

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد حطاب البلدية
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

En Électronique
Spécialité Electronique des Systèmes Embarqués

Présenté par

Ouzeri Abderrezak

&

Kettab Imad Eddine

Deep Learning pour la détection de la rétinopathie

Proposé par : Bougherira Hamida

Année Universitaire 2017-2018

Remerciements

Nous remercions tout d'abord ALLAH le tout puissant qui nous a donné la force de mener à bon terme ce modeste travail.

Nous tenons à remercier notre promotrice et enseignante Madame Bougherira Hamida pour son soutien et ses précieux conseils.

Notre reconnaissances s'adresse également à toutes personnes qui sont chères et ceux qui ont contribué à l'aboutissement de ce projet de près ou de loin.

ملخص : قمنا بتطوير خوارزمية ذكاء الاصطناعي على وجه التحديد التعلم العميق لتحديد بعض اعتلالات الشبكية باستخدام تصوير قاع العين من قاعدة بيانات Kaggle. استخدمنا بيئة تطوير Python و المكتبتين Keras و Tensorflow. حصلنا على معدل استكشاف 97 % للتعلم و 73 % للمصادقة.

كلمات المفاتيح : اعتلال الشبكية ؛ الذكاء الاصطناعي ؛ التعلم العميق ؛ Keras.

Résumé : Nous avons développé un algorithme d'intelligence artificielle plus précisément l'apprentissage profond afin d'identifier certaines rétinopathies à l'aide de photographie du fond d'œil de la base de données Kaggle. Nous avons utilisé comme environnement de développement Python et les bibliothèques Keras et Tensorflow. Nous avons obtenu un taux de détection de 97 % à l'apprentissage et 73 % à la validation.

Mots clés : rétinopathie ; intelligence artificielle ; apprentissage profond ; Keras.

Abstract : We developed an artificial intelligence algorithm specifically deep learning to identify certain retinopathies using fundus photography from the Kaggle database. We used Python as development environment and the Keras and Tensorflow libraries. We obtained a detection rate of 97% for learning and 73% for validation.

Keywords : Retinopathy ; Artificial Intelligence ; Deep Learning ; Keras.

Introduction Générale

L'imagerie médicale est certainement l'un des domaines de la médecine qui a le plus progressé ces vingt dernières années. Ces récentes découvertes permettent non seulement un meilleur diagnostic mais offrent aussi de nouveaux espoirs de traitement pour de nombreuses maladies. Cancer, épilepsie... l'identification précise de la lésion facilite déjà le recours à la chirurgie, seule solution thérapeutique pour certains malades. De telles techniques permettent également de mieux comprendre le fonctionnement de certains organes encore mystérieux, comme le cerveau. La qualité de l'acquisition d'image et l'interprétation détermine la précision du diagnostic. En effet, l'apparition d'ordinateurs de plus en plus performants, ont un impact énorme sur l'acquisition d'images médicales. Elles remplissent des fonctions à plusieurs volets, tels que, le contrôle du matériel d'imagerie, d'effectuer la transmission, le stockage, la visualisation, l'extraction automatique d'information, la reconstruction, le post-traitement des données d'image...etc. En revanche, le rôle de l'informatique dans l'interprétation des images médicales a jusqu'ici été limité. L'interprétation reste un domaine presque exclusivement humain. Mais l'émergence de l'intelligence artificielle dans le secteur médical bouleverse le concept.

Apparue dans divers domaines l'IA, intelligence artificielle, a fait ses preuves notamment dans la finance, le transport, l'éducation, l'informatique, le biomédical...etc. Dans ce dernier domaine, en offrant de mieux détecter et déceler plus tôt des pathologies, elle permettra de prévenir et de s'attaquer plus rapidement aux pathologies et de gagner du temps dans leur traitement... voire d'intervenir avant même l'apparition d'une maladie. C'est dans ce contexte que notre projet intervient, l'équipe Biorétine du laboratoire LATSI à l'université de Blida 1 vise à réaliser un photocoagulateur laser embarqué, muni d'un détecteur de pathologies rétinienne embarqué. Le détecteur est un réseau de neurone convolutionnel (CNN : convolutional neural network), qui reçoit en entrée une angiographie de la rétine, et détecte les zones affectées. Le CNN est entraîné par un algorithme de Machine Learning, le Deep Learning.

Notre travail consiste à effectuer le Deep Learning sur une base de données d'images rétinienne affectées par des pathologies, et saines.

Le CNN obtenu permettra de détecter les zones d'intérêt, sur une angiographie rétinienne, concernées par le traitement laser.

Cette intelligence artificielle prend forme de logiciels ou d'algorithmes, parmi les sous-ensembles de l'IA les plus utilisés on peut citer : l'apprentissage automatique ou l'apprentissage machine (Machine Learning en anglais), et l'apprentissage profond (Deep Learning).

Le Machine Learning comme son nom l'indique, est un algorithme basé sur le concept de donner à l'ordinateur (Machine en général) la capacité d'apprendre. Par la suite différentes techniques de Machine Learning ont été développées pour créer des algorithmes capables d'apprendre et de s'améliorer de manière autonome.

Introduction Générale

Parmi ces techniques, on compte les réseaux de neurones artificiels. C'est sur ces algorithmes que reposent le Deep Learning, mais aussi des technologies comme la reconnaissance d'images. Les réseaux de neurones artificiels sont inspirés par les neurones du cerveau humain. Ils sont constitués de plusieurs neurones artificiels connectés entre eux. Plus le nombre de neurones est élevé, plus le réseau est profond. Le Deep Learning a été développé en 1998, mais a dû être abandonné à cause de la quantité de traitement requise.

Ce n'est qu'à l'avènement des processeurs multi-cœurs puissants et de GPU, que son exécution a été possible, et ses résultats et son efficacité démontrés ; cela a permis de relancer les algorithmes neuromorphique et l'intelligence artificielle.

Ce qui nous mène à notre projet, qui est de fournir un programme de Deep Learning capable d'identifier plusieurs rétinopathies en se basant sur une base de données comportant des images de celle-ci.

Pour décrire notre projet nous avons organisé notre mémoire comme suit :

Chapitre I

Dans le premier chapitre on présentera des généralités de notre projet, à savoir l'aspect médical de la rétinopathie, sa définition élargie, ses causes, ses symptômes et ses traitements.

Chapitre II

Dans le second chapitre on s'approfondira dans l'Intelligence Artificielle notamment dans le Machine Learning et le Deep Learning.

Chapitre III

Le dernier chapitre est consacré au Deep Learning sous Python, Outils, Bibliothèques... Nous présentons l'implémentation et la mise en œuvre de notre code, et les résultats de son exécution.

Table des matières

Introduction Générale	1
Chapitre I : Aspect Médical et Généralités de l'IA.....	4
1. Introduction.....	5
2.Description Générale de l'Œil	5
2.1. Anatomie de l'œil	6
2.2. La rétine	7
3.Les rétinopathies	8
3.1.Les Rétinopathies les plus courantes et leurs Causes	8
3.1.1. La Rétinopathie Diabétique (RD).....	8
3.1.2. La rétinopathie hypertensive.....	8
3.1.3. Autre Rétinopathies	9
3.2. Symptômes et Complications	9
3.3. Diagnostic.....	10
3.4. Traitement et Prévention	12
3.4.1. La Photo-coagulation au laser.....	13
4. L'Intelligence Artificielle.....	14
4.1. Introduction	14
4.2. Quelques exemples d'usage d'IA	14
4.3. L'Apprentissage	15
4.3.1. Machine Learning	15
4.3.2. Deep Learning.....	16
5. Conclusion.....	17
Chapitre II :L'Intelligence Artificielle.....	19
1. Introduction	19
2. Définition Intelligence Artificielle.....	19
2.1. Raison de l'évolution de l'Intelligence Artificielle.....	19
2.2. L'Apprentissage	21
3. Machine Learning	22
3.1. Types d'apprentissage	22
3.1.1. Apprentissage supervisé.....	22
3.1.2. Apprentissage non supervisé.....	23
3.1.3. Apprentissage par renforcement.....	23
3.2. Méthode des k plus proches voisins.....	24
3.3. Arbre de décision	25
3.4. Méthode des K-moyennes	26
4. Deep Learning	27
4.1. Réseaux de Neurones Convolutifs	28
4.1.1. Principe et Architecture des CNN.....	28
4.1.2. Les blocs de construction	29
4.1.3. Choix des hyper-paramètres.....	31
4.1.4. Avantages de CNN.....	32
5. Conclusion.....	33

Table des matières

Chapitre III : Implémentation et Résultats

1. Deep Learning et Python.....	35
2. Python sur Windows avec Pycharm.....	36
3. La bibliothèque Keras	36
4. La bibliothèque Tensorflow	37
5. Préparation des données	38
6. Le réseau de neurone à convolution	39
7. Exécution du code Python.....	43
8. Résultat final	48
ConclusionGénéral.....	49
Annexes.....	50
Bibliographie.....	53

Liste des Acronyms et Abbreviations

IA: Intelligence Artificielle.

RD : Rétinopathie Diabétique.

CPU : Central Processing Unit (unités centrales de traitement).

GPU : Graphics Processing Unit (unités de traitement graphique).

ML : Machine Learning.

DL : Deep Learnig.

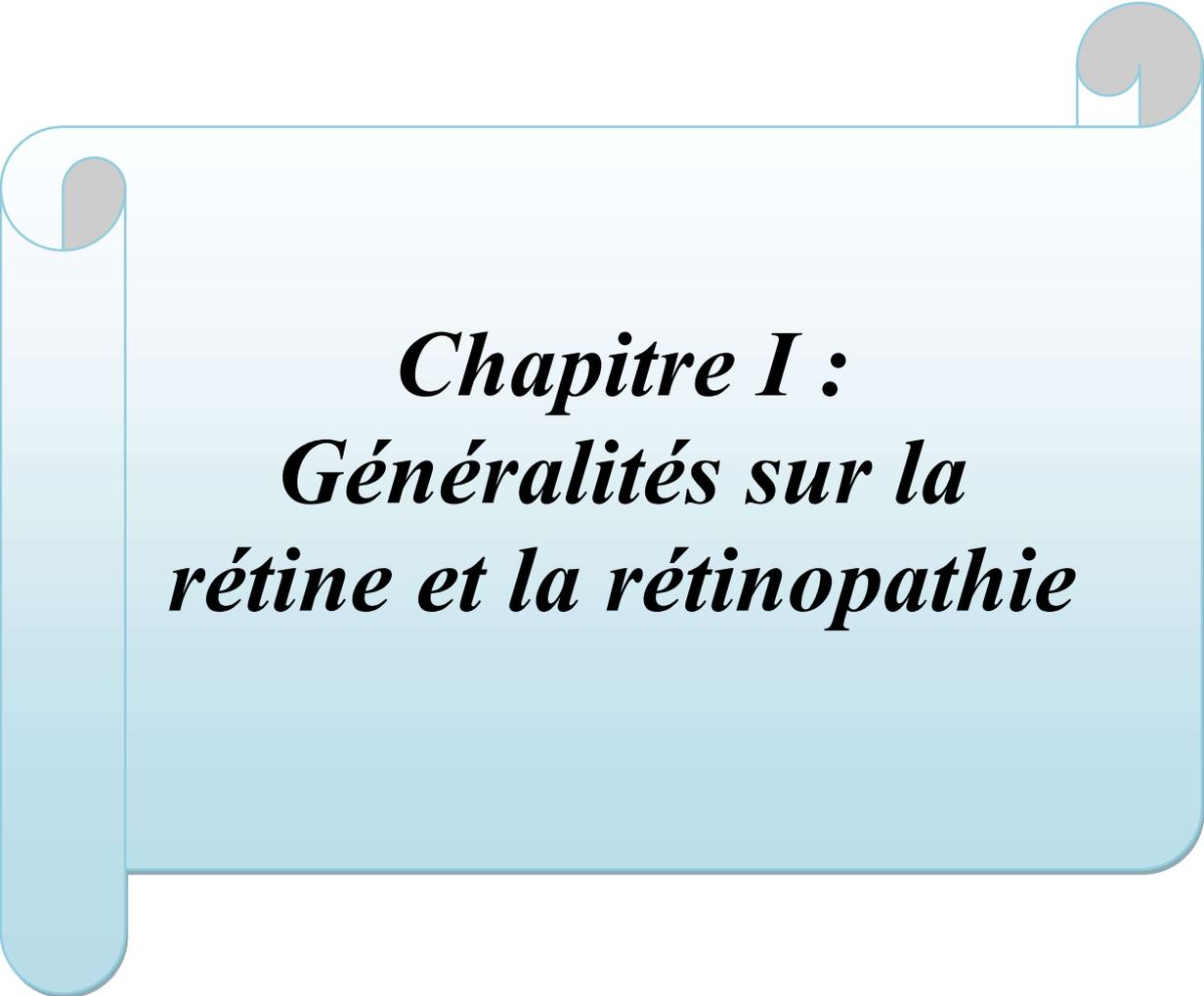
CNN : Convolutional Neural Network (réseaux de neurones convolutifs).

CONV : Convolutional.

POOL : Pooling.

ReLU : Rectified Linear Units.

FC : Fully Connected.



Chapitre I :
Généralités sur la
rétine et la rétinopathie

1.1.Introduction :

Le but de notre projet étant de développer un algorithme de deep learning, pour la détection de rétinopathie à partir d'une image de la rétine (angiographie), le présent chapitre montre une description de la rétine et de ses rétinopathies.

1.2.Description Générale de l'Œil:

Pour bien comprendre ce qu'est la rétine et où elle se situe dans l'œil une description de l'œil humain s'impose.

L'œil est un organe sensoriel du corps humain qui reçoit 80% des informations extérieures. Sa fonction est de transformer l'information lumineuse en influx nerveux transmis au cerveau. Il est de faible volume (6.5 cm³), il pèse 7 grammes, de forme d'une sphère d'environ 25 mm de diamètre, complétée vers l'avant par une autre sphère de 8 mm de rayon (la cornée) [1].

La figure I-1 illustre les composants de l'œil :

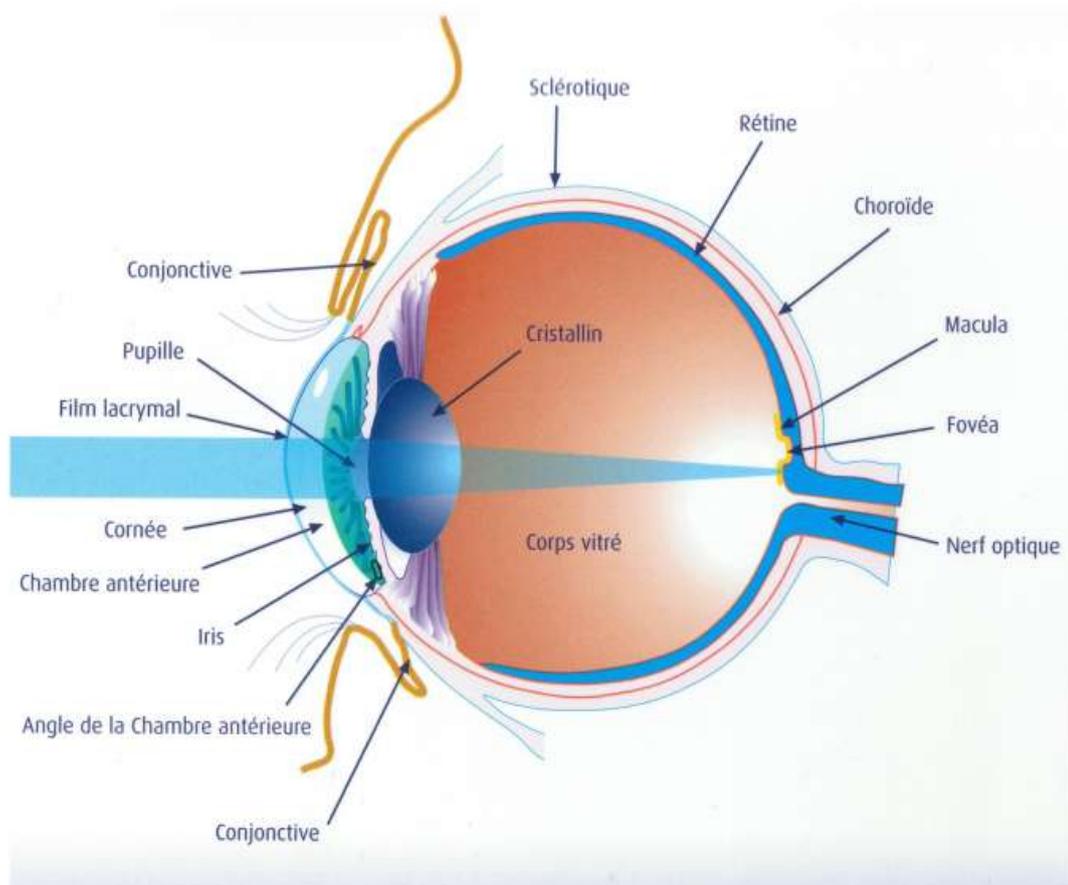


Figure I-1 : Coupe horizontale de l'œil

1.2.1. Anatomie de l'œil :

Selon la Figure I-1 on peut diviser l'anatomie de l'œil en trois parties[2] :

a) Les parties externes de l'œil :

➤ La cornée :

La cornée est la lentille antérieure de l'œil. Elle est parfaitement transparente. C'est sur cette partie que l'on pose les lentilles de contact.

➤ La sclère :

La sclère correspond au blanc des yeux. C'est une structure très résistante qui donne la structure de l'œil.

➤ La conjonctive :

La conjonctive est une membrane transparente qui recouvre la sclère. Elle est bien vascularisée. Quand on a les yeux rouges, ce sont ces vaisseaux-là qui se gonflent de sang. Comme elle est en contact direct avec l'extérieur, la conjonctive est sujette aux inflammations. C'est ce qu'on appelle la conjonctivite.

➤ Les paupières :

Les paupières sont formées de peau, de cartilage et de conjonctive. Elles ont pour rôle de protéger l'œil des agressions extérieures et aussi de le lubrifier. L'œil a besoin d'être toujours humide. De petites glandes dans les paupières sécrètent des substances lubrifiantes.

➤ Les larmes :

Les larmes sont essentielles au confort des yeux. Elles recouvrent en permanence la surface de l'œil. Les larmes sont composées d'eau, de lipides et de mucus. Ces substances sont sécrétées par des glandes dans les paupières et la conjonctive. Les larmes ont plusieurs rôles : lubrification, défense immunitaire, évacuation des poussières.

b) Parties internes visibles :

➤ L'iris :

C'est une membrane pigmentaire qui donne la couleur aux yeux. L'iris permet de réguler la quantité de lumière qui rentre dans l'œil. S'il y a beaucoup de lumière, l'iris se contracte et la pupille devient petite pour ne pas être éblouie. S'il fait sombre, l'iris se dilate pour nous permettre de mieux voir dans l'obscurité.

➤ La pupille :

La pupille est le trou noir au centre de l'iris. Il détermine la quantité de lumière qui rentre dans l'œil.

c) Parties internes non visibles :

➤ La rétine :

Couche réceptrice de l'œil qui perçoit l'image. La rétine peut être comparée au capteur d'un appareil photo digital. La rétine est tapissée de récepteurs (les cônes pour les couleurs et les bâtonnets pour la luminosité) (figureI-2). Ces récepteurs captent l'image et la transmettent ensuite au cerveau par l'intermédiaire du nerf optique.

➤ La fovéa ou macula :

C'est la zone centrale de la rétine qui permet de voir les détails et les couleurs. Il y a plusieurs millions de récepteurs dans cette zone. On appelle ces récepteurs les cônes qui sont capables de distinguer les couleurs : rouge, vert, bleu.

➤ Le nerf optique :

Le nerf optique transmet au cerveau les informations visuelles captées par les récepteurs. Le cerveau ensuite les déchiffre. C'est aussi grâce au cerveau que l'on peut voir.

➤ Le corps vitré :

Il s'agit d'un gel aqueux transparent qui permet à l'œil de garder sa forme et sa fermeté. Il devient plus liquide avec les années, on perçoit alors des particules qui flottent devant la vision. Ce phénomène s'appelle les mouches volantes.

➤ La rétine :

La rétine est une membrane fine et transparente tapissant le fond de l'œil destinée à recevoir les impressions lumineuses qui délivrent la vision. C'est cette partie qui nous intéresse.

1.2.2. La rétine :

C'est la rétine qui est responsable de transformer les rayons lumineux en un signal nerveux et de le transmettre au système nerveux central. Elle est constituée de 10 couches : 4 couches de cellules visuelles qui permettent la réception de la lumière, puis 6 autres couches permettant sa transmission vers le cerveau par le nerf optique[3].

La rétine se caractérise par deux rétines : centrale et périphérique.

- La rétine centrale se trouve dans l'axe de vision, avec plus précisément la macula : portion non vascularisée uniquement composée de photorécepteurs appelés cônes qui sont directement exposés à la lumière. En son centre se trouve une petite zone de 1,5 mm de diamètre : la fovéa. La fovéa contient environ 50 000 cônes, qui, stimulés par la lumière, permettent d'obtenir la meilleure précision de vision, la vision des formes, des couleurs, et la vision diurne (de jour).
- La rétine périphérique est chargée de détecter les informations dans la totalité du champ de vision, les mouvements, la vision est moins précise. La rétine périphérique est essentielle pour la vision nocturne grâce aux photorécepteurs spécialisés dans la réception lumineuse en basse luminance : les bâtonnets.

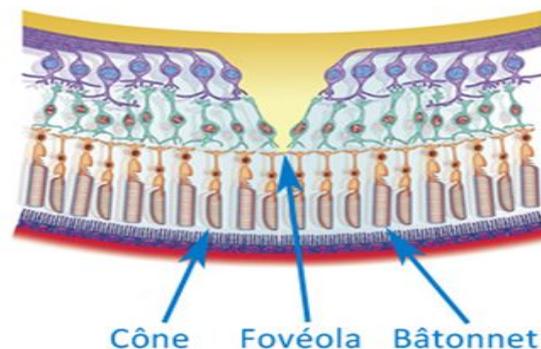


Figure I-2 : Schéma en coupe de la rétine centrale

1.3. Les rétinopathies:

Les rétinopathies c'est les infections et les maladies qui touchent la rétine des yeux pouvant entraîner une déficience visuelle. Elle fait souvent référence à une maladie vasculaire rétinienne autrement dit à des dommages survenus sur des lésions des vaisseaux sanguins.

La rétine, à l'arrière de l'œil, fournit une fenêtre sur le système circulatoire. En l'examinant, un médecin peut inspecter un échantillon des vaisseaux sanguins du corps et détecter les signes précoces de complications liées au diabète ou à l'hypertension, ainsi que de nombreuses autres maladies (Drépanocytose, anémie, lupus etc.). La rétinopathie peut également être observée chez les nouveau-nés prématurés[4].

Les rétinopathies sont généralement un signe d'une autre affection médicale. Bien que plusieurs affections médicales (p. Ex., Drépanocytose, lupus) puissent causer la rétinopathie, les causes les plus fréquentes sont le diabète et l'hypertension.

1.3.1. La Rétinopathie Diabétique (RD) :

La rétinopathie diabétique est une détérioration des vaisseaux sanguins de la rétine qui affecte généralement les deux yeux. Presque toutes les personnes atteintes de diabète montrent des signes de lésions rétinienne après environ 20 ans de vie avec la maladie.

La rétinopathie diabétique est une complication du diabète. Le diabète provoque des niveaux élevés de sucre dans le sang, ce qui peut endommager les vaisseaux sanguins (figure I-4). Les vaisseaux endommagés autour de la rétine peuvent laisser échapper des protéines et des graisses, formant des dépôts pouvant interférer avec la vision. Les vaisseaux sanguins endommagés ne sont pas aussi efficaces pour transporter l'oxygène vers la rétine, ce qui peut également causer des dommages.

Au stade avancé, appelé rétinopathie proliférative, de nouveaux vaisseaux sanguins se développent dans l'œil. Cependant, ils sont faibles et éclatent souvent, provoquant des saignements dans les yeux. Le saignement peut provoquer des lésions oculaires et endommager la vision.

1.3.2. La rétinopathie hypertensive :

Il est inhabituel que l'hypertension altère la vision, mais la rétinopathie hypertensive peut entraîner un blocage des artères ou des veines de la rétine, ce qui peut entraîner la perte de la vision. Le tabagisme et le diabète augmentent le risque de développer une rétinopathie hypertensive.

La rétinopathie hypertensive est une complication de l'hypertension qui prend généralement plusieurs années à se développer. Une pression artérielle élevée endommage les parois des vaisseaux sanguins, les épaississant et les rétrécissant (figure I-5). Cela réduit l'apport sanguin disponible pour la rétine, entraînant des dommages à la rétine. En fin de compte, le sang peut s'infiltrer dans la rétine, causant des dommages supplémentaires.

1.3.3. Autres Rétinopathies :

Bien que les rétinopathies les plus fréquentes restent La rétinopathie diabétique et La rétinopathie hypertensive, ils existent d'autres rétinopathies. Tel que :

➤ La rétinopathie du prématuré :

La rétinopathie du prématuré est une maladie consistant en un dysfonctionnement des vaisseaux sanguins de la rétine chez le pré maturé.

Une des causes est une oxygénothérapie mal contrôlée lors de la réanimation néonatale.

➤ La rétinopathie photique :

La rétinopathie photique est une affection de la rétine de l'œil, en particulier de la macula, à la suite d'une exposition prolongée au rayonnement solaire ou d'autres lumières intenses, par exemple, des lasers ou de la soudure à l'arc.

➤ La rétinopathie des radiations :

La rétinopathie des radiations ou rétinopathie radiée est une lésion de la rétine due à une exposition aux rayonnements ionisants. La rétinopathie par rayonnement a un retard d'apparition, généralement après des mois ou des années de radiation, et est lentement progressive. En général, la rétinopathie par rayonnement est observée environ 18 mois après le traitement par rayonnement externe et par curiethérapie. Le délai d'apparition de la rétinopathie par rayonnement se situe entre 6 mois et 3 ans.

1.3.4. Symptômes et Complications :

Au début de la rétinopathie, il peut ne pas y avoir de symptômes du tout. Au fur et à mesure que la maladie progresse que les symptômes apparaissent. Les patients deviennent souvent symptomatiques en cas de dommages irréversibles. Les symptômes ne sont généralement pas douloureux et se ressemblent dans tous les cas, on peut distinguer ceux de la rétinopathie diabétique et La rétinopathie hypertensive.

➤ la rétinopathie diabétique :

- Vision Floue
- Vision Fluctuante
- Angles Morts
- Voir des taches Flottantes
- Changements de perception de la Couleur
- Perte de vision soudaine
- Vision double
- Douleur oculaire dans les cas avancés

➤ La rétinopathie hypertensive :

- Maux de tête
- Changements de Vision
- Perte de vision soudaine d'un ou des deux yeux
- Vision double

1.3.5. Diagnostic :

La rétinopathie est diagnostiquée par un ophtalmologue lors de l'examen oculaire. La photographie stéréoscopique du fond d'œil est l'étalon-or pour le diagnostic de la rétinopathie. La Fondoscopie dilatée, ou la visualisation directe du fond d'œil, s'est également avérée efficace. En clair la photographie du fond d'œil révèle tous.

La rétinopathie diabétique est une complication à long terme du diabète. Si une personne est atteinte de diabète, il est extrêmement important que ses yeux soient surveillés et examinés.

Le signe le plus précoce de rétinopathie diabétique que l'ophtalmologue pourrait détecter est la formation de microanévrismes (figure I-4). Ce sont des petits ballons dans la rétine qui apparaissent sous la forme de minuscules points rouges à l'arrière de l'œil; ils se cassent parfois, provoquant un saignement dans la rétine et une vision trouble.

Une forme plus avancée de rétinopathie diabétique, appelée rétinopathie diabétique proliférative, peut entraîner des cicatrices qui diminuent la vision. Dans la rétinopathie proliférative, de nouveaux vaisseaux sanguins se développent au-dessus de la rétine et pénètrent dans le corps vitré (la substance gélatineuse située entre le cristallin et la rétine). Ces vaisseaux sanguins peuvent gonfler et éclater, provoquant des saignements et des lésions oculaires. Dans certains cas, le sang peut être réabsorbé, mais dans de nombreux cas, la rétine peut se détacher, provoquant une cécité totale.

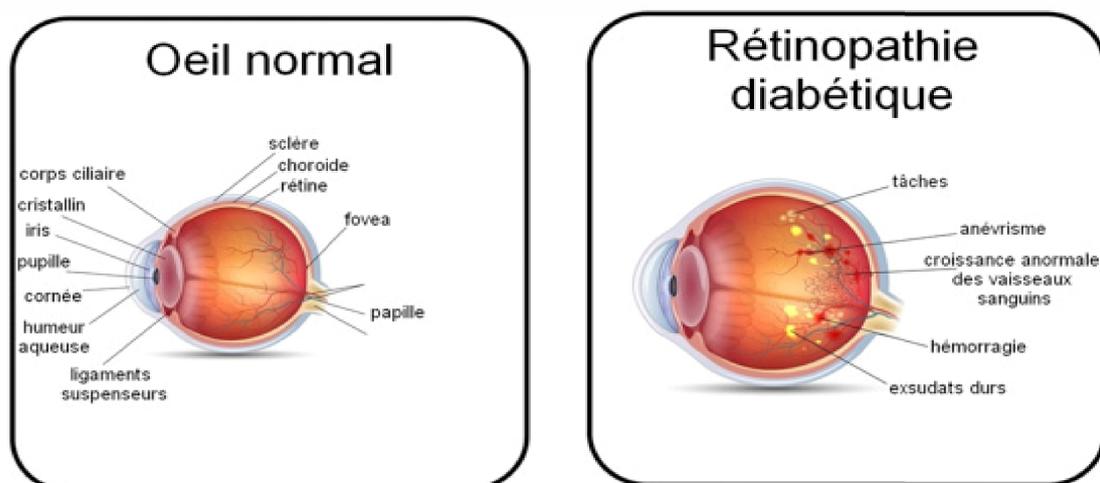


Figure I-3 : Différence entre un œil normal et un œil atteint d'une RD

La Figure I-4 montre ce qu'on obtient d'une photographie du fond d'œil d'une RD.

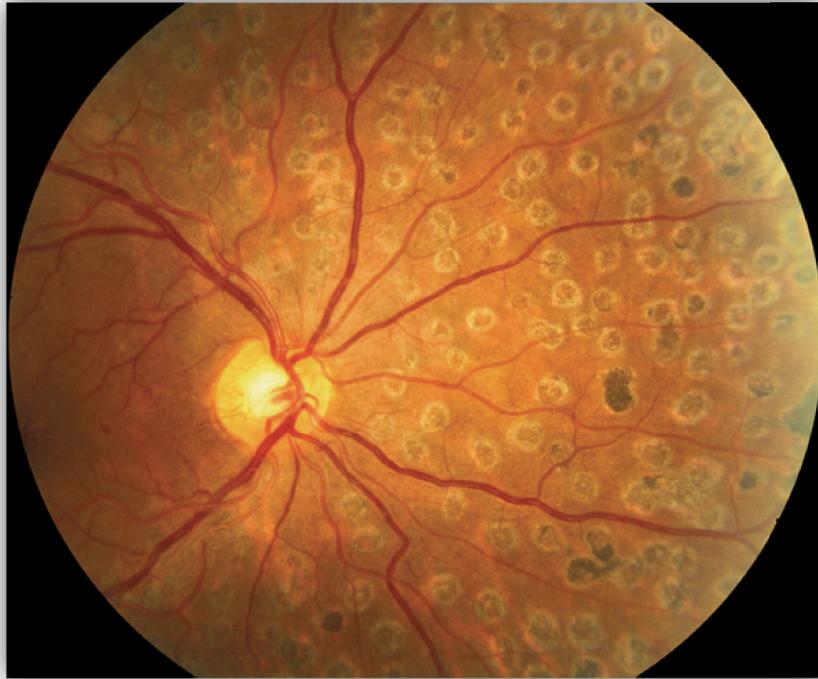


Figure I-4 : Photographie du fond d'œil d'une RD

En cas de rétinopathie hypertensive, un ophtalmologiste recherchera de minuscules plaques contenant du cholestérol dans les vaisseaux sanguins de la rétine (Exsudat), ainsi que d'autres modifications des vaisseaux sanguins telles que le rétrécissement et l'épaississement.

La Figure I-5 illustre mieux les signes que provoque la rétinopathie hypertensive sur la rétine.

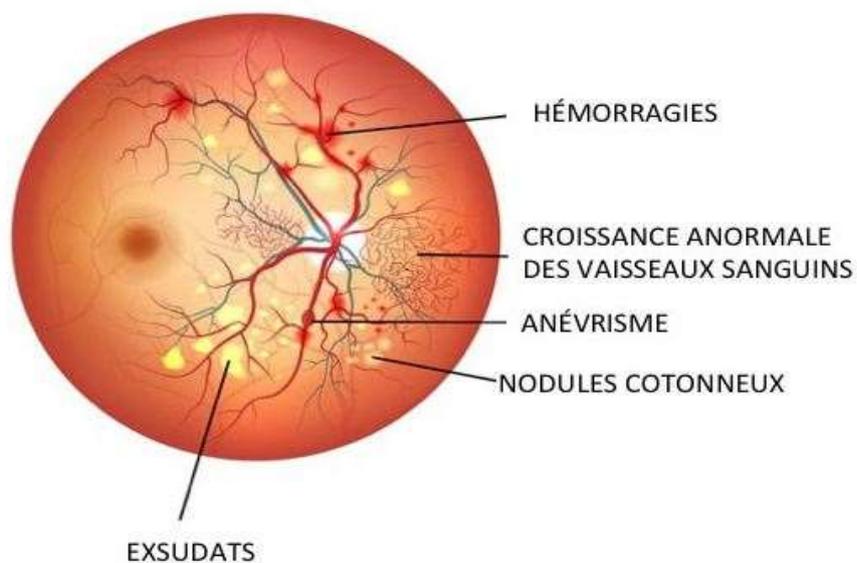


Figure I-5 : Rétine atteinte d'une rétinopathie hypertensive

1.4. Traitement et Prévention :

La clé du traitement de la rétinopathie consiste à gérer les causes sous-jacentes de cette affection par exemple :

Le contrôle de la glycémie dans le diabète est essentiel pour retarder l'apparition de la rétinopathie diabétique. Une prise en charge adéquate du diabète implique de prendre les traitements prescrits, tels que l'insuline ou d'autres médicaments contre le diabète, ainsi que de suivre un régime alimentaire sain et un programme d'exercices.

Maintenir la pression artérielle sous contrôle aidera à prévenir la rétinopathie hypertensive. Réduire la tension artérielle avec des médicaments appropriés aidera à prévenir les complications. L'exercice régulier, un régime alimentaire approprié et d'autres changements de style de vie, tels que l'abandon du tabac, contribueront grandement à réduire le risque de rétinopathie.

Et concernant Le traitement dans la plupart des cas sa peut inclure un traitement au laser de la rétine. La thérapie par photo-coagulation au laser est le traitement standard de nombreux types de rétinopathie. Les données montrent que la thérapie au laser est généralement sûre et améliore les symptômes visuels dans la rétinopathie diabétique.

Le traitement par laser a évolué de manière importante ces dernières années. Les technologies de pointe permettent d'utiliser les impulsions laser non seulement pour la photocoagulation au niveau des hémorragies, mais également pour prévenir les nodules, anévrismes et croissance anormale de vaisseaux sanguins.

1.4.1. La Photocoagulation au laser :

La photocoagulation est une technique utilisée en ophtalmologie par laquelle on produit des cicatrices dans la rétine par moyen d'une brûlure thérapeutique réalisée avec l'application d'un faisceau laser sur ce tissu. La cicatrice résultante permet d'arrêter le développement de certaines maladies rétinienne[5]. Ça consiste à utiliser l'effet cicatrisant et coagulant d'un laser dont le rayon est focalisé précisément sur la rétine. Cette technique comporte trois objectifs distincts :

- Créer au moment de la cicatrisation un accolement des différentes couches de la rétine afin d'empêcher, par exemple, une déchirure rétinienne de s'étendre.
- Coaguler des vaisseaux qui fuient ou qui sont anormaux.
- Détruire très précisément des zones malades de la rétine qui, sans traitement, pourraient entraîner une diminution de la vision.

La technique se déroule à l'aide d'un verre de contact et d'un microscope doté d'un laser.

Parmi Les maladies pour lesquelles on l'utilise le plus sont[6]:

- Les déchirures rétinienne : Elles doivent être traitées le plus tôt possible. Autrement, du liquide peut s'accumuler au-dessous des déchirures. La photo-coagulation arrête le développement de ces déchirures afin d'éviter des décollements de rétine.

- Rétinopathies : Il est le cas de la rétinopathie diabétique, des occlusions des veines ou des pathologies exsudatives telles que la vitréorétinopathie exsudative familiale. La photo-coagulation facilite le scellement des vaisseaux sanguins ou tissus anormales, réduit dans une certaine mesure la croissance de nouveaux vaisseaux sanguins anormaux et facilite l'absorption du liquide.
- Glaucome : La photo-coagulation réduit la pression oculaire tout en facilitant l'issue de l'humeur aqueuse dans les cas de glaucome.
- Tumeurs : La photo-coagulation peut arrêter la croissance de tumeurs réiniennes ou même les réduire.

La Figure schématise un exemple de Photocoagulation par laser lors d'un traitement d'une rétinopathie diabétique. Afin de coaguler des vaisseaux qui sont anormaux.

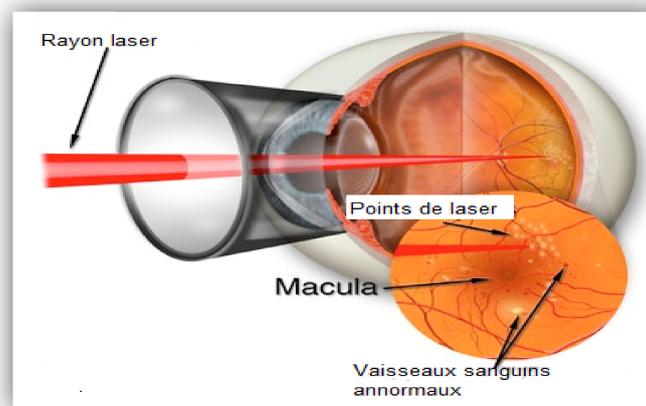


Figure I-6 : Photocoagulation par laser d'une RD

1.4.2. Prévention et traitement modernes à l'aide d'impulsions Laser :

Les lasers en tant que mode de prise en charge des maladies réiniennes continuent d'évoluer en termes de protocole de prise en charge, d'innovations et d'indications toujours plus nombreuses. Ils ont offert une certaine variabilité dans la durée des impulsions et des traitements plus précisément ciblés. Ils ont été appliqués avec succès sur le plan thérapeutique, mais une destruction chori-rétinienne intense et des hémorragies fréquentes se sont rapidement révélées être un problème.

Avec l'avènement des technologies laser à microimpulsion et à nanoimpulsion, de haute précision, la thérapie au laser retrouve son importance, avec des indications de plus en plus nombreuses dans le traitement et la prévention de nombreuses maladies de la rétine, sans les effets de destruction de zones saines de la rétine, grâce à la modulation de fréquence des lasers, la réduction du temps d'exposition de la rétine, au traitement parallèle de zones altérées (lasers multi spots), parmi d'autres.

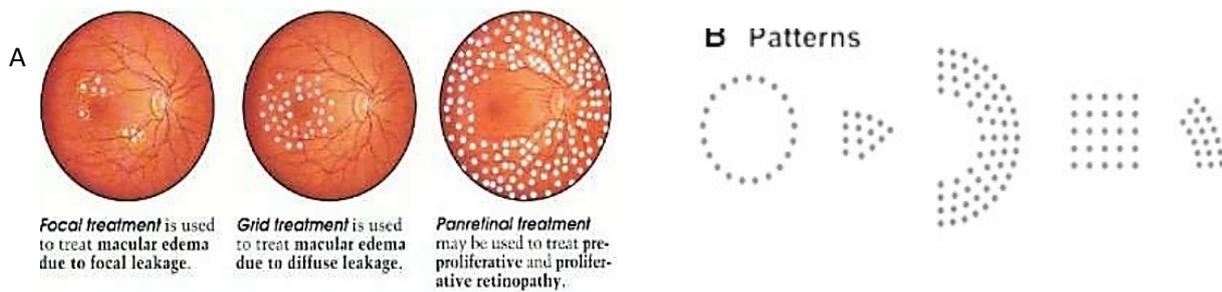


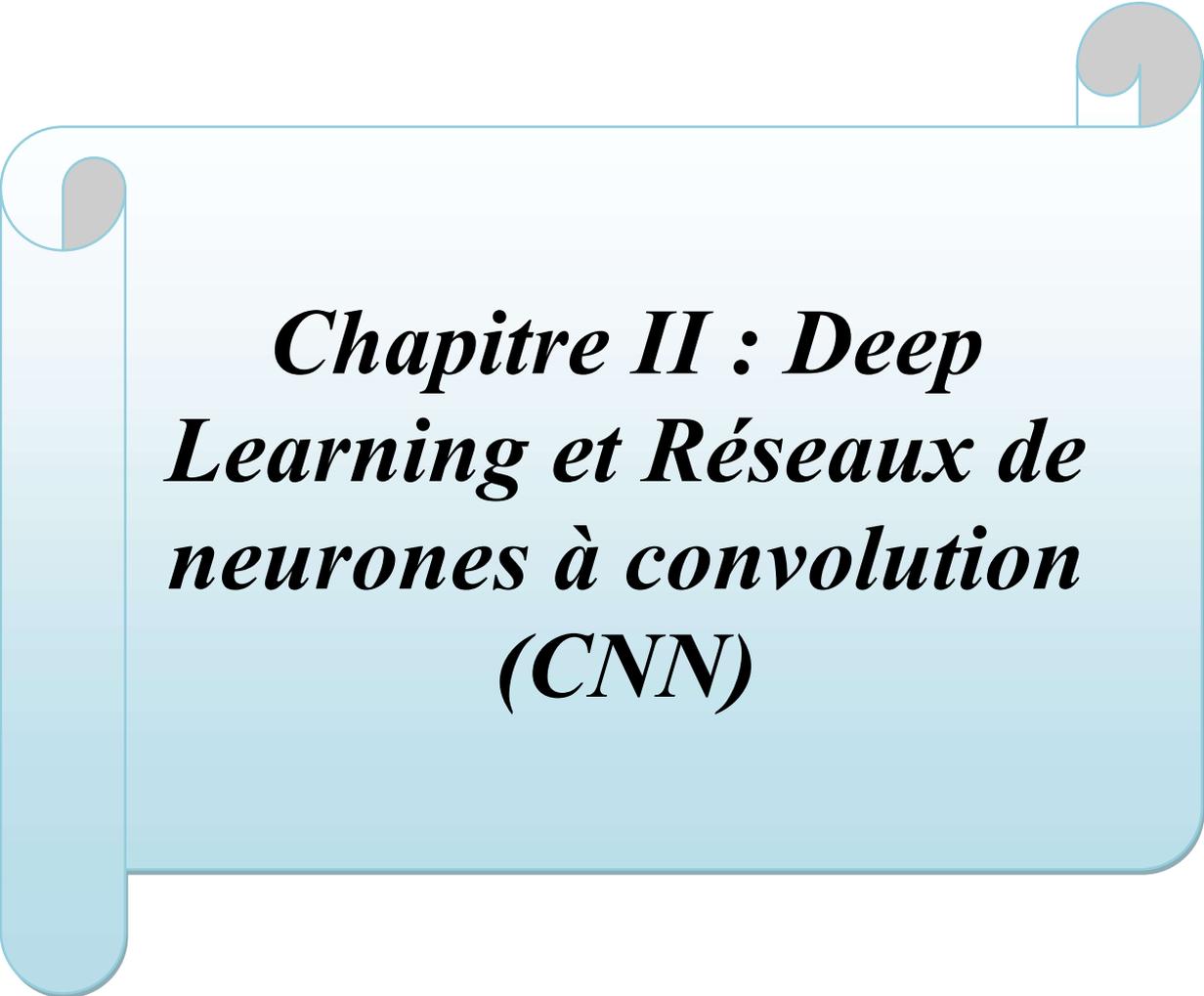
Fig. I-7 : Exemples : a) d'utilisations de laser multispots, b) de motifs pour laser multi-spots

De tels équipements thérapeutiques requièrent une détection précise des zones rétiniennes endommagées, d'où la nécessité d'un algorithme d'aide au diagnostic efficace, rapide, et précis. Nous avons choisi, pour notre travail, les algorithmes de neurones à convolution (CNN) entraînés par l'algorithme d'apprentissage profond, le Deep Learning.

1.5. Conclusion :

Ce chapitre illustre les généralités sur l'aspect médical de la rétinopathie : l'anatomie de l'œil pour mieux comprendre où se trouve la rétine et son rôle. Les rétinopathies définition, les causes des rétinopathies, symptômes ainsi que le traitement par Photocoagulation un résumé de cette méthode qui a fait ces preuves dans le monde de l'ophtalmologie.

L'apprentissage profond, et les réseaux de neurones à convolution ont prouvé leur efficacité dans le domaine de traitement d'image. Nous utiliserons cet algorithme d'intelligence artificielle pour la détection des rétinopathies et le présentons dans le deuxième chapitre. On s'approfondira plus dans celle-ci, notamment par le Machine Learning et le Deep Learning.



***Chapitre II : Deep
Learning et Réseaux de
neurones à convolution
(CNN)***

2.1.Introduction :

Notre travail consiste à utiliser l'apprentissage profond afin de détecter la présence de rétinopathie dans une angiographie de la rétine. Le principe vise à déterminer les paramètres du CNN, lequel une fois construit et embarqué sur le système (de l'équipe biorétine, ce sera un photocoagulateur laser embarqué) permettra la détection en temps réel des points d'anomalie.

Le présent chapitre définit le deep learning, qui est une branche du machine learning, lui-même une branche de l'intelligence artificielle.

Il décrit également le fonctionnement théorique des réseaux de neurones, ainsi que la structure des réseaux de neurones à convolution, qui subiront l'apprentissage profond.

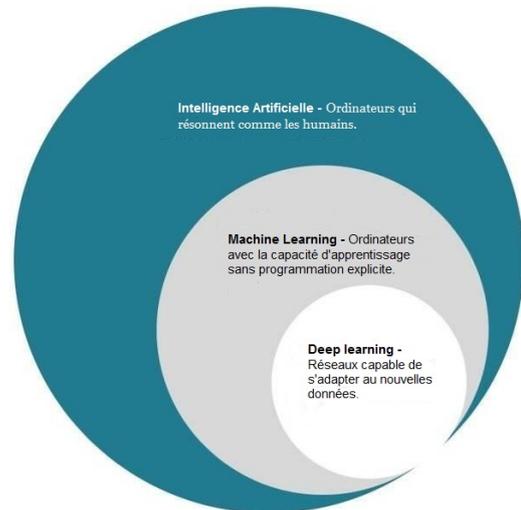


Figure II-1 : Relation entre IA, Machine Learning et Deep Learning

2.2.L'intelligence artificielle :

C'est un ensemble de méthodes dans le but est l'imitation du comportement et de l'intelligence humains. Les algorithmes neuromorphiques, en particulier les réseaux de neurones à convolution, qui représentent une branche importante de l'intelligence artificielle, seront décrits dans ce chapitre.

2.2.1. Machine Learning :

L'apprentissage automatique (Machine Learning ou ML) fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques. Chaque observation, comme par exemple « j'ai mangé tels et tels aliments à tel moment de la journée pendant telle période ce qui a causé telle maladie » est décrite au moyen de deux types de variables :

- Les premières sont appelées les variables prédictives (ou attributs ou caractéristiques), dans notre exemple mon âge, mon dossier médical, mes antécédents médicaux. Ce sont les variables à partir desquelles on espère pouvoir faire des prédictions. Les n variables prédictives associées à une observation seront notées comme un vecteur $x=(x_1 ; \dots ; x_n)$ à n composantes. Un ensemble de M observations sera constitué de M tels vecteurs $x^{(1)}; \dots ; x^{(M)}$.
- Une variable cible dont on souhaite prédire la valeur pour des événements non encore observés. Dans notre exemple, il s'agirait de la maladie contractée. on notera (y) cette variable cible.

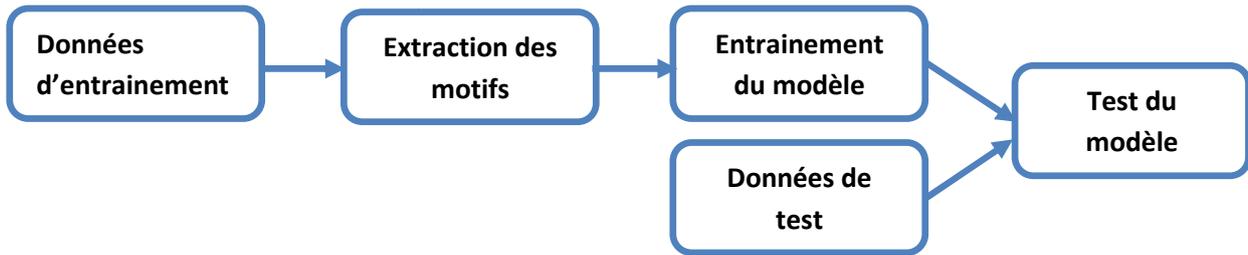


Figure II-2 : Processus typique du ML

En résumé, la valeur de la variable y dépend de :

- Une fonction $F(x)$ déterminée par les variables prédictives.
- Un bruit $\varepsilon(x)$ qui est le résultat d'un nombre de paramètres dont on ne peut pas tenir compte.

Aussi bien F que ε ne seront jamais connues mais l'objectif d'un modèle de ML est d'obtenir la meilleure approximation possible de F à partir des observations disponibles. Cette approximation sera notée f , on l'appelle la fonction de prédiction.

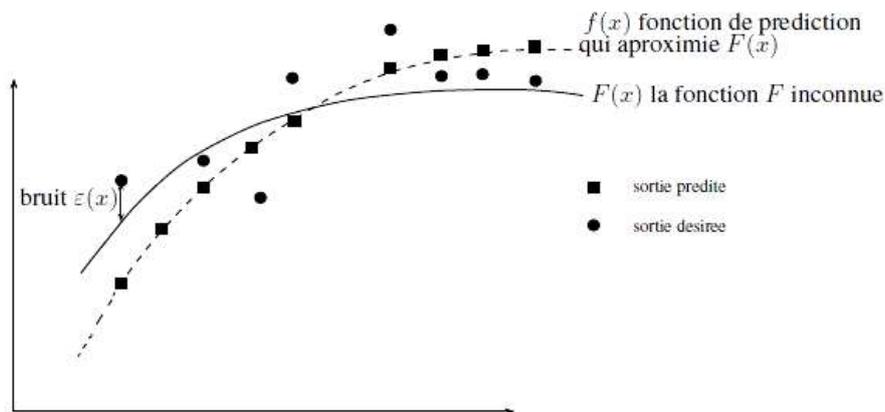


Figure II-3 : Modèle de ML pour approximation de F

2.2.2. Deep Learning :

L'apprentissage profond (en anglais Deep Learning) est un sous-ensemble de l'apprentissage automatique qui a été introduit dans le but de rapprocher le ML de son objectif principal : l'intelligence artificielle. Il concerne les algorithmes inspirés par la structure et le fonctionnement du cerveau. Ils peuvent apprendre plusieurs niveaux de représentation dans le but de modéliser des relations complexes entre les données.

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

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

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

Mais ce n'est qu'après que de plus grandes quantités de données ne soit disponibles grâce notamment au Big Data et aux objets connectés et que les machines de calcul soient devenues plus puissantes qu'on a pu comprendre le potentiel réel du Deep Learning.

Une des grandes différences entre le Deep Learning et les algorithmes de ML traditionnelles c'est qu'il s'adapte bien, plus la quantité de données fournie est grande plus les performances d'un algorithme de Deep Learning sont meilleures. Contrairement à plusieurs algorithmes de ML classiques qui possèdent une borne supérieure à la quantité de données qu'ils peuvent recevoir des fois appelée plateau de performance, les modèles de Deep Learning n'ont pas de telles limitations (théoriquement) et ils sont même allés jusqu'à dépasser la performance humaine.

Autre différence entre les algorithmes de ML traditionnelles et les algorithmes de Deep Learning c'est l'étape de l'extraction de caractéristiques. Dans les algorithmes de ML traditionnelles l'extraction de caractéristiques est faite manuellement, c'est une étape difficile et coûteuse en temps et requiert un spécialiste en la matière alors qu'en Deep Learning cette étape est exécutée automatiquement par l'algorithme.

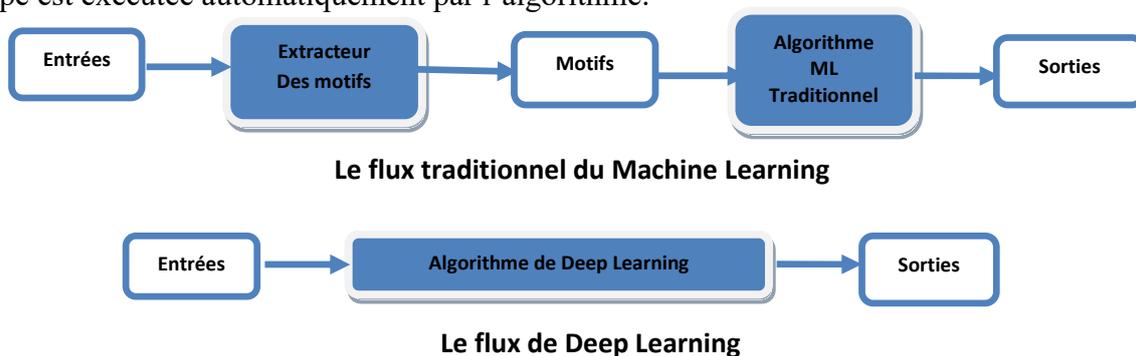


Figure II-4 : Le procédé du Machine Learning comparé à celui du Deep Learning

2.3. Les réseaux de neurones artificiels :

Inspirée du fonctionnement des neurones biologiques, les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle.

Chaque processeur élémentaire (neurone artificiel) calcule une sortie unique sur la base des informations qu'il reçoit.

Grâce à leur capacité d'apprentissage, les réseaux neuronaux permettent de découvrir des relations complexes non-linéaires entre un grand nombre de variables, sans intervention externe. De ce fait, ils sont largement utilisés dans de nombreux problèmes de classification (ciblage marketing, reconnaissance de formes, traitement de signal,...) d'estimation (modélisation de phénomènes complexes,...) et prévision (bourse, ventes,...).

Il existe un compromis entre clarté du modèle et pouvoir prédictif. Plus un modèle est simple, plus il sera facile à comprendre, mais moins il sera capable de prendre en compte des dépendances trop variées.

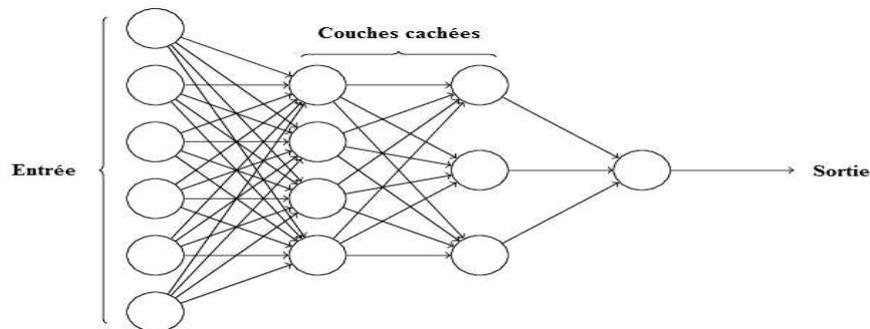


Figure II-5 : Architecture d'un Réseaux de neurones artificiels

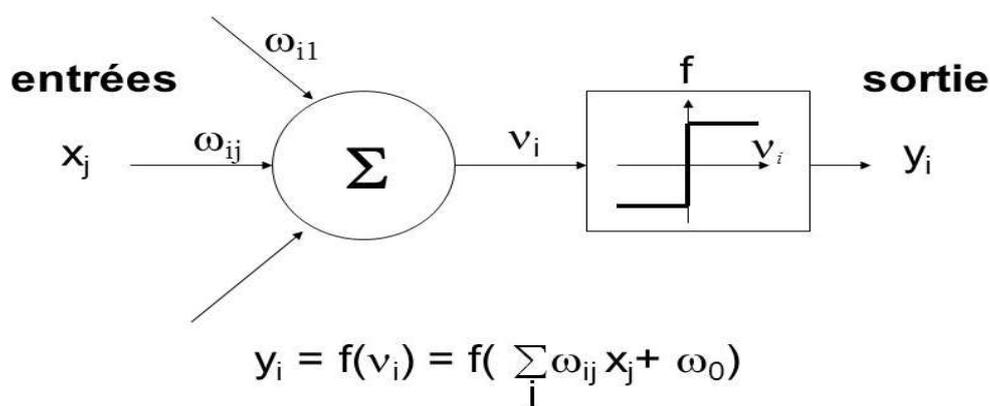
2.3.1. Neurone artificiel :

Neurone artificiel aussi appelé neurone formel, est une fonction algébrique non linéaire et bornée, dont la valeur dépend de paramètres appelés coefficients ou poids. Les variables de cette fonction sont habituellement appelées entrées du neurone, et la valeur de la fonction est appelée sortie. Un neurone est donc avant tout un opérateur mathématique, dont on peut calculer la valeur numérique par quelques lignes de programme informatique[7].

Autrement dit le neurone reçoit un certain nombre de variables d'entrées provenant de neurones en amont. À chacune de ces entrées, est associé un poids qui représente la force de la connexion entre les deux neurones.

Chaque neurone est doté d'une unique sortie qui se ramifie ensuite pour représenter les entrées qui alimenteront d'autres neurones en aval. Pour résumer, chaque neurone calcule une sortie unique en se basant sur les informations qui lui sont données [8].

Neurone artificiel



Pondération ➡ **Sommation** ➡ **Transfert**

Figure II-6 : Schéma montrant la composition d'un neurone artificiel

Le neurone en tant qu'unité élémentaire agit de la façon suivante :

- 1^{ère} phase : Le neurone fait le calcul de la somme pondérée des entrées (en fonction de la force des connexions). L'apprentissage ayant été réalisé auparavant, le poids des connexions est ici déjà déterminé et fixe.
- 2^e phase : À partir de la valeur obtenue, une fonction d'activation ou de transfert va calculer la valeur de l'état du neurone. Le neurone compare la somme pondérée des entrées à une valeur de seuil et fournit alors une réponse en sortie.

La majorité des fonctions sont continues et offrent donc une infinité de valeurs possibles comprises dans les intervalles $[0;+1]$ ou $[-1;+1]$.

Il existe différents types de fonctions de transfert pour le neurone artificiel [9]. Les plus courantes et les plus citées dans la littérature sont énumérées dans la figure II-7.

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

Figure II-7 : Tableau des Fonctions de transfert pour le neurone artificiel

Les trois les plus utilisées sont les fonctions seuil (a), linéaire (b) et sigmoïde(c) comme représentées Figure II-8 :

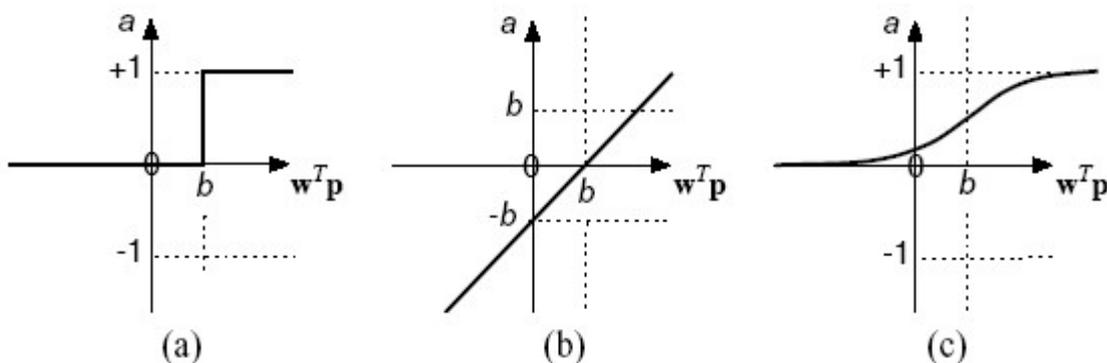


Figure II-8 : Fonctions de transfert les plus utilisées

2.3.2. Topologies de réseaux de neurones artificiels :

Les connexions entre les neurones qui composent le réseau décrivent la topologie du modèle. Elle peut être quelconque, mais le plus souvent il est possible de distinguer une certaine régularité.

a) Réseau multicouche:

Les neurones sont arrangés par couche. Il n'y a pas de connexion entre neurones d'une même couche et les connexions ne se font qu'avec les neurones des couches avales (figure II-9). Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et celle-ci seulement.

Ceci nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et donc définir les concepts de neurone d'entrée, neurone de sortie. Par extension, on appelle couche d'entrée l'ensemble des neurones d'entrée, couche de sortie l'ensemble des neurones de sortie. Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelés couches cachées.

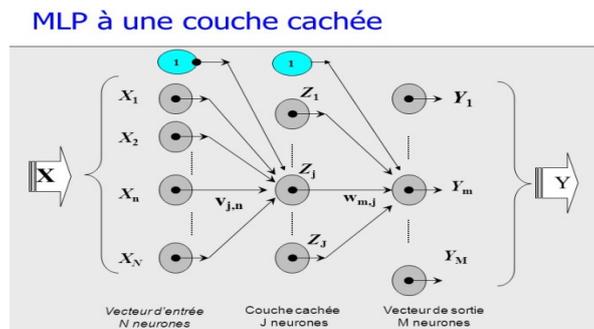


Figure II-9 : Définition des couches d'un réseau multicouche

b) Réseau à connexions locales :

Il s'agit d'une structure multicouche, mais qui à l'image de la rétine, conserve une certaine topologie. Chaque neurone entretient des relations avec un nombre réduit et localisé de neurones de la couche avale (figure II-10). Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique.

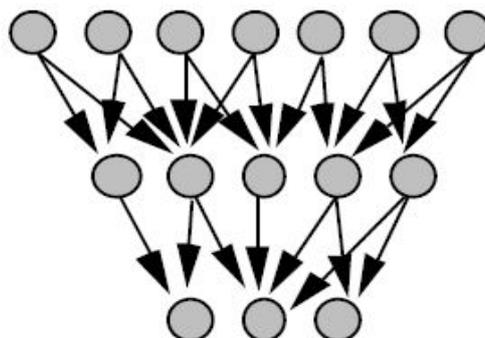


Figure II-10 : Réseau à connexions locales

Réseau à connexions récurrentes :

Les connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouche. Ces connexions sont le plus souvent locales (figure II-11).

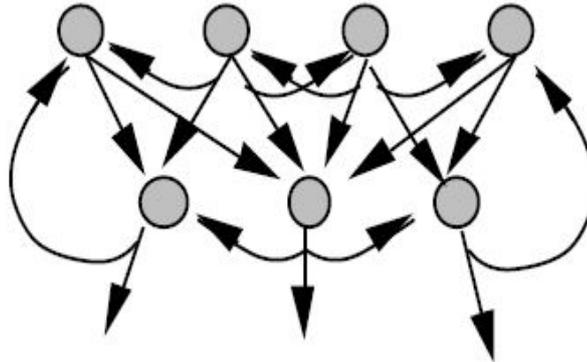


Figure II-11 : Réseau à connexions récurrentes

Réseau à connexion complète :

C'est la structure d'interconnexion la plus générale (figure II-12). Chaque neurone est connecté à tous les neurones du réseau (et à lui-même).

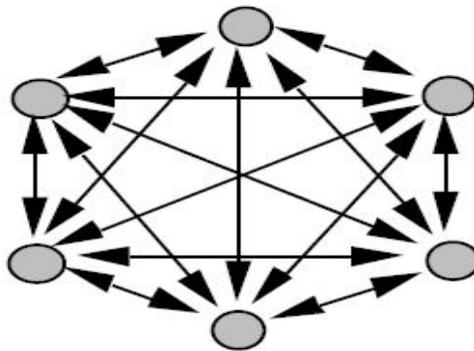


Figure II-12 : Réseau à connexion complète

Il existe de nombreuses autres topologies possibles, mais elles n'ont pas eu à ce jour la notoriété des quelques unes que nous avons décrites ici [10].

2.3.3. L'Apprentissage :

L'apprentissage est vraisemblablement la propriété la plus intéressante des réseaux neuronaux. Elle ne concerne cependant pas tous les modèles, mais les plus utilisés.

a) Définition :

L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré.

L'apprentissage neuronal fait appel à des exemples de comportement.

Dans le cas des réseaux de neurones artificiels, on ajoute souvent à la description du modèle l'algorithme d'apprentissage. Le modèle sans apprentissage présente en effet peu d'intérêt.

Dans la majorité des algorithmes actuels, les variables modifiées pendant l'apprentissage sont les poids des connexions.

L'apprentissage est la modification des poids du réseau dans l'optique d'accorder la réponse du réseau aux exemples et à l'expérience. Il est souvent impossible de décider a priori des valeurs des poids des connexions d'un réseau pour une application donnée. A l'issue de l'apprentissage, les poids sont fixés : c'est alors la phase d'utilisation. Certains modèles de réseaux sont improprement dénommés à apprentissage permanent. Dans ce cas il est vrai que l'apprentissage ne s'arrête jamais, cependant on peut toujours distinguer une phase d'apprentissage (en fait de remise à jour du comportement) et une phase d'utilisation. Cette technique permet de conserver au réseau un comportement adapté malgré les fluctuations dans les données d'entrées.

Au niveau des algorithmes d'apprentissage, il a été défini deux grandes classes selon que l'apprentissage est dit supervisé ou non supervisé. Cette distinction repose sur la forme des exemples d'apprentissage. Dans le cas de l'apprentissage supervisé, les exemples sont des couples (Entrée, Sortie associée) alors que l'on ne dispose que des valeurs (Entrée) pour l'apprentissage non supervisé. Remarquons cependant que les modèles à apprentissage non supervisé nécessite avant la phase d'utilisation une étape de labellisation effectuée l'opérateur, qui n'est pas autre chose qu'une part de supervision.

b) Types d'apprentissage :

Les algorithmes d'apprentissage peuvent se catégoriser selon deux grandes classes:

➤ Apprentissage supervisé :

L'apprentissage supervisé consiste à apprendre une fonction de prédiction à partir d'exemples annotés. On distingue les problèmes de régression des problèmes de classification. Ainsi, on considère que les problèmes de prédiction d'une variable quantitative sont des problèmes de régression tandis que les problèmes de prédiction d'une variable qualitative sont des problèmes de classification.

Les exemples annotés constituent une base d'apprentissage, et la fonction de prédiction apprise peut aussi être appelée hypothèse ou modèle. On suppose cette base d'apprentissage représentative d'une population d'échantillons plus large et le but des méthodes d'apprentissage supervisé est de bien généraliser, c'est-à-dire d'apprendre une fonction qui fasse des prédictions correctes sur des données non présentes dans l'ensemble d'apprentissage.

L'apprentissage supervisé se découpe en deux parties :

- La première détermine un modèle de données étiquetées.
- La deuxième consiste à prédire l'étiquette d'une nouvelle donnée, connaissant le modèle préalablement appris.

Donald Hebb a proposé la première stratégie d'apprentissage supervisé en 1949, inspirée par des observations neurophysiologiques.

La loi de Hebb s'applique aux connexions entre neurones, comme le représente la figure II-13.

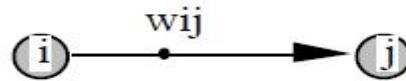


Figure II-13 : Connexion entre deux neurones

i le neurone amont, j le neurone aval et w_{ij} le poids de la connexion.

Elle s'exprime de la façon suivante Si 2 cellules sont activées en même temps alors la force de la connexion augmente.

On peut la décrire par la formule suivante $w_{ij} = x_i x_j$.

Où w_{ij} est le poids de la connexion de neurone i jusqu'à neurone j et x_i l'entrée de neurone i .

➤ Apprentissage non supervisé :

Différent de l'apprentissage supervisé, l'application d'un algorithme basé sur l'apprentissage non supervisé ne nécessite aucune connaissance des sorties souhaitées respectives.

Ainsi, le réseau doit s'organiser lorsqu'il y a des particularités existantes entre les éléments qui composent la totalité de l'échantillon, en identifiant les sous-ensembles (ou groupement) présentant des similitudes. L'algorithme d'apprentissage ajuste les poids synaptiques et les seuils du réseau afin de refléter ces grappes au sein du réseau lui-même

Alternativement, le concepteur de réseau peut spécifier (a priori) la quantité maximale de ces groupements possibles, en utilisant ses connaissances sur le problème.

La méthode des k moyennes est un exemple de l'apprentissage non supervisé.

Description :

- Choisir k points qui représentent la position moyenne des partitions $m_1^{(1)}, \dots, m_k^{(1)}$ initiales (au hasard par exemple).
- Répéter jusqu'à ce qu'il y ait convergence :
 - assigner chaque observation à la partition la plus proche :
$$S_i^{(t)} = \{x_j : \|x_j - m_i^{(t)}\| \leq \|x_j - m_{i^*}^{(t)}\| \forall i^* = 1 \dots, k\}$$
 - mettre à jour la moyenne de chaque groupement :

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

2.4. Le réseau de neurones à convolution (CNN) :

La topologie des CNN a été définie par Yann LeCun en 1998. La non disponibilité de processeurs puissants à l'époque ont écarté les CNN vu que leur implémentation logicielle et leur apprentissage demandaient une logistique (Mémoire, puissance de calcul...) impossible à fournir.

A partir de 2010, avec l'avènement de GPU et de processeurs multi-cœurs puissants, le deep learning et le CNN ont pu être expérimentés et leurs résultats testés.

Les résultats ont, en fait, été tellement concluants, que l'intelligence artificielle elle-même a été relancée (alors qu'elle était arrivée à une impasse).

2.4.1. Principe des CNN :

Désignés par l'acronyme CNN, de l'anglais Convolutional Neural Network, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a deux dimensions pour une image aux niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales (Rouge, Vert, Bleu).

La première partie d'un CNN est la partie convolutive à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers d'une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. En fin, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN.

Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée de couches entièrement connectées. Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image.

La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories. Voir Figure II-14.

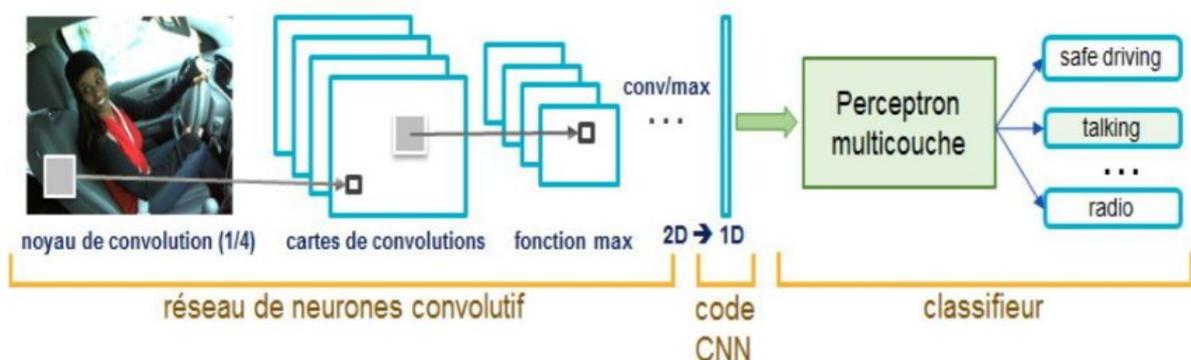


Figure II-14 : Exemple CNN

2.4.2. Architecture des CNN :

Une architecture CNN est formée par un empilement de couches de traitement

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

a) La couche de convolution (CONV) :

La couche de convolution est le bloc de construction de base d'un CNN. Le détail de son fonctionnement est précisé dans le paragraphe précédent.

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

- Profondeur de la couche : nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).
- Le pas contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.
- La marge (à 0): parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d'entrée.

Si le pas et la marge appliquée à l'image d'entrée permettent de contrôler le nombre de champs récepteurs à gérer (surface de traitement), la profondeur permet d'avoir une notion de volume de sortie, et de la même manière qu'une image peut avoir un volume, si on prend une profondeur de 3 pour les trois canaux RGB d'une image couleur, la couche de convolution va également présenter en sortie une profondeur.

C'est pour cela que l'on parle plutôt de volume de sortie et de volume d'entrée, car l'entrée d'une couche de convolution peut être soit une image soit la sortie d'une autre couche de convolution.

La taille spatiale du volume de sortie peut être calculée en fonction de la taille du volume d'entrée W_i la surface de traitement K (nombre de champs récepteurs), le pas S avec lequel ils sont appliqués, et la taille de la marge P La formule pour calculer le nombre de neurones du volume de sortie est $W_0 = \frac{W_i - K + 2P}{S} + 1$.

Si W_0 n'est pas entier, les neurones périphériques n'auront pas autant d'entrée que les autres. Il faudra donc augmenter la taille de la marge (pour recréer des entrées virtuelles).

Souvent, on considère un pas $S=1$, on calcule donc la marge de la manière suivante :

$p = \frac{K-1}{2}$ Si on souhaite un volume de sortie de même taille que le volume d'entrée. Dans ce cas particulier la couche est dite connectée localement.

b) Couche de pooling (POOL) :

Un autre concept important des CNN est le pooling (mise en commun), ce qui est une forme de sous-échantillonnage de l'image. L'image d'entrée est découpée en une série de rectangles de n pixels de côté ne se chevauchant pas (pooling). Chaque rectangle peut être vu comme une tuile. Le signal en sortie de tuile est défini en fonction des valeurs prises par les différents pixels de la tuile.

Le pooling réduit la taille spatiale d'une image intermédiaire, réduisant ainsi la quantité de paramètres et de calcul dans le réseau. Il est donc fréquent d'insérer périodiquement une couche de pooling entre deux couches convolutives successives d'une architecture CNN pour contrôler l'overfitting (sur-apprentissage). L'opération de pooling crée aussi une forme d'invariance par translation.

La couche de pooling fonctionne indépendamment sur chaque tranche de profondeur de l'entrée et la redimensionne uniquement au niveau de la surface. La forme la plus courante est une couche de mise en commun avec des tuiles de taille 2×2 (largeur/hauteur) et comme valeur de sortie la valeur maximale en entrée. On parle dans ce cas de Max-Pool 2×2 .

Il est possible d'utiliser d'autres fonctions de pooling que le maximum. On peut utiliser un average pooling (la sortie est la moyenne des valeurs du patch d'entrée), du L2-norm pooling. Dans les faits, même si initialement l'average pooling était souvent utilisé il s'est avéré que le max-pooling était plus efficace car celui-ci augmente plus significativement l'importance des activations fortes. En d'autres circonstances, on pourra utiliser un pooling stochastique.

Le pooling permet de gros gains en puissance de calcul. Cependant, en raison de la réduction agressive de la taille de la représentation (et donc de la perte d'information associée), la tendance actuelle est d'utiliser de petits filtres (type 2×2). Il est aussi possible d'éviter la couche de pooling mais cela implique un risque sur-apprentissage plus important.

Voir la figure II-15 pour exemple de Max-Pool 2×2 .

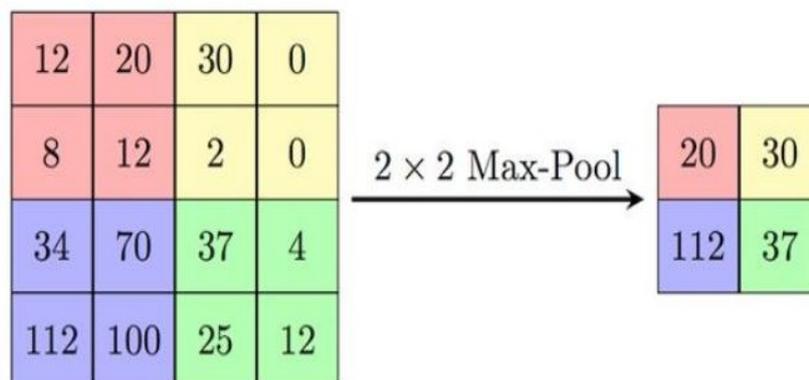


Figure II-15 : Exemple de Max-Pool 2×2

c) Couche de correction (ReLU):

Souvent, il est possible d'améliorer l'efficacité du traitement en intercalant entre les couches de traitement une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie. On a notamment :

- La correction ReLU (abréviation d'Unités Rectifié linéaires) : $f(x) = \max(0, x)$. Cette fonction, appelée aussi fonction d'activation non saturante, augmente les propriétés non linéaires de la fonction de décision et de l'ensemble du réseau sans affecter les champs récepteurs de la couche de convolution.
- La correction par tangente hyperbolique : $f(x) = \tanh(x)$.
- La correction par la tangente hyperbolique saturante : $f(x) = |\tanh(x)|$.
- La correction par la fonction sigmoïde : $f(x) = (1 + e^{-x})^{-1}$.

Souvent, la correction Relu est préférable, car il en résulte la formation de réseau neuronal plusieurs fois plus rapide, sans faire une différence significative à la généralisation de précision.

d) Couche entièrement connectée(FC):

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

e) Couche de perte (LOSS) :

La couche de perte spécifie comment l'entraînement du réseau pénalise l'écart entre le signal prévu et réel. Elle est normalement la dernière couche dans le réseau. Diverses fonctions de perte adaptées à différentes tâches peuvent y être utilisées. La perte Soft max est utilisée pour prédire une seule classe parmi K classes mutuellement exclusives. La perte par entropie croisée sigmoïde est utilisée pour prédire K valeurs de probabilité indépendante dans [0,1].

La perte euclidienne est utilisée pour régresser vers des valeurs réelles.

2.4.3. Choix des hyper-paramètres :

Les CNN utilisent plus d'hyper-paramètres qu'un perceptron multicouche standard. Même si les règles habituelles pour les taux d'apprentissage et des constantes de régularisation s'appliquent toujours, il faut prendre en considération les notions de nombre de filtres, leur forme et la forme du max pooling.

a) Nombre de filtres :

Comme la taille des images intermédiaires diminue avec la profondeur du traitement, les couches proches de l'entrée ont tendance à avoir moins de filtres tandis que les couches plus proches de la sortie peuvent en avoir davantage. Pour égaliser le calcul à chaque couche, le produit du nombre de caractéristiques et le nombre de pixels traités est généralement choisi pour être à peu près constant à travers les couches. Pour préserver l'information en entrée, il faudrait maintenir le nombre de sorties intermédiaires (nombre d'images intermédiaire multiplié par le nombre de positions de pixel) pour être croissante (au sens large) d'une couche à l'autre.

Le nombre d'images intermédiaires contrôle directement la puissance du système, dépend du nombre d'exemples disponibles et la complexité du traitement.

b) Forme de filtres :

Les formes de filtre varient grandement dans la littérature. Ils sont généralement choisis en fonction de l'ensemble de données. Les meilleurs résultats sur les images de MNIST (28x28) sont habituellement dans la gamme de 5x5 sur la première couche, tandis que les ensembles de données d'images naturelles (souvent avec des centaines de pixels dans chaque dimension) ont tendance à utiliser de plus grands filtres de première couche de 12x12, voire 15x15. Le défi est donc de trouver le bon niveau de granularité de manière à créer des abstractions à l'échelle appropriée et adaptée à chaque cas.

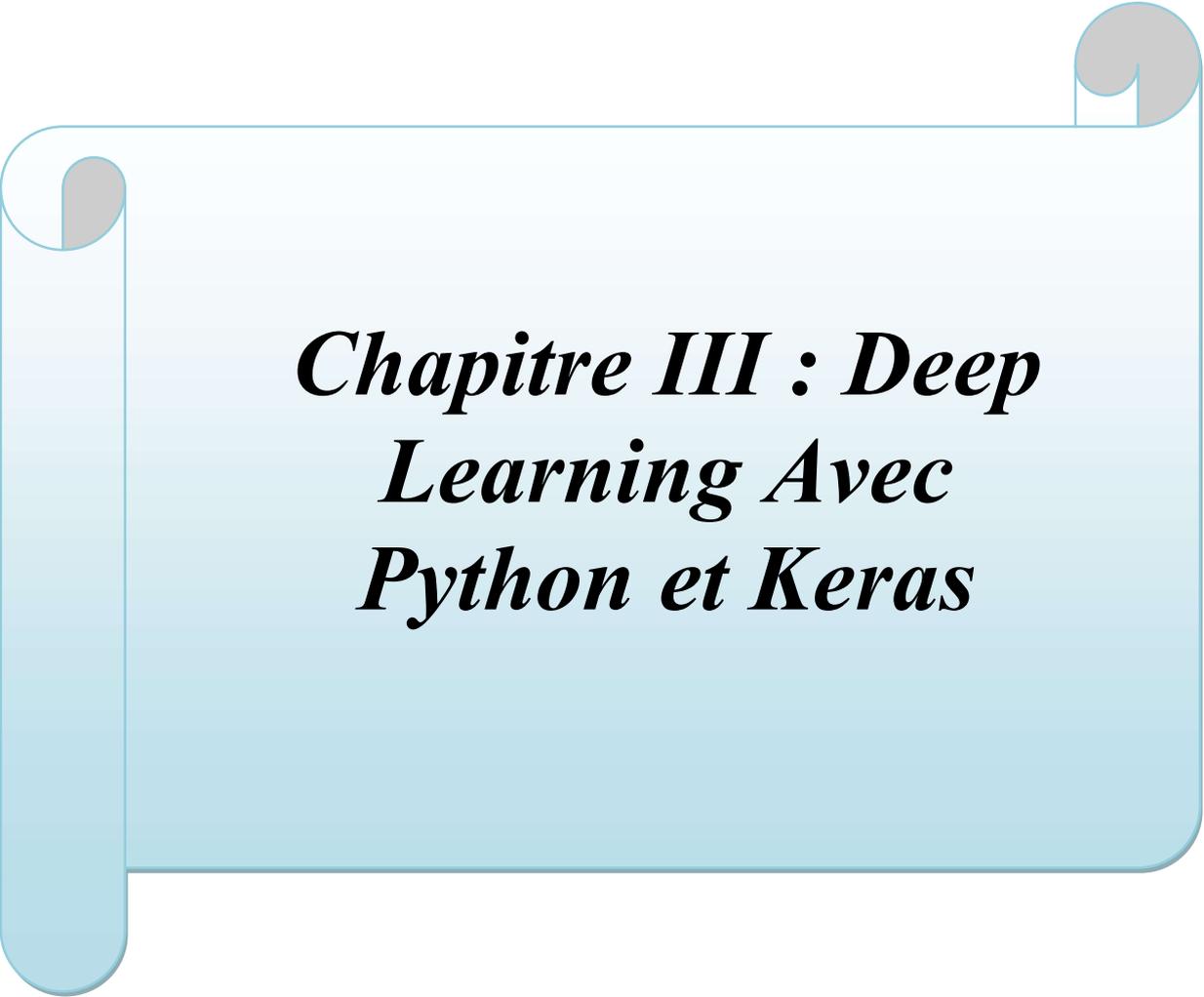
c) Forme de Max pooling :

Les valeurs typiques sont 2x2 (Figure II-15). De très grands volumes d'entrée peuvent justifier un pooling 4x4 dans les premières couches. Cependant, le choix de formes plus grandes va considérablement réduire la dimension du signal, et peut entraîner la perte de trop d'information.

2.5. Conclusion :

Dans ce chapitre on a présenté les notions importantes qui sont en relation avec l'apprentissage profond (Deep Learning). Notamment les réseaux de neurones artificiels ses types et topologies, et le CNN qui a pour avantage majeur d'utiliser peu de prétraitement. Cela signifie que le réseau est responsable de faire évoluer tout seul ses propres filtres (apprentissage sans supervision), ce qui n'est pas le cas d'autres algorithmes plus traditionnels.

Le prochain chapitre, traite la réalisation de l'algorithme sous Python, ainsi que les bibliothèques et les outils utilisés.



***Chapitre III : Deep
Learning Avec
Python et Keras***

3.1 Introduction

Pour implémenter notre modèle de Deep Learning pour la détection de la rétinopathie nous utilisons le langage de programmation Python et ses bibliothèques Keras et Tensorflow, d'abord on doit installer Python sur notre PC et le logiciel Pycharm qui permet de faciliter la programmation, ensuite on installe toutes les bibliothèques nécessaires, enfin on peut commencer la conception du modèle, la préparation des données et les fonctions.

Ce chapitre présente les différents logiciels et outils utilisés, les données, l'architecture du réseau CNN et les résultats obtenus.

3.2 Deep Learning et Python:

Bien qu'il ne soit pas le langage de programmation le plus rapide ni le plus sûr, Python est le plus utilisé dans le domaine de l'intelligence artificielle et surtout en deep learning.

Python contient un nombre important de bibliothèques performantes et utiles pour la vision artificielle, réseaux de neurones et pour la gestion des données, notamment Tensorflow et Theano pour le deep learning, Opencv pour la vision artificielle.

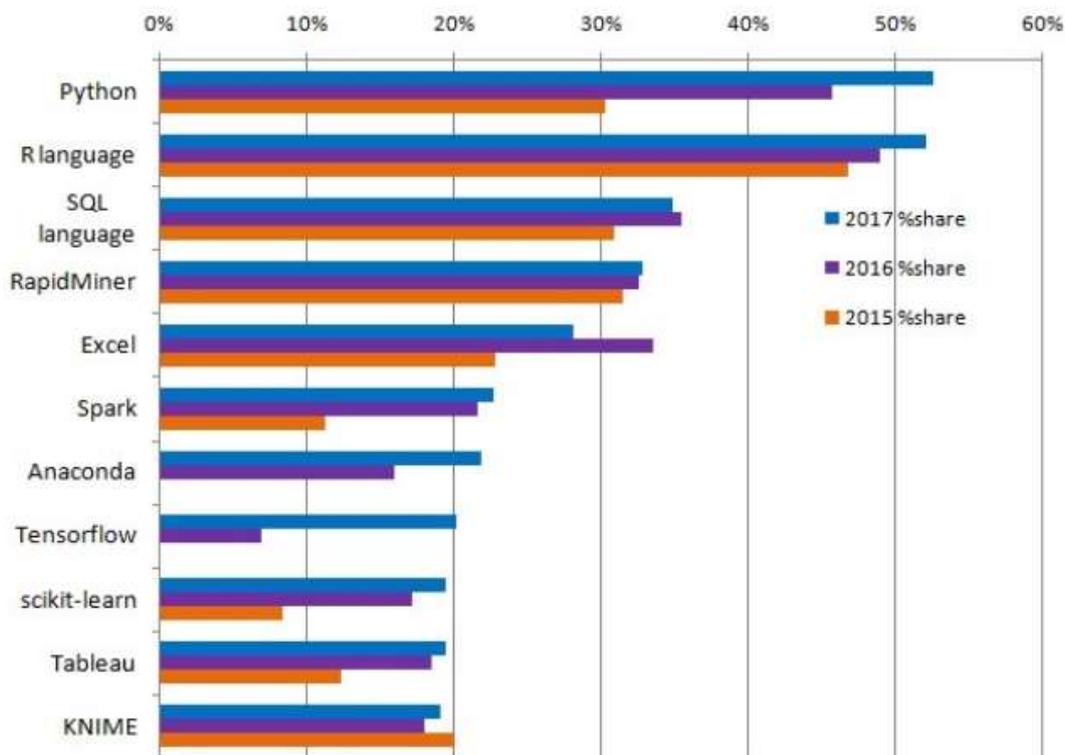


Figure III.1. Résultats du sondage des outils les plus utilisés en machine Learning par KDnuggets

3.3 Python sur Windows avec Pycharm :

Sur Windows, plusieurs options se présentent pour utiliser Python, parmi elles on distingue l'IDE Pycharm de JetBrains qui possède de nombreuses fonctionnalités dont l'analyse de

code complètement automatique, un débogueur Python intégré et navigation des projets et code pour ne citer qu'eux, pour l'installer, il suffit de le télécharger sur le site officiel et de le lancer, notons qu'il faut avoir une version récente de Python installé sur le système.

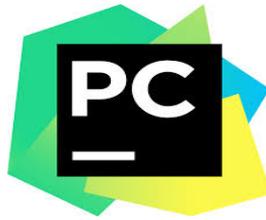


Figure III.2. Logo de pycharm

3.4 La bibliothèque Tensorflow:

Tensorflow est un logiciel bibliothèque open-source utilisé dans les applications des réseaux de neurones, il a été développé par l'équipe Google Brain de Google et le code a été ouvert en 2015.

Pour utiliser Tensorflow il est recommandé d'avoir un GPU NVIDIA® pour réduire le temps d'exécution énormément comparé au CPU (de quelques jours à quelque heures)(figIII.3).



Figure III.3. Logo des bibliothèques TensorFlow et Keras

3.5 La bibliothèque Keras :

Keras est une bibliothèque open source des réseaux de neurones écrite en Python et qui peut être exécuté avec Tensorflow, theano et d'autres bibliothèques, son auteur est François Chollet, un ingénieur de Google(figIII.3)

3.6Préparation des données :

On utilise une base de données téléchargée du site Kaggle(<https://www.kaggle.com/c/diabetic-retinopathy-contest/data>)avec 1276 images de 1.3 go divisé en deux parties pour les malades et non-malades, d'abord on utilise le logiciel JPEGcrops pour recadrer les images pour réduire les bordures noires, ensuite on utilise XnConvert pour transformer les images en niveau de gris, maintenant, on sépare les images en deux dossier, un pour l'entraînement et l'autre pour la validation, chacun possède ses

propre sous-dossiers pour malades et non malades, avec 868 images d'entraînement, et 408 pour la validation.

Les autres transformations se feront à l'exécution par ce que notre programme exige la transformation en tableau Numpy (une autre bibliothèque Python).

Le temps d'apprentissage dépendra du matériel de l'architecture, dans notre cas avec

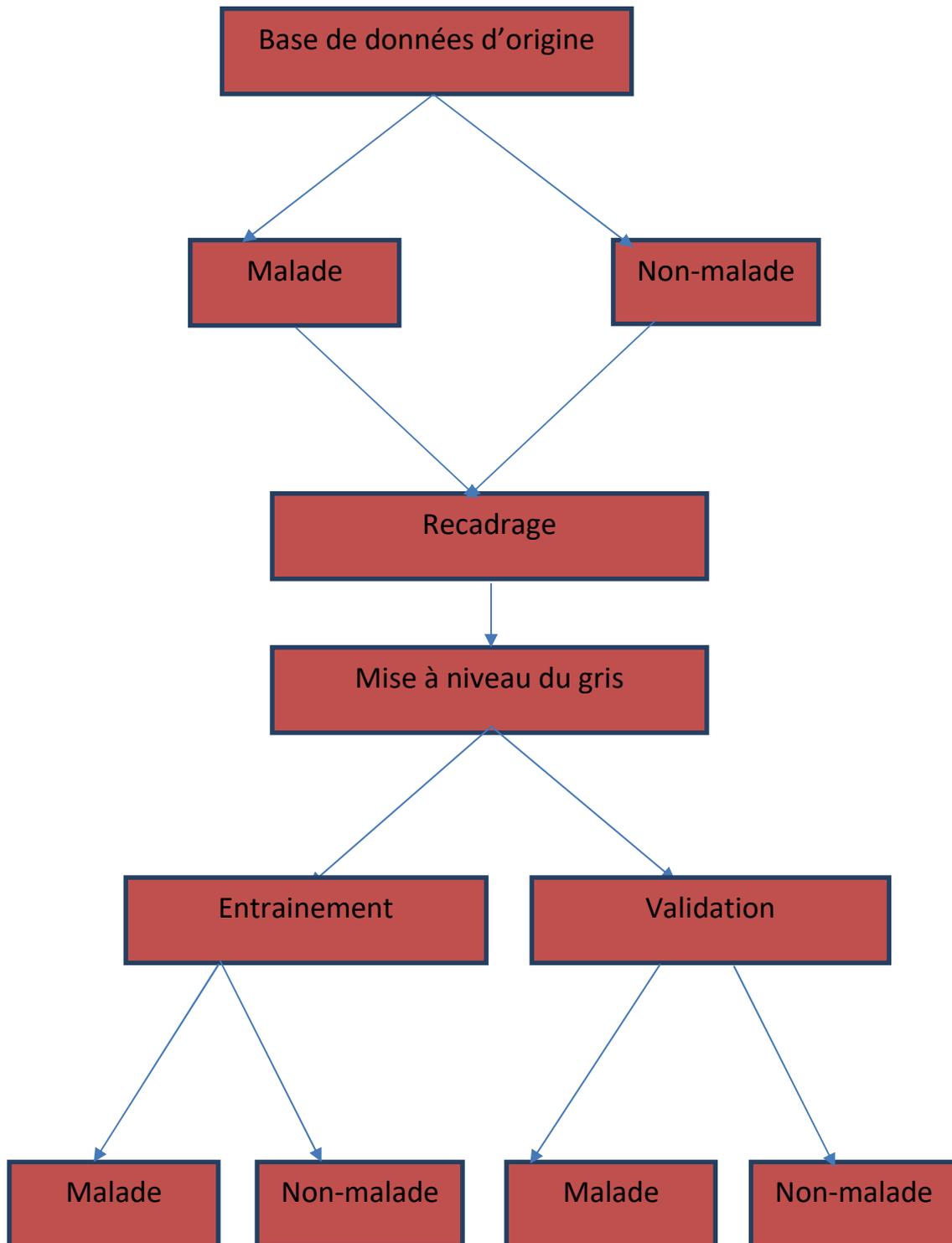


Figure III.4. Les étapes de la préparation des données.

3.7 Le réseau de neurone à convolution CNN utilisé :

Le réseau de neurone se compose de 25 couches, 22 du réseau VGG16 et 3 du classifieur. Le réseau VGG16 se compose de couches de convolution poursuivi par des couches MaxPooling, ensuite la couche Flatten transforme le résultat en une dimension qui passe des couches Dense et une couche Dropout et enfin la couche Dense finale qui a pour fonction d'activation la fonction sigmoid.

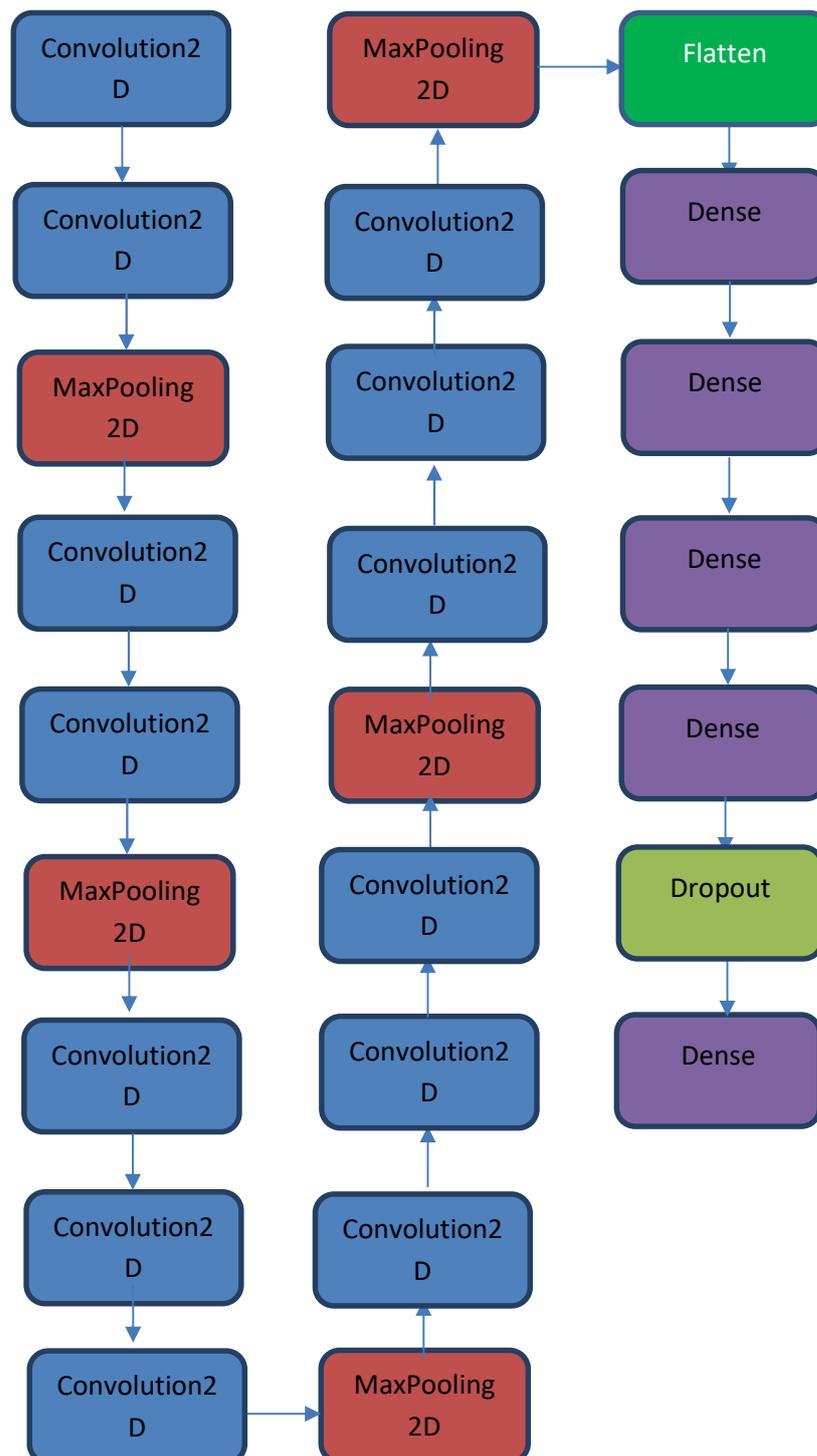


Figure III.5. L'architecture du réseau de neurone complet

pour la détection de rétinopathie sur une angiographie, image de la rétine.

3.8.1 Importation des bibliothèques (figIII.6) :

Tout d'abord on va importer les bibliothèques à utiliser :

- Image pour la préparation des données images.
- Models et Layers pour créer le réseau et utiliser les couches.
- ImageDataGenerator pour créer la boucle de parcours d'images.
- Adam le compilateur choisi pour ce réseau.
- matplotlib.pyplot pour tracer les courbes.
- Numpy pour manipuler les données et les matrices.

```
from keras.preprocessing import image
from keras import models
from keras import layers

from keras.preprocessing.image import ImageDataGenerator
from keras.optimizers import Adam

import matplotlib.pyplot as plt
import numpy as np
```

Figure III.6. L'importation des bibliothèques.

3.8.2 Appel du modèle VGG16 (figIII.7) :

On définit la base de notre réseau avec le modèle VGG16 qui prend comme paramètres les poids de la base de données imagenet, l'exclusion du haut de modèles (on prend que les zones d'intérêts), et le format des entrées.

```
conv_base = VGG16(weights='imagenet', include_top=False, input_shape=(256, 256, 3))
```

Figure III.7. L'appel du modèle VGG16.

3.8.3 Variables des chemins des dossiers (figIII.8) :

On déclare deux variables qui stockent l'emplacement des fichiers d'images comme chaîne de caractères, chaque variable pointe vers les dossiers d'entraînement et validation respectivement.

```
train = 'E:\\retinopathy data\\train'  
validation = 'E:\\retinopathy data\\validation'
```

Figure III.8. La déclaration des variables des chemins.

3.8.4 Construction du classeur (figIII.9) :

On construit notre classeur lié au réseau VGG16 en sortie, qui se compose de 3 couches :

- Une couche Dense avec 256 paramètres et la fonction d'activation relu avec la dimension d'entrée 4 * 4 * 512.
- Une couche Dropout fixée à 20%.
- Une couche Dense finale avec 1 paramètre pour donner un résultat binaire avec la fonction sigmoid entre 0 et 1.

On utilise `model.summary()` pour afficher le nombre de paramètres à l'exécution.

```
model = models.Sequential()
model.add(layers.Dense(256, activation='relu', input_dim=4 * 4 * 512))
model.add(layers.Dropout(0.2))
model.add(layers.Dense(1, activation='sigmoid'))

model.summary()
```

Figure III.9. La construction du classeur.

3.8.5 Définition des boucles d'images (fig III.10) :

On définit nos générateurs d'images qui sont des boucles infinies (que nous arrêtons au nombre d'images) qui vont retourner les images des dossiers, nous configurons la source et la taille des entrées, le mode des couleurs (niveau de gris), la fréquence de retour d'image et la classe (binaire).

```
datagen = ImageDataGenerator(rescale=1./255)
validation_generator, train_generator = datagen.flow_from_directory(
    train,
    target_size=(256, 256),
    color_mode='grayscale',
    batch_size=1,
    class_mode='binary')
```

Figure III.10. Le générateur d'images.

3.8.6 Extraction des zones d'intérêts (fig III.11) :

On crée une fonction qui va extraire les zones d'intérêts de notre réseau VGG16, cette fonction reçoit les images des dossiers d'entraînement et validation comme entrée, elle va ensuite créer deux matrices qui seront initialisées à 0, ces matrices recevront le résultat de la prédiction du réseau VGG16 limité par le nombre d'images

```
def extract_features(directory, sample_count):
    features = np.zeros(shape=(sample_count, 4, 4, 512))
    labels = np.zeros(shape=(sample_count))
    generator = datagen.flow_from_directory(
        directory, target_size=(150, 150), batch_size=batch_size, class_mode='binary')
    i = 0
    for inputs_batch, labels_batch in generator:
        features_batch = conv_base.predict(inputs_batch)
        features[i * batch_size : (i + 1) * batch_size] = features_batch
        labels[i * batch_size : (i + 1) * batch_size] = labels_batch
        i += 1
        if i * batch_size >= sample_count:
            break
    return features, labels
train_features, train_labels = extract_features(train, 868)
validation_features, validation_labels = extract_features(validation, 408)
```

Figure III.11. La fonction d'extraction des données.

3.8.7 Changement du type des zones d'intérêts (figIII.12) :

On change la forme des matrices issues de la fonction `extract_features` pour les envoyer vers la fonction d'entraînement du réseau.

```
train_features = np.reshape(train_features, (868, 4 * 4 * 512))
validation_features = np.reshape(validation_features, (408, 4 * 4 * 512))
#
```

Figure III.12. Figure montrant le changement du type des zones d'intérêts.

3.8.8 Configuration de l'optimisateur (figIII.13) :

Il existe plusieurs algorithmes d'optimisation qui permettent d'ajuster les poids du réseau, choisir le plus performant reste un des défis de conception.

On configure le compilateur à l'appel, on utilise l'algorithme Adam avec un taux d'apprentissage de 0.004, on a essayé plusieurs compilateurs avec plusieurs taux, celui la donne le meilleur résultat, on choisit la fonction de calcul de perte `binary_crossentropy` et la métrique de précision.

```
model.compile(optimizer=Adam(lr=.0004),
              loss='binary_crossentropy',
              metrics=['acc'])
```

Figure III.13. La Configuration de l'optimisateur.

3.8.9 Entraînement du modèle (figIII.14) :

On commence l'entraînement de notre modèle, on passe comme argument à la fonction fit les données d'entraînement et de validation, 50 itérations et des groupes de 80 images.

```
history = model.fit(train_features, train_labels,  
                    epochs=50,  
                    batch_size=80,  
                    validation_data=(validation_features, validation_labels))
```

Figure III.14. L'entraînement du modèle.

3.8.10 Traçage des courbes (figIII.15) :

Enfin, pour tracer les courbes de la précision et de perte d'entraînement et de validation on utilise les fonctions de la bibliothèque matplotlib.pyplot.

```
acc = history.history['acc']  
val_acc = history.history['val_acc']  
loss = history.history['loss']  
val_loss = history.history['val_loss']  
epochs = range(1, len(acc) + 1)  
plt.plot(epochs, acc, 'bo', label='Training acc')  
plt.plot(epochs, val_acc, 'b', label='Validation acc')  
plt.title('Training and validation accuracy')  
plt.legend()  
plt.figure()  
plt.plot(epochs, loss, 'bo', label='Training loss')  
plt.plot(epochs, val_loss, 'b', label='Validation loss')  
plt.title('Training and validation loss')  
plt.legend()  
plt.show()
```

Figure III.15. Le code pour le traçage des courbes.

3.9 Résultat final :

L'exécution du programme prend 15 minutes sur notre machine.

Nous avons essayé plusieurs paramètres et optimisations avec plusieurs résultats, voici quelques essais :

*En utilisant l'optimisateur RMSprop avec dropout plus important on obtient une précision de 84% d'entraînement et 67% de validation.

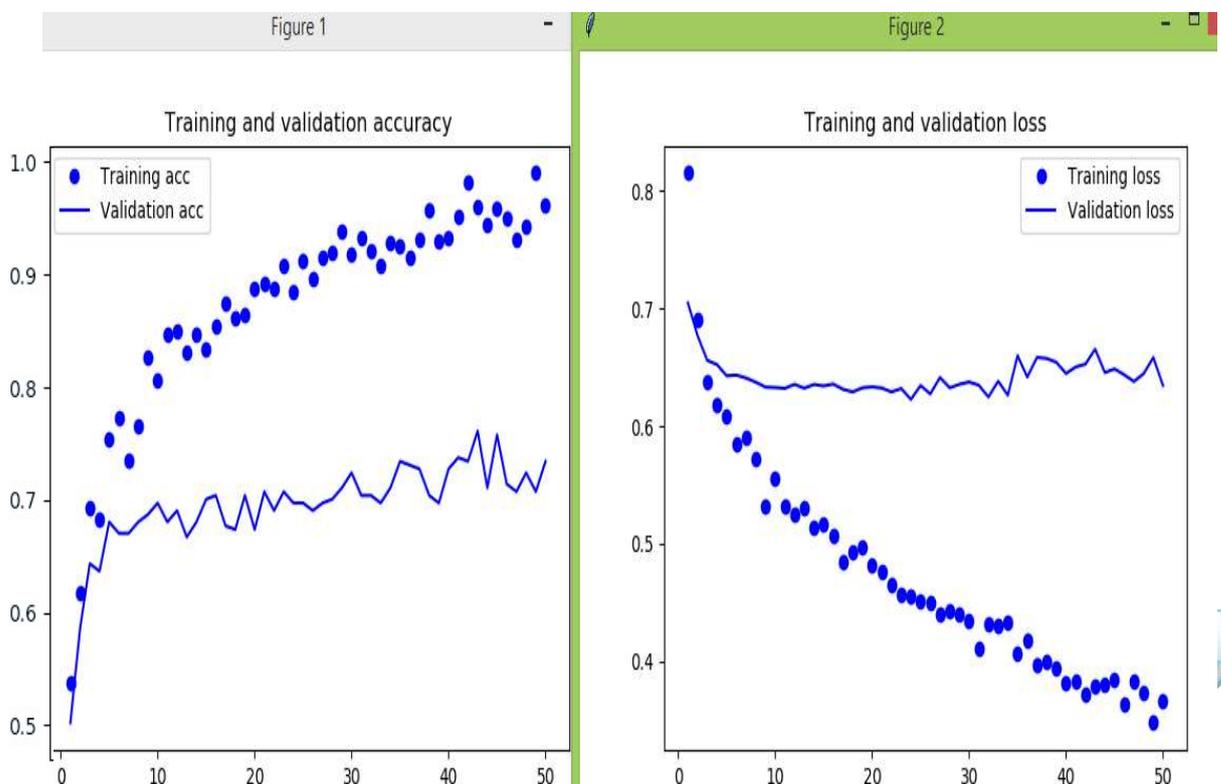


Figure III.16. Un premier résultat.

*En utilisant une couche Dense supplémentaire avec 128 paramètres on obtient
Une précision de 56% d'entraînement et 64% de validation.

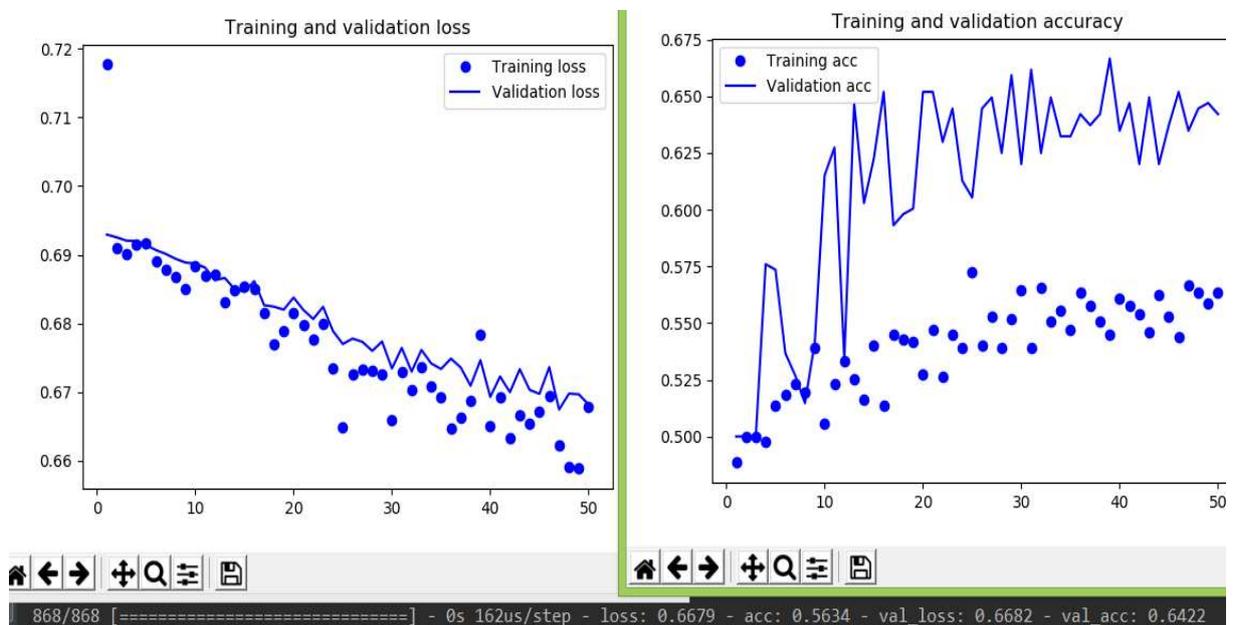
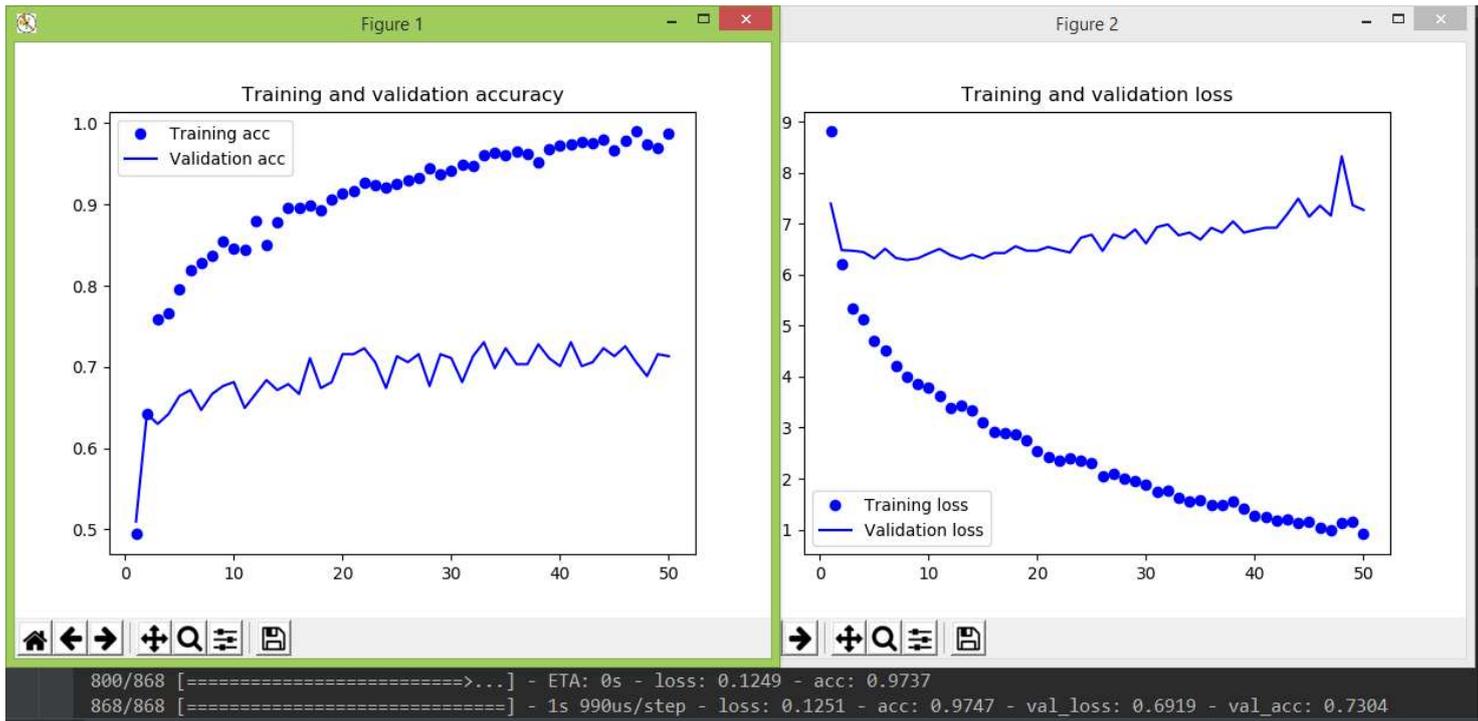


Figure III.17. Un deuxième résultat.

*On obtient une précision d'entraînement de 97% et de validation de 73% avec l'optimisateur Adam.



3.10 Conclusion:

Malgré l'utilisation de 5% de la taille des données recommandées et être limité par l'absence d'un GPU on a réussi à obtenir des résultats respectables en terme de précision, la prochaine étape sera de relier le programme entraîné à un appareil laser qui traite les images captées en temps réel et donner un résultat rapide et fiable qui montre si le patient est atteint de rétinopathie et qui montre les zones exactes où la maladie se propage, pour faciliter cela un PC robuste avec une quantité de RAM importante, un CPU puissant I7 et une GPU NVIDIA dernière génération seront nécessaires, ou on pourra ré-entraîner notre modèle sur une base de données plus grande et avec une architecture plus complexe.

Conclusion Générale

Les rétinopathies c'est les infections qui touchent la rétine pouvant entraîner une déficience visuelle. Elles sont généralement causées par d'autres pathologies telles le diabète ou l'hypertension. Les patients deviennent souvent symptomatiques en cas de dommages irréversibles. Seul un contrôle régulier chez l'ophtalmologue peut prévenir cela.

C'est dans ce cadre que s'inscrit notre travail, qui a pour objectif de concevoir un algorithme d'intelligence artificielle qui permet de détecter les rétinopathies.

Pour réaliser notre travail de détection de rétinopathies on a utilisé le deep learning, la méthode d'apprentissage qui a montré ses performances ces dernières années. Plus précisément le réseau de neurones à convolution (CNN), ce choix est justifié par l'efficacité de ce réseau dans la reconnaissance et la classification d'image.

Les paramètres du réseau sont difficiles à définir a priori. C'est pour cette raison que nous avons défini différents modèles avec des architectures différentes afin d'obtenir des meilleurs résultats en terme de précision et d'erreur.

Nous avons rencontré quelques problèmes dans la phase d'implémentation, l'utilisation d'un CPU a fait que le temps d'exécution était trop couteux. Afin de régler ce problème on doit utiliser des réseaux de neurones convolutionnels plus profonds déployé sur un GPU au lieu d'un CPU sur des bases plus importantes.

A travers ce projet nous avons fourni la partie détection pour le photocoagulateur laser embarqué, dont l'équipe Biorétine du laboratoire LATSI à l'université de Blida 1 vise à réaliser. Reste le photocoagulateur difficile mais faisable, ce qui est sur c'est qu'une fois réalisé il aura un impact capital dans le domaine médical.

Bibliographie

1. Sight Savers: The structure of the human eye. Dernier Accès, septembre 2018, Du site internet : <https://www.sightsavers.org/protecting-sight/the-eyes/>
2. Fleury Opticiens à Bulle : Anatomie de l'œil. Dernier Accès, septembre 2018, Du site internet : <https://www.fleury.ch/anatomie-de-l-oeil/anatomie-de-l-oeil>.
3. Le Guide De La Vue : Rétine, Dernier Accès, septembre 2018, Du site internet : <https://www.guide-vue.fr/glossaire/retine>.
4. SantéChezNous : Rétinopathie, Dernier Accès, septembre 2018, Du site internet : <https://santecheznous.com/condition/getcondition/retinopathie>.
5. Clinique Ophtalmologique (Institut Català de Retina) : Photo-coagulation au laser, Dernier Accès, septembre 2018, Du site internet : <https://icrcat.com/fr/traitements-et-tests-de-diagnostic/photocoagulation-au-laser/>
6. Institut de l'œil des Laurentides : Photo-coagulation au laser, Dernier Accès, septembre 2018, Du site internet : <http://www.institutdeloeil.com/traitement-de-loeil/traitement-de-la-retine/photocoagulation-au-laser.html>.
7. RÉSEAUX DE NEURONES. Dernier Accès, septembre 2018, Du site internet : <https://www.universalis.fr/encyclopedie/reseaux-de-neurones-formels/2-quelques-definitions/>
8. Réseaux de neurones : Qu'est-ce qu'un neurone ? Dernier Accès, septembre 2018, Du site internet : https://fr.wikiversity.org/wiki/R%C3%A9seaux_de_neurones/Qu%27est-ce_qu%27un_neurone_%3F.
9. Cours d'analyse numérique : réseaux de neurones formels. Dernier Accès, septembre 2018, Du site internet : <http://informatique.coursgratuits.net/methodes-numeriques/reseaux-de-neurones-formels.php>.
10. LES RESEAUX DE NEURONES ARTIFICIELS, INTRODUCTION AU CONNEXIONNISME Claude Touzet. Dernier Accès, septembre 2018, Du site internet : <http://www.touzet.org/Claude/Web-Fac-Claude/enseignement/RNA/>.
11. François Chollet: 'Deep Learning with Python', Manning, 2018.
12. Antonio Gulli et Sujit Pal : 'Deep Learning with Keras', Packt Publishing, 2017.

Le code Python complet :

```
from keras.preprocessing import image
from keras import models
from keras import layers
from keras.preprocessing.image import ImageDataGenerator
from keras.optimizers import Adam
import matplotlib.pyplot as plt
import numpy as np
from keras.applications import VGG16

conv_base = VGG16(weights='imagenet', include_top=False, input_shape=(150, 150,3))

train = 'E:\\retinopathy data\\train'
validation = 'E:\\retinopathy data\\validation'

model = models.Sequential()
model.add(layers.Dense(256, activation='relu', input_dim=4 * 4 * 512))
model.add(layers.Dropout(0.2))
model.add(layers.Dense(1, activation='sigmoid'))
model.summary()

datagen = ImageDataGenerator(rescale=1./255)
train_generator = datagen.flow_from_directory(
    train,
    target_size=(150, 150),
    color_mode='grayscale',
    batch_size=1,
    class_mode='binary')
validation_generator = datagen.flow_from_directory(
    train,
    target_size=(150, 150),
    color_mode='grayscale',
    batch_size=1,
    class_mode='binary')
    batch_size = 10

def extract_features(directory, sample_count):
    features = np.zeros(shape=(sample_count, 4, 4, 512))
    labels = np.zeros(shape=(sample_count))
    generator = datagen.flow_from_directory(
    directory,target_size=(150, 150),batch_size=batch_size,class_mode='binary')
    i = 0
    for inputs_batch, labels_batch in generator:
        features_batch = conv_base.predict(inputs_batch)
        features[i * batch_size : (i + 1) * batch_size] = features_batch
        labels[i * batch_size : (i + 1) * batch_size] = labels_batch
        i += 1
    if i * batch_size >= sample_count:
        break
    return features, labels
```

Annexes

```
train_features, train_labels = extract_features(train, 868)
validation_features, validation_labels = extract_features(validation, 408)
```

```
train_features = np.reshape(train_features, (868, 4 * 4 * 512))
validation_features = np.reshape(validation_features, (408, 4 * 4 * 512))
```

```
model.compile(optimizer=Adam(lr=.0004),
              loss='binary_crossentropy',
              metrics=['acc'])
history = model.fit(train_features, train_labels,
                   epochs=50,
                   batch_size=80,
                   validation_data=(validation_features, validation_labels))
```

```
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(acc) + 1)
plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()
plt.figure()
plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()
plt.show()
```