond d'œil. Cet examen permet d'identifier les différents signes de la RD [9].

I.2.6.1 Microanévrismes

Les microanévrismes sont les premiers signes, d'un point de vue ophtalmologique, de la rétinopathie diabétique. Ils apparaissent sous forme de lésions punctiformes rouges de petite taille, le plus souvent situées au pôle postérieur. Ils correspondent à une prolifération des cellules endothéliales de certains capillaires [10].

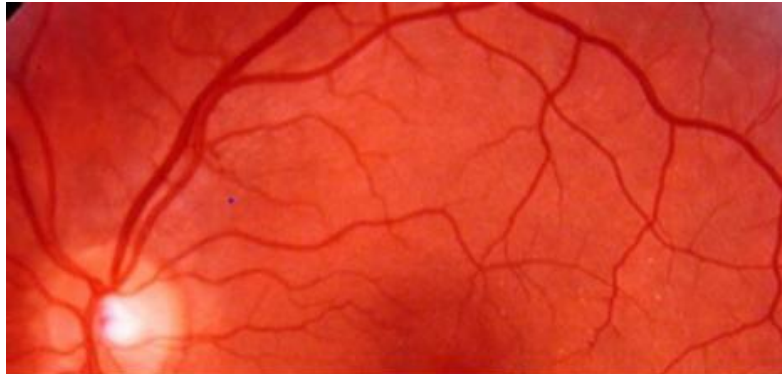


Figure I.5 : Microanévrismes [11].

I.2.6.2 Hémorragies rétinienne punctiformes

Elles peuvent être associées aux microanévrismes. Elles se présentent de manière superficielle, sont de petite taille et ont une forme arrondie. Il est parfois difficile de les distinguer des microanévrismes [10].

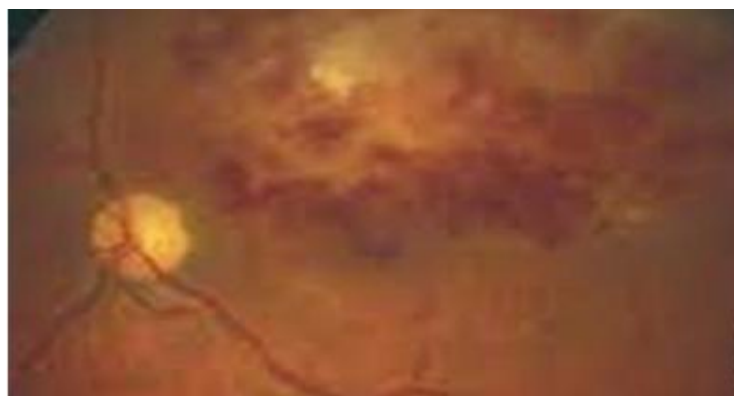


Figure I.6 : Hémorragies rétinienne punctiformes [11].

I.2.6.3 Nodules cotonneux

Ils témoignent souvent d'une poussée évolutive de la rétinopathie diabétique. Ils apparaissent comme des taches blanches superficielles. Leur taille et leur localisation sont variables, mais la présence de nombreux nodules cotonneux disposés « en couronne » péri-papillaires doit faire rechercher une hypertension artérielle associée à la rétinopathie diabétique [10].

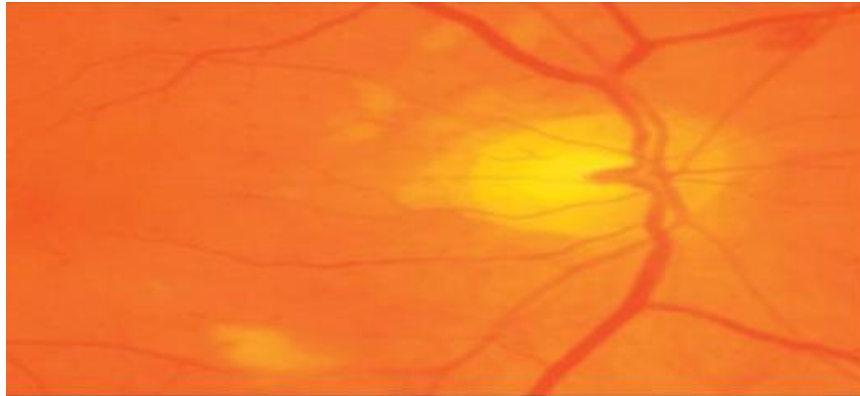


Figure I.7 : Nodules cotonneux [11].

I.2.6.4 Hémorragies en flammèche

Elles sont très superficielles et prennent l'aspect peigné des fibres ganglionnaires dont elles suivent les trajets. Elles évoquent une rétinopathie hypertensive associée [10].

I.2.6.5 Hémorragies en tâche

Elles siègent soit au pôle postérieur, soit en périphérie rétinienne, et sont localisées en bordure ou au sein des territoires rétiens ischémiques. Elles sont plus profondes que les hémorragies en flammèche. Elles apparaissent avec des bords irréguliers et sont de grande taille [10].

I.2.6.6 Anomalies de la microcirculation intra-rétinienne (AMIR)

Elles correspondent à un début de néovascularisation intra-rétinienne, développée à partir des veinules, sans que les vaisseaux franchissent la membrane limitante interne. Elles se présentent comme un réseau rougeâtre, irrégulier et digitiforme, situé en bordure d'un territoire ischémique [10].

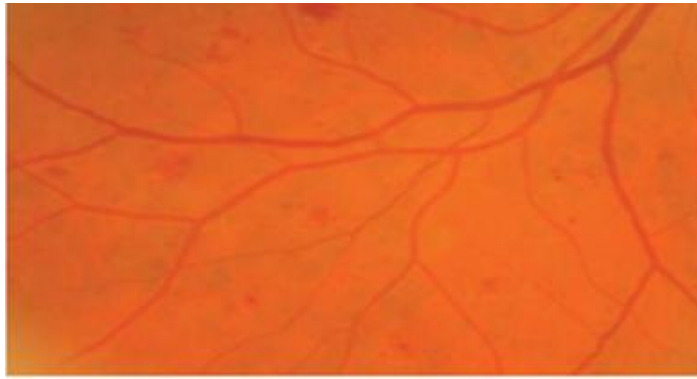


Figure I.8 : Hémorragies en taches et anomalies de la microcirculation intrarétinienne (AMIR) [11].

I.2.6.7 Anomalies veineuses

Les veines en chapelet, qui sont des alternances de rétrécissements et de dilatations, aussi appelées omniiformes, s'observent toujours au sein de territoires étendus de non-perfusion rétinienne. Les veines en oméga ou boucles veineuses se développent sur des veines de moyen ou gros calibre. Ces deux anomalies sont assez spécifiques du diabète et peuvent régresser après réalisation d'une photocoagulation panrétinienne [10].



Figure I.9 : Anomalies veineuses [11].

I.2.6.8 Néo-vaisseaux

Ils sont situés devant la rétine, au-dessus de la membrane limitante interne, soit au niveau prépapillaire, soit ailleurs sur la rétine. Ils proviennent d'un bourgeonnement à partir des

veines et apparaissent sous forme de réseau. Le risque lié à la néovascularisation est proportionnel à l'étendue de la non-perfusion en moyenne périphérie rétinienne [10].

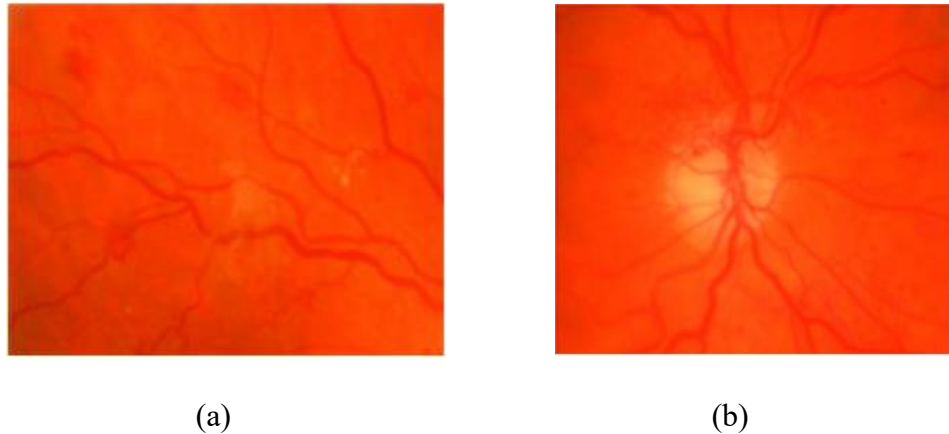


Figure I.10 : Néo vaisseaux prépapillaires (a), et néo-vaisseaux prérétiniens (b) [11].

I.2.6.9 Œdème maculaire

Il traduit l'épaississement de la rétine centrale, secondaire à l'accumulation de liquide dans les secteurs intra- et extra-rétiniens. Il témoigne de la rupture de la barrière hémato-rétinienne interne, conséquence des altérations vasculaires induites par l'hyperglycémie chronique. Il est qualifié de cystoïde lorsque le liquide est disposé en logettes dans le tissu rétinien [10].

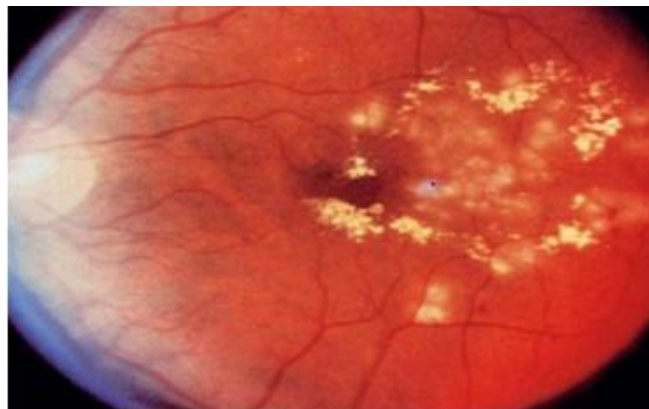


Figure I.11 : Œdème maculaire [11].

I.2.6.10 Exsudats

Ils correspondent à des dépôts de lipides intra- ou sous-rétiniens, provenant de l'extravasation plasmatique, liée à une perméabilité anormale des vaisseaux rétinien. Ils sont jaunes et de forme souvent spéculée.

Ils peuvent s'associer à l'œdème maculaire cystoïde. Dans ce cas, ils sont disposés en couronne autour des anomalies vasculaires dont ils sont issus (exsudats circinés).

S'ils se situent au niveau de la fovéa, ils aboutissent à la destruction de la rétine externe sus-jacente.



Figure I.12 : Les exsudats [11].

I.3 Traitement d'images

I.3.1 Notions de base

I.3.1.1 Images

Une image est une représentation planaire d'une scène ou d'un objet généralement situé dans un espace tridimensionnel. Elle résulte de la captation des rayons lumineux émis ou réfléchis par les objets formant la scène à l'aide d'un capteur (caméra, scanner, rayons X, etc.). Il s'agit en réalité d'une représentation spatiale de la lumière.

L'image est considérée comme un ensemble de points auxquels est associée une grandeur physique (luminance, couleur). Ces grandeurs peuvent être continues (image analogique) ou discrètes (image numérique), comme le montre la figure (I.13).

Mathématiquement, une image est modélisée par une fonction continue FI, appelée fonction image, de deux variables spatiales. Elle est représentée par $FI(x,y)$, qui mesure l'intensité du niveau de gris de l'image aux coordonnées (x,y) . La fonction image peut être définie comme suit :

$$FI: R^2 \rightarrow R$$

Figure I.14 : Exemple d'image en niveaux de gris, représentée par des entiers compris entre 0 (noir) et 255 (blanc).

B) Représentation des images numériques

On distingue généralement deux grandes catégories d'images [12] :

B.1) Images bitmap (ou images matricielles)

Il s'agit d'images pixellisées, c'est-à-dire constituées d'un ensemble de points (pixels) organisés dans un tableau, chacun de ces points possédant une ou plusieurs valeurs décrivant sa couleur.

B.2) Images vectorielles

Ce sont des représentations d'entités géométriques telles qu'un cercle, un rectangle ou un segment. Celles-ci sont définies par des formules mathématiques (un rectangle est défini par deux points, un cercle par un centre et un rayon, une courbe par plusieurs points et une équation). Le principal inconvénient des images matricielles est qu'elles ne peuvent pas être redimensionnées sans perte de qualité. En revanche, les images vectorielles peuvent facilement être affichées à différentes échelles et s'adaptent à la résolution du dispositif d'affichage [13].

C) Caractéristiques des images numériques

C.1) Pixel

Le pixel est l'élément de base d'une image ou d'un écran, c'est-à-dire un point. Le pixel se caractérise par ses coordonnées et son intensité lumineuse (niveau de gris). Il constitue donc le plus petit élément que peuvent manipuler les matériels et logiciels d'affichage et d'impression [14].

140	144	147	140	140	155	179	175
144	152	140	147	140	148	167	179
152	155	136	167	163	162	152	172
168	145	156	160	152	155	136	160
162	148	156	148	140	136	147	162
147	167	140	155	155	140	136	162
136	156	123	167	162	144	140	147
148	155	136	155	152	147	147	136

Pixel

Figure I.15 : Pixel d'une image.

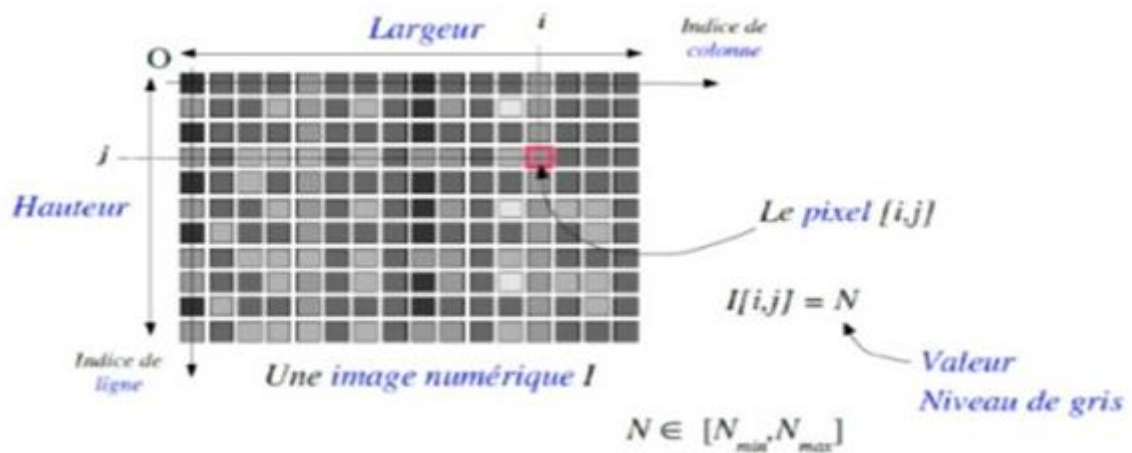


Figure 1.16 : Représentation des pixels dans une image.

C.2) Voisinage

Le plan de l'image est divisé en formes rectangulaires ou hexagonales, ce qui permet d'exploiter la notion de voisinage [15]. Le voisinage d'un pixel correspond à l'ensemble des pixels situés autour de lui. On appelle également fenêtre de voisinage l'ensemble des pixels pris en compte autour d'un pixel donné pour une opération de traitement d'image [16].

On distingue deux types principaux de voisinage :

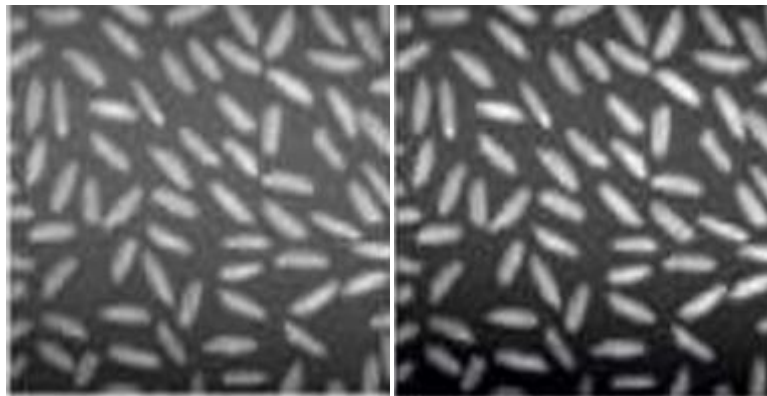
- **Voisinage à 4** : seuls les pixels ayant un côté commun avec le pixel central sont pris en compte (haut, bas, gauche, droite).
- **Voisinage à 8** : tous les pixels ayant au moins un point de contact avec le pixel central sont pris en compte (y compris les coins).

C.3) Contraste

Le contraste correspond à la différence entre deux régions d'une image. Une image bien contrastée présente une bonne répartition des valeurs de gris sur toute l'échelle disponible, avec des blancs très clairs et des noirs profonds. À l'inverse, une image peu contrastée a une faible dynamique lorsque la majorité des pixels ont des niveaux de gris proches les uns des autres.

Si L_1 et L_2 représentent les niveaux de luminosité de deux zones voisines d'une image, le contraste C peut être défini par le rapport suivant :

$$C = \frac{L_1 - L_2}{L_1 + L_2} \quad (I.2)$$



(a) Contraste faible

(b) Contraste élevé

Figure I.17 : Deux images possédant un contraste différent.

C.4) Luminance

La luminance est le degré de luminosité des points d'une image. Elle est également définie comme le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface. Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes).

- Un bon contraste : il faut éviter les images dont la gamme de contraste est trop réduite vers le blanc ou le noir, car cela entraîne une perte de détails dans les zones sombres ou très claires.
- L'absence de parasites [17].

C.5) Résolution d'une image

La résolution est le nombre de pixels constituant une image, c'est-à-dire sa dimension (le nombre de colonnes multiplié par le nombre de lignes). Par exemple, une image de 640 pixels en largeur et 480 en hauteur aura une résolution de 640×480 pixels.

La résolution est souvent confondue avec la définition d'une image, qui désigne le nombre de points par unité de surface, exprimé en points par pouce (PPP ou DPI, pour dots per inch en anglais). Un pouce équivaut à 2,54 cm. La résolution établit donc le rapport entre le nombre de pixels d'une image et la taille réelle de son affichage sur un support physique.

Par exemple, une résolution de 300 dpi signifie qu'il y a 300 colonnes et 300 lignes de pixels par pouce carré, soit 90 000 pixels sur cette surface. Une résolution de référence de 72 dpi signifie qu'un pixel correspond à 1/72 de pouce.

C.6) Bruit

Le bruit dans une image est défini comme une variation brusque de l'intensité d'un pixel par rapport à ses voisins. Il se manifeste par de petites taches de faible dimension, réparties de manière aléatoire sur l'image.

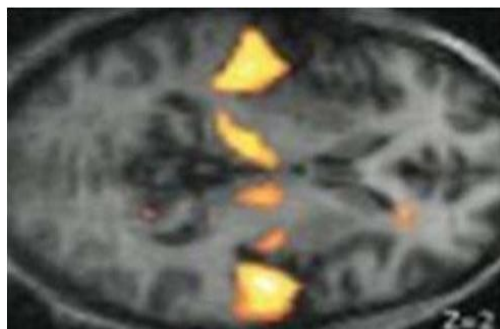


Figure I.18 : Image bruitée.

C.7) Textures

La texture d'une image désigne l'apparence visuelle d'une surface, caractérisée par la répétition ou la variation de motifs, de structures, ou de motifs de luminance. Elle reflète

l'organisation spatiale des pixels et permet de distinguer les différentes régions ou objets d'une image, même sans contours nets. La texture peut être régulière ou irrégulière.

C.8) Niveaux de gris

Le niveau de gris correspond à la valeur de l'intensité lumineuse en un point de l'image. Un pixel peut prendre des valeurs allant du noir (0) au blanc (255), en passant par un nombre fini de niveaux intermédiaires. Ainsi, pour représenter une image en niveaux de gris, on attribue à chaque pixel une valeur numérique qui correspond à la quantité de lumière renvoyée.

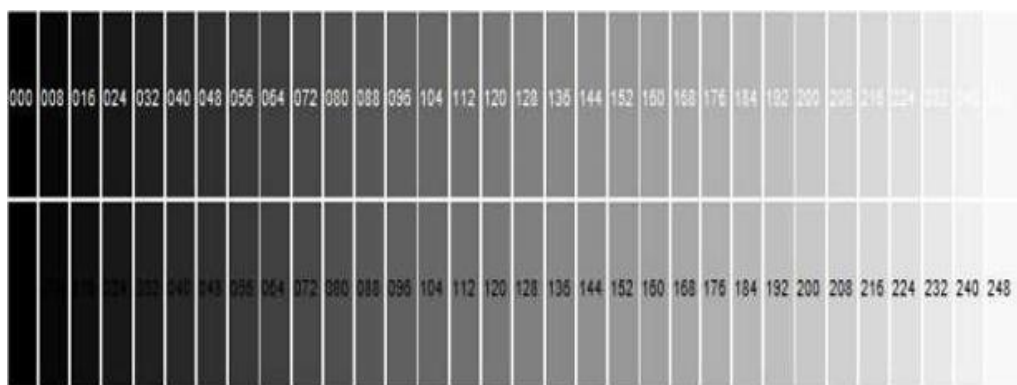


Figure I.19 : Différents niveaux de gris.

C.9) Couleurs

L'espace colorimétrique est basé sur la synthèse additive des couleurs, c'est-à-dire que le mélange de différentes couleurs produit une nouvelle couleur. La plupart des images en couleur reposent sur trois couleurs primaires : le Rouge, le Vert et le Bleu (RVB ou RGB en anglais), et utilisent généralement 8 bits pour chaque composante. Ainsi, chaque pixel nécessite $3 \times 8 = 24$ bits pour coder les trois composantes, et chaque composante peut prendre une valeur dans l'intervalle $[0, 255]$.

