



Université A. Mira de Béjaïa  
Faculté des Sciences Exactes  
Département d'Informatique

## *MÉMOIRE DE MASTER RECHERCHE*

Option : Intelligence artificielle

### Thème

---

Application d'aide au diagnostic de la rétinopathie diabétique par  
réseau de neurones convolutifs en utilisant l'apprentissage par  
transfert

---

Réalisé par *M<sup>elles</sup>* :

KEBIR Yasmine

MAAKNI Katia

Devant le jury composé de :

Présidente : *M<sup>me</sup>* ADEL Karima

*Encadrante* : *M<sup>me</sup>* BOUKREDERA Djamila

Examinatrice : *M<sup>me</sup>* EL BOUHISSI Houda

*Co-encadrante* : *M<sup>me</sup>* AIT KACI AZZOU Samira

## *Remerciements*

Nos remerciements vont à notre encadrant Mme Boukredera et Co-encadrant Mme Ait Kaci Azzou pour la préciosité de leurs conseils, leurs orientation, ainsi que le temps-consacré et qui ont constitué un apport considérable grâce auquel ce travail a pu être mené à bon port.

Nous tenons à remercier chacun des membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Nos remerciements s'étendent à tous nos enseignants et les membres du département-d'Informatique de l'université ABDERRAHMANE MIRA. Ainsi qu'à tous ceux et celles-qui ont contribué de près ou de loin à l'accomplissement de ce travail.

---

## *Dedicaces*

Je voudrais avant tout exprimer ma reconnaissance envers mes très chers parents, qui ont toujours été là pour moi. Je remercie mes frères, et ma soeur pour leur soutien constant et leurs encouragements.

Je remercie mes amis qui ont toujours été là pour moi. Leur soutien inconditionnel et leurs encouragements ont été d'une grande aide.

J'aimerais exprimer ma gratitude à tous ceux qui sont intervenus de près ou de loin, qui ont pris le temps de discuter de mon sujet. Chacun de ces échanges m'a aidé à faire avancer mon projet.

MAAKNI Katia

---

## *Dedicaces*

Je tiens en premier lieu à exprimer ma gratitude et mon amour envers mes parents, sans qui rien de tout ça n'aurait pu aboutir. Votre soutien m'a été d'une aide incroyable et je vous en remercie jamais assez ainsi que pour tous vos sacrifices afin que je puisse avoir ce que j'ai aujourd'hui.

Je dois cette réussite aussi à mes chères grands-mères qui m'ont toujours soutenu et cru en moi, je vous aime inconditionnellement.

Un hommage à mes chers grands-pères qui nous ont malheureusement quittés mais qui j'espère sont fiers de moi de là où ils sont. Vous êtes ma force et c'est grâce à vous que je continue de me battre.

Un énorme merci à toute ma famille, mes frères et ma soeur, ainsi que mes tantes et oncles, cousins et cousines. Je vous aime tous.

Et enfin, je remercie tous mes amis pour leur contribution de près ou de loin à cette réussite, votre soutien m'a été d'une énorme aide.

KEBIR Yasmine

# Table des matières

<b>Table des matières</b>	<b>vi</b>
<b>Liste des figures</b>	<b>viii</b>
<b>Liste des tableaux</b>	<b>ix</b>
<b>Liste des acronymes et des abréviations</b>	<b>x</b>
<b>1 Rétinopathie diabétique (RD)</b>	<b>5</b>
1.1 L'anatomie de l'oeil . . . . .	6
1.2 La cause de la Rétinopathie Diabétique . . . . .	6
1.3 Les stades de la rétinopathie diabétique . . . . .	7
<b>2 Traitement d'images</b>	<b>9</b>
2.1 Définition de l'image numérique . . . . .	10
2.2 Types d'image . . . . .	10
2.2.1 Image binaire . . . . .	10
2.2.2 Image en niveaux de gris . . . . .	10
2.2.3 Image RVB . . . . .	11
2.3 Caractéristiques d'une image . . . . .	11
2.3.1 Dimension . . . . .	11
2.3.2 Résolution . . . . .	11
2.3.3 Bruit d'image . . . . .	12
2.3.4 Histogramme . . . . .	12
2.3.5 Luminance . . . . .	12
2.3.6 Contraste . . . . .	12
2.4 Imagerie médicale . . . . .	13

2.4.1	Acquisition d'images médicales . . . . .	13
2.4.2	Prétraitement d'image . . . . .	13
2.4.3	Segmentation . . . . .	18
2.4.4	Extraction des caractéristiques . . . . .	19
2.4.5	La Reconnaissance . . . . .	19
<b>3</b>	<b>Apprentissage Automatique et Apprentissage Profond</b>	<b>21</b>
3.1	Les Réseaux de Neurones . . . . .	22
3.1.1	Definition . . . . .	22
3.1.2	Apprentissage . . . . .	23
3.1.3	Fonctions d'activation . . . . .	24
3.1.4	Tests . . . . .	26
3.2	Apprentissage Automatique : . . . . .	27
3.2.1	Techniques de pré-traitement . . . . .	27
3.2.2	Caractéristiques utilisées . . . . .	27
3.2.3	classification en apprentissage automatique . . . . .	29
3.2.4	Mesures de performances . . . . .	29
3.2.5	Problèmes rencontrés en apprentissage automatique . . . . .	30
3.3	Apprentissage Profond . . . . .	31
3.3.1	Differentes architectures de l'apprentissage profond . . . . .	32
3.3.2	Comparaison entre apprentissage automatique et apprentissage pro- fond en traitement d'images . . . . .	35
3.4	Méthodes de détection de la RD . . . . .	36
3.4.1	L'approche d'apprentissage automatique . . . . .	36
3.4.2	L'approche d'apprentissage profond . . . . .	38
3.4.3	Approches d'apprentissage profond utilisant l'apprentissage par trans- fert . . . . .	39
3.4.4	Approches d'apprentissage profond utilisant de nouvelles architec- tures d'apprentissage profond . . . . .	40
3.4.5	L'approche d'apprentissage automatique et d'apprentissage profond	41
<b>4</b>	<b>Conception et Réalisation</b>	<b>45</b>
4.1	Conception du modèle . . . . .	46

4.1.1	CNN Architecture . . . . .	46
4.1.2	Modélisation du réseau de neurones . . . . .	47
4.1.3	Modélisation du VGG16 pour la classification de la RD . . . . .	48
4.2	Implémentation . . . . .	50
4.2.1	Importer VGG16 . . . . .	50
4.2.2	Adapter VGG16 . . . . .	51
4.2.3	Choix de la base de données . . . . .	52
4.3	Entraînement . . . . .	55
4.3.1	Propagation vers l'avant . . . . .	55
4.3.2	Descente de Gradient . . . . .	56
4.3.3	Fonctions d'activation . . . . .	57
4.3.4	La fonction de perte . . . . .	57
4.3.5	Propagation vers l'arrière . . . . .	58
4.4	Evaluation . . . . .	58
4.4.1	Résultats d'apprentissage . . . . .	60
4.5	Tests et Observations . . . . .	61
4.6	Présentation des interfaces du système . . . . .	63
4.6.1	Definition : . . . . .	69
4.6.2	Utilisation : . . . . .	69
	<b>Conclusion générale</b>	<b>71</b>
	<b>Bibliographie</b>	<b>71</b>