Il existe plusieurs méthodes pour convertir une image couleur en niveaux de gris. La plus simple consiste à utiliser la formule suivante :

$$Gris = \frac{Rouge + Vert + Bleu}{3} \quad (I.3)$$

I.3.2 Traitement d'images

I.3.2.1 Définition du traitement d'images

Le traitement d'images est une discipline à l'intersection de l'informatique et des mathématiques appliquées. Elle s'intéresse à l'analyse et à la transformation des images numériques dans le but d'améliorer leur qualité ou d'extraire des informations pertinentes. Il s'agit d'une branche spécifique du traitement du signal, dédiée aux images et aux données visuelles telles que les vidéos, ce qui la distingue des autres domaines qui traitent des signaux unidimensionnels, comme l'audio.

Dans le cadre de la vision par ordinateur, le traitement d'images intervient après la phase d'acquisition et de numérisation des images. Il englobe les opérations de modification et d'analyse qui permettent d'interpréter le contenu visuel. Cette étape d'interprétation est de plus en plus enrichie par l'intelligence artificielle, qui exploite des connaissances spécifiques au domaine pour mieux comprendre la signification des images traitées.

Pour bien comprendre le traitement d'images, il faut d'abord savoir ce qu'est une image numérique. Les modalités d'acquisition et de numérisation influencent fortement les traitements nécessaires à l'extraction d'informations. Plusieurs éléments sont importants, notamment :

- La résolution et le mode de codage, qui impactent la précision des mesures.
- Les réglages optiques, comme la mise au point, qui affectent la netteté.
- Les conditions d'éclairage, responsables d'une grande partie de la variabilité des images.
- Le bruit généré lors de la transmission ou de la capture des images.

Le traitement d'image fait partie d'une chaîne de traitement généralement résumée en quatre étapes principales [18].

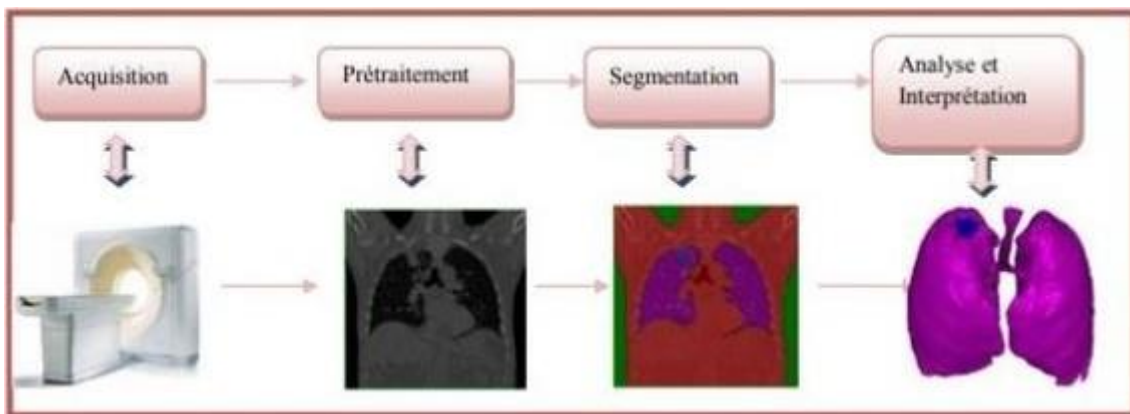


Figure II.20 : Chaîne de traitement d'images (exemple d'une image médicale).

I.3.2.2 Exemples de traitement d'images

Le traitement d'images constitue un domaine vaste et multidisciplinaire qui englobe une multitude de méthodes destinées à manipuler et à analyser des images numériques. Parmi les principales techniques utilisées, on peut citer :

- **Les filtres**, qui servent à améliorer la qualité visuelle des images en corrigeant les défauts, en atténuant le bruit, en renforçant la netteté ou en ajustant les couleurs.
- **La détection des contours**, qui vise à repérer les limites des objets présents dans une image ; une étape cruciale en vision par ordinateur pour l'identification des formes et des structures.
- **La segmentation**, qui consiste à diviser une image en différentes régions homogènes selon des critères tels que la couleur, la texture ou la forme, facilitant ainsi une analyse ciblée.
- **La compression**, technique essentielle pour réduire la taille des fichiers image tout en conservant une qualité acceptable, particulièrement utile lors de la transmission sur des réseaux à bande passante limitée.
- **La reconstruction**, qui permet de restaurer des images altérées ou incomplètes grâce à des algorithmes spécifiques de traitement.
- **L'augmentation de résolution**, méthode employée pour améliorer la définition d'une image en appliquant des techniques d'interpolation avancées ; elle est idéale pour améliorer des images à basse résolution.
- **La reconnaissance de formes**, qui utilise des algorithmes de classification ou d'apprentissage automatique pour identifier et catégoriser des objets ou des motifs dans une image.

- **La superposition d'images**, qui consiste à fusionner plusieurs clichés afin de créer une image composite ; cette technique est fréquemment utilisée en photographie pour réaliser des panoramas ou éliminer des éléments indésirables. [19]

I.3.2.3 Système de traitement d'images

Un système de traitement d'images comprend plusieurs étapes, telles que l'acquisition, le prétraitement, la segmentation, l'extraction de caractéristiques, la classification et la reconnaissance. Chaque étape est réalisée à l'aide d'algorithmes spécifiques.

A) Acquisition et numérisation

Le processus d'acquisition et de numérisation consiste à capturer une image numérique, généralement sous la forme d'une matrice bidimensionnelle, à partir d'une scène en trois dimensions à l'aide d'un système optique. Cette capture peut être effectuée de manière statique, comme avec un appareil photo ou un scanner, ou de façon dynamique, à l'aide d'une caméra ou d'une webcam, ce qui permet d'obtenir une séquence vidéo.

L'image continue $f(x,y)$ est convertie en un ensemble d'échantillons discrets par la discrétisation des coordonnées spatiales (x,y) , appelée échantillonnage, puis par la quantification des niveaux d'intensité de ces échantillons. Le résultat est un ensemble de données numériques brutes, qui peuvent nécessiter des traitements supplémentaires pour améliorer la qualité ou extraire des informations.

Il est important de noter que cette représentation numérique est toujours imparfaite, car le processus d'acquisition introduit inévitablement du bruit dans l'image.

B) Prétraitement des images

Le prétraitement est une étape permettant de corriger certains défauts des images afin d'en faciliter l'analyse ou l'apprentissage automatique. En général, cela se traduit par l'application de filtres ou par des opérations sur l'histogramme de l'image.

Les méthodes de prétraitement couramment utilisées pour extraire les caractéristiques liées à la rétinopathie diabétique incluent :

- Le filtre gaussien.
- Le filtre anisotrope.
- La décomposition de l'espace colorimétrique RVB.
- L'expansion de l'histogramme.

- L'égalisation d'histogramme (CLAHE) et le filtrage médian.
- Les opérations morphologiques.

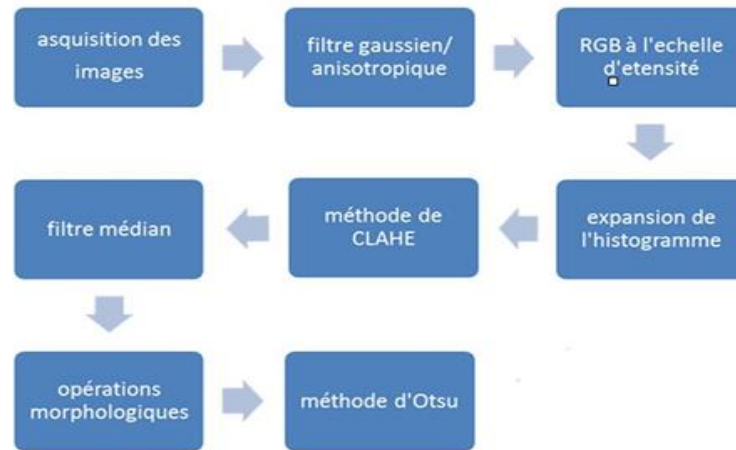


Figure I.21 : Exemple d'un diagramme de prétraitement des images de rétinopathie.

B.1) Filtre gaussien

En électronique et en traitement du signal, un filtre gaussien est un filtre dont la réponse impulsionnelle suit une fonction gaussienne, ou une approximation de celle-ci (car une véritable fonction gaussienne aurait une réponse impulsionnelle infinie).

Les filtres gaussiens sont largement utilisés en traitement d'images pour adoucir ou lisser les images tout en évitant les transitions brusques. Lorsqu'une entrée en forme de marche (ou échelon) leur est appliquée, ils ne génèrent pas de dépassement brutal, ce qui les rend efficaces pour réduire le bruit sans altérer fortement les contours. De plus, ils assurent une transition rapide entre les niveaux de signal, tout en introduisant un retard minimal dans la transmission des fréquences.

Ce comportement le rend idéal comme filtre dans le domaine temporel pour de nombreuses applications. La réponse impulsionnelle d'un filtre gaussien unidimensionnel est donnée par l'expression suivante [20] :

$$g(x) = \sqrt{\frac{\alpha}{\pi}} e^{-\alpha x^2} \quad (\text{I.4})$$

Dans cette formule :

- $g(x)$ représente la forme du filtre dans le domaine temporel.

- α est un paramètre qui contrôle la largeur de la courbe : plus α est grand, plus le filtre est étroit (et donc plus le lissage est localisé).
- $e^{-\alpha x^2}$ est une fonction exponentielle gaussienne qui donne au filtre sa forme caractéristique en cloche.

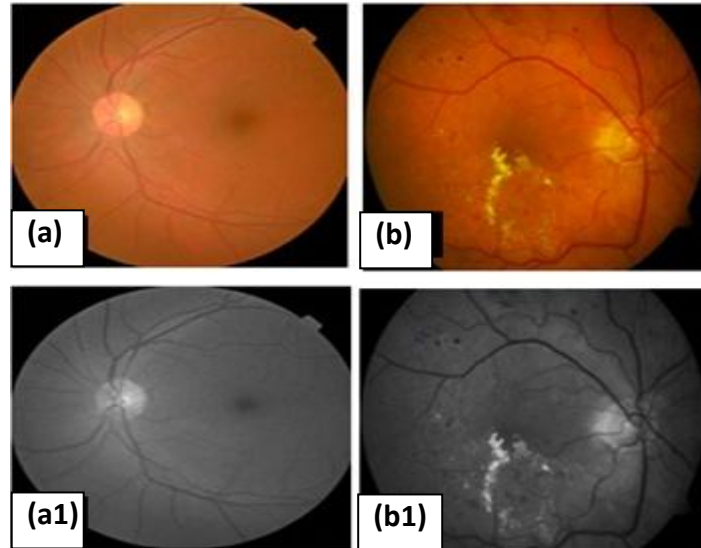


Figure I.22 : (a) Image saine ; (a1) image saine après application du filtre gaussien; (b) image malade ; (b1) image malade après application du filtre gaussien.

B.2) Filtre anisotrope

Le filtrage anisotrope est une technique de traitement d'image qui permet de lisser les régions homogènes tout en préservant les contours et les détails importants. Contrairement aux filtres classiques qui appliquent un lissage uniforme, le filtre anisotrope adapte le traitement en fonction du contenu local de l'image : il atténue le bruit dans les zones uniformes sans flouter les bords, ce qui le rend particulièrement utile pour des applications comme la détection de structures ou l'analyse médicale [21].

B.3) Filtre médian

Le filtre médian est un filtre non linéaire basé sur le calcul de la valeur médiane des pixels dans son voisinage. Pour l'appliquer, il est nécessaire de choisir une taille de masque, c'est-à-dire la région considérée autour de chaque pixel.

La fonction principale de ce filtre est d'éliminer les pics d'intensité isolés présents dans la zone du masque. Il est donc particulièrement utile lorsqu'une image présente du bruit.

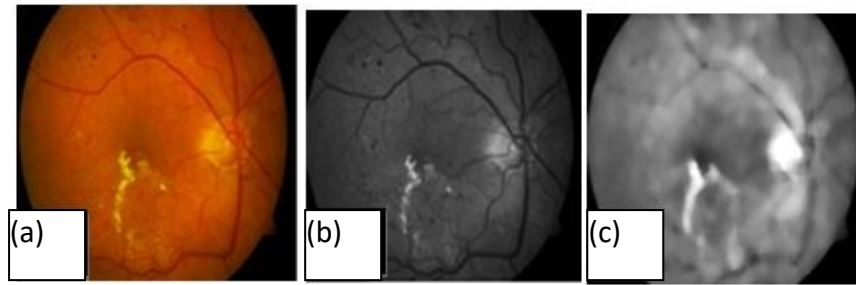


Figure I.23 : (a) Image RVB ; (b) Image en niveaux de gris ; (c) Image après application du filtre médian.

La médiane est la valeur (ou l'une des deux valeurs centrales) qui se situe au milieu d'un ensemble de données triées. Autrement dit, au moins la moitié des données est inférieure ou égale à la médiane, et l'autre moitié est supérieure ou égales à celle-ci.

Le filtre médian permet de réduire les distorsions impulsives dans une image sans altérer fortement les contours. C'est une méthode efficace pour supprimer les bruits isolés tout en préservant les arêtes nettes.

L'opération de filtrage médian consiste à remplacer chaque pixel par la médiane des pixels dans une petite fenêtre glissante centrée sur celui-ci.

C) Segmentation des images

La segmentation est un traitement de bas niveau qui consiste à diviser l'image observée en un certain nombre de régions [22]. La segmentation d'image est avant tout un processus naturel, réalisé en permanence par notre cerveau à partir de nos perceptions visuelles [23].

Elle constitue une étape essentielle et critique dans l'analyse d'images. En effet, une bonne segmentation conduit généralement à une bonne analyse, car c'est à partir de l'image segmentée que sont effectuées les mesures permettant d'extraire des paramètres discriminants, en vue de la classification ou de l'interprétation.

L'objectif de la segmentation est de produire une description compacte et représentative du contenu informationnel de l'image, plus exploitable que l'ensemble brut de ses pixels. Il s'agit d'extraire des indices visuels (ou primitives) pertinents, suffisamment corrélés avec les entités présentes dans la scène représentée.

La diversité des travaux menés sur la compréhension de la vision humaine témoigne de la complexité de la notion d'indice visuel et de la difficulté à en donner une définition précise. Un indice visuel peut être défini comme une information directement perceptible dans l'image, telle que des contours ou des régions homogènes.

C.1) Objectifs de la segmentation

- Fournir des régions homogènes selon un critère donné (intensité, couleur, texture, etc.).
- Délimiter précisément les contours des différentes régions.
- Faciliter l'étude et l'interprétation des structures anatomiques.
- Réduire le bruit dans l'image [24].

C.2) Méthodes de segmentation

Le choix d'une méthode de segmentation dépend de plusieurs facteurs [25], tels que : la nature de l'image, les conditions d'acquisition (notamment le bruit), les primitives à extraire (contours, textures, etc.), et les contraintes d'exploitation (traitement en temps réel, ressources matérielles comme la mémoire vive ou le type d'appareil utilisé).

Il existe plusieurs méthodes de segmentation d'images, parmi lesquelles:

- La segmentation par classification des pixels.
- La segmentation par détection de contours.
- La segmentation basée sur les régions.

D) Interprétation des images

L'interprétation des images médicales dans le cadre de la rétinopathie diabétique consiste à analyser les photographies du fond d'œil afin de détecter et caractériser les lésions spécifiques, telles que les microanévrismes, les hémorragies intrarétiniennes, les exsudats, l'œdème maculaire ou la néovascularisation. Cette analyse repose principalement sur l'examen visuel des images en couleur du fond d'œil, souvent complété par des examens complémentaires tels

que l'angiographie à la fluorescéine ou la tomographie en cohérence optique (OCT), permettant une évaluation plus précise des lésions et de leur étendue.

L'objectif de cette étape est de classifier la rétinopathie selon son degré de gravité, afin d'orienter la prise en charge thérapeutique appropriée. Elle est cruciale pour le dépistage précoce et la prévention des complications visuelles sévères.

Par ailleurs, des systèmes automatisés de traitement d'images assistés par l'intelligence artificielle sont de plus en plus intégrés à cette étape. Ils facilitent la détection rapide et fiable des lésions, tout en allégeant la charge de travail des spécialistes et en maintenant une haute précision diagnostique.

I.4 Conclusion

Ce chapitre a permis de mieux comprendre la rétinopathie diabétique, ses différents stades et son importance croissante dans le domaine de la santé publique. La forte prévalence de cette maladie et ses conséquences graves sur la vision montrent à quel point il est crucial de mettre en place des méthodes de dépistage précises et rapides.

D'un point de vue technique, les images du fond d'œil doivent être traitées avec soin pour permettre une analyse efficace. Le traitement d'images médicales joue ici un rôle central, car il permet d'améliorer la qualité des données visuelles et d'en extraire les informations pertinentes nécessaires au diagnostic.

Ces notions fondamentales sont indispensables pour comprendre comment les technologies d'intelligence artificielle, et en particulier le Deep Learning, peuvent être utilisées pour détecter automatiquement la rétinopathie diabétique. Elles constituent la base sur laquelle repose l'automatisation du diagnostic, en combinant expertise médicale et puissance des outils numériques.

Chapitre 2

Deep learning

(Apprentissage profond)

II.1 Introduction

Le deep learning, ou apprentissage profond, représente une sous-catégorie majeure de l'apprentissage automatique (voir [figure II.1](#)), reposant principalement sur l'utilisation de réseaux de neurones artificiels, inspirés du fonctionnement du cerveau humain. Cette approche vise à modéliser, de manière informatique, certains aspects de l'intelligence humaine, notamment la capacité à apprendre à partir de grandes quantités de données. Bien que le concept d'apprentissage profond ne soit pas récent, il connaît un essor considérable ces dernières années, principalement grâce à l'augmentation exponentielle de la puissance de calcul et à la disponibilité massive de données numériques. Ces avancées technologiques ont permis au deep learning de s'imposer comme un domaine de recherche incontournable, notamment dans des secteurs tels que la santé.

Dans ce chapitre, nous explorerons en détail les principes fondamentaux du deep learning, en mettant l'accent sur les réseaux de neurones convolutifs, qui jouent un rôle central dans notre méthodologie de détection du diabète.

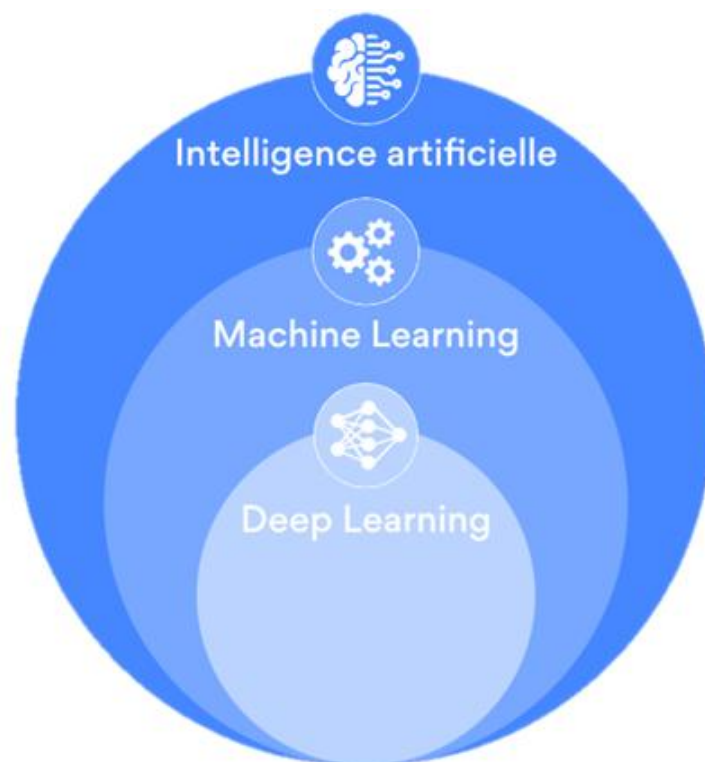


Figure II.1 Hiérarchie entre l'intelligence artificielle, l'apprentissage automatique (ou machine learning) et le deep learning [\[26\]](#).

II.2 Définition du Deep Learning

Le Deep Learning est une branche de l'intelligence artificielle et un sous-domaine de l'apprentissage automatique, qui repose sur l'utilisation de réseaux de neurones artificiels comportant plusieurs couches successives. Ces réseaux, inspirés du fonctionnement du cerveau humain, permettent aux ordinateurs d'apprendre à partir de grandes quantités de données et de résoudre des tâches complexes telles que la reconnaissance d'images, la compréhension du langage ou la prédiction de tendances, souvent sans intervention humaine directe. Grâce à leur architecture profonde, ces modèles sont capables d'extraire automatiquement des représentations hiérarchisées des données, rendant possible l'analyse et l'interprétation de signaux variés de façon performante.

II.3 Pourquoi le Deep Learning?

Il s'agit d'une combinaison de facteurs dont :

- **L'omniprésence des données** : Nous vivons à l'ère de l'informatisation (Internet of Things), et le propre du Deep Learning est de tirer parti d'une grande quantité de données pour en estimer une représentation abstraite exploitable.
- **La puissance de calcul** : La théorie des réseaux de neurones existe depuis plusieurs décennies, mais c'est grâce à la puissance de calcul aujourd'hui accessible notamment avec la démocratisation des GPU, devenus la plateforme de choix pour le Deep Learning que ces modèles ont pu se développer rapidement.
- **Des besoins croissants dans le domaine de l'IA** : vision par ordinateur, reconnaissance vocale, traitement automatique du langage, etc.
- **Un effet de mode** : On a tendance à vouloir appliquer le Deep Learning partout, alors qu'il ne s'agit que d'un moyen, et non d'une fin. Certains problèmes peuvent être efficacement résolus par d'autres méthodes d'apprentissage statistique. Cela dit, si de nombreuses personnes investissent dans le Deep Learning, il est normal qu'il devienne si populaire.
- **La capacité de stockage** : Les capacités de stockage sont devenues beaucoup plus accessibles et abordables.
- **L'apparition de plateformes et de communautés fortes** : Celles-ci encouragent le développement de ce domaine et favorisent sa démocratisation.

II.4 Domaines d'application du Deep Learning

Le Deep Learning s'impose aujourd'hui comme une technologie clé dans de nombreux secteurs, grâce à sa capacité à traiter de vastes ensembles de données et à reconnaître des motifs complexes. Voici quelques-uns de ses principaux domaines d'application :

- **Reconnaissance d'images et de vidéos** : Le Deep Learning est largement utilisé pour analyser et interpréter des images ou des vidéos. Cela inclut la reconnaissance faciale, la détection d'objets, la surveillance de sécurité, ainsi que l'imagerie médicale pour le diagnostic de maladies.
- **Traitement automatique du langage naturel (NLP)** : Les modèles de Deep Learning permettent de comprendre, générer et traduire du texte. Ils sont utilisés pour l'analyse de sentiments, la traduction automatique, la génération de contenu, ainsi que dans les chatbots et assistants virtuels.
- **Reconnaissance vocale et assistants intelligents** : Le Deep Learning améliore la précision des systèmes de reconnaissance vocale, utilisés dans les assistants personnels (comme Siri, Alexa ou Google Assistant), la transcription automatique et les commandes vocales pour appareils intelligents.
- **Véhicules autonomes** : Les voitures autonomes s'appuient sur le Deep Learning pour percevoir leur environnement, reconnaître les panneaux de signalisation, détecter les obstacles et prendre des décisions de conduite en temps réel.
- **Santé et médecine personnalisée** : Dans le domaine médical, le Deep Learning est utilisé pour analyser des images médicales, diagnostiquer des maladies, prédire l'évolution des patients et personnaliser les traitements en fonction des données génétiques et médicales des individus.
- **Détection de fraude et cybersécurité** : Les institutions financières et les entreprises utilisent le Deep Learning pour détecter les transactions frauduleuses, repérer les comportements suspects et renforcer la sécurité des systèmes informatiques.
- **Systèmes de recommandation** : Les plateformes de streaming, d'e-commerce et de réseaux sociaux exploitent le Deep Learning pour proposer des recommandations personnalisées à leurs utilisateurs, en analysant leurs préférences et comportements passés.

- **Marketing et analyse de données clients** : Le Deep Learning optimise la segmentation du marché, la personnalisation des campagnes publicitaires et l'analyse du comportement des consommateurs, permettant ainsi des stratégies marketing plus efficaces.
- **Villes intelligentes et environnement** : Cette technologie contribue à la gestion du trafic, à la prévision des phénomènes météorologiques, à l'optimisation de la consommation d'énergie et à la surveillance de l'environnement grâce à l'analyse de données issues de capteurs et d'images satellitaires.
- **Exploration spatiale** : Le Deep Learning est employé pour analyser les données des missions spatiales, détecter de nouveaux objets célestes et développer des systèmes autonomes pour l'exploration de l'espace.

II.5 Principe et fonctionnement du Deep Learning

La [figure II.2](#) présente une vue d'ensemble simple et compréhensible de l'architecture générale du Deep Learning. Comme mentionné précédemment, le Deep Learning repose essentiellement sur l'utilisation de réseaux de neurones artificiels. Cette figure illustre qu'un tel réseau se compose de trois types de couches de neurones (ou nœuds) : une couche d'entrée (en vert), plusieurs couches cachées (en bleu) et une couche de sortie (en orange). Les couches adjacentes sont connectées par des liens pondérés (lignes noires) [\[27\]](#).

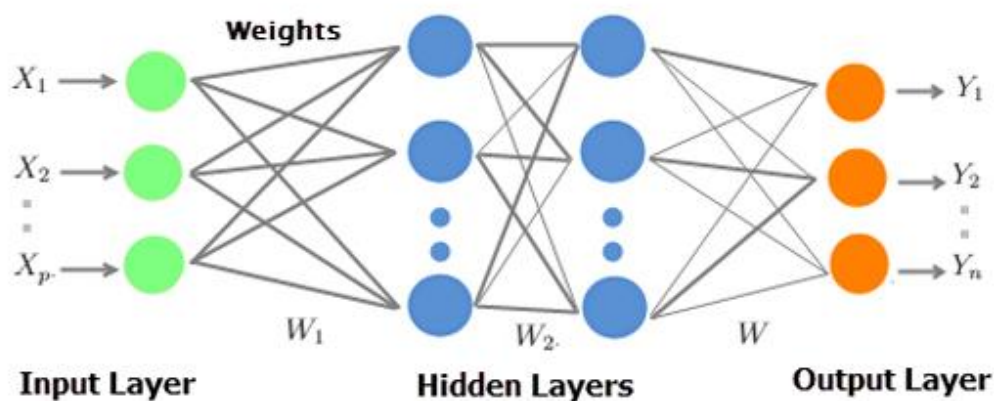


Figure II.2 : Architecture d'un modèle de Deep Learning [\[27\]](#).

II.6 Réseaux de neurones classiques

Le cerveau humain, fruit de millions d'années d'évolution, représente le système de traitement de l'information le plus avancé. Il est capable d'accomplir des tâches complexes telles que la lecture, la reconnaissance faciale ou la prise de décision. L'unité fondamentale qui compose ce système est le neurone, nom donné par les biologistes à cette cellule nerveuse essentielle (voir [figure II.3](#)).

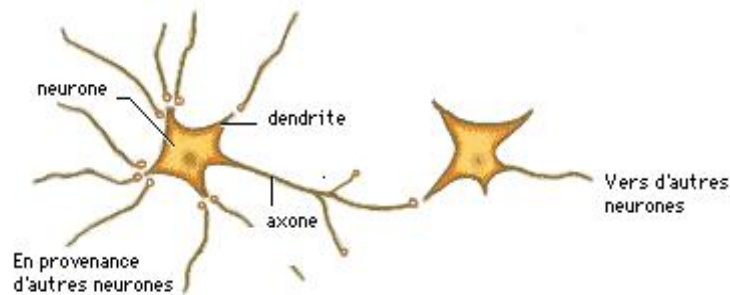
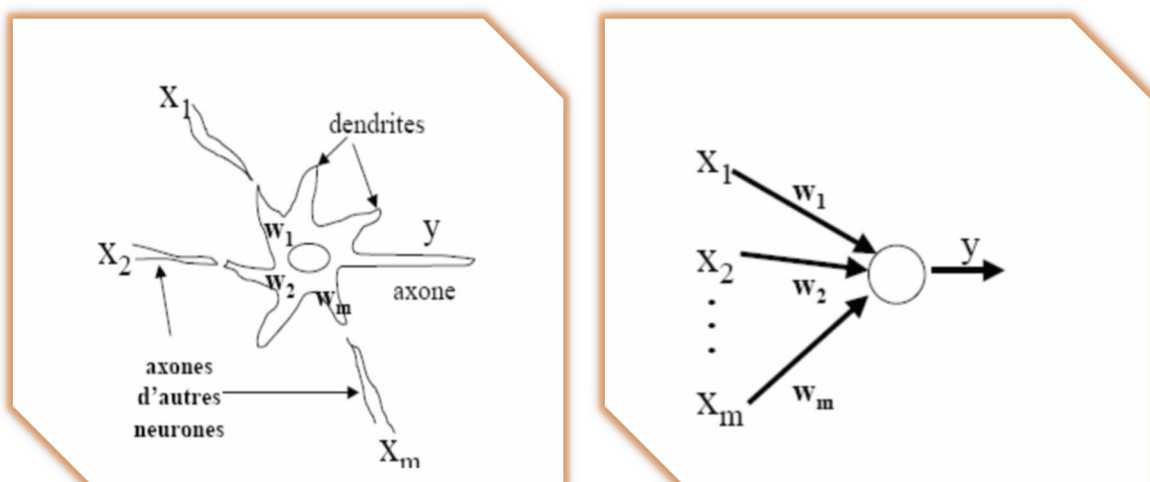


Figure II.3 : Connexions inter-neurones chez l'être humain.

L'interaction simultanée et collective de milliards de neurones, ou cellules nerveuses, confère au cerveau des capacités bien supérieures à celles des ordinateurs, même les plus performants. Pourtant, chaque neurone accomplit une opération relativement simple : il traite les signaux reçus via ses dendrites. En fonction de l'intensité de ces signaux, il décide s'il doit transmettre ou non un signal électrique à ses voisins, via son axone. Cette communication entre axones et dendrites se produit au niveau de points spécifiques appelés synapses.

Les réseaux de neurones artificiels s'inspirent de ce mécanisme biologique pour concevoir des architectures capables de traiter l'information de manière similaire. La [figure II.4](#) illustre la comparaison entre les réseaux de neurones biologiques et artificiels.



(a)

(b)

Figure II.4 : (a) Neurone biologique et (b) Neurone artificiel.

Dans cette section, nous proposons une vue d'ensemble de ce type d'architecture. Nous commencerons par décrire la modélisation d'un neurone biologique, puis nous présenterons plusieurs architectures reposant sur ce modèle.

III.6.1 Définition d'un réseau de neurones

Un réseau de neurones peut être interprété comme une tentative de reproduire le fonctionnement d'une portion du cerveau, lequel est constitué d'environ 10^{11} cellules appelées neurones. Ces neurones sont interconnectés par des liaisons entre l'axone d'un neurone et la dendrite d'un autre, formant un réseau complexe estimé à 10^{15} connexions inter-neurales dans le cerveau humain.

L'axone, prolongement du neurone émetteur, transmet un signal à la dendrite du neurone récepteur. L'intensité de ces connexions varie, certaines étant plus fortes que d'autres. Lorsque la somme des signaux chimiques reçus par un neurone dépasse un certain seuil, ce dernier émet à son tour un signal via son axone, propageant ainsi l'information aux neurones auxquels il est connecté.

II.6.2 Neurone biologique

Les êtres humains sont capables de traiter une grande variété d'informations bien plus rapidement que les machines. Par exemple, comprendre le langage parlé ou identifier des objets dans une scène visuelle ne prend que quelques centaines de millisecondes au cerveau, alors qu'un ordinateur mettrait beaucoup plus de temps, voire ne parviendrait pas à accomplir ces tâches aussi efficacement.

Les unités fondamentales de traitement dans le cerveau humain, appelées neurones, fonctionnent à une vitesse de l'ordre de quelques millisecondes, ce qui est nettement plus lent que celle des transistors utilisés dans les ordinateurs modernes. [28]

II.6.2.1 Cellule nerveuse

Le type de neurone biologique le plus courant se présente comme une cellule dont le corps est doté de deux types de prolongements (voir figure II.5) : (1) Des ramifications courtes et buissonnantes, appelées dendrites. (2) Une longue fibre se terminant par une arborisation,

appelée axone. Ces prolongements permettent au neurone d'établir des connexions avec d'autres cellules nerveuses (ou avec des cellules musculaires, dans le cas des neurones moteurs).

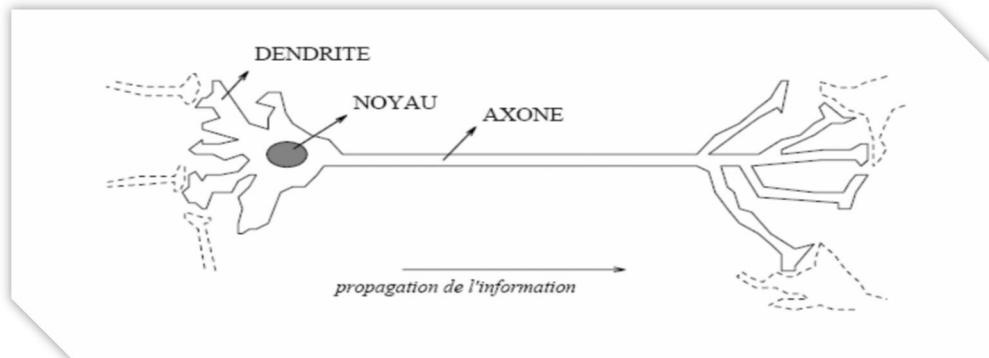


Figure II.5 : Représentation d'une cellule nerveuse.

II.6.2.2 Corps cellulaire

Le corps cellulaire contient le noyau du neurone ainsi que la machinerie biochimique nécessaire à la synthèse des enzymes. De forme sphérique ou pyramidale, il renferme également d'autres molécules essentielles à la vie de la cellule. Sa taille est de l'ordre de quelques microns de diamètre.

II.6.2.3 Dendrites

Les dendrites sont de fines extensions tubulaires qui se ramifient autour du neurone, formant une vaste arborescence. Elles captent les signaux envoyés au neurone par d'autres cellules. Leur taille est de l'ordre de quelques dizaines de microns.

II.6.2.4 Axone

C'est le long de l'axone que les signaux sont transmis depuis le neurone. Contrairement aux dendrites, qui se ramifient autour du corps cellulaire, l'axone est plus long et se ramifie à son extrémité, où il se connecte aux dendrites d'autres neurones. Sa longueur peut varier de quelques millimètres à plusieurs mètres.

II.6.2.5 Transmission synaptique

Comment se fait la transmission de l'influx nerveux lorsqu'il arrive à l'extrémité d'un axone (la synapse) ? Les expériences montrent que les cas de transmission électrique sont rares. En général, la transmission repose sur des processus chimiques impliquant des molécules organiques : les neuromédiateurs.

La terminaison synaptique contient des vésicules de neuromédiateurs (voir figure II.6). L'arrivée de l'onde de dépolarisation provoque une augmentation de la perméabilité de la membrane, ce qui entraîne la libération des neuromédiateurs dans l'espace intersynaptique. Ceux-ci viennent alors se fixer sur des récepteurs chimiques situés sur le neurone cible. [ref ajout 1]

Lorsque ces récepteurs captent les neuromédiateurs, ils provoquent une augmentation de la perméabilité de la membrane du neurone cible aux ions sodium. L'entrée des ions sodium dans la cellule peut ainsi déclencher un nouveau potentiel d'action, selon les principes évoqués précédemment. (En pratique, plusieurs dépolarisations successives sont souvent nécessaires pour libérer une quantité suffisante de neuromédiateurs.)

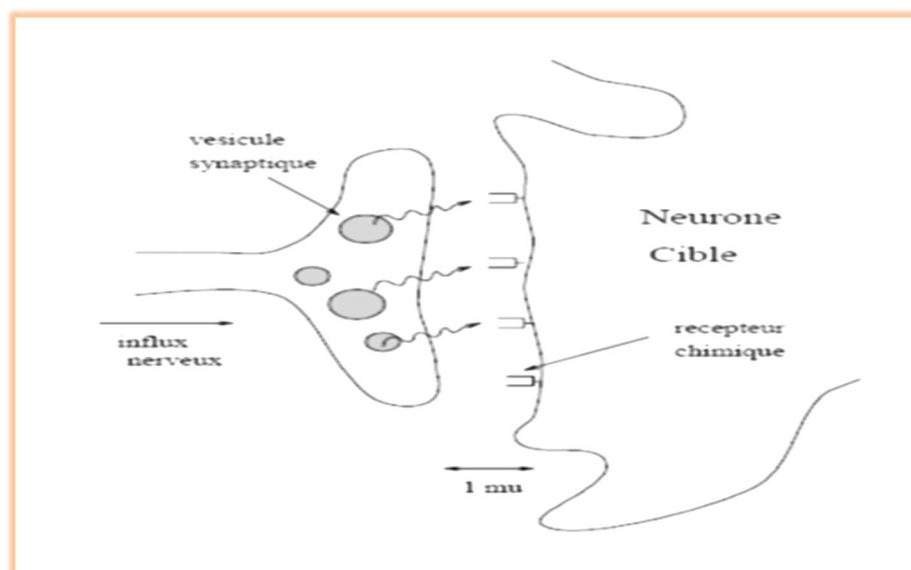


Figure II.6 : Transmission synaptique.

II.6.3 Réseau de neurones artificiels

Un réseau de neurones artificiels (RNA) est un ensemble de neurones formels (unités de calcul simples, ou nœuds processeurs) organisés en couches (ou sous-groupes) et fonctionnant en parallèle (voir figure II.7).

Dans un réseau, chaque sous-groupe effectue un traitement indépendant des autres et transmet le résultat de son analyse au sous-groupe suivant. L'information donnée au réseau se propage ainsi couche par couche, de la couche d'entrée à la couche de sortie, en passant par une ou plusieurs couches intermédiaires (appelées couches cachées). Il est à noter qu'en fonction de l'algorithme d'apprentissage, il est également possible d'avoir une propagation de l'information en sens inverse (rétropropagation ou backpropagation). En général (à l'exception des couches d'entrée et de sortie), chaque neurone dans une couche est connecté à tous les neurones des couches précédente et suivante.

Les RNA ont la capacité de stocker des connaissances empiriques et de les rendre disponibles à l'usage. Les capacités de traitement (et donc la connaissance) du réseau sont stockées dans les poids synaptiques, obtenus par des processus d'adaptation ou d'apprentissage. En ce sens, les RNA ressemblent au cerveau, car non seulement la connaissance est acquise par apprentissage, mais elle est également stockée dans les connexions entre les entités, c'est-à-dire dans les poids synaptiques. [29]

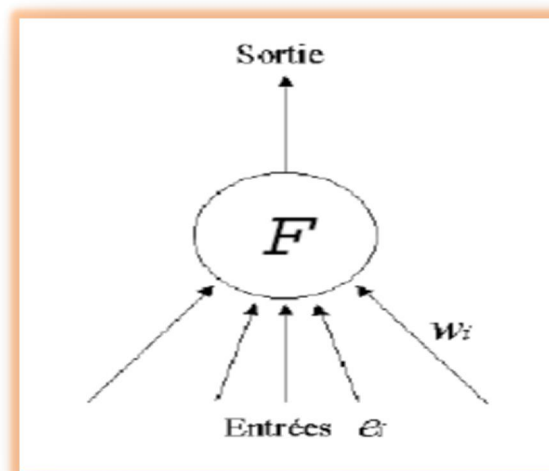


Figure II.7 : Neurone artificiel.

On peut classer les RNA en deux grandes catégories : les réseaux feedforward et les réseaux feedback.

II.6.3.1 Réseaux « Feed-Forward »

Appelés aussi « réseaux de type Perceptron », ce sont des réseaux dans lesquels l'information se propage de couche en couche sans retour en arrière possible. Ils sont divisés en trois classes :

- **Le perceptron monocouche** : C'est historiquement le premier RNA, celui de Rosenblatt. C'est un réseau simple, puisqu'il ne se compose que d'une couche d'entrée et d'une couche de sortie. Il est calqué, à l'origine, sur le système visuel et a donc été conçu dans un but initial de reconnaissance des formes. Cependant, il peut également être utilisé pour effectuer des tâches de classification ou pour résoudre des opérations logiques simples (telles que « ET » ou « OU »). Sa principale limite est qu'il ne peut résoudre que des problèmes linéairement séparables. Il suit généralement un apprentissage supervisé selon la règle de correction de l'erreur.
- **Le perceptron multicouche (PMC)** : C'est une extension du précédent, comportant une ou plusieurs couches cachées entre l'entrée et la sortie. Chaque neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante (à l'exception des couches d'entrée et de sortie), et il n'y a pas de connexions entre les cellules d'une même couche. Les fonctions d'activation utilisées dans ce type de réseau sont principalement des fonctions à seuil ou sigmoïdes. Il peut résoudre des problèmes non linéairement séparables ainsi que des problèmes logiques plus complexes, notamment le célèbre problème du XOR. Il utilise également un apprentissage supervisé basé sur la règle de correction de l'erreur.
- **Les réseaux à fonction radiale** : Ce sont les réseaux appelés également RBF (Radial Basis Function). Leur architecture est similaire à celle des PMC, cependant, les fonctions de base utilisées ici sont des fonctions gaussiennes. Les RBF sont donc utilisés dans les mêmes types de problèmes que les PMC, à savoir la classification et l'approximation de fonctions. L'apprentissage le plus couramment utilisé pour les RBF est le mode hybride, et les règles d'apprentissage sont soit la règle de correction de l'erreur, soit la règle d'apprentissage par compétition.

II.6.3.2 Réseaux « Feed-Back »

Appelés aussi « réseaux récurrents », ce sont des réseaux dans lesquels il y a un retour en arrière de l'information.

- **Les cartes auto-organisatrices de Kohonen** : Ce sont des réseaux à apprentissage non supervisé qui établissent une carte discrète, ordonnée topologiquement, en

fonction des patterns d'entrée. Le réseau forme ainsi une sorte de treillis, dont chaque nœud est un neurone associé à un vecteur de poids. La correspondance entre chaque vecteur de poids est calculée pour chaque entrée. Par la suite, le vecteur de poids ayant la meilleure corrélation, ainsi que certains de ses voisins, sera modifié afin d'augmenter encore cette corrélation [30].

- **Les réseaux de Hopfield :** Les réseaux de Hopfield sont des réseaux récurrents et entièrement connectés. Dans ce type de réseau, chaque neurone est connecté à tous les autres, et il n'y a aucune distinction entre les neurones d'entrée et de sortie. Ils fonctionnent comme une mémoire associative non linéaire et sont capables de retrouver un objet stocké à partir de représentations partielles ou bruitées. L'application principale des réseaux de Hopfield est le stockage de connaissances, mais aussi la résolution de problèmes d'optimisation. Le mode d'apprentissage utilisé ici est non supervisé [31].

II.6.4 Apprentissage

Pour un RNA, l'apprentissage peut être considéré comme le processus de mise à jour des poids des connexions au sein du réseau, afin d'accomplir correctement la tâche qui lui est assignée. L'apprentissage constitue la caractéristique principale des RNA, et il peut s'effectuer de différentes manières, selon diverses règles. On distingue généralement quatre types d'apprentissage [32]:

II.6.4.1 Apprentissage supervisé

Dans ce type d'apprentissage, le réseau s'adapte en comparant le résultat qu'il a calculé à partir des entrées fournies avec la réponse attendue en sortie. Ainsi, le réseau se modifie progressivement jusqu'à ce qu'il parvienne à produire la sortie correcte, c'est-à-dire celle correspondant à l'entrée donnée.

II.6.4.2 Apprentissage par renforcement

Le renforcement est en réalité une forme particulière d'apprentissage supervisé, et certains auteurs le classent d'ailleurs dans cette catégorie. Dans cette approche, le réseau doit apprendre la corrélation entrée/sortie à partir d'une estimation de son erreur, c'est-à-dire du rapport entre échec et succès. Le réseau cherche donc à maximiser un indice de performance qui lui est fourni, appelé signal de renforcement. Le système est capable de déterminer si la réponse qu'il fournit est correcte ou non, mais il ne connaît pas la bonne réponse exacte.

II.6.4.3 Apprentissage non-supervisé (ou auto-organisationnel)

Dans ce cas, l'apprentissage est basé sur des probabilités. Le réseau se modifie en fonction des régularités statistiques présentes dans les données d'entrée et établit des catégories en leur attribuant et en optimisant une valeur de qualité.

II.6.4.4 Apprentissage hybride

Le mode hybride combine en réalité les deux autres approches : une partie des poids est déterminée par apprentissage supervisé, tandis que l'autre partie est déterminée par apprentissage non supervisé.

II.6.5 Neurone formel

Le premier neurone formel a été proposé en 1943 par les deux scientifiques McCulloch et Pitts (voir [figure II.8](#)). Le neurone formel est une modélisation mathématique qui s'inspire des principes de fonctionnement du neurone biologique, en particulier de la sommation des entrées. Sachant qu'au niveau biologique, les synapses n'ont pas toutes la même valeur (les connexions entre neurones étant plus ou moins fortes), les auteurs ont élaboré un algorithme qui pondère la somme des entrées à l'aide de poids synaptiques (coefficients de pondération). De plus, les valeurs 1 et -1 sont utilisées en entrée pour représenter respectivement une synapse excitatrice ou inhibitrice [\[33\]](#).

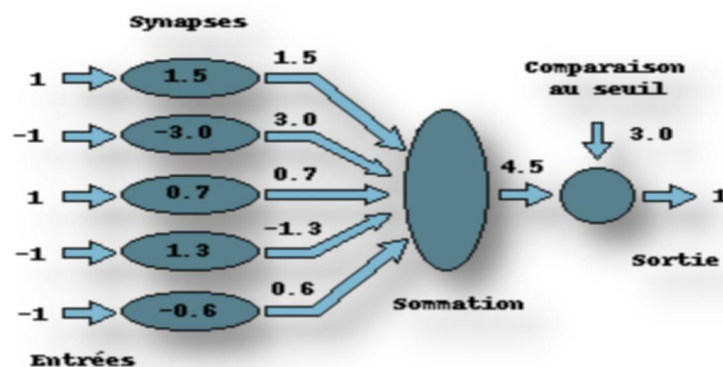


Figure II.8 : Neurone formel.

II.6.5.1 Interprétation mathématique

D'un point de vue mathématique, le neurone formel peut être représenté de la manière suivante :

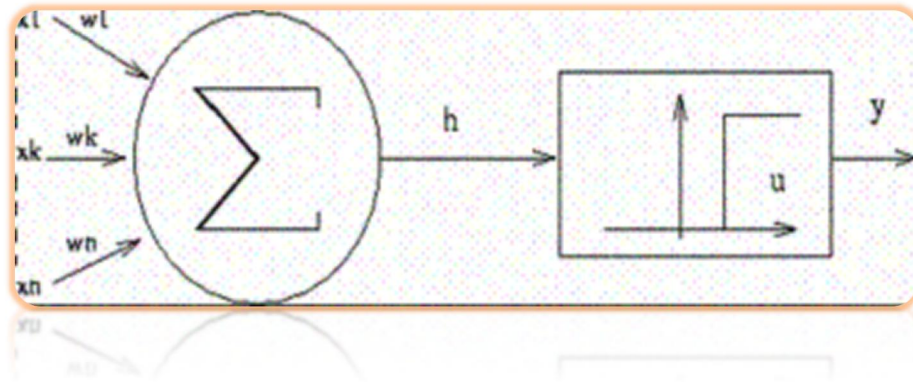


Figure II.9 : Représentation mathématique du neurone formel.

Pour un nombre d'entrées compris entre $j=1$ et un nombre quelconque n , le neurone formel calcule la somme de ses entrées (x_1, \dots, x_n), pondérées par les poids synaptiques (w_1, \dots, w_n), puis la compare à un seuil (θ). Si le résultat est supérieur à ce seuil, la valeur renvoyée est 1 ; sinon, la valeur renvoyée est 0. D'où la formule (avec f = fonction seuil) :

$$y = f \sum_{j=1}^n w_j x_i - \theta \quad (\text{II.1})$$

II.6.5.2 Fonctions d'activation

Dans sa première version, le neurone formel était implémenté avec une fonction à seuil (figure II.10 a), mais de nombreuses variantes existent. Ainsi, le neurone de McCulloch et Pitts a été généralisé de différentes manières, en utilisant d'autres fonctions d'activation, telles que les fonctions linéaires par morceaux (figure II.10 b), les sigmoïdes (figure II.10 c) ou les gaussiennes (figure II.10 d), par exemple.

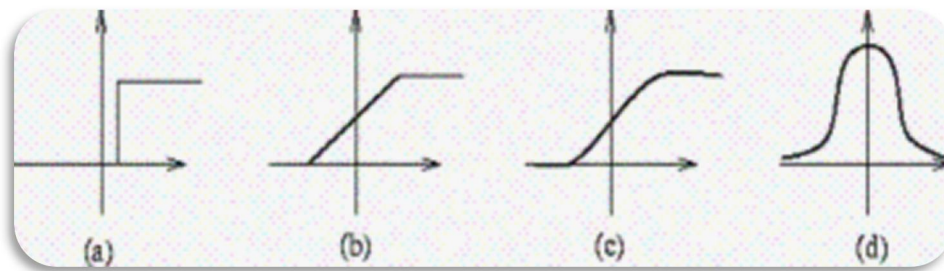


Figure II.10 : Les fonctions d'activation : à seuil (a), linéaire par morceaux (b), sigmoïde (c) et gaussienne (d).

II.6.6 Réseau de neurones multicouches

Le réseau de neurones multicouches, également appelé MLP (Multi-Layer Perceptron), est un modèle conçu pour imiter les fonctions d'apprentissage du cerveau humain. Dans les réseaux neuronaux, les neurones sont regroupés en couches, appelées couches neuronales. Habituellement, chaque neurone d'une couche est connecté à tous les neurones des couches précédentes et suivantes (à l'exception des couches d'entrée et de sortie). L'information envoyée dans un réseau de neurones se propage de couche en couche, de l'entrée vers la sortie, en passant ou non par une ou plusieurs couches cachées.

Selon l'algorithme utilisé, l'information peut également être propagée en sens inverse (rétropropagation). La **figure II.11** montre un réseau de neurones avec une couche cachée [34].

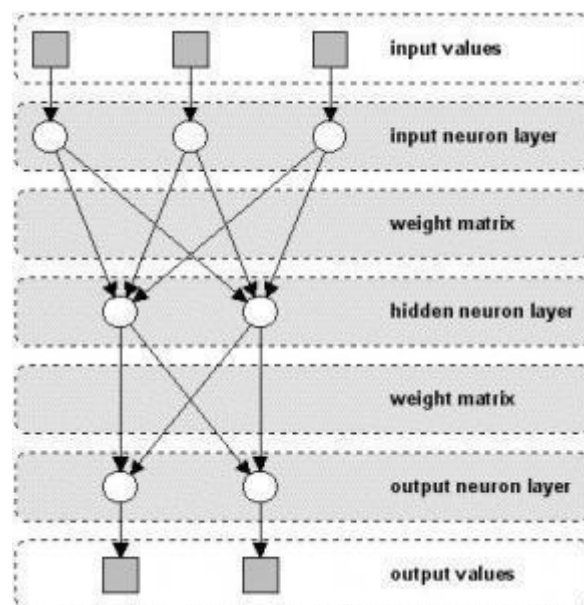


Figure II.11 : Architecture d'un réseau de neurones multicouches avec une seule couche cachée.

Le MLP est un type de réseau très populaire, généralement utilisé pour des tâches de classification, et produisant de bons résultats. Il est composé d'une couche d'entrée, de plusieurs couches cachées, et d'une couche de sortie. La figure II.12 présente l'architecture d'un seul nœud du MLP. L'indice i représente le numéro du neurone dans le réseau, et g désigne sa fonction d'activation.

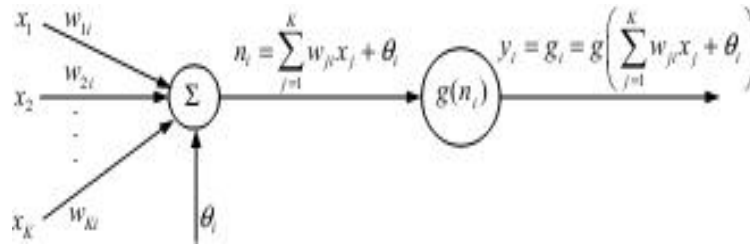


Figure II.12 : Représentation d'un nœud unique dans un MLP.

Soit $x_k, k=1\dots, K$ l'entrée du modèle, w la matrice des poids, et θ le biais. Le résultat n_i est une entrée pour activer la fonction d'activation g . il existe plusieurs fonctions d'activation mais les deux couramment utilisées dans les MLP sont la tangente hyperbolique (\tanh) et la fonction sigmoïde. Les tangentes hyperboliques sont définies comme suit :

$$\tanh(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (\text{II.2})$$

Ainsi, la sortie devient:

$$y_i = g_i = g\left(\sum_{j=1}^k w_{ji} x_j + \theta_i\right) \quad (\text{II.3})$$

En connectant plusieurs sorties en série et en parallèle, le réseau MLP est entraîné comme le montre la [figure II.13](#).

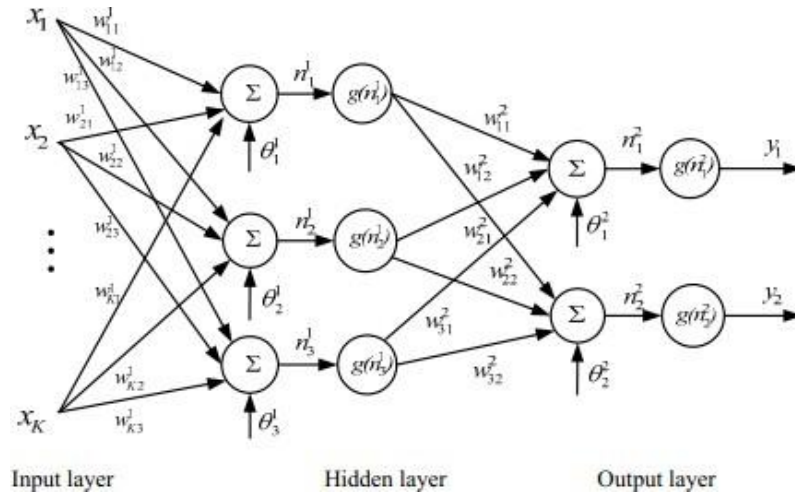


Figure II.13 : MLP avec une seule couche cachée.

La sortie y_i , $i=1,2$ du MLP de la **figure II.13** est donnée par :

$$y_i = g \left(\sum_{j=1}^3 w_{ji}^2 g(n_j^1) + \theta_j^2 \right) = g \left(\sum_{j=1}^3 w_{ji}^2 g \left(\sum_{k=1}^K w_{kj}^1 x_k + \theta_j^1 \right) + \theta_j^2 \right) \quad (\text{II.4})$$

À partir de cette équation, on peut conclure que le MLP est un réseau à paramètres non linéaires. Les paramètres sont les poids w_k et les biais θ_k . La fonction d'activation g est généralement supposée identique pour chaque couche.

II.7 Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (en anglais Convolutional Neural Networks ou CNN) sont un type particulier de réseaux de neurones artificiels, principalement utilisés pour le traitement d'images, de vidéos ou de données ayant une structure spatiale ou temporelle.

Le terme « réseau de neurones convolutif » fait référence à l'utilisation, par le réseau, d'une opération mathématique connue sous le nom de convolution. Les réseaux convolutifs remplacent la multiplication générale de matrices, utilisée dans les réseaux classiques, par une ou plusieurs convolutions dans certaines couches.

Le CNN est l'un des meilleurs algorithmes d'apprentissage pour effectuer cette opération, ce qui facilite l'extraction de caractéristiques pertinentes à partir de données localement connectées. La sortie des noyaux convolutifs est ensuite transmise à une fonction d'activation

(une unité de traitement non linéaire), qui permet à la fois l'apprentissage d'abstractions et l'introduction de non-linéarité dans l'espace des caractéristiques. Cette non-linéarité permet la génération de divers motifs d'activation, facilitant ainsi la détection et l'apprentissage des différences de signification dans les images.

La topologie d'un CNN est généralement structurée en plusieurs étapes d'entraînement, comprenant des couches convolutives, des fonctions d'activation non linéaires et des couches de sous-échantillonnage (ou de pooling) [35] [36]. La structure générale d'un réseau CNN est représentée dans la figure II.14.

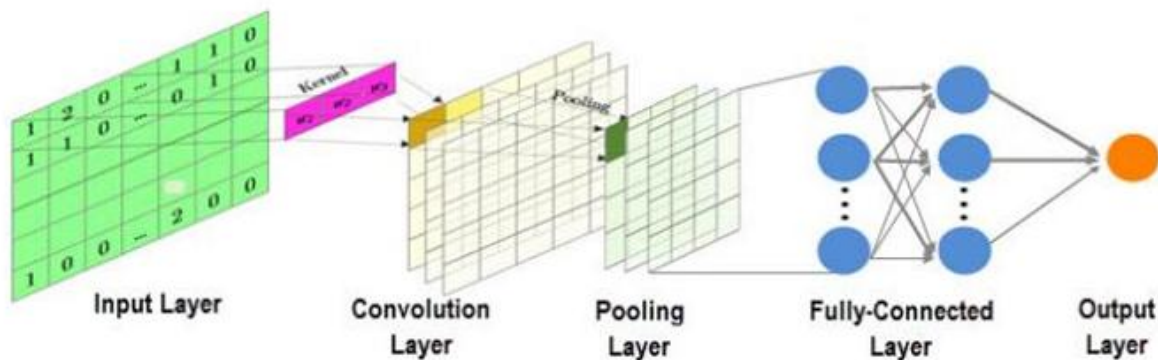


Figure II.14: Architecture d'un CNN [27].

II.7.1 Architecture des CNN

Les CNN représentent aujourd'hui les architectures profondes les plus performantes pour la classification d'images. Leur conception multicouche s'inspire des mécanismes biologiques. Ces réseaux apprennent automatiquement, et de manière hiérarchique, des caractéristiques invariantes, ce qui leur permet de reconnaître et de combiner ces traits pour modéliser des structures complexes.

L'architecture des CNN comprend de nombreuses couches, comme illustré à la figure II.15, allant de la couche d'entrée à la couche de sortie, en passant par plusieurs étapes essentielles qui peuvent être répétées. Chacune de ces étapes joue un rôle crucial dans l'efficacité globale du réseau [37].

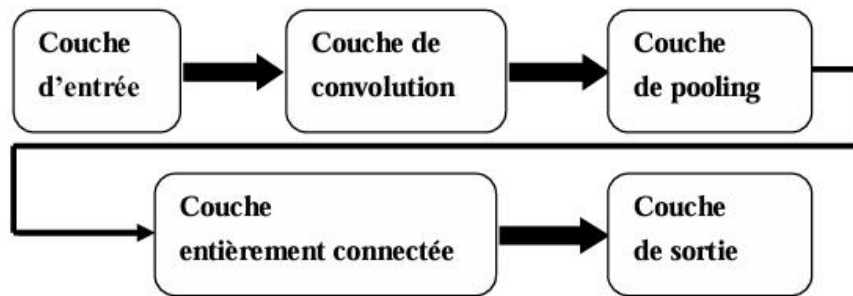


Figure II.15 : Différentes couches d'un CNN.

II.7.1.1 Couche d'entrée

La couche d'entrée représente la première étape du traitement dans un CNN. Son rôle est de recevoir les données brutes, qu'il s'agisse de texte, de signaux audio ou, plus fréquemment, d'images.

La couche d'entrée constitue la première étape du traitement dans un réseau de neurones convolutifs (CNN). Son rôle est de recevoir les données brutes, qu'il s'agisse de texte, de signaux audio ou, plus fréquemment, d'images.

Dans le cas du traitement d'images, cette couche accepte une matrice tridimensionnelle représentant l'image à analyser. Cette matrice est généralement structurée selon trois dimensions principales (voir [figure II.16](#)) : la hauteur de l'image (en pixels), la largeur de l'image (en pixels), et la profondeur, qui correspond au nombre de canaux de couleur.

Par exemple pour une image en niveaux de gris, la profondeur est égale à 1 (un seul canal), et pour une image en couleur RGB, la profondeur est égale à 3 (rouge, vert, bleu).

La couche d'entrée ne modifie pas directement les valeurs des pixels, mais elle prépare les données pour les couches suivantes (convolutives, d'activation, etc.) en standardisant leur format et leurs dimensions.

Dans certains cas, une normalisation préalable est également appliquée afin d'améliorer la stabilité de l'apprentissage (par exemple en ramenant les valeurs des pixels entre 0 et 1 ou en les centrant autour de zéro).

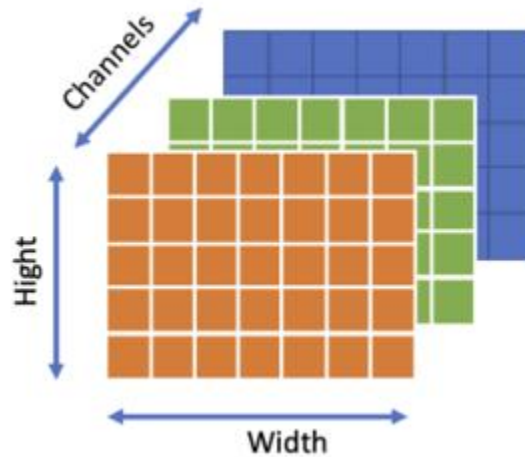


Figure II.16 : Trois dimensions d'une image.

II.7.1.2 Couche de convolution

Pour extraire des caractéristiques d'une image d'entrée. La convolution maintient l'association entre les pixels en apprenant les caractéristiques de l'image via les données d'entrée de petits carrés. Cette opération mathématique a deux entrées, à savoir un noyau ou filtre et une matrice d'image [38].

La couche de convolution constitue une étape clé dans les CNN. Son rôle principal est d'extraire automatiquement des caractéristiques pertinentes (ou features) à partir de l'image d'entrée, comme les contours, les textures ou d'autres motifs visuels.

L'opération de convolution repose sur deux éléments :

- Une matrice d'entrée, représentant l'image.
- Un noyau de convolution ou filtre, généralement de petite taille (par exemple 3×3 ou 5×5).

Le filtre glisse (ou convole) sur l'image d'entrée, en multipliant les valeurs des pixels de la région locale par les poids du filtre, puis en sommant les produits obtenus. Cette somme devient la valeur d'un pixel dans l'image de sortie, appelée carte de caractéristiques (feature map). Plusieurs filtres peuvent être appliqués en parallèle, chacun apprenant à détecter un type de motif spécifique pendant l'entraînement du réseau. La figure II.17 illustre un exemple de convolution.

L'opération de convolution présente plusieurs avantages : elle permet de préserver les relations spatiales entre les pixels, de réduire les dimensions de l'image lorsqu'elle est

combinée à un stride supérieur à 1 ou à l'absence de padding, et de capturer des motifs locaux, essentiels à la reconnaissance visuelle.

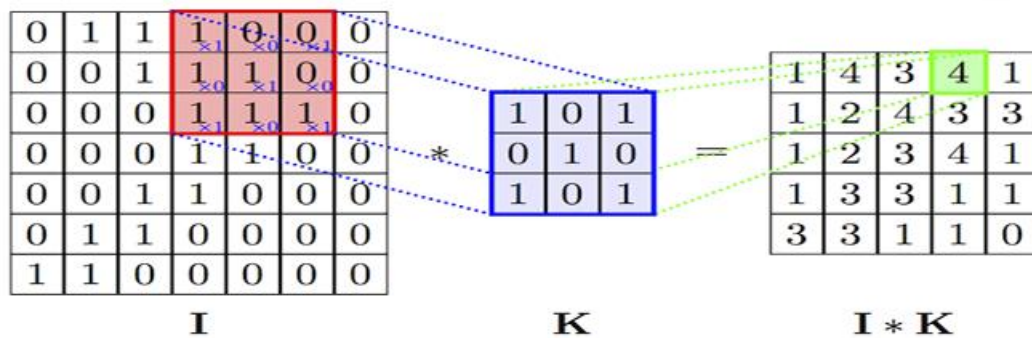


Figure II.17 : Exemple de convolution d'une image d'entrée I avec un filtre K .

II.7.1.3 Couche de pooling

La couche de pooling (ou sous-échantillonnage) est une méthode utilisée pour réduire la dimensionnalité de l'entrée, tout en conservant les informations les plus importantes. Elle est généralement placée entre deux couches de convolution. Contrairement à ces dernières, les couches de pooling ne comportent pas de poids à apprendre.

Le pooling permet de diminuer la charge de calcul du CNN et de limiter le risque de surapprentissage en réduisant la complexité du modèle. Il agit en agrégeant des valeurs selon une fonction spécifique. Il existe différents types de pooling [39], les plus courants sont :

- le maximum pooling qui sélectionne la valeur maximale parmi les pixels d'une région donnée.
- l'average pooling qui calcule la moyenne des valeurs des pixels dans cette même région.

La figure II.18 montre le résultat de l'application de ces deux types de pooling (max et moyen) sur une portion de matrice d'image.

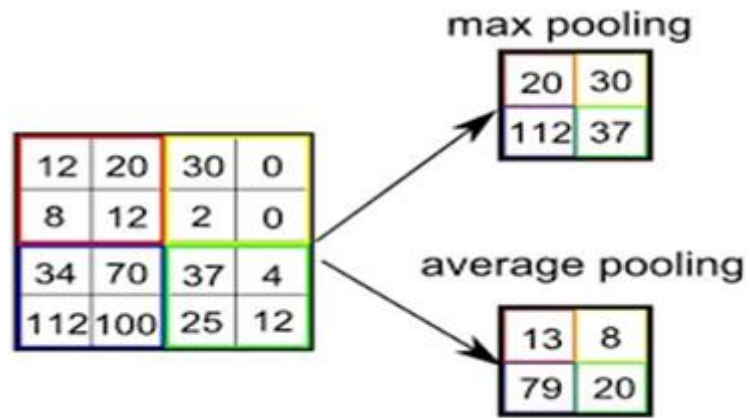


Figure II.18 : Deux principaux types de pooling (max et moyen).

II.7.1.4 Couche de normalisation (Batch normalisation)

La couche de normalisation, appelée en anglais Batch Normalization (BN), est une étape importante dans les réseaux de neurones profonds, en particulier dans les CNN. Introduite pour améliorer la stabilité et la vitesse de l'apprentissage, cette couche permet de normaliser les activations intermédiaires d'un réseau au sein d'un mini-batch.

Lors de l'entraînement d'un réseau de neurones, les distributions des sorties d'une couche peuvent changer, ce qui ralentit l'apprentissage. Ce phénomène est connu sous le nom de "covariate shift" interne. La couche de normalisation atténue cet effet en maintenant les distributions stables, ce qui permet d'utiliser des taux d'apprentissage plus élevés et rend l'entraînement plus efficace. Concrètement, la batch normalization agit en normalisant les sorties de la couche précédente, c'est-à-dire en recentrant les valeurs autour de la moyenne (généralement 0) et en les échelonnant selon leur écart-type (généralement 1), avant d'appliquer une transformation linéaire apprenable (avec des paramètres de mise à l'échelle γ et de translation β).

II.7.1.4 Couche d'aplatissement (Flatten)

La couche d'aplatissement, appelée Flatten en anglais, constitue une étape de transition entre les couches de convolution (et éventuellement de pooling) et les couches entièrement connectées (ou fully connected layers) dans une architecture CNN.

Après l'extraction des caractéristiques par les couches précédentes, les données sont encore organisées sous forme de matrices ou de tenseurs à plusieurs dimensions. Or, les couches entièrement connectées requièrent une entrée sous forme de vecteur unidimensionnel. C'est précisément le rôle de la couche Flatten : transformer une matrice 2D ou un tenseur 3D en un

vecteur 1D. Par exemple, si la sortie d'une couche de convolution ou de pooling est une matrice de dimensions (hauteur \times largeur \times profondeur), la couche Flatten va convertir cette matrice en un vecteur de taille égale au produit de ces trois dimensions. Cette opération ne modifie pas les valeurs des éléments, elle change uniquement leur organisation.

Cette étape est cruciale car elle permet de relier les caractéristiques extraites par les couches de convolution à la phase de classification ou de régression finale du réseau. Elle marque ainsi le passage du traitement spatial au traitement global de l'information. La **figure II.19** illustre le principe de l'aplatissement (flattening).

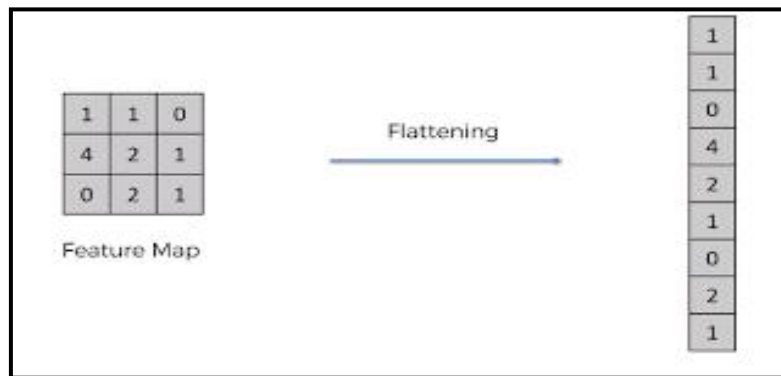


Figure II.19 : Opération d'aplatissement (flattening).

II.7.1.5 Couche entièrement connectée

Dans les modèles classiques, la couche entièrement connectée (ou fully connected layer, abrégée FC) fonctionne comme un réseau de neurones dense où chaque neurone est connecté à tous les neurones de la couche précédente (voir la **figure II.20**). À ce stade, la sortie de la phase précédente qui est composée de plusieurs couches de convolution et de pooling est aplatie en un vecteur unidimensionnel, puis transmise à la couche FC.

Dans cette couche, un produit scalaire est effectué entre le vecteur d'entrée et un vecteur de poids, auquel est ajoutée une valeur de biais, pour produire la sortie. Cette opération permet de combiner les caractéristiques extraites précédemment afin de générer une prédiction finale (comme une probabilité de classe) [38].

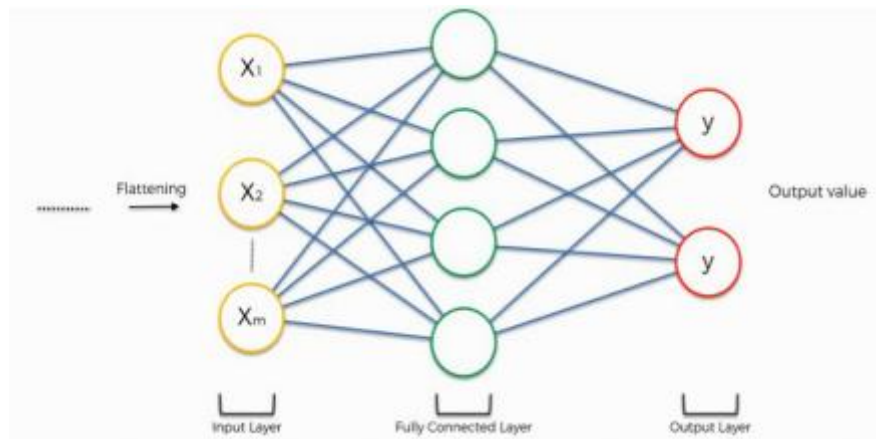


Figure II.20 : Architecture de la couche entièrement connectée.

II.7.1.6 Couche de sortie

La couche de sortie (ou output layer en anglais) transforme les résultats finaux du réseau en un vecteur dont la dimension correspond au nombre de classes à prédire. Chaque composante de ce vecteur représente une étiquette de classe, généralement codée sous forme numérique.

La fonction d'activation utilisée dans cette couche dépend du type de tâche de classification :

Pour une classification multiclasse, on utilise généralement la fonction d'activation softmax (voir la [figure II.21](#)), également appelée fonction exponentielle normalisée. Cette fonction transforme un vecteur de scores réels (appelés logits) en une distribution de probabilités sur l'ensemble des classes. Elle procède en exponentiant chaque valeur d'entrée, puis en les divisant par la somme totale des exponentielles. Ainsi, chaque sortie est comprise entre 0 et 1, et la somme des sorties est égale à 1. Ces valeurs peuvent être interprétées comme des probabilités d'appartenance à chaque classe, la classe ayant la plus grande probabilité étant choisie comme prédiction finale.

Pour une classification binaire, on utilise plutôt la fonction sigmoïde (voir la [figure II.22](#)). Cette fonction transforme un score réel en une valeur comprise entre 0 et 1, représentant directement la probabilité d'appartenance à la classe positive. Si cette probabilité dépasse un

seuil (généralement 0,5), l'instance est classée comme appartenant à cette classe ; sinon, elle est attribuée à la classe négative.

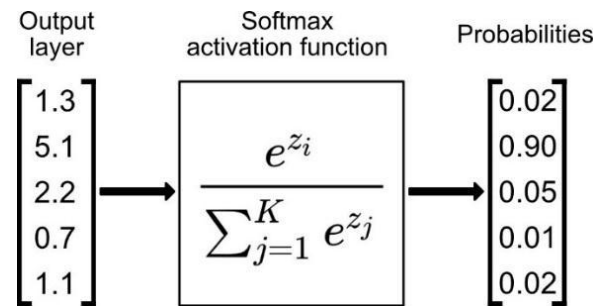


Figure II.21 : Fonction softmax [40].

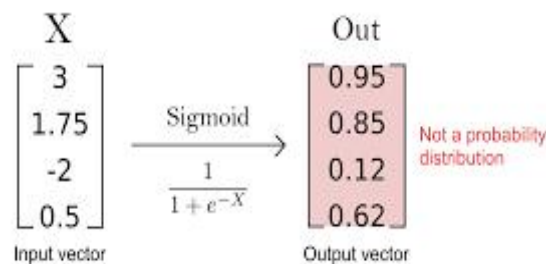


Figure II.22 : Fonction sigmoïde [40].

II.7.2 Autres composantes et techniques utilisées dans les CNN

II.7.2.1 Fonction de perte (ou fonction d'erreur)

La fonction de perte (également appelée fonction d'erreur ou fonction coût, en anglais loss function) détermine la manière dont l'apprentissage du réseau pénalise l'écart entre la sortie prédite et la sortie réelle. Elle mesure l'erreur entre la sortie attendue (ou connue) et celle produite par le réseau, et guide ainsi la mise à jour des poids au cours de l'apprentissage.

Il existe différentes fonctions de perte, chacune étant adaptée à un type spécifique de tâche [41]:

- Lorsque la fonction Softmax est utilisée pour prédire une seule classe parmi K classes mutuellement exclusives, on applique généralement la fonction d'entropie croisée (en anglais categorical cross-entropy).

- Si on cherche à prédire K probabilités indépendantes dans l'intervalle $[0, 1]$, la fonction d'entropie croisée sigmoïde (ou sigmoid cross-entropy) est alors plus appropriée.
- La fonction de perte euclidienne (ou erreur quadratique moyenne) est couramment utilisée dans les tâches de régression, où l'objectif est de prédire des valeurs numériques continues.
- Dans le cas d'une classification binaire (deux classes), la fonction de perte la plus couramment utilisée est l'entropie croisée binaire (binary cross-entropy), qui est définie comme suit :

$$\text{fonction d'erreur} = - \sum_{i=1}^{C=2} t_i \log(s_i) = t_1 \log(s_1) - (1 - t_1) \log(1 - s_1) \quad (\text{II.5})$$

Où C représente l'ensemble des classes, s_i est la probabilité prédite pour la classe i , et t est la valeur réelle (0 ou 1) associée à cette classe.

II.7.2.2 Abandon aléatoire (Dropout)

Le problème de surapprentissage (overfitting en anglais) survient généralement lorsque certains neurones deviennent trop fortement corrélés ou dépendants les uns des autres au fil de l'entraînement. Cette dépendance excessive limite la capacité de généralisation du modèle.

La technique de dropout permet de réduire ce phénomène en supprimant de manière aléatoire certaines connexions neuronales pendant l'apprentissage. À chaque itération, un sous-ensemble des neurones est désactivé temporairement, ce qui force le réseau à apprendre des représentations plus robustes et indépendantes.

Un paramètre clé de cette méthode est le taux de dropout, qui définit la probabilité de désactivation d'un neurone. Ce taux est une valeur comprise entre 0 et 1. En pratique, il est généralement fixé entre 0,2 et 0,75, selon la complexité du modèle et la taille des données d'entraînement.

II.7.2.3 Normalisation par lots (Batch Normalization)

La normalisation par lots (batch normalization en anglais) est une technique utilisée pour améliorer à la fois la vitesse d'entraînement et la performance finale d'un réseau de neurones artificiel. Elle consiste à normaliser les données en entrée de chaque couche, en les recentrant autour de zéro et en les mettant à l'échelle (généralement avec une variance unitaire).

Cette opération est appliquée sur les mini-lots (batches) de données pendant l'entraînement, ce qui permet de réduire la variance interne du covariate qui est un phénomène qui peut ralentir l'apprentissage. La normalisation par lots a plusieurs avantages :

- ✓ Elle permet de stabiliser l'apprentissage du réseau.
- ✓ Elle accélère la convergence.
- ✓ Elle agit comme une forme de régularisation, limitant le surapprentissage.

En ajustant les valeurs des activations intermédiaires, elle évite que certaines valeurs très grandes ne dominent le processus d'apprentissage et permet également aux petites valeurs qui sont souvent porteuses d'information d'être mieux exploitées par le modèle.

II.7.2.4 Arrêt anticipé (Early Stopping)

L'arrêt anticipé (early stopping en anglais) est une méthode de régularisation utilisée pour éviter le surapprentissage. Elle consiste à interrompre l'entraînement d'un modèle lorsque la fonction de perte sur un ensemble de validation commence à augmenter, indiquant ainsi une dégradation des performances de généralisation. En d'autres termes, bien que le modèle continue à s'améliorer sur les données d'entraînement, ses performances sur les données qu'il n'a jamais vues (les données de validation) se détériorent. L'arrêt anticipé permet donc de conserver le modèle dans un état où il généralise le mieux, en stoppant l'entraînement avant qu'il ne commence à surajuster les données d'apprentissage. La [figure II.23](#) explique cette procédure [42].

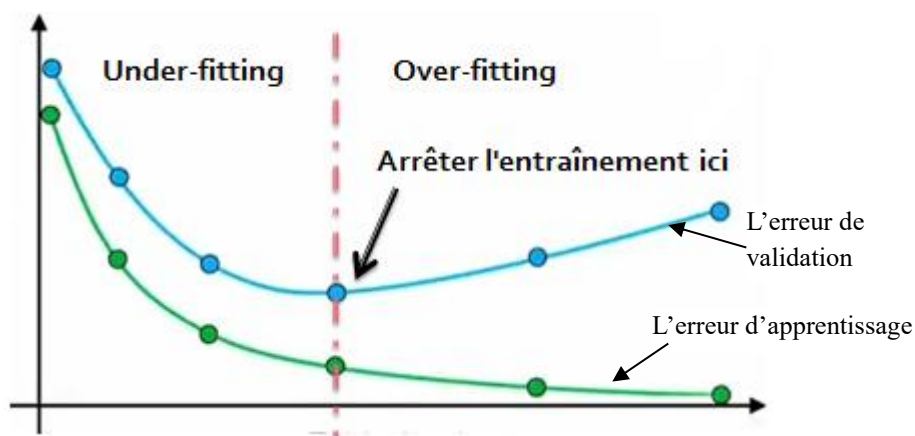


Figure II.23 : Utilisation de l'arrêt anticipé durant la phase d'entraînement d'un modèle.

II.7.2.5 Augmentation des données (Data Augmentation)

L'augmentation des données, en analyse de données, est une technique utilisée pour accroître la quantité d'informations disponibles en ajoutant des copies légèrement modifiées de données existantes ou des données synthétiques générées à partir de celles-ci.

Elle agit comme une méthode de régularisation et aide à réduire le surapprentissage lors de l'entraînement d'un modèle d'apprentissage automatique. Cette approche est étroitement liée à la technique de suréchantillonnage en analyse de données [43].

II.7.2.6 Apprentissage par transfert (Transfer Learning)

L'apprentissage par transfert, ou transfer learning en anglais, désigne les méthodes qui permettent de réutiliser les connaissances acquises lors de la résolution d'un problème pour en résoudre un autre.

Le transfer learning s'est largement développé avec l'essor du deep learning. En effet, les modèles utilisés dans ce domaine demandent souvent beaucoup de calcul et de ressources. En utilisant des modèles préentraînés comme point de départ, l'apprentissage par transfert permet de créer rapidement des modèles efficaces et de résoudre des problèmes complexes plus facilement [44].

II.7.3 Optimisation dans les CNN

II.7.3.1 Descente de gradient stochastique

La méthode de descente de gradient stochastique (Stochastic Gradient Descent, ou SGD) et ses variantes comptent parmi les algorithmes d'optimisation les plus répandus en apprentissage automatique, en particulier en apprentissage profond.

Plutôt que de calculer le gradient exact sur l'ensemble des données, SGD l'estime en utilisant un mini-lot (ou mini-batch) d'échantillons tirés aléatoirement de la distribution des données. L'algorithme ajuste alors les paramètres du modèle en suivant cette estimation, dans le but de minimiser la fonction de coût.

Un paramètre crucial de SGD est le taux d'apprentissage, qui détermine la taille des pas effectués lors de la mise à jour des poids. Bien que SGD utilise souvent un taux d'apprentissage fixe au départ, il est courant en pratique de le diminuer progressivement au fil de l'entraînement pour favoriser la convergence.

À titre de comparaison, le calcul du gradient exact sur l'ensemble des données (batch gradient) produit un gradient qui tend vers zéro à mesure que le modèle s'approche du minimum de la fonction de coût [45].

II.7.3.2 Optimiseur RMSprop

RMSprop est une méthode d'optimisation introduite par le professeur Geoffrey Hinton lors de ses cours sur les réseaux de neurones. Contrairement à l'algorithme Momentum (une extension de la descente de gradient classique), qui accumule les gradients sur l'ensemble de l'historique, RMSprop ne conserve que les gradients récents dans une fenêtre temporelle limitée.

Cette approche permet de réduire les oscillations pendant l'apprentissage, mais selon un mécanisme différent de celui de Momentum. Un avantage majeur de RMSprop est sa capacité à ajuster automatiquement le taux d'apprentissage, évitant ainsi la nécessité de le régler manuellement. De plus, il attribue un taux d'apprentissage distinct à chaque paramètre du modèle.

Les mises à jour des paramètres sont effectuées individuellement à l'aide de formules spécifiques, ce qui améliore la stabilité et l'efficacité de l'optimisation [46].

II.7.3.3 Optimiseur Adam

Adam (acronyme de "Adaptive Moment Estimation") est l'un des algorithmes d'optimisation les plus couramment utilisés en Deep Learning. Il se distingue par sa manière innovante d'exploiter les gradients passés pour ajuster les mises à jour actuelles des paramètres. En combinant les concepts de Momentum (moyenne des gradients) et de RMSprop (moyenne des carrés des gradients), Adam intègre des moyennes pondérées des gradients précédents afin de stabiliser et d'accélérer l'apprentissage.

Cette approche a largement gagné en popularité et constitue aujourd'hui une norme dans l'entraînement des réseaux de neurones profonds. Bien que certains optimiseurs plus récents puissent offrir des performances supérieures dans certains cas, Adam reste très apprécié pour sa robustesse, notamment face à la sensibilité aux hyperparamètres. Toutefois, il peut parfois être nécessaire d'ajuster le taux d'apprentissage par rapport à la valeur par défaut pour obtenir de meilleurs résultats [47].

II.7.3 Avantages et inconvénients des CNN

II.7.3.1 Avantages des CNN

- Les CNN sont capables de traiter des images de grande taille avec des ressources de calcul limitées, grâce à l'utilisation de couches de sous-échantillonnage et de pooling.
- Les CNN sont hautement parallélisables, ce qui permet un entraînement efficace sur des architectures de calcul modernes, telles que les GPU et les TPU.
- Les CNN ont été largement étudiés et développés ces dernières années, ce qui a conduit à la disponibilité de modèles préentraînés de haute qualité pour une grande variété de tâches.

II.7.3.2 Inconvénients des CNN

- Les CNN peuvent être sensibles aux variations d'éclairage, de positionnement et de résolution des images, ce qui peut affecter les performances du modèle.
- Les CNN sont basés sur une architecture fixe, ce qui les rend peu flexibles pour traiter des entrées de tailles et de formes variables.
- Les CNN peuvent être coûteux en termes de ressources de calcul et de temps d'entraînement, en particulier pour les tâches de vision par ordinateur complexes.
- Les CNN peuvent être sujets au surapprentissage s'ils sont entraînés sur des ensembles de données trop petits ou déséquilibrés.

III.8 Conclusion

Dans ce chapitre, nous avons présenté les fondements du deep learning, en mettant l'accent sur ses principes, ses techniques principales, ainsi que ses multiples domaines d'application. Une attention particulière a été accordée aux CNN, en détaillant leur architecture, leurs composants essentiels, ainsi que les méthodes d'optimisation et les techniques d'amélioration des performances.

Cette étude théorique nous a permis de mieux comprendre le fonctionnement des CNN et leur pertinence dans les tâches de classification et de détection, notamment dans le domaine médical. Ces notions constituent une base essentielle pour la mise en œuvre des modèles que nous allons développer dans le chapitre suivant, dédié à l'application concrète des CNN pour la détection du diabète à partir d'images du fond d'œil.

Chapitre 3

Expériences et résultats

III.1 Introduction

Après avoir présenté la théorie du deep learning dans le chapitre précédent, ce chapitre est consacré à la mise en œuvre d'un système basé sur le deep learning pour la détection de la rétinopathie diabétique. Pour cela, nous utilisons une base d'images de fond d'œil contenant deux classes : la première représente des cas positifs de diabète, la seconde des cas négatifs. Nous présentons également des modèles CNN développés pour la classification de cette maladie. Les résultats des tests des différentes architectures seront également présentés et détaillés dans ce chapitre.

III.2 Environnement de travail

Dans cette section, nous présenterons le matériel et les logiciels utilisés dans notre travail.

III.2.1 Environnement matériel

Afin de mettre en œuvre ce projet, nous avons utilisé un ensemble de matériel dont les caractéristiques sont les suivantes :

- ✓ Un ordinateur portable HP
- ✓ Processeur : Intel(R) Celeron(R) CPU N3060 @ 1.60GHz
- ✓ Mémoire (RAM) : 8 Go
- ✓ Système d'exploitation: Microsoft Windows 10 (64bits).

III.2.2 Langage de programmation Python

Python a été développé à l'Institut néerlandais de mathématiques et d'informatique (CWI) à Amsterdam par Guido van Rossum à la fin des années 1980, et sa première annonce remonte à 1991. Le noyau du langage est écrit en langage C. Van Rossum a appelé son langage « Python » pour exprimer son admiration pour un célèbre groupe de sketches britannique, les Monty Python [48].

Python est un langage de programmation de haut niveau, conçu pour être facile à lire et à implémenter. Il est open source, ce qui signifie qu'il est gratuit, même pour des applications commerciales. Python peut fonctionner sur les systèmes Mac, Windows et Unix [49].



Figure III.1 : Logo de Python.

III.2.3 Google Colab

Google Colab, ou Colaboratory, est un service cloud gratuit offert par Google, basé sur Jupyter Notebook et destiné à la formation et à la recherche en apprentissage automatique.

Cette plateforme permet d'entraîner des modèles de machine learning directement dans le cloud, sans avoir besoin d'installer quoi que ce soit sur l'ordinateur, à l'exception d'un navigateur [50].

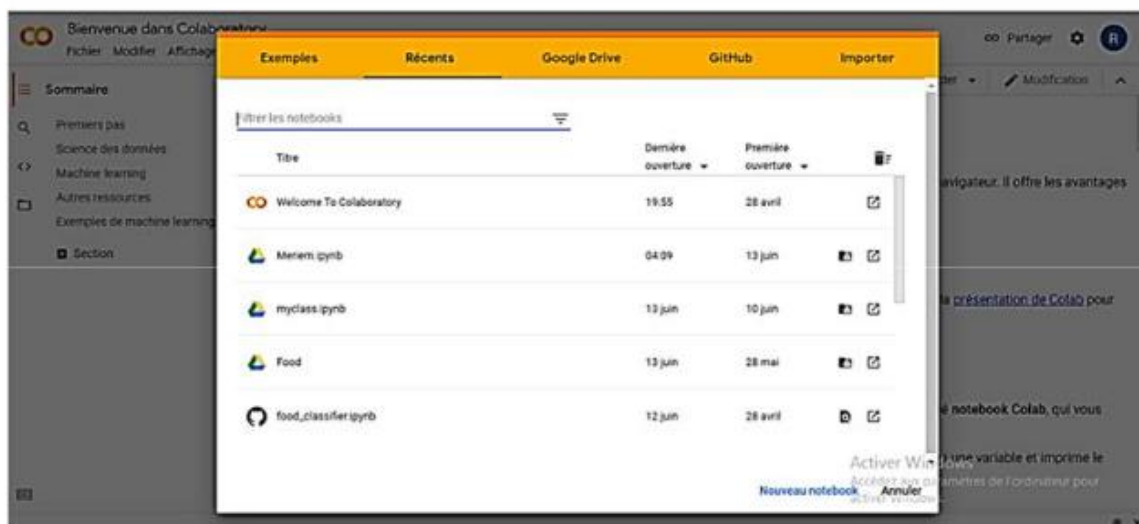


Figure III.2 : Interface de Google Colab.

III.2.4 Navigateur Anaconda

Le Navigateur Anaconda est une interface graphique (GUI) incluse dans la distribution Anaconda. Elle permet aux utilisateurs de lancer des applications ainsi que de gérer les bibliothèques Conda. Le Navigateur peut également accéder à des bibliothèques disponibles sur le cloud Anaconda ou dans un dépôt Anaconda local, afin de les installer dans un environnement, les exécuter et les mettre à jour. Il est disponible pour Windows, macOS et Linux [51].

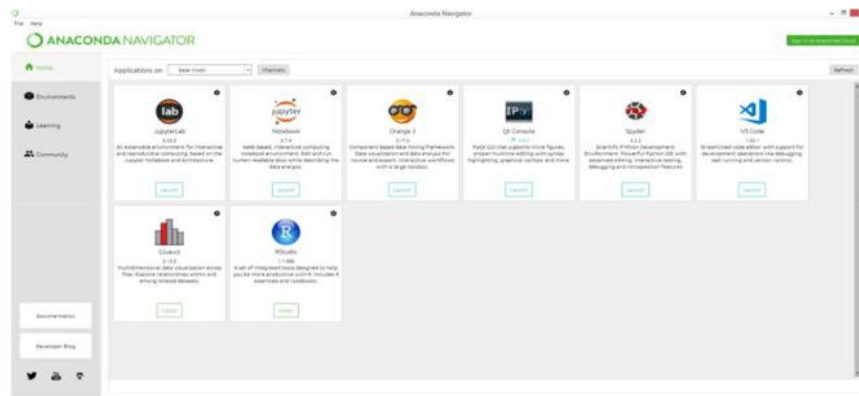


Figure III.3 : Navigateur Anaconda.

III.2.5 Jupyter Notebook

Jupyter Notebook est une application web open source qui permet de créer et de partager des documents contenant du code en direct, des équations, des visualisations et du texte narratif.

Parmi ses utilisations figurent le nettoyage et la transformation des données, la simulation numérique, la modélisation statistique, la visualisation de données, l'apprentissage automatique et bien plus encore [52].

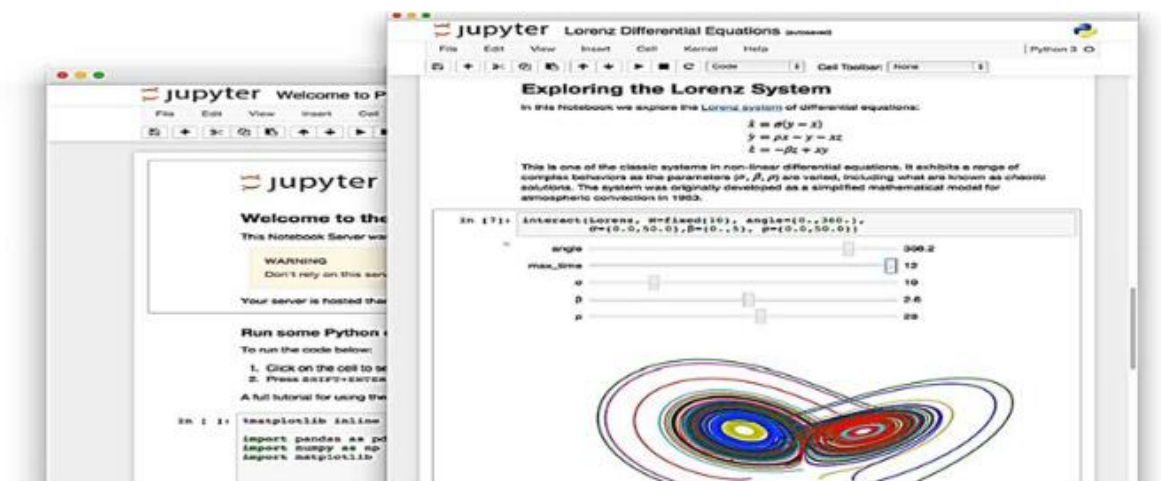


Figure III.4 : Jupyter notebook.

III.2.6 Keras

Keras est une API de haut niveau permettant de créer et d'entraîner des modèles d'apprentissage profond. Elle peut s'exécuter sur TensorFlow et est utilisée pour le prototypage rapide, la recherche avancée et la mise en production.

Les modèles Keras sont construits en reliant entre eux des blocs de construction configurables. Ainsi, Keras permet de créer de nouvelles couches, de définir des fonctions personnalisées et de développer des modèles à la pointe de la technologie, avec peu de restrictions et en seulement quelques lignes de code [53].



Figure III.5: Logo de Keras.

III.2.7 TensorFlow

TensorFlow est un outil open source d'apprentissage automatique développé par Google. Le code source a été rendu public le 9 novembre 2015 et publié sous licence Apache. Il est fondé sur l'infrastructure DistBelief, initiée par Google en 2011, et dispose d'une interface pour Python, Julia et R.

TensorFlow est l'un des outils les plus utilisés en intelligence artificielle dans le domaine de l'apprentissage automatique.



Figure III.6: Logo de TensorFlow.

III.2.8 OpenCV

OpenCV (Open Source Computer Vision Library) est une bibliothèque logicielle open source dédiée à la vision par ordinateur et à l'apprentissage automatique. Elle a été créée pour fournir une infrastructure commune aux applications de vision par ordinateur et pour accélérer l'intégration de la perception machine dans les produits commerciaux.

Sous licence BSD, OpenCV permet aux entreprises d'utiliser et de modifier facilement le code. La bibliothèque contient plus de 2 500 algorithmes optimisés, couvrant un ensemble complet d'algorithmes de vision par ordinateur et d'apprentissage automatique, allant des classiques aux plus avancés.

Ces algorithmes peuvent être utilisés pour :

- détecter et reconnaître des visages ;
- identifier des objets ;
- classifier des actions humaines dans des vidéos ;
- suivre les mouvements de caméra ;
- suivre des objets en mouvement ;
- extraire des modèles 3D d'objets ;
- produire des nuages de points 3D à partir de caméras stéréo ;
- assembler des images pour créer une image haute résolution d'une scène entière ;
- trouver des images similaires dans une base de données ;
- supprimer les yeux rouges des photos prises avec un flash ;
- suivre les mouvements des yeux ;
- reconnaître des paysages ;
- établir des marqueurs à superposer en réalité augmentée, etc.

OpenCV compte plus de 47000 membres dans sa communauté et son nombre de téléchargements est estimé à plus de 18 millions. Elle est largement utilisée par des entreprises, des groupes de recherche et des organismes gouvernementaux.

III.2.9 Numpy

La bibliothèque NumPy (Numerical Python) est une bibliothèque populaire en Python, qui fournit des structures de données et des fonctions permettant de manipuler efficacement des tableaux multidimensionnels.

Elle est largement utilisée dans les domaines scientifiques et numériques, tels que le calcul mathématique, la manipulation de tableaux, l'algèbre linéaire, le traitement du signal et la simulation.

III.2.10 Pandas

Pandas est une bibliothèque open source de manipulation et d'analyse de données pour le langage de programmation Python. Elle offre des structures de données flexibles et expressives, principalement les DataFrames et les Series, qui facilitent la gestion de données tabulaires et de séries temporelles.

Pandas est particulièrement utile pour nettoyer, transformer et analyser les données, grâce à ses fonctionnalités puissantes de sélection, filtrage, agrégation et visualisation.

III.3 Dataset utilisé

Le dataset Messidor-2 [54] est une base de référence largement utilisée pour l'évaluation des algorithmes de détection de la rétinopathie diabétique (RD). Il regroupe un grand nombre d'images issues de 874 examens (deux images par œil, centrées sur la macula), aux formats PNG ou JPG.

Un fichier CSV accompagne chaque image d'examen, avec les colonnes suivantes :

- ✓ **image_id** : nom du fichier dans Messidor-2.
- ✓ **adjudicated_dr_grade** : note de la RD selon l'échelle ICDR (International Clinical Diabetic Retinopathy) à 5 niveaux (0 : aucun signe, 1 : RD légère, 2 : RD modérée, 3 : RD sévère, 4 : RD proliférante).
- ✓ **adjudicated_dme** : indication de la présence de DME (Diabetic Macular Edema ou œdème maculaire diabétique) (0 : non, 1 : oui), défini par des exsudats durs dans un rayon d'un disque optique autour de la macula.
- ✓ **adjudicated_gradable** : qualité de l'image (1 : exploitable, 0 : non exploitable). Si la valeur est 0, les champs RD et DME sont laissés vides, car l'image ne peut être utilisée.

Les annotations ont été réalisées selon le protocole d'arbitrage décrit dans Krause et al. [55], qui a montré une amélioration de la qualité des annotations par rapport à Gulshan et al. [56], grâce à l'utilisation de données arbitrées.

La colonne adjudicated_gradable est essentielle pour filtrer les images exploitables : seules celles dont la valeur est à 1 doivent être incluses dans l'entraînement ou l'évaluation du modèle.

Messidor-2 présente plusieurs avantages pour la recherche :

- Une base d'images réelles et variées, utile pour évaluer la robustesse des modèles de détection.
- Des annotations fiables, réalisées par des experts, garantissant la rigueur des labels RD et DME.
- Une échelle de gravité étendue (RD ICDR 0 à 4) et des annotations DME, permettant une classification multiclassées.

Dans le cadre de cette étude, seules les images gradables (adjudicated_gradable = 1) ont été conservées. Après filtrage, les colonnes « adjudicated_dr_grade » et « adjudicated_dme » sont entièrement renseignées, assurant l'intégrité du jeu de données pour l'entraînement et l'évaluation des performances des modèles.

III.3.1 Simplification de la classification ICDR dans notre étude

Afin de faciliter l'apprentissage automatique et d'améliorer la robustesse du modèle de classification, nous avons choisi de regrouper les niveaux de la classification ICDR en deux classes : présence ou absence de rétinopathie diabétique (RD).

L'échelle ICDR (International Clinical Diabetic Retinopathy) comprend cinq niveaux :

Niveau ICDR	Signification
0	Aucun signe de rétinopathie (pas de RD)
1	RD légère non proliférante
2	RD modérée non proliférante
3	RD sévère non proliférante
4	RD proliférante (PDR)

Table III.1 : Classification ICDR.

Pour simplifier le problème de classification en une tâche binaire, nous avons appliqué le regroupement suivant :

- Classe 0 : Absence de rétinopathie → correspond au niveau 0 de l'échelle ICDR.
- Classe 1 : Présence de rétinopathie → regroupe les niveaux 1 à 4 de l'échelle ICDR, soit toutes les formes de rétinopathie, de légère à proliférante.

Ce regroupement permet de traiter la détection de la RD comme une tâche de classification binaire, plus adaptée à une première étape de dépistage automatisé. Cette approche est particulièrement pertinente dans un contexte clinique où l'objectif est de repérer les patients nécessitant un examen ophtalmologique approfondi, quel que soit le stade exact de la maladie.

La **figure III.7** montre quelques images de notre dataset.

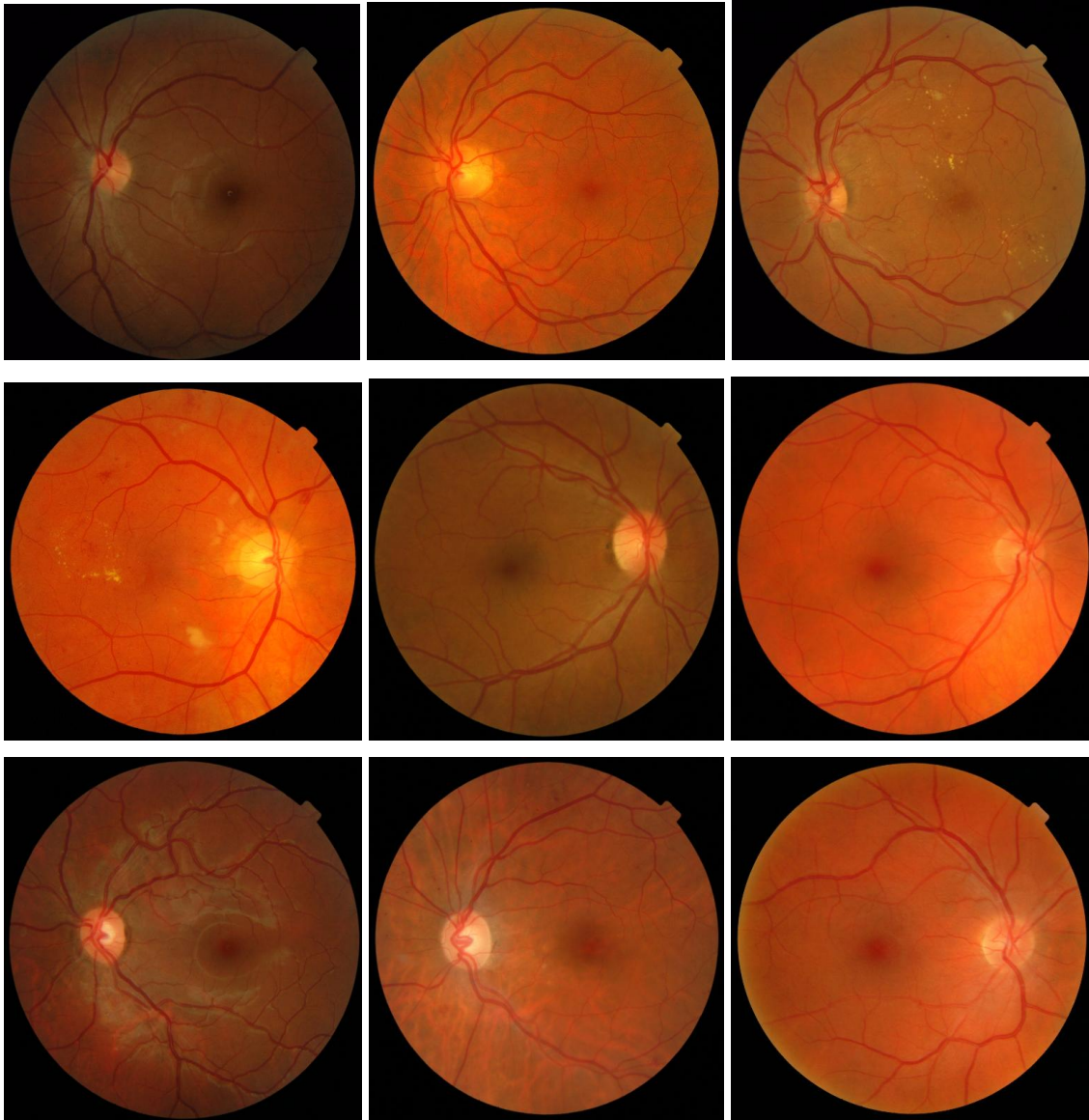


Figure III.7: Exemple d'images du dataset Messidor-2.

III.4 Augmentation des données

L'augmentation des données est une technique qui consiste à accroître la diversité et la quantité d'images d'entraînement en appliquant un ensemble de transformations aléatoires (telles que des rotations, des zooms, des renversements horizontaux, etc.). L'objectif est que le modèle ne voie jamais deux fois exactement la même image, ce qui permet de réduire le surapprentissage et d'améliorer la capacité de généralisation du modèle.

Dans Keras, cette opération peut être réalisée à l'aide de la classe `keras.preprocessing.image.ImageDataGenerator`.

`ImageDataGenerator` est un utilitaire fourni par les bibliothèques TensorFlow et Keras, qui permet d'effectuer de l'augmentation de données en temps réel pendant l'entraînement du modèle. Plutôt que de générer et stocker un grand nombre de nouvelles images, cet outil applique des transformations à la volée, juste avant que les images ne soient transmises au modèle. Cette classe permet de configurer différentes opérations de transformation et de normalisation à appliquer sur les images. Elle permet également de générer des lots d'images (batches) augmentées avec leurs étiquettes associées, facilitant ainsi l'apprentissage supervisé.

Une opération de normalisation est également appliquée afin de ramener les valeurs des pixels dans l'intervalle $[0, 1]$, ce qui est crucial pour assurer une convergence stable du modèle lors de l'entraînement. Cette normalisation se fait généralement en divisant les valeurs des pixels (initialement dans $[0, 255]$) par 255.

Les transformations incluent :

- Redimensionnement à (256×256) pixels.
- Rotation aléatoire (jusqu'à 20°),
- Décalages horizontaux et verticaux (jusqu'à 20% de l'image),
- Zoom, cisaillement (shear) et flip horizontal,
- Mise à l'échelle des pixels entre 0 et 1 ($\text{rescale}=1./255$).

Une séparation de 80/20 a été appliquée pour générer les jeux d'entraînement et de validation à partir du répertoire contenant les images classées. La figure III.8 illustre différentes opérations d'augmentation appliquées sur notre jeu de données d'apprentissage.

```

# === Préparation des générateurs ===
train_datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2, # 80% train, 20% val
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
test_datagen = ImageDataGenerator(rescale=1./255)

```

Figure III.8: Opérations d'augmentation des données.

III.5 Modèles de détection de la rétinopathie

La structure générale de nos modèles repose sur plusieurs couches de convolutions, qui extraient automatiquement les caractéristiques discriminantes à partir des images rétiniennes brutes. Chaque couche de convolution applique un ensemble de filtres afin de produire des cartes de caractéristiques (feature maps), représentant les éléments visuels essentiels tels que les bords, textures, micro-anévrismes ou exsudats. Ces couches sont suivies de couches de pooling, généralement du type max pooling, qui réduisent la dimensionnalité des cartes tout en conservant les informations les plus saillantes. Cette opération permet non seulement de diminuer la complexité computationnelle, mais aussi d'introduire une certaine invariance aux translations locales, ce qui est essentiel dans l'analyse d'images rétiniennes où les anomalies peuvent apparaître à différentes positions.

En fin de réseau, des couches entièrement connectées assurent la phase de classification. Chaque neurone dans ces couches est relié à l'ensemble des activations de la couche précédente, ce qui permet une prise de décision globale à partir des caractéristiques extraites. Ces couches permettent de prédire la présence ou non de rétinopathie diabétique (classification binaire).

Contrairement aux réseaux de neurones traditionnels entièrement connectés, les couches de convolution exploitent la structure spatiale des images, ce qui les rend particulièrement adaptées à la détection de motifs pathologiques dans les fonds d'œil.

La structure générale des CNN est montrée sur la figure III.9.

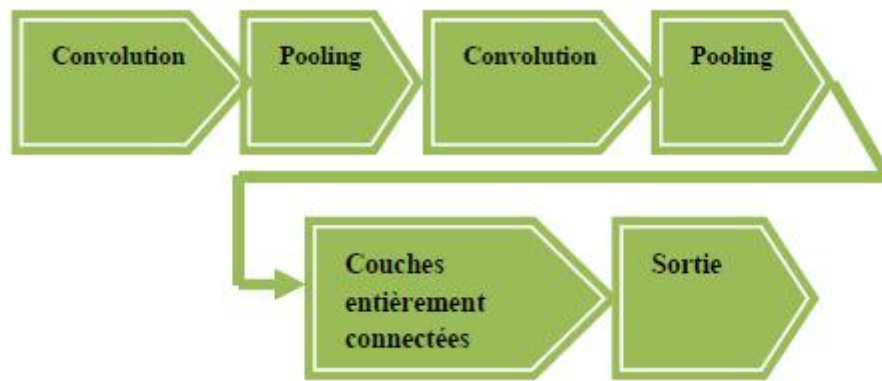


Figure III.9 : Structure de base d'un CNN.

III.5.1 Architecture de notre modèle 1 (modèle personnalisé)

Notre premier modèle se compose de trois blocs de convolution successifs. Le premier bloc commence par une couche de convolution avec 32 filtres de taille 3×3 et une fonction d'activation ReLU. Cette couche est immédiatement suivie d'une opération de max pooling de taille 2×2 , qui réduit la dimension spatiale des cartes de caractéristiques tout en conservant les informations essentielles.

Le deuxième bloc de convolution reprend la même structure, cette fois avec 64 filtres. Le troisième bloc, plus profond, comporte 128 filtres, toujours avec des noyaux 3×3 et une activation ReLU, suivi également d'un max pooling.

Après ces blocs convolutionnels, les cartes de caractéristiques sont aplaties via une couche Flatten, pour être transmises à une couche dense (fully connected) de 64 neurones avec une activation ReLU. Un dropout de 50 % est appliqué ensuite, pour réduire le risque de surapprentissage en désactivant aléatoirement la moitié des neurones lors de l'entraînement.

Enfin, la dernière couche est une couche de sortie dense avec un seul neurone, activé par une fonction sigmoïde. Cette sortie unique permet de produire une probabilité indiquant la présence ou l'absence de rétinopathie diabétique, ce qui en fait un modèle adapté à la classification binaire. La figure III.10 donne l'architecture détaillée de notre modèle.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 254, 254, 32)	896
max_pooling2d (MaxPooling2D)	(None, 127, 127, 32)	0
conv2d_1 (Conv2D)	(None, 125, 125, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 64)	0
conv2d_2 (Conv2D)	(None, 60, 60, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 30, 30, 128)	0
flatten (Flatten)	(None, 115200)	0
dense (Dense)	(None, 64)	7,372,864
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

Total params: 7,466,177 (28.48 MB)
Trainable params: 7,466,177 (28.48 MB)

Figure III.10: Architecture du modèle 1 (modèle personnalisé).

III.5.2 Modèles utilisant le transfer learning

Le transfert learning est une technique puissante du machine learning qui consiste à réutiliser un modèle préalablement entraîné sur une tâche source comme point de départ pour une tâche cible similaire. Cette approche est particulièrement avantageuse lorsque les données disponibles pour la nouvelle tâche sont limitées. En capitalisant sur les représentations déjà apprises par un modèle performant, on peut à la fois améliorer la précision et accélérer l'entraînement sur la nouvelle problématique.

Le transfert learning est largement utilisé dans des domaines tels que la vision par ordinateur et le traitement automatique du langage naturel, en raison de son efficacité démontrée dans des contextes de données restreintes. Il présente plusieurs avantages majeurs :

- Réduction du temps d'entraînement : le modèle commence avec des poids préalablement optimisés, ce qui permet une convergence plus rapide ;
- Amélioration des performances : les connaissances issues d'un apprentissage sur de grands ensembles de données génériques (comme ImageNet) sont transférées et réutilisées efficacement ;

- Moins de données nécessaires : il est possible d'obtenir de bonnes performances même avec des jeux de données de taille modeste.

Dans le cadre de notre travail, nous avons appliqué cette méthode à la détection de la rétinopathie diabétique, en utilisant quatre architectures de CNN pré-entraînés : VGG19, InceptionV3, EfficientNetB0 et ResNet152. Ces modèles, largement utilisés dans des tâches de classification d'images médicales ou naturelles, sont réputés pour leur robustesse, leur profondeur, ainsi que leur capacité à extraire des caractéristiques complexes.

Ces architectures ont été adaptées à notre tâche spécifique par remplacement de la tête de classification par de nouvelles couches denses, compatibles avec la classification binaire (présence ou absence de rétinopathie). Cette stratégie permet de tirer parti des couches convolutionnelles pré-entraînées, tout en adaptant la dernière partie du réseau aux spécificités de notre base de données (Messidor 2).

III.5.2.1 Transfer learning avec VGG19 (Modèle 2)

VGG19 [57] est un modèle de CNN développé par le Visual Geometry Group de l'Université d'Oxford. Ce réseau est réputé pour sa structure simple mais profonde, composée de 19 couches de convolution suivies de couches de pooling et de couches entièrement connectées.

Il se distingue notamment par l'utilisation systématique de filtres de taille 3×3 dans les couches convolutionnelles, ce qui permet de capturer efficacement les caractéristiques locales tout en maintenant une architecture cohérente. Les couches de classification en fin de réseau sont entièrement connectées, ce qui facilite l'adaptation du modèle à différentes tâches de classification. Bien que VGG19 soit un modèle relativement profond, il reste facile à comprendre et à mettre en œuvre, ce qui en fait un choix populaire dans de nombreuses applications de vision par ordinateur.

L'architecture adaptée de notre modèle 3, basé sur VGG19, est illustrée dans les figures III.11 et III.12.

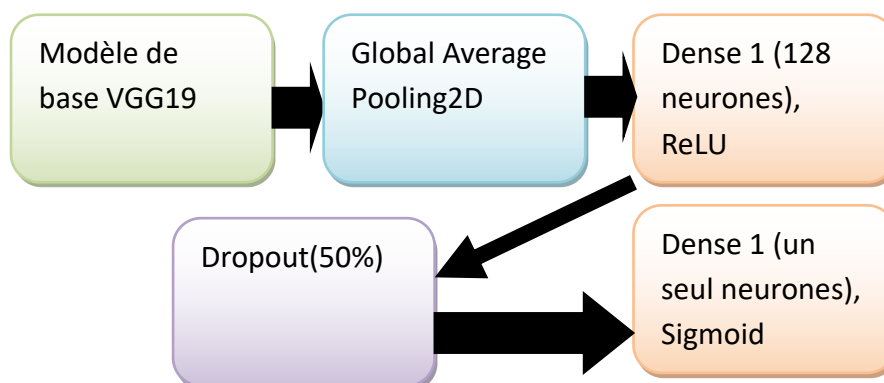


Figure III.11: Architecture du modèle de transfer learning avec VGG19.

block3_conv4 (Conv2D)	(None, 56, 56, 256)	590,080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 512)	0
dense_2 (Dense)	(None, 128)	65,664
dropout_1 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 1)	129
Total params: 20,090,177 (76.64 MB)		
Trainable params: 65,793 (257.00 KB)		
Non-trainable params: 20,024,384 (76.39 MB)		

Figure III.12: Différents composants et paramètres du modèle de transfer learning avec VGG19.

III.5.2.2 Transfer learning avec EfficientNet (Modèle 3)

EfficientNet [58] est une famille de modèles de réseaux neuronaux convolutifs conçus pour optimiser à la fois la précision et l'efficacité computationnelle. Ce modèle repose sur une approche de redimensionnement uniforme, qui ajuste simultanément la profondeur, la largeur et la résolution du réseau, ce qui permet une montée en échelle plus harmonieuse et plus efficace.

EfficientNet se distingue par l'utilisation de blocs MBConv (Mobile Inverted Bottleneck Convolution), qui combinent des couches de convolution optimisées pour réduire le nombre

de paramètres tout en maintenant des performances élevées. Grâce à cette architecture, EfficientNet parvient à surpasser des modèles classiques tels que VGG19 ou InceptionV3 en termes de précision, tout en consommant moins de mémoire et de puissance de calcul.

Dans notre cas, nous utilisons EfficientNet comme base pour le modèle 3, illustré dans la figure III.13.

(Activation)	(1152)		
block7a_se_squeeze (GlobalAveragePool...)	(None, 1152)	0	block7a_activati...
block7a_se_reshape (Reshape)	(None, 1, 1, 1152)	0	block7a_se_squee...
block7a_se_reduce (Conv2D)	(None, 1, 1, 48)	55,344	block7a_se_resha...
block7a_se_expand (Conv2D)	(None, 1, 1, 1152)	56,448	block7a_se_reduc...
block7a_se_excite (Multiply)	(None, 7, 7, 1152)	0	block7a_activati... block7a_se_expan...
block7a_project_co... (Conv2D)	(None, 7, 7, 320)	368,640	block7a_se_excit...
block7a_project_bn (BatchNormalizatio...)	(None, 7, 7, 320)	1,280	block7a_project_...
top_conv (Conv2D)	(None, 7, 7, 1280)	409,600	block7a_project_...
top_bn (BatchNormalizatio...)	(None, 7, 7, 1280)	5,120	top_conv[0][0]
top_activation (Activation)	(None, 7, 7, 1280)	0	top_bn[0][0]
global_average_poo... (GlobalAveragePool...)	(None, 1280)	0	top_activation[0...
dense (Dense)	(None, 128)	163,968	global_average_p...
dropout (Dropout)	(None, 128)	0	dense[0][0]
dense_1 (Dense)	(None, 1)	129	dropout[0][0]

Total params: 4,213,668 (16.07 MB)
Trainable params: 164,097 (641.00 KB)
Non-trainable params: 4,049,571 (15.45 MB)

Figure III.13: Différents composants et paramètres du modèle de transfer learning avec EfficientNet.

III.5.2.3 Transfert learning avec InceptionV3 (Modèle 4)

InceptionV3 [59] est un modèle de réseau de CNN par Google. Il fait partie de la famille des architectures Inception, connues pour leur capacité à réduire le coût computationnel tout en maintenant de bonnes performances. L'originalité d'InceptionV3 réside dans sa structure modulaire, composée de blocs Inception qui permettent de capturer des informations à différentes échelles à l'intérieur d'une même couche. Chaque bloc effectue des convolutions 1×1 , 3×3 et 5×5 en parallèle, suivies d'un regroupement des résultats.

Ce modèle se distingue également par l'utilisation de techniques avancées telles que le Factorization into smaller convolutions, la normalisation batch, et les convolutions asymétriques, qui contribuent à une plus grande efficacité en termes de calcul.

Dans notre travail, InceptionV3 est utilisé en tant que base pré-entraînée sur ImageNet. Les couches supérieures ont été remplacées par des couches personnalisées adaptées à la classification binaire de la rétinopathie diabétique. L'utilisation d'InceptionV3 nous permet de bénéficier de sa capacité à extraire des caractéristiques riches et hiérarchiques tout en réduisant le risque de surapprentissage. L'architecture du modèle 4 est illustrée dans la figure III.14.

batch_normalizatio_ (BatchNormalizatio_	{Nmmu, 8, 8, 384}	1,152	conv2d_87[0][0]
batch_normalizatio_ (BatchNormalizatio_	{Nmmu, 8, 8, 384}	1,152	conv2d_88[0][0]
batch_normalizatio_ (BatchNormalizatio_	{Nmmu, 8, 8, 384}	1,152	conv2d_91[0][0]
batch_normalizatio_ (BatchNormalizatio_	{Nmmu, 8, 8, 384}	1,152	conv2d_92[0][0]
conv2d_93 (Conv2D)	{Nmmu, 8, 8, 192}	393,216	average_pooling2_
batch_normalizatio_ (BatchNormalizatio_	{Nmmu, 8, 8, 320}	960	conv2d_85[0][0]
activation_87 (Activation)	{Nmmu, 8, 8, 384}	0	batch_normalizat_
activation_88 (Activation)	{Nmmu, 8, 8, 384}	0	batch_normalizat_
activation_91 (Activation)	{Nmmu, 8, 8, 384}	0	batch_normalizat_
activation_92 (Activation)	{Nmmu, 8, 8, 384}	0	batch_normalizat_
batch_normalizatio_ (BatchNormalizatio_	{Nmmu, 8, 8, 192}	576	conv2d_93[0][0]
activation_85 (Activation)	{Nmmu, 8, 8, 320}	0	batch_normalizat_
mixed9_1 (Concatenate)	{Nmmu, 8, 8, 768}	0	activation_87[0]_ activation_88[0]_
concatenate_1 (Concatenate)	{Nmmu, 8, 8, 768}	0	activation_91[0]_ activation_92[0]_
activation_93 (Activation)	{Nmmu, 8, 8, 192}	0	batch_normalizat_
mixed10 (Concatenate)	{Nmmu, 8, 8, 2048}	0	activation_85[0]_ mixed9_1[0][0]_ concatenate_1[0]_ activation_93[0]_
global_average_poo_ (GlobalAveragePool_	{Nmmu, 2048}	0	mixed10[0][0]
dense (Dense)	{Nmmu, 128}	262,272	global_average_p_
dropout (Dropout)	{Nmmu, 128}	0	dense[0][0]
dense_1 (Dense)	{Nmmu, 1}	129	dropout[0][0]

Total params: 22,065,185 (84.17 MB)
Trainable params: 262,401 (1.00 MB)
Non-trainable params: 21,802,784 (83.17 MB)

Figure III.13: Différents composants et paramètres du modèle de transfer learning avec InceptionV3.

III.5.2.4 Transfert learning avec ResNet152 (Modèle 5)

ResNet152 [60] est une architecture de réseau de neurones profonds développée par Microsoft Research. Elle appartient à la famille des Residual Networks (ResNet), qui ont introduit le concept de connexions résiduelles (ou raccourcis). Ces connexions permettent au gradient de circuler plus facilement lors de l'entraînement de réseaux très profonds, évitant ainsi les problèmes de dégradation de performance généralement observés au-delà d'un certain nombre de couches. ResNet152 se compose de 152 couches profondes, organisées en blocs résiduels, où chaque bloc apprend uniquement la différence (le résidu) entre l'entrée et la sortie attendue. Cela facilite l'entraînement de réseaux très profonds tout en maintenant une haute précision. Ce modèle s'est avéré particulièrement efficace pour des tâches de classification d'images complexes.

Dans notre application, ResNet152 est utilisé comme extracteur de caractéristiques en tirant parti des poids pré-entraînés sur ImageNet. Les couches finales ont été remplacées par des couches denses adaptées à notre tâche de classification binaire. Grâce à sa profondeur et à ses performances éprouvées, ResNet152 constitue une base solide pour l'identification automatique de la rétinopathie diabétique. L'architecture du modèle 5 est représentée dans la figure III.14.

conv5_block2_2_relu {Activation}	{None, 7, 7, 512}	8	conv5_block2_2_bn
conv5_block2_3_conv {Conv2D}	{None, 7, 7, 2048}	1,058,624	conv5_block2_2_relu
conv5_block2_out {Add}	{None, 7, 7, 2048}	8	conv5_block1_out_ conv5_block2_3_c
conv5_block3_preac_ {BatchNormalizatio...	{None, 7, 7, 2048}	8,192	conv5_block2_out
conv5_block3_preac_ {Activation}	{None, 7, 7, 2048}	8	conv5_block3_pre
conv5_block3_1_conv {Conv2D}	{None, 7, 7, 512}	1,048,576	conv5_block3_pre
conv5_block3_1_bn {BatchNormalizatio...	{None, 7, 7, 512}	2,048	conv5_block3_1_c
conv5_block3_1_relu {Activation}	{None, 7, 7, 512}	8	conv5_block3_1_b
conv5_block3_2_pad {ZeroPadding2D}	{None, 9, 9, 512}	8	conv5_block3_1_r
conv5_block3_2_conv {Conv2D}	{None, 7, 7, 512}	2,359,296	conv5_block3_2_p
conv5_block3_2_bn {BatchNormalizatio...	{None, 7, 7, 512}	2,048	conv5_block3_2_c
conv5_block3_2_relu {Activation}	{None, 7, 7, 512}	8	conv5_block3_2_b
conv5_block3_3_conv {Conv2D}	{None, 7, 7, 2048}	1,058,624	conv5_block3_2_r
conv5_block3_out {Add}	{None, 7, 7, 2048}	8	conv5_block2_out_ conv5_block3_3_c
post_bn {BatchNormalizatio...	{None, 7, 7, 2048}	8,192	conv5_block3_out
post_relu {Activation}	{None, 7, 7, 2048}	8	post_bn[0][0]
global_average_pool_ {GlobalAveragePool...	{None, 2048}	8	post_relu[0][0]
dense_2 {Dense}	{None, 128}	262,272	global_average_p
dropout_1 {Dropout}	{None, 128}	8	dense_2[0][0]
dense_3 {Dense}	{None, 1}	129	dropout_1[0][0]

Total params: 58,594,040 (223.52 MB)
 Trainable params: 262,401 (1.00 MB)
 Non-trainable params: 58,331,648 (222.52 MB)

Figure III.14: Différents composants et paramètres du modèle de transfer learning avec ResNet152.

III.6 Résultats obtenus avec nos modèles

III.6.1 Résultats du modèle 1 (modèle personnalisé)

Cette section présente les résultats obtenus par notre modèle 1. Ces résultats sont exprimés en termes de précision (accuracy) et de perte (loss).

Le modèle est compilé avec l'optimiseur Adam, reconnu pour sa rapidité de convergence, avec un taux d'apprentissage de 0.001. La fonction de perte utilisée est la binary crossentropy, adaptée à une tâche de classification binaire. La métrique d'évaluation choisie est la précision (accuracy).

Pour optimiser l'entraînement, nous avons utilisé deux callbacks :

- EarlyStopping : interrompt l'entraînement si la performance sur l'ensemble de validation cesse de s'améliorer pendant 10 époques consécutives.
- ModelCheckpoint : sauvegarde automatiquement le modèle ayant obtenu la meilleure précision sur l'ensemble de validation.

L'entraînement est effectué sur 20 époques, avec des batchs de 32 images. La taille des images est fixée à 256×256 pixels.

Les valeurs d'accuracy et de perte du modèle, pour les phases d'entraînement et de validation, sont illustrées dans la figure III.15.

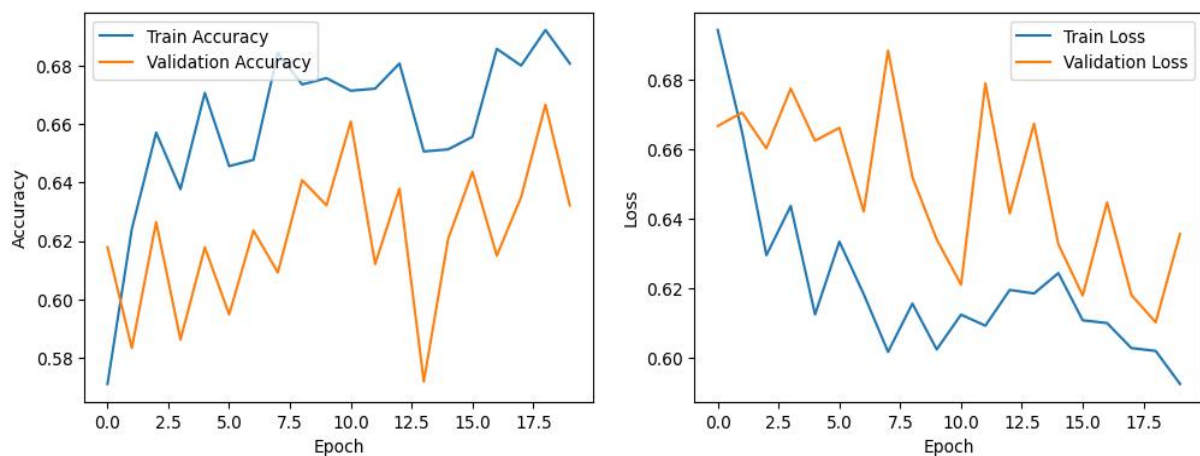


Figure III.15: Évolution de l'accuracy et de la perte pendant l'entraînement et la validation du modèle 1.

Après l'entraînement du modèle 1 et la sauvegarde de ses poids finaux, nous l'avons évalué sur les images du jeu de test. Le modèle a atteint une valeur d'accuracy de 0,6777 (68 %) et une perte de 0,6006.

III.6.2 Résultats obtenus avec nos modèles de transfer learning

Dans cette section, nous présentons les performances obtenues par les différents modèles utilisant la technique de transfert learning. Ces résultats sont évalués en termes d'accuracy et de perte sur les jeux d'entraînement, de validation et de test.

Le modèle VGG19 a été entraîné sur notre base de données prétraitée. La figure III.16 montre l'évolution des valeurs d'accuracy et de perte durant les phases d'entraînement et de validation.

Lors de l'évaluation finale sur le jeu de test, le modèle a atteint une précision de 0,7181 (soit environ 72 %) et une perte de 0,5636.

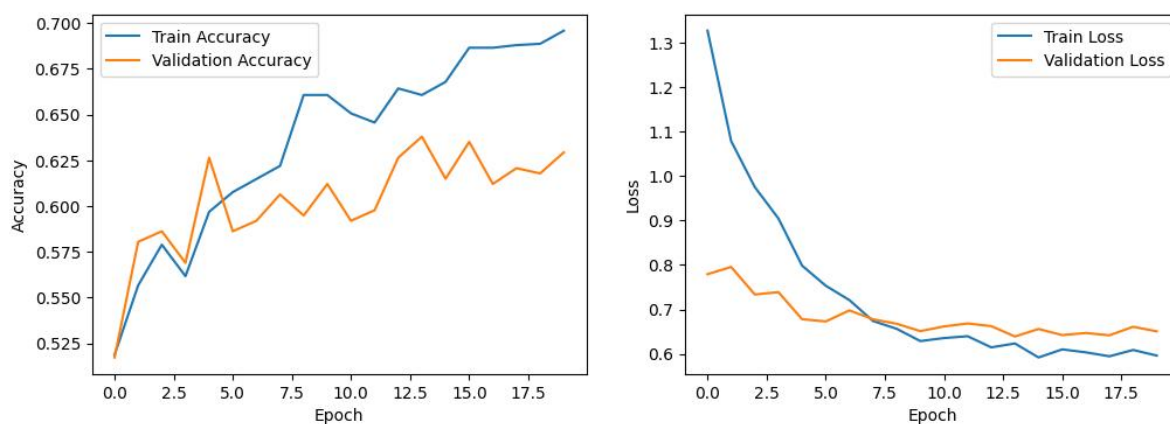


Figure III.16: Valeurs d'accuracy et de perte au cours de l'entraînement et de la validation pour le modèle 2 (transfer learning avec VGG19).

Le modèle EfficientNet a également été entraîné avec les mêmes paramètres d'optimisation. La figure III.17 illustre les performances obtenues.

Après l'évaluation sur les données de test, le modèle a atteint une précision (accuracy) de 0,7532 (environ 75 %) avec une perte de 0,5708.

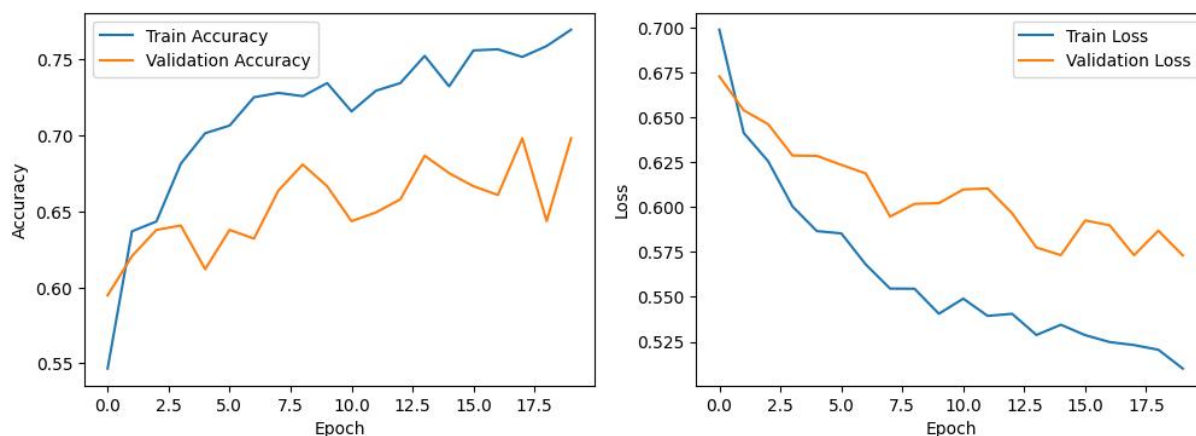


Figure III.17: Valeurs d'accuracy et de perte au cours de l'entraînement et de la validation pour le modèle 3 (transfer learning avec EfficientNet).

Le modèle InceptionV3 a présenté des performances plus modestes, comme le montre la figure III.18.

Après l'entraînement, l'évaluation sur le jeu de test a révélé une précision de 0,5172 (environ 52 %) et une perte de 0,6678.

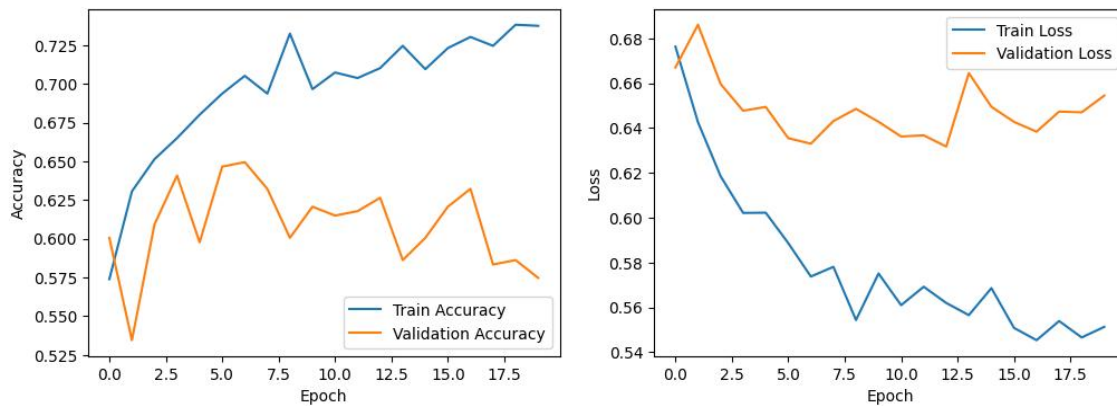


Figure III.18: Valeurs d'accuracy et de perte au cours de l'entraînement et de validation pour le modèle 4 (transfer learning avec InceptionV3).

Enfin, le modèle ResNet152 a montré des performances modérées. La figure III.19 présente les courbes d'accuracy et de perte.

Lors du test final, ce modèle a obtenu une précision de 0,6923 (environ 69 %) et une perte de 0,4823.

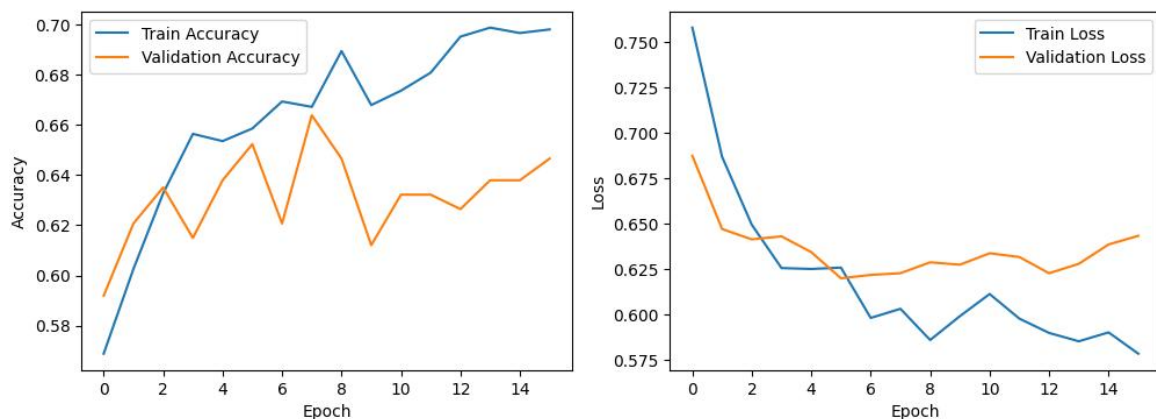


Figure III.19: Valeurs d'accuracy et de perte dans les phases d'entraînement et de validation pour le modèle 5 (transfer learning avec ResNet152).

III.7 Comparaison des différents résultats

Après l'évaluation de tous les modèles développés, nous avons effectué une analyse comparative de leurs performances. Les résultats obtenus montrent que les modèles basés sur le transfert learning ont apporté une amélioration notable par rapport à l'entraînement classique sur un modèle personnalisé. Malgré des scores de précision parfois modestes (variant entre 0,60 et 0,75), plusieurs facteurs liés aux données et au problème traité justifient ces performances et en soulignent la valeur.

Tout d'abord, il est important de rappeler que le dataset utilisé Messidor-2 présente plusieurs défis : un nombre d'images relativement limité, une forte variabilité inter-image, ainsi que des caractéristiques visuelles complexes et fines propres à la rétinopathie diabétique. Ces éléments rendent la tâche de classification difficile, même pour des architectures avancées. De nombreux travaux antérieurs utilisant ce même dataset rapportent des précisions similaires, souvent comprises entre 60 % et 80 %, ce qui confirme la difficulté intrinsèque de cette tâche.

Malgré cela, notre approche présente plusieurs points forts :

- L'intégration de quatre modèles de transfert learning (VGG19, EfficientNet, InceptionV3 et ResNet152), chacun entraîné et comparé sur des bases identiques, permet une analyse claire et approfondie.
- L'application de techniques d'augmentation de données a permis de simuler une base plus variée et a probablement limité les effets du surapprentissage.
- Le modèle personnalisé que nous avons conçu a montré une performance compétitive, illustrant que des architectures simples mais bien structurées peuvent rivaliser avec des réseaux pré-entraînés dans certains contextes.

Les résultats les plus prometteurs ont été obtenus avec EfficientNet et ResNet152, confirmant leur capacité à équilibrer profondeur, efficacité et précision. Néanmoins, les performances globales peuvent encore être améliorées. Cela ouvre des perspectives pour des améliorations futures, notamment :

- L'exploitation des dataset plus volumineux et diversifiés.
- L'utilisation de techniques d'optimisation et de régularisation récentes, ou l'adoption de modèles hybrides combinant CNN et les modèles d'attention.
- L'intégration de techniques de prétraitement plus efficaces pour renforcer les contrastes ou isoler les zones pertinentes.

III.8 Conclusion

Dans ce chapitre, nous avons présenté une méthode pour détecter la rétinopathie diabétique en utilisant les CNN. Pour cela, cinq modèles différents ont été construits et testés, et leurs résultats ont été comparés en fonction de l'accuracy et de la perte.

Les résultats obtenus montrent que plusieurs éléments influencent la performance des modèles, comme la taille du dataset, la profondeur du réseau, et le type d'architecture utilisé. On a vu aussi que certains modèles donnent de meilleurs résultats quand ils sont mieux adaptés aux images à analyser.

En plus, le transfert learning a montré son utilité. Il permet aux modèles de commencer avec des connaissances déjà apprises, ce qui les aide à apprendre plus vite et à obtenir de bons résultats, même avec un dataset limité.

Conclusion générale

La rétinopathie diabétique représente l'une des principales causes de cécité évitable dans le monde, affectant de plus en plus de patients diabétiques. Le diagnostic repose encore largement sur l'expertise humaine, qui peut être limitée par la disponibilité des spécialistes et la complexité des images médicales. Face à ces limites, ce mémoire a exploré l'apport des techniques d'intelligence artificielle, en particulier des réseaux de neurones convolutifs (CNN), pour automatiser la détection de cette pathologie.

Nous avons proposé et évalué cinq modèles, dont un modèle personnalisé et quatre basés sur le transfert learning (VGG19, EfficientNet, InceptionV3, ResNet152). Malgré des résultats parfois modestes, nos expérimentations ont montré que même avec un dataset limité, l'utilisation de modèles pré-entraînés permet d'obtenir des performances intéressantes, confirmant l'intérêt du transfert learning. Le modèle personnalisé a offert de bonnes performances, mettant en évidence l'importance du choix d'architecture et du réglage des hyperparamètres. EfficientNet s'est également distingué par un bon compromis entre précision et efficacité.

Ces résultats soulignent plusieurs points forts de notre travail, notamment la diversité des architectures testées, l'utilisation de techniques d'augmentation de données pour pallier le manque de données, et l'intégration de bonnes méthodes d'entraînement comme l'arrêt anticipé (early stopping) ou la sauvegarde du meilleur modèle. Toutefois, certains défis restent à relever : la qualité et la quantité des données utilisées ont limité la performance globale des

modèles, et l'évaluation sur un dataset plus large serait nécessaire pour mieux valider leur robustesse.

Plusieurs pistes peuvent être envisagées pour améliorer et approfondir ce travail :

- ✓ Collecte de données plus riches et variées, incluant des images de différentes sources, de meilleure qualité et annotées de manière plus précise.
- ✓ Amélioration des architectures en intégrant, des modèles plus récents comme les vision transformers (ViT) ou en combinant des données d'imagerie avec des données cliniques (âge, durée du diabète, etc.).
- ✓ Utilisation de l'apprentissage fédéré permettant d'entraîner les modèles sur plusieurs sources de données (téléphones, hôpitaux, ou serveurs d'un centre médical) sans les centraliser pour garder les données privées tout en améliorant la performance des modèles.
- ✓ Déploiement des modèles dans des contextes réels, notamment en milieu hospitalier ou dans des centres de dépistage, afin de tester l'intégration pratique des systèmes développés.
- ✓ Exploration de la détection multi-pathologies, pour permettre à un même système de détecter plusieurs maladies oculaires à partir d'une seule image.

Références bibliographiques

- [1] M. moumouni, A. karim. Poursuite laser de cibles multiples par micromiroirs, thèse de doctorat ,université saad dahlaeb-blida, 2018/2019.
- [2] J.W. Harbour, T.G. Murray, J.L. Davis. Rétine et vitré, article, France, société française d'aphtalmologie, 2014.
- [3] B. Wang, J. Shen. La Rétinopathie Diabétique Et Les Maladies Des Yeux, article scientifique , La Fédération Française des Diabétiques.
- [4] P. Fischer. Encyclopédie de la vue, livre , SNOF, 10 rue Schweighaeuser.
- [5] G. Zago. Dépistage de la rétinopathie diabétique, la Haute Autorité de Santé , France, Décembre 2010.
- [6] M. Martini. Prise en charge de la Rétinopathie Diabétique par l'orthoptiste, 2016.
- [7] S. Mehta, Rétinopathie diabétique, Le manuel MSD, 2020.
- [8] D. Bourdet. L'examen du fond d'œil, 2019.
- [9] femmes santé, «Fond d'oeil», 09/03/2020.

- [10] Collège des Ophtalmologistes : Rétinopathie diabétique, France, 2013.
- [11] Studio Cassette: Réalités ophtamologique, 2016.
- [12] A. scalante, P. Andrés. Une approche métrique pour la segmentation d'images, Thèse de doctorat, Université Paris-Dauphine, 2005.
- [13] A. hakim, B. bilal. Système d'aide à la détection de maladies sur image médicale basé sur les réseaux de neurones , mémoire d'ingénieur d'état en informatique , 2010.
- [14] B. Mouloud. Contribution à la restauration et fermeture de contours par une approche neuronale, Mémoire de magister, ESI 2001.
- [15] F. Hadjila, R. Bouabdellah. Reconnaissance des visages par les réseaux de neurones, Mémoire de PFE, Université de Tlemcen, 2003.
- [16] H. Naciri, N. Chaoui. Conception et Réalisation d'un système automatique d'identification des empreintes digitales, Mémoire de PFE, Université de Tlemcen, 2003.
- [17] K. Tabari, S. Tagma. Compression d'images animées à très faible débit par la géométrie des fractales, Projet de fin d'études, USTBH, 1996.
- [18] M. Al-Mahfuz, M. AliMoni, P. Lio, M. Sheikh, S. Islam, S. Berkovsky, M. Khushi, J. Quinn. Deep convolutional neural networks-based ECG beats classification to diagnose cardiovascular conditions. Biomedical engineering letters, vol. 11, no. 2, pp. 147–162, 2021.
- [19] A. Gupta. Current research opportunities of image processing and computer vision. Computer Science. vol 20, no. 4, 2019.
- [20] Python Definition. Techterms.com , [en ligne]. Disponible sur : <https://techterms.com/definition/python> (consulté en juin 2025).
- [21] Google Colab . Techterms.com , [en ligne]. Disponible sur : <https://research.google.com/colaboratory/faq.html> (consulté en juin 2025).
- [22] U. Acharya, S. LihOhh , Y. Hagiwara, J. Hong Tan, M. Adam, A. Gertych , R. Tan. A deep convolutional neural network model to classify heart beats. Computers in biology and medicine, vol. 89, pp. 389–396, 2017.
- [23] T. Rachida. Segmentation des images médicales par ondelettes. Mémoire de magister en imagerie vision artificielle et robotique médicale, Université des sciences et de la technologie d'Oran, 2011.
- [24] Y. Wu, F. Yang, Y. Liu, X. Zha, S. Yuan. A Comparison of 1-D and 2-D Deep Convolutional Neural Networks in ECG Classification. Conference on Computer Vision and Pattern Recognition, pp.324-327, 2018.
- [25] S. Gehrke, E. McClure. Comment lire un-ECG: 9 étapes (avec images) - wikiHow. [en ligne]. Disponible sur : <https://fr.wikihow.com/lire-un-ECG> (consulté en février 2025).

- [26] Deep learning : c'est quoi ?. [en ligne]. Disponible sur : www.nordvpn.com/fr/blog/apprentissage-profond/ (consulté en février 2025).
- [27] H. Djallel. Intrusion Detection System for Cyber Security, Thèse de Master, Université 8 Mai 1945, Guelma, 2020.
- [28] Neurones artificiels et neurones biologiques. [en ligne]. Disponible sur : <https://www.tree-learning.fr/plateforme-lms-elearning/neurones-artificiels/> (consulté en mars 2025).
- [29] Réseaux de neurones. [en ligne]. Disponible sur <https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-app-rn.pdf> (consulté en mai 2025).
- [30] Présentation de la mise en place d'une analyse par cartes auto-organisatrices. [en ligne]. Disponible sur : <https://hal.science/hal-04075400v1/document> (consulté en mars 2025).
- [31] Réseau Hopfield. [en ligne]. Disponible sur : <https://prezi.com/p/sdw4mtthyvrt/reseau-hopfield/> (consulté en mars 2025).
- [32] Les Différents Modèles de Machine Learning. [en ligne]. Disponible sur : <https://www.clicdata.com/fr/blog/modeles-machine-learning/> (consulté en avril 2025).
- [33] Découvrez le neurone formel. [en ligne]. Disponible sur : <https://openclassrooms.com/fr/courses/5801891-initiez-vous-au-deep-learning/5801898-decouvrez-le-neurone-formel> (consulté en avril 2025).
- [34] Cours - Réseaux de neurones multicouches. [en ligne]. Disponible sur : <https://cedric.cnam.fr/vertigo/Cours/ml/coursReseauxNeuronesMulticouches.html> (consulté en mai 2025).
- [35] O. Chapelle, B. Schölkopf, A. Zien. Semi-Supervised Learning
- [36] R. A. Ivan, N. da Silva. Artificial Neural Networks A Practical Course. Springer International Publishing Switzerland, 2017.
- [37] Les architectures CNN performantes pour la classification deep learning en imagerie médicale en 2021. [en ligne]. Disponible sur : <https://www.imaios.com/fr/ressources/blog/classification-en-imagerie-medecale-les-architectures-cnn-les-plus-performantes-en-2021> (consulté en juin 2025).
- [38] A. Khan, A. Sohail, U. Zahoor, A. S. Qureshi. A survey of the recent architectures of deep convolutional neural networks. Artificial intelligence review, vol. 53, no. 8, pp. 5455-5516, 2020.
- [39] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, Y. LeCun. What is the best multi-stage architecture for object recognition?. In 2009 IEEE 12th international conference on computer vision, pp. 2146-2153, Septembre 2009.
- [40] I. Goodfellow, Y. Bengio, A. Courville. 6.2.2.3 Softmax Units for Multinoulli Output Distributions. Deep Learning, MIT Press, pp. 180–184, 2016.

- [41] A. Esteva, K. Chou, S. Yeung, N. Naik, A. Madani, A. Mottaghi, Y. Liu, E. Topol, J. Dean, R. Socher. Deep learning-enabled medical computer vision. *Deep learning-enabled medical computer vision*. npj Digit. Med. 4, 5 (2021). <https://doi.org/10.1038/s41746-020-00376-2>.
- [42] Use Early Stopping to Halt the Training of Neural Networks At the Right Time. [en ligne]. Disponible sur : <https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/> (consulté en mai 2025).
- [43] Data augmentation : Qu'est-ce que c'est ? À quoi ça sert ? [en ligne]. Disponible sur : <https://datascientest.com/data-augmentation-tout-savoir> (consulté en mai 2025).
- [44] Découvrez le Transfer Learning : quand l'IA ne part pas de zéro. [en ligne]. Disponible sur : <https://www.innovatiana.com/post/transfer-learning-101> (consulté en avril 2025).
- [45] S. Nagendram, , A. Singh, G. Harish Babu, R. Joshi, S. D. Pande, S. K. H. Ahammad, D. Dhabliya, A. Bisht. Stochastic gradient descent optimisation for convolutional neural network for medical image segmentation. *Open life sciences*, vol. 18, no.1, 20220665, 2023. <https://doi.org/10.1515/biol-2022-0665>
- [46] Understanding the RMSprop Optimizer: A Guide. [en ligne]. Disponible sur : <https://builtin.com/articles/rmsprop-optimizer> (consulté en avril 2025).
- [47] Adaptive Moment Estimation : comprendre Adam et l'utiliser correctement. [en ligne]. Disponible sur : <https://konfuzio.com/fr/estimation-adaptative-du-moment/> (consulté en avril 2025).
- [48] Python. [en ligne]. Disponible sur : <https://www.python.org/> (consulté en juin 2025).
- [49] Python. [en ligne]. Disponible sur : <https://techterms.com/definition/python> (consulté en mai 2025).
- [50] Multiclass Classification using Keras and TensorFlow on Food-101 Dataset. [en ligne]. Disponible sur : https://colab.research.google.com/github/theimgclist/examples/blob/MultiClassTF2.0/community/en/multi_class_classification/food_classifier.ipynb (consulté en mars 2025).
- [51] Conda. [en ligne]. Disponible sur : <http://conda.pydata.org/docs/> (consulté en juin 2025).
- [52] Outils de développement logiciel. [en ligne]. Disponible sur : <https://outils-developpement-logiciel.sodevlog.com/2020/10/cest-quoi-loutil-jupyter-notebook.html> (consulté en mars 2025).
- [53] Keras. [en ligne]. Disponible sur : <https://www.tensorflow.org/guide/keras?hl=fr> (consulté en juin 2025).
- [54] E. Decencière, D. Etienne, Z. Xiwei. Feedback on a publicly distributed image database: the Messidor database. *Image Anal Stereol*, vol. 33, no. 3, pp. 231-234, 2014. doi:10.5566/ias.1155

- [55] J. Krause et al. Grader variability and the importance of reference standards for evaluating machine learning models for diabetic retinopathy. *Ophthalmology*, 2018. doi:10.1016/j.ophtha.2018.01.034
- [56] V. Gulshan et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA* 316, pp. 2402–2410, 2016.
- [57] K. Simonyan, , A. Zisserman,. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. <https://arxiv.org/abs/1409.1556>
- [58] M. Tan, Q. V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 6105–6114, 2019. <https://arxiv.org/abs/1905.11946>
- [59] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [60] K. He, X. Zhang, S. Ren, J. Sun,. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.