# Table des figures

1.1	Description de l'anatomie de l'oeil [28]. . . . .	6
1.2	Microanévrismes [3] . . . . .	7
1.3	Rétinopathie non proliférante modérée [3] . . . . .	7
1.4	Rétinopathie non proliférante sévère [3] . . . . .	8
1.5	Rétinopathie proliférante [38] . . . . .	8
2.1	La transformation d'une image couleur en une image en niveau de gris [20] . . . . .	14
2.2	Espace couleur RVB [20] . . . . .	14
2.3	Résultat du filtrage de l'image améliorée [20]. . . . .	16
2.4	Représentation d'histogramme du canal rouge d'une image rétinienne [20]. . . . .	17
2.5	Egalisation d'histogramme appliqué sur l'image rétinienne [20]. . . . .	18
3.1	Architecture d'un réseau de neurones [39] . . . . .	23
3.2	Hyperbolic Tangent Function [9]. . . . .	25
3.3	Fonction ReLU [9] . . . . .	25
3.4	Fonction Sigmoid [9] . . . . .	26
3.5	Schéma représentant le type et quelques algorithmes de classification en ML . . . . .	29
3.6	Architecture du DNN [52] . . . . .	33
3.7	Diagramme du RNN [33] . . . . .	34
3.8	Exemple d'architecture CNN [36]. . . . .	35
3.9	Méthodes existantes pour la détection de la rétinopathie diabétique RD . . . . .	36
3.10	Modèle basé sur l'apprentissage automatique [2] . . . . .	37
3.11	Modèle basé sur l'apprentissage profond [2] . . . . .	38
4.1	Architecture d'un CNN [25]. . . . .	47
4.2	Architecture de VGG16[23] . . . . .	48

4.3	Résumé du modèle VGG-16 . . . . .	50
4.4	Importer VGG16 . . . . .	51
4.5	Adapter VGG-16 . . . . .	51
4.6	Adapter VGG-16 -2- . . . . .	52
4.7	Structure de la base de données . . . . .	53
4.8	NO_DR . . . . .	53
4.9	Rétinopathie non poliférante minime . . . . .	54
4.10	Rétinopathie non poliférante modérée . . . . .	54
4.11	Rétinopathie non poliférante sévère . . . . .	54
4.12	Rétinopathie poliférante . . . . .	55
4.13	Code permettant de relier la base de données au programme . . . . .	56
4.14	Figure illustrant le résultat obtenu pour l'apprentissage avec la première méthode . . . . .	59
4.15	Processus du fine tuning partiel sur le modèle utilisé . . . . .	60
4.16	Resultats de test pour DR Proliférante . . . . .	62
4.17	Resultats de test pour DR Modérée . . . . .	62
4.18	Resultats de test pour DR Sévère . . . . .	63
4.19	Interface d'accueil . . . . .	64
4.20	Interface pour sélectionner une image de la base de données . . . . .	65
4.21	Résultat d'une NO-DR . . . . .	66
4.22	Interface de la plateforme ANACONDA . . . . .	67
4.23	L'interface d'accueil de Jupyter Notebook . . . . .	68

# Liste des tableaux

3.1	Tableau comparatif entre Apprentissage Automatique et Apprentissage Profond [35]. . . . .	35
3.2	Tableau représentatif de l'état de l'art de quelques techniques de détection des symptômes de la R.D. . . . .	43
4.1	Résumé des résultats d'apprentissage sur le modèle . . . . .	61

# Liste des acronymes et des abréviations

<b>Acc</b>	<b>A</b> ccuracy
<b>ANN</b>	<b>A</b> rtificial <b>N</b> eural <b>N</b> etwork
<b>API</b>	<b>A</b> pplication <b>P</b> rogramming <b>I</b> nterface
<b>AUC</b>	<b>A</b> rea <b>U</b> nder the <b>C</b> urve
<b>CNN</b>	<b>C</b> onvolutional <b>N</b> eural <b>N</b> etwork
<b>DL</b>	<b>D</b> eep <b>L</b> earning
<b>DNN</b>	<b>D</b> eep <b>N</b> eural <b>N</b> etwork
<b>GRNN</b>	<b>G</b> eneralized <b>R</b> egression <b>N</b> eural <b>N</b> etwork
<b>HSV</b>	<b>H</b> ue <b>S</b> aturation <b>V</b> alue 2
<b>IA</b>	<b>I</b> ntelligence <b>A</b> rtificielle
<b>MFV</b>	<b>M</b> ethod of <b>F</b> irst <b>V</b> iolation
<b>ML</b>	<b>M</b> achine <b>L</b> earning
<b>NG</b>	<b>N</b> iveau de <b>G</b> ris
<b>RD</b>	<b>R</b> étinopathie <b>D</b> iabétique
<b>RNN</b>	<b>R</b> eurrent <b>N</b> eural <b>N</b> etwork
<b>RVB</b>	<b>R</b> ouge <b>V</b> ert <b>B</b> leu 2
<b>SE</b>	<b>S</b> ensibilité
<b>SGD</b>	<b>S</b> tochastic <b>G</b> radient <b>D</b> escent
<b>SP</b>	<b>S</b> pécificité
<b>VGG</b>	<b>V</b> isual <b>G</b> eometry <b>G</b> roup

# Résumé

La rétinopathie diabétique (RD) est l'une des principales causes de cécité dans le monde chez les diabétique âgés de 20 à 64 ans. Des examens ophtalmologiques réguliers sont d'une haute importance pour diminuer la fréquence de cette maladie. Ainsi, nous pouvons comprendre l'importance de l'efficacité du diagnostique pour la détection et la prévention de cette pathologie. Ces dernières années, de nombreuses recherches visent à améliorer la qualité de l'interprétation de l'image médicale grâce à l'intelligence artificielle (IA) [1].

L'IA et l'apprentissage automatique ainsi que l'apprentissage profond, ont transformé le secteur des soins de santé dans de nombreux domaines, notamment le diagnostic des maladies par imagerie médicale [1].

Dans notre projet, nous allons donc travailler sur la détection de la rétinopathie diabétique (RD) en utilisant un réseau neuronal convolutif avec la méthode d'apprentissage par transfert. Ça consiste à prendre des images pré-traitées du fond de l'oeil comme entrées du réseau de neurones afin de pouvoir faire une extraction automatique des caractéristiques. On aura de ce fait en sortie, une classification par rapport aux types de la maladie (no-DR, minime, modéré, sévère, et proliférante). Le but est de réaliser un réseau de neurones qui constituera un système d'aide au diagnostic efficace.

# Abstract

Diabetic retinopathy (DR) is one of the leading causes of blindness worldwide in diabetics aged between 20 to 64 years. Regular ophthalmological examinations are of high importance to decrease the frequency of this disease. Thus, we can understand the importance of effective diagnosis for the detection and prevention of this disease. In recent years, a lot of research aimed at improving the quality of medical image interpretation through artificial intelligence (AI) [1].

AI and machine learning as well as deep learning have transformed the healthcare sector in many areas, including the diagnosis of diseases through medical imaging [1].

In our project, we will therefore work on DR detection using a convolutional neural network with the transfer learning method. This consists of taking pre-processed fundus images as inputs to the neural network in order to perform automatic feature extraction. The output will be a classification according to the type of disease (no-DR, minimal, moderate, severe, and proliferate). The aim is to create a neural network that will constitute an effective diagnostic aid system.

# Introduction Générale

La rétinopathie diabétique (RD) ou la maladie oculaire diabétique est une condition médicale dans laquelle les dommages se produisent à la rétine en raison du diabète causant la cécité [8]. La détection de la RD à un stade précoce est très importante pour sauver la vision du patient. De nombreux chercheurs ont montré que 90% des patients diabétiques peuvent être sauvés de la RD grâce à un diagnostic précoce. Ce dernier peut être effectué soit manuellement par un ophtalmologiste, soit par un système automatisé. Ces deux méthodes de détection de la RD présentent des avantages et des inconvénients. Le seul avantage de la détection manuelle est qu'elle ne nécessite aucune assistance informatique pour le processus de détection de la RD, alors qu'elle exige que l'ophtalmologiste soit un expert en la matière. Parfois, les signes de la RD dans sa phase initiale sont si petits que même un ophtalmologiste expert ne peut les reconnaître correctement [2].

Grâce aux progrès de l'intelligence artificielle (IA), la détection précoce de la maladie par un système automatisé est plus probable et présente plus d'avantages que le système manuel de détection de la RD. Ces avantages peuvent inclure une diminution de la charge de travail de l'ophtalmologiste, une réduction du risque d'erreur humaine. De plus, un système automatisé peut être capable de détecter des lésions et des anomalies avec beaucoup plus de facilité et d'efficacité que ce qui est possible manuellement, ainsi que la mise en évidence précoces de lésions rétinienne minimes ne pouvant être détectées par les examens de routine chez un ophtalmologue et donc cette dernière permet une prise en charge optimale et efficace précocement à ce stade réversible de la maladie. L'automatisation de la détection de la RD est donc essentielle. Les systèmes automatisés de RD peuvent être développés soit par des approches d'apprentissage automatique, soit par des approches d'apprentissage profond [2]. Pour notre projet, nous avons choisi les approches d'apprentissage profond.

L'objectif principal de notre travail est d'utiliser l'apprentissage profond, plus précisément l'apprentissage par transfert pour détecter les signes cliniques produits par la RD. Ces lésions sont : les exsudats, les micro-anévrismes et les hémorragies. Ces complications peuvent aboutir à des retentissements pouvant aller d'un flou visuel à une cécité irréversible si elles ne sont pas diagnostiquées à un stade précoce. Depuis une dizaine d'années, l'utilisation des réseaux de neurones s'est développée dans de nombreuses disciplines, notamment en médecine, et son efficacité a été démontrée à maintes reprises.

Notre projet consiste, en premier lieu, à implémenter l'architecture VGG16, puis l'adapter à notre problème pour que grâce à ça, l'extraction des caractéristiques des images d'entrées se fasse automatiquement, pour ensuite les utiliser comme entrées du réseau de neurones entièrement connecté. On aura, de ce fait, en sortie une prédiction du stade de la rétinopathie diabétique. Le système décrit dans ce travail est conçu pour aider les ophtalmologues à établir un diagnostic en lui fournissant un deuxième avis et fonctionne également comme un outil pour le dépistage de la rétinopathie diabétique.

De ce fait, le mémoire sera décomposé en quatre chapitres, représentant les différents aspects de ce travail. Le premier chapitre expliquera la maladie de la Rétinopathie Diabétique et ses différents stades. Le second, abordera les aspects du traitement d'images. Dans le troisième chapitre, nous présenterons les méthodes utilisées dans la littérature pour la détection de la RD. Le dernier chapitre fera l'objet de la conception et des résultats obtenus lors de l'expérimentation, ainsi que l'environnement et les outils utilisés pour la mise en œuvre de notre système.

# Chapitre 1

## Rétinopathie diabétique (RD)

### Sommaire

---

1.1	L'anatomie de l'oeil . . . . .	6
1.2	La cause de la Rétinopathie Diabétique . . . . .	6
1.3	Les stades de la rétinopathie diabétique . . . . .	7

---

## 1.1 L'anatomie de l'oeil

La rétine est une structure sphérique présente à l'intérieur, à l'arrière de l'œil. Sa fonction est de traiter les informations visuelles par l'intermédiaire des photorécepteurs présents dans l'œil, appelés bâtonnets et cônes. Au centre de la rétine se trouve une zone ronde et sombre appelée macula. Le point central de la macula appelé fovéa, est chargé de fournir une vision nette. Comme tout autre tissu de l'organisme, le tissu rétinien est alimenté en sang par le système vasculaire du corps. Le disque optique, composé de la cupule optique est une structure où le nerf optique quitte l'œil. C'est également le site où toutes les principales artères et veines entrent et sortent de l'œil [2].

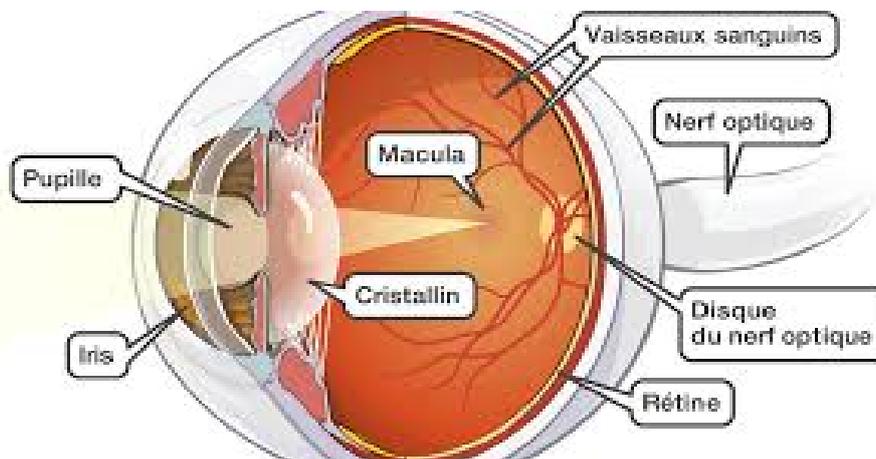


FIGURE 1.1 – Description de l'anatomie de l'œil [28].

## 1.2 La cause de la Rétinopathie Diabétique

Le sang est transporté vers les couches du tissu rétinien par des micro-vaisseaux sanguins qui nécessitent un taux de sucre sanguin maintenu et un flux sanguin non obstrué. Si de grandes quantités de sucres, tels que le glucose ou le fructose s'accumulent dans le sang, ces vaisseaux sanguins commencent à se détériorer en raison d'une mauvaise distribution de l'oxygène aux cellules. Des anomalies structurelles se produisent à la suite d'une diminution du taux métabolique, ce qui entraîne la RD [2].

### 1.3 Les stades de la rétinopathie diabétique

- **Rétinopathie non proliférante minime** Le signe principal de ce stade est la présence de microanévrismes.

La figure (1.2) montre un microanévrisme, et donc signe d'une Rétinopathie non proliférante minime.

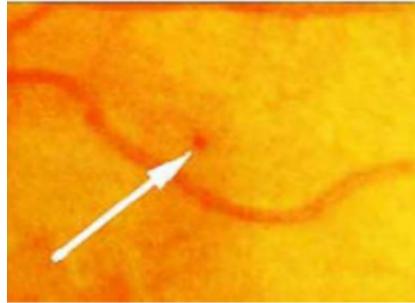


FIGURE 1.2 – Microanévrismes [3]

- **Rétinopathie non proliférante modérée**

Les symptômes qui caractérisent cette classe sont : les microanévrismes, exsudats secs, nodules cotonneux (« nodules dysoriques ») et hémorragies rétinienne punctiformes.

La figure (1.3) montre la présence de microanévrismes et d'hémorragies de petite taille en nombre modéré, ainsi que de rares nodules cotonneux.

FIGURE 1.3 – Rétinopathie non proliférante modérée [3]

- **Rétinopathie non proliférante sévère (rétinopathie préproliférante) :**

Pour ce stade, les signes peuvent être des Hémorragies intrarétiniennes sévères et microanévrismes, ou bien un perlage veineux défini dans deux quadrants ou plus, ou des Hémorragies intrarétiniennes étendues, volumineuses ou en tache.

La figure (1.4) montre les signes énoncés précédemment et qui caractérisent la Rétinopathie non proliférante sévère.

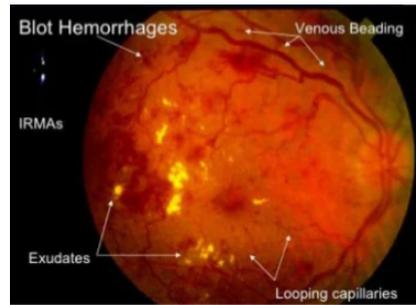


FIGURE 1.4 – Rétinopathie non proliférante sévère [3]

### — Rétinopathie proliférante

Ce stade de la RD se caractérise par la néovascularisation, Hémorragie vitréenne ou prérétinienne et le décollement de rétine par traction.[4,5]

La figure (1.5) montre l'anatomie de l'oeil, en rouge c'est les signes de la rétinopathie proliférante.

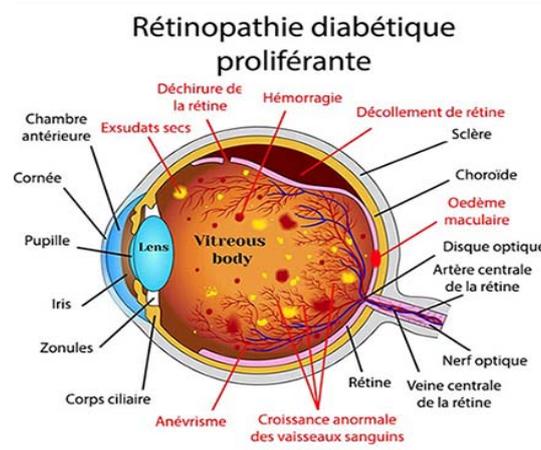


FIGURE 1.5 – Rétinopathie proliférante [38]

## Conclusion

Dans ce chapitre, nous avons présenté et expliqué ce qu'est la rétinopathie diabétique, son origine ainsi que ses stades et signes cliniques. L'objectif principal de ce travail est de développer un système d'analyse des images numériques du fond d'oeil qui sera capable de détecter la présence ainsi que les premiers signes cliniques produits par la RD. C'est pourquoi, le chapitre suivant va nous donner un aperçu sur comment traiter l'image afin d'extraire les paramètres nécessaires à la reconnaissance des anomalies rétinienne.

# Chapitre 2

## Traitement d'images

### Sommaire

---

2.1	Définition de l'image numérique . . . . .	10
2.2	Types d'image . . . . .	10
2.3	Caractéristiques d'une image . . . . .	11
2.4	Imagerie médicale . . . . .	13

---

## Introduction

Le traitement de l'image est un domaine très vaste qui a connu et continue de connaître un développement important. Cependant, le traitement d'image numérique spécifie toutes les techniques susceptibles de modifier les images numériques pour les améliorer ou en extraire des informations [9]. Dans ce chapitre, nous allons définir l'image ainsi que ses caractéristiques et types, puis nous allons nous introduire dans le traitement d'imagerie médicale qui sont le type d'images que nous allons traiter tout au long de notre travail.

### 2.1 Définition de l'image numérique

Une image numérique est une image composée de pixels. L'intensité ou niveau gris de chaque élément d'image a une représentation numérique finie et discrète. Le terme image numérique désigne généralement les images matricielles ou bitmaps (par opposition aux images vectorielles) [9].

### 2.2 Types d'image

En général, il existe 3 types d'images numériques :

#### 2.2.1 Image binaire

L'image binaire est une image en noir et blanc avec des valeurs de pixels prenant 0 ou 1. Ce type d'image est rarement utilisé et est souvent appelé "bit map image". Les images binaires sont utilisés comme masques pour indiquer les pixels d'intérêt dans de nombreuses tâches de traitement d'image [9].

#### 2.2.2 Image en niveaux de gris

En photographie numérique, imagerie et colorimétrie générées par ordinateur, les images en niveaux de gris sont des images dans lesquelles la valeur de chaque pixel est un échantillon unique qui représente uniquement la quantité de lumière. En d'autres termes, il ne porte que des informations d'intensité. Images en niveaux de gris (un monochroma-

tique noir et blanc ou un gris) sont constitués de niveaux de gris seulement. Le contraste se situe entre la plus faible intensité de noir à la plus forte intensité de blanc [9].

### 2.2.3 Image RVB

RVB (Rouge, Vert, Bleu) se réfère à trois nuances qui peuvent être mélangées pour créer des couleurs différentes. La combinaison de lumière rouge, verte et bleue est la méthode standard pour générer des images couleur sur les écrans comme les téléviseurs, les écrans d'ordinateur et les écrans de téléphone intelligent. Le modèle de couleur RVB est un modèle "additif". Lorsque chaque couleur est mélangée à 100%, la lumière blanche est produite. Lorsque 0% de chaque couleur est combiné, aucune lumière n'est générée et le noir est généré [9].

## 2.3 Caractéristiques d'une image

La qualité d'une image est fortement affectée par de nombreux paramètres. Quelques-uns des paramètres les plus courants traitent de la qualité de l'image sont décrits ci-dessous [9].

### 2.3.1 Dimension

Il s'agit de la taille de l'image, qui est présentée sous la forme d'une matrice dont les éléments sont des valeurs numériques représentatives des intensités de lumière (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image 2D, ce nombre est multiplié par le nombre de calques pour avoir le nombre total de pixels dans une image 3D [9].

### 2.3.2 Résolution

La résolution est un terme qui désigne le nombre de pixels utilisés pour construire une image. Lorsque nous disons qu'une image numérique a une résolution plus élevée qu'une autre image, il signifie que l'image à haute résolution est composée de plus de pixels que l'autre image inférieure. La résolution permet de déterminer la qualité d'une image et décrire en détail comment un objet peut être représenté par l'image. C'est une mesure

pour déterminer la taille d'un objet pour qu'un système d'imagerie le détecte.

En imagerie médicale, le terme résolution peut être utilisé pour décrire la résolution d'imagerie. La résolution d'un système d'imagerie médicale est la capacité du système à représenter les microstructures [9].

### 2.3.3 Bruit d'image

Le bruit dans une image est considéré comme un phénomène de variation soudaine de l'intensité d'un pixel relatif à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur [9].

### 2.3.4 Histogramme

L'histogramme en niveaux de gris ou en couleurs d'une image, est une fonction qui indique la fréquence de chaque niveau de gris(couleur) qui se produit dans l'image. Il donne beaucoup d'informations sur la distribution des niveaux de gris (couleur), et de voir entre quelles limites sont réparties la majorité des niveaux de gris (couleur) dans le cas d'une image trop claire ou trop sombre [9].

### 2.3.5 Luminance

C'est le degré de luminosité des points dans l'image. Il est également défini comme étant le quotient de l'intensité lumineuse d'une surface par la surface apparente de cette surface, pour un observateur éloigné. Le mot luminance est substitué du mot luminosité, qui correspond à la luminosité d'un objet. Une bonne luminance se caractérise par : des images brillantes. Un bon contraste : des images doivent être évités lorsque la plage de contraste tend vers le blanc ou le noir ; ces images provoquent une perte de détails dans les zones sombres ou claires [9].

### 2.3.6 Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions obscures et les régions lumineuses de cette image. Le contraste est défini en fonction de la luminance de deux zones d'image. Si  $L_1$  et  $L_2$  sont les degrés de luminosité

respectivement de deux zones voisines  $A_1$  et  $A_2$  d'une image, le contraste  $C$  est défini par le rapport :  $C = (L_1 - L_2) / (L_1 + L_2)$  [9].

## 2.4 Imagerie médicale

Un système de traitement d'images suit les étapes suivantes :

### 2.4.1 Acquisition d'images médicales

Il s'agit des techniques et des processus utilisés pour créer des images du corps humain à des fins cliniques, comme chercher à révéler, diagnostiquer ou examiner blessure, dysfonctionnement ou pathologie [9].

Dans le contexte clinique, l'imagerie médicale est généralement assimilée à la radiologie. Le responsable de l'interprétation (et parfois de l'acquisition) des images est le radiologue [9].

### 2.4.2 Prétraitement d'image

L'œil humain est essentiellement sensible aux forts contrastes, les méthodes d'amélioration de contraste tentent à augmenter ce dernier, dans le but d'obtenir une image rétinienne de meilleure qualité visuelle. Dans ce paragraphe nous décrivons les différentes méthodes permettant d'augmenter le contraste [20]

## a- Représentation de l'image dans les espaces couleur

### — a-1 Transformation de l'image couleur en niveau de gris (NG)

A partir d'une image couleur obtenue par rétinographie, on peut transformer l'image en niveaux de gris (NG). Cette dernière renferme 256 teintes de gris codé sur un octet, c'est une image à 256 couleurs, simplement chacun de ces 256 couleurs est définie dans la gamme des gris dans le but d'augmenter le contraste [20]

La figure (2.1) montre une image originale en couleur (figure 2.1.a), et sa forme en niveau de gris (figure 2.1.b)

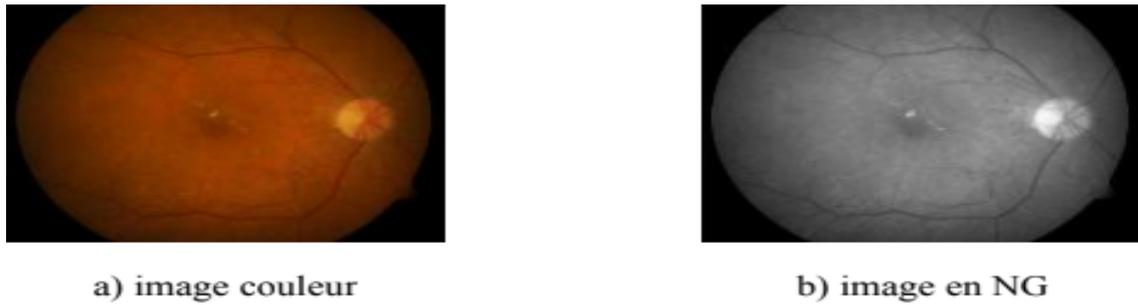


FIGURE 2.1 – La transformation d'une image couleur en une image en niveau de gris [20]

La transformation en NG n'a pas donné de bons résultats en termes de contraste et luminosité des lésions lumineuse. Pour résoudre ce problème, on fait appel aux espaces couleurs comme : RVB, HSV

- **a-2 Espace couleur (RVB)** L'espace couleur RVB (Rouge vert bleu), est un format de codage des couleurs. Ces trois couleurs sont les couleurs primaires en synthèse additive. Elles correspondent en fait aux trois longueurs d'ondes auxquelles répondent les trois types de cônes de l'œil humain. L'addition des trois couleurs donne du blanc pour l'œil humain [20].

La figure (2.2) montre une image en couleur perçue par l'oeil humain (2.2.a), et ses trois canaux à savoir rouge (2.2.b),vert(2.2.c) et bleu (2.2.d).

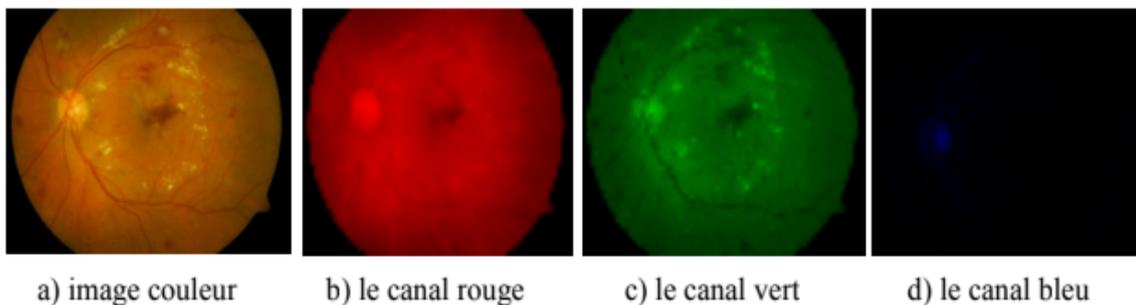


FIGURE 2.2 – Espace couleur RVB [20]

Dans le canal rouge les lésions claires apparaissent très floues (figure b). alors que dans le canal bleu, les lésions ne sont presque pas présentes, il est très bruité (figure d). Cependant, le canal vert donne plus au moins un bon contraste des pathologies (figure c). Pour mieux améliorer la qualité de l'image, on utilise l'espace couleur HSV [20].

— **a-3 Espace couleur (HSV)**

L'espace couleur HSV signifie « teinte, saturation, valeur ». On utilise aussi les termes anglais HSV (hue, saturation, value). L'espace de couleur HSV est un modèle de représentation dit "naturel", c'est-à-dire proche de la perception physiologique de la couleur par l'œil humain. Il consiste à décomposer la couleur selon des critères physiologiques. Cet espace est une transformation non linéaire de l'espace de couleur RVB et c'est un espace colorimétrique, défini en fonction de ses trois composantes [20].

**Teinte** : on code la teinte suivant l'angle qui lui correspond sur le cercle des couleurs :

- 0° ou 360° : rouge ; 60° : jaune ; 120° : vert ; 180° : cyan ; 240 : bleu , 300° : magenta.
- La teinte varie entre 0 et 360, mais est parfois normalisée en 0–100 [20].

**Saturation** : l'intensité de la couleur :

- Elle varie entre 0 et 100%
- Elle est parfois appelée « pureté »
- Plus la saturation d'une couleur est faible, plus l'image sera « grisée » [20].

**Valeur** : la « brillance » de la couleur

- Elle varie entre 0 et 100%
- Plus la valeur d'une couleur est faible, plus la couleur est sombre. Une valeur de 0 correspond au noir [20].

## **b- Filtrage de l'image**

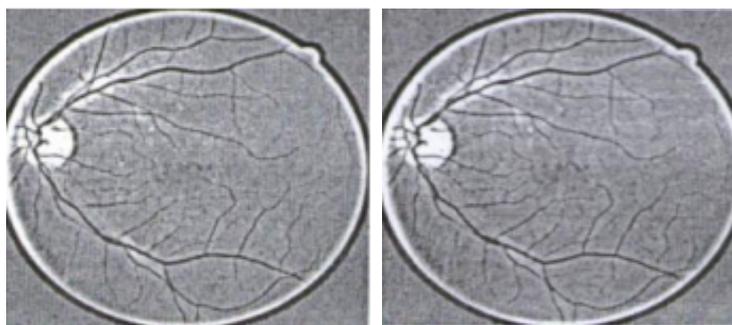
Le filtrage est une étape de prétraitement. Dans notre travail nous faisons appel aux différents types de filtres utilisés en traitement d'image pour éliminer le bruit de l'image et aux différentes méthodes d'amélioration dont le but est d'obtenir une image de meilleure qualité visuelle [20]. Parmi les filtres les plus utilisés en traitement d'image, nous citons :

- **b-1 Filtre passe haut** Le filtre passe-haut est un filtre linéaire principalement utilisé pour traiter les images dont les petits détails ne sont pas assez présents. Ce filtrage est strictement identique à une convolution dans le domaine spatial. Ce filtre

est particulièrement adapté pour traiter le flou car il fait ressortir les petits détails que le flou a estompé. Comme le filtrage n'est pas un outil magique, il a des effets négatifs, dont le principal est d'amplifier le bruit de l'image . En effet, le bruit est particulièrement présent dans les hautes fréquences et traiter une image par un filtre passe –haut augmentera indifféremment les petits détails et bruit. Pour résoudre ce problème, on utilise le filtre de gauss [20].

— **b-2 Le filtre gauss**

Ce filtre est de la même famille que le filtre moyennneur, qui permet de réduire le bruit grâce à la localisation du bruit dans les hautes fréquences. Il est basé sur le même principe .En effet, l'intensité d'un pixel dépend de celle de ses voisins (figure 2.3) :



(a) Image améliorée du canal vert (b) Le filtrage par la gaussienne de l'image précédente

FIGURE 2.3 – Résultat du filtrage de l'image améliorée [20].

D'après les résultats obtenus, on constate que le filtre de gauss (figure 2.6) a permis d'atténuer le bruit mais ajouter un effet de flou. Pour cela, on utilise le filtre médian qui combine à la fois une suppression efficace du bruit et la conservation des détails suffisamment importants [20].

— **b-3 Filtre médian**

On considère les  $N$  pixels contenus dans une fenêtre d'analyse, par exemple une zone de  $3 \times 3$  pixels. La valeur de pixel central de la fenêtre est remplacée par la valeur médiane à l'ensemble des valeurs contenues dans la fenêtre. La valeur médiane est la valeur pour laquelle on observe  $(N-1)/2$  pixels de valeur supérieur et  $(N-1)/2$  de valeur inférieure. Dans la partie suivante, nous présentons des opérations

pour améliorer les images rétiniennes basées notamment sur les modifications de l'histogramme pour augmenter le contraste et la luminosité globale de l'image [20].

### c- Histogramme

En imagerie numérique, l'histogramme représente la distribution des intensités (ou des couleurs) de l'image [20]. C'est un outil fondamental du traitement d'images.

La figure (2.4) représente l'histogramme du canal rouge dont les valeurs des pixels se situent principalement entre [100-200].

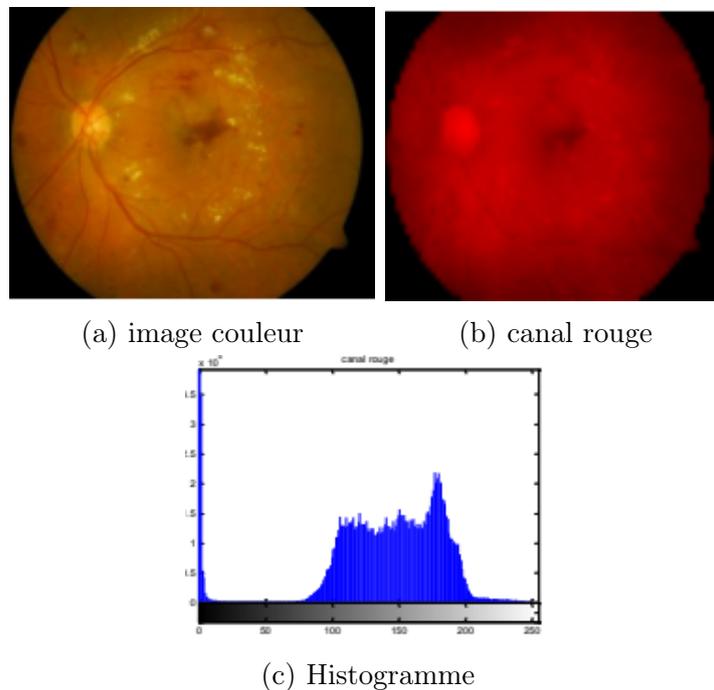


FIGURE 2.4 – Représentation d'histogramme du canal rouge d'une image rétinienne [20].

— Egalisation d'histogramme :

On peut améliorer l'image d'une meilleure façon, en utilisant la technique d'Egalisation d'histogramme, Celle-ci obtenue en appliquant une transformation ponctuelle d'intensité qui consiste à rendre l'histogramme le plus plat possible, ce qui correspond à une répartition idéale des niveaux de gris dans l'image [14].

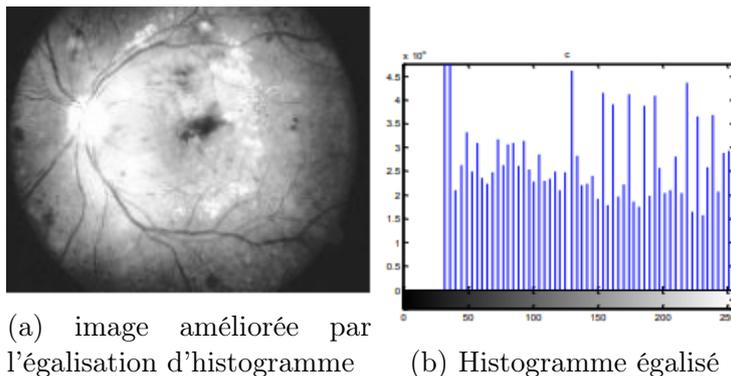


FIGURE 2.5 – Egalisation d'histogramme appliqué sur l'image rétinienne [20].

D'après la figure (2.5) , On constate que là où le contraste est augmenté, on peut mieux visualiser où se situe la pathologie lumineuse.

### 2.4.3 Segmentation

La segmentation est une des étapes critiques de l'analyse d'images qui conditionne la qualité des mesures effectuées ultérieurement. C'est généralement une première étape d'un traitement plus complexe comme la reconnaissance de formes. Elle permet de cerner les formes des objets sur lesquels doit porter l'analyse, de délimiter des régions (l'intérêt et de les extraire du fond). Une bonne méthode de segmentation sera celle qui permettra d'arriver à une bonne interprétation. Elle devra donc avoir simplifié l'image sans pour autant en avoir trop réduit le contenu [16].

En conclusion, la segmentation consiste à :

- Regrouper les pixels de l'image qui partagent une même propriété pour former des régions homogènes.
- Répartir l'ensemble de pixels de l'image en différents groupes.
- Découper l'image en région. Une région est caractérisée par contours et par homogénéité (par exemple, même couleur).
- Partitionner une image en un ensemble de régions connexes et disjointes.
- La recherche de zones de l'image possédant des attributs communs, comme la luminosité, la couleur ou plus rarement la texture [16].

## Différentes approches de segmentation

Généralement, les méthodes de segmentation sont regroupées en trois approches chacune ayant des avantages et ses domaines d'application et elles sont parfois complémentaires, ces approches sont :

**Segmentation basée sur les contours (en anglais : edge-based segmentation) :** L'approche contour consiste à identifier les changements entre les régions. En général, un élément de contours est un point de l'image appartenant à la frontière de deux ou plusieurs objets ayant des niveaux de gris différents..[16].

**Segmentation basée sur les régions (en anglais : régions-based segmentation) :** La segmentation d'image par l'approche région consiste à découper l'image en régions. Les pixels adjacents sont regroupés en régions distinctes, selon un critère d'homogénéité ou de similarité donnée. Ce critère peut être par exemple, le niveau de gris, couleur, texture...etc [16].

**Segmentation en utilisant la classification :** qui consiste à affecter à chaque pixel de l'image une classe qui définit les régions à extraire de l'image [16].

### 2.4.4 Extraction des caractéristiques

Consiste à extraire les paramètres nécessaires à la reconnaissance des formes. Ces caractéristiques peuvent être de type :

- Statistique : Moyenne, variance..
- Forme : squelette, segment, cercle...
- Texture : contraste, homogénéité... [2].

### 2.4.5 La Reconnaissance

Une fois l'image traitée, les régions détectées et les primitives extraites, l'image est prête à être classée. Parmi les techniques de classification, nous retrouvons :

- Les approches mathématiques (k plus proche voisin, ) [26].
- Les approches par réseau de neurones (CNN..) qui font l'objet de notre travail.

## Conclusion

Dans ce chapitre, nous avons expliqué ce qu'est l'image numérique, ainsi que ses types, ses caractéristiques et le processus permettant de la traiter. Une fois que l'image est traitée, il reste à l'exploiter afin de pouvoir diagnostiquer l'existence éventuelle de la rétinopathie diabétique. Pour cela, on va exploiter les techniques de l'IA, à savoir l'apprentissage automatique et l'apprentissage profond. Ceci fera l'objet du prochain chapitre.

# Chapitre 3

## Apprentissage Automatique et Apprentissage Profond

### Sommaire

---

<b>3.1</b>	<b>Les Réseaux de Neurones . . . . .</b>	<b>22</b>
<b>3.2</b>	<b>Apprentissage Automatique : . . . . .</b>	<b>27</b>
<b>3.3</b>	<b>Apprentissage Profond . . . . .</b>	<b>31</b>
<b>3.4</b>	<b>Méthodes de détection de la RD . . . . .</b>	<b>36</b>

---

## Introduction

L'apprentissage Automatique est une application de l'intelligence artificielle (IA) qui procure aux systèmes la capacité d'apprendre et d'améliorer automatiquement à partir de l'expérience sans être explicitement programmé.

Aujourd'hui, l'apprentissage profond est beaucoup utilisé dans le traitement d'images, et la reconnaissance d'objets qui nous montrent des progrès dans l'utilisation de réseaux convolutionnels profonds. Selon les fondateurs Yann LeCun, Yoshua Bengio Geoffrey Hinton[51], l'apprentissage profond permet à des modèles informatiques composés de plusieurs couches de traitement d'apprendre les représentations de données avec plusieurs niveaux d'abstraction.

Dans ce chapitre, nous allons présenter les réseaux de neurones, l'apprentissage automatique et l'apprentissage profond, ainsi que leurs méthodes utilisées dans la littérature pour la détection de la rétinopathie diabétique.

## 3.1 Les Réseaux de Neurones

### 3.1.1 Définition

De nombreuses inventions ont été inspirées par la nature, il semble donc logique de regarder l'architecture du cerveau pour s'inspirer de la façon de construire des machines intelligentes.

Un «réseau de neurones artificiels» est un système de calcul qui tente d'imiter (ou du moins, s'inspirer) des connexions neuronales de notre système nerveux. Les réseaux de neurones artificiels sont également appelés « réseaux de neurones » ou « systèmes de neurones artificiels ».

Dans un réseau de neurones on trouve 3 couches :

- **La couche d'entrée** qui prend les données pour le réseau, dans le cas d'une image par exemple comme entrée, ce sont tous les pixels de cette image qui seront dans cette couche du réseau ( $x_1, x_2, x_3$  dans la figure (3.1)).
- **La couche de sortie** : qui est responsable de regrouper les résultats calculés par le réseau ( la classe a laquelle une image appartient par exemple).

- **Couches cachées** : qui se trouve entre ces deux couches déjà présenté ci-dessus, elles permettent d'extraire les caractéristiques a partir des données d'entrées pour les utiliser par la couche de sortie [39].

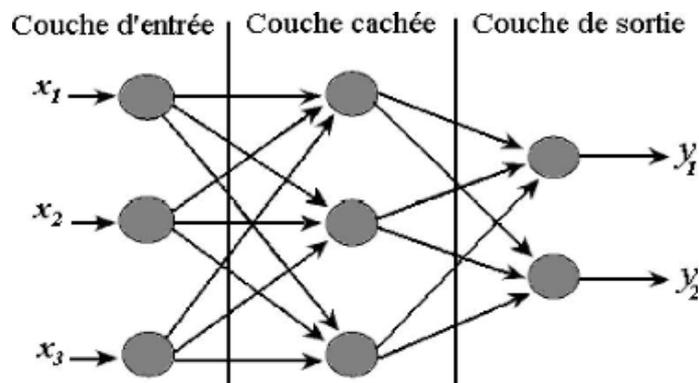


FIGURE 3.1 – Architecture d'un réseau de neurones [39]

Chaque nœud effectue un calcul simple. Chaque connexion transporte ensuite un signal (c'est-à-dire la sortie du calcul) d'un nœud à un autre, étiqueté par un poids indiquant dans quelle mesure le signal est amplifié ou diminué. Certaines connexions ont des poids positifs importants qui amplifient le signal, indiquant que le signal est très important lors d'une classification. D'autres ont des poids négatifs, diminuant la force du signal, spécifiant ainsi que la sortie du nœud est moins importante dans la classification finale. Nous appelons un tel système un réseau de neurones artificiels s'il consiste en une structure de graphe avec des poids de connexion modifiables à l'aide d'un algorithme d'apprentissage.

La classification par réseaux de neurones s'effectue en 2 étapes :

- Apprentissage pour qui chaque couche de sortie donne un résultat
- Test afin d'évaluer la validité du système.

### 3.1.2 Apprentissage

Nous avons vu certains des composants des réseaux de neurones, nous allons citer et expliquer ici les types d'apprentissage qu'un modèle peut effectuer pour pouvoir accomplir

la tâche qui lui est attribué.

— **Apprentissage supervisé :**

Dans ce cas, le réseau est formé en lui fournissant des modèles d'entrée et de sortie correspondants [27].

— **Apprentissage non-supervisé :**

Contrairement au paradigme de l'apprentissage supervisé, il n'y a pas d'ensemble a priori de catégories dans lesquelles les modèles doivent être classés ; le système doit plutôt développer sa propre représentation des stimuli d'entrée [27].

— **Apprentissage par renforcement :**

Ici le modèle d'apprentissage effectue une action sur l'environnement et obtient une réponse en retour de l'environnement. Le système d'apprentissage note son action bonne (gratifiante) ou mauvaise (punissable) en fonction de la réponse environnementale et ajuste en conséquence ses paramètres. Généralement, l'ajustement des paramètres est poursuivi jusqu'à ce qu'un état d'équilibre se produise[27].

Afin de convertir les entrées en sorties, on utilise des fonctions d'activation.

### 3.1.3 Fonctions d'activation

Les fonctions d'activation sont des équations mathématiques qui déterminent la sortie d'un réseau neuronal.

La fonction est attachée à chaque neurone du réseau et détermine s'il doit être activé ou non, si l'entrée de chaque neurone est pertinente pour la prédiction du modèle.

Les fonctions d'activation aident également à normaliser la sortie de chaque neurone à une plage comprise entre 1 et 0 ou entre -1 et 1. Un autre aspect des fonctions d'activation est qu'elles doivent être efficaces sur le plan informatique parce qu'elles sont calculées à travers des milliers voir des millions de neurones pour chaque échantillon de données. Les réseaux de neurones modernes utilisent une technique appelée rétropropagation (propagation vers l'arrière) pour former le modèle, qui place une pression de calcul accrue sur la fonction d'activation, et sa fonction dérivée. Le besoin de vitesse a conduit au développement de nouvelles fonctions telles que ReLu et Swish. [10]

Parmi les fonctions d'activation les plus utilisées, on trouve :

**Fonction Hyperbolic tangent** : La fonction tangente hyperbolique, ou tanh pour faire court, est une fonction d'activation non linéaire qui produit des valeurs entre -1.0 et 1.0 [9].

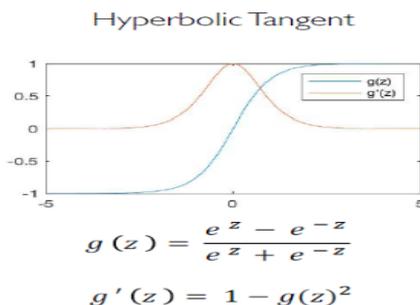


FIGURE 3.2 – Hyperbolic Tangent Function [9].

**Fonction ReLU** : C'est une des fonctions d'activation les plus utilisées dans les réseaux de conception aujourd'hui. En plus d'être non linéaire, par rapport à d'autres fonctions d'activation, l'avantage principal de l'utilisation de la fonction ReLU est qu'il n'active pas tous les neurones en même temps. Si l'entrée est négative, elle sera convertie en 0 et le neurone ne sera pas activé. Cela signifie que seuls quelques neurones sont activés à la fois, ce qui rend le réseau rare et très efficace [9].

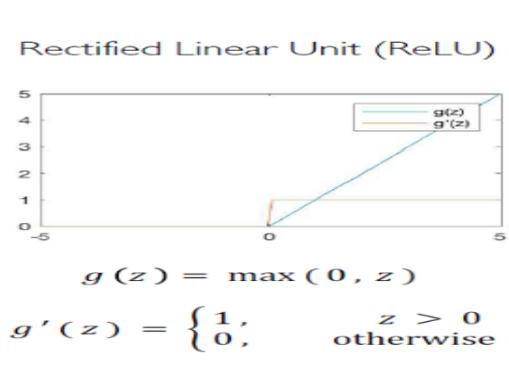


FIGURE 3.3 – Fonction ReLU [9]

**Fonction Sigmoid** La fonction d'activation Sigmoid, également connue sous le nom de fonction logistique, a traditionnellement été une fonction d'activation populaire pour les réseaux neuronaux. L'entrée de cette fonction sera convertie en une valeur comprise entre 0,0 et 1,0 [9].

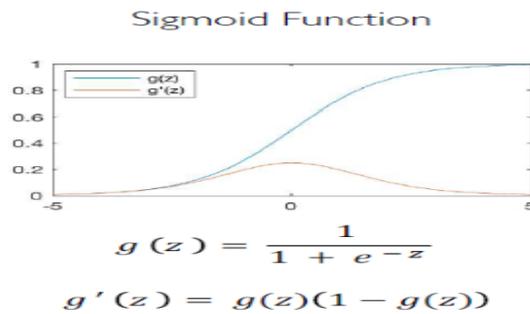


FIGURE 3.4 – Fonction Sigmoid [9]

**Fonction Softmax :** Softmax est une fonction mathématique qui convertit un vecteur de nombres en un vecteur de probabilités, où les probabilités de chaque valeur sont proportionnelles à l'échelle relative de chaque valeur dans le vecteur [29].

L'utilisation la plus courante de la fonction softmax dans l'apprentissage automatique appliqué est son utilisation en tant que fonction d'activation dans un modèle de réseau de neurones. Plus précisément, le réseau est configuré pour sortir N valeurs, une pour chaque classe dans la tâche de classification, et la fonction softmax est utilisée pour normaliser les sorties, en convertissant des valeurs de somme pondérées en probabilités dont la somme est égale à un. Chaque valeur dans la sortie de la fonction softmax est interprétée comme la probabilité d'appartenance à chaque classe [29].

### 3.1.4 Tests

Une fois le réseau de neurones entraîné (après apprentissage), il est nécessaire de le tester sur une base de données différente de celle utilisée pour l'apprentissage. Ce test permet à la fois d'apprécier les performances du système neuronal et de détecter le type de données qui pose problème. Si les performances ne sont pas satisfaisantes, il faudra soit

modifier l'architecture du réseau, soit modifier la base d'apprentissage (caractéristiques discriminantes ou représentativité des données de chaque classe).

## 3.2 Apprentissage Automatique :

L'apprentissage automatique est un sous-domaine de l'informatique dans lequel les machines apprennent à exécuter des tâches pour qu'ils n'ont pas été explicitement programmés. En bref, les machines observent un modèle et tentent de l'imiter d'une manière qui peut être directe ou indirecte [10].

### 3.2.1 Techniques de pré-traitement

Il existe plusieurs techniques de pré-traitement en apprentissage automatique , parmi elles on retrouve :

- L'extraction du canal vert.
- L'amélioration du contraste.
- La correction de l'illumination.
- Le filtre gaussien.
- Le redimensionnement.
- La suppression du disque optique et des vaisseaux sanguins.
- L'égalisation de l'histogramme.
- L'augmentation.
- La conversion en niveaux de gris.
- Le recadrage de la région d'intérêt [2].

### 3.2.2 Caractéristiques utilisées

L'extraction des caractéristiques est l'un des concepts clés de l'apprentissage automatique, qui a un effet considérable sur l'efficacité de l'algorithme et les résultats obtenus. Cette étape a plusieurs avantages, notamment :

1. Diminution du surapprentissage

2. Augmentation de la précision : moins de données inexactes signifie une amélioration de la précision.
3. Diminue le temps de traitement : moins de points de données diminue la complexité de l'algorithme et l'entraînement des algorithmes devient plus rapide [15].

Les caractéristiques :

- **Forme et structure** Ces caractéristiques comprennent la forme et la structure de diverses lésions de la rétinopathie diabétique, notamment les exsudats durs et mous, les hémorragies et les microanévrismes[2]. Par exemple, pour la détection de microanévrismes, Zhou et Wu [13] ont utilisé la surface et le périmètre, la longueur de l'axe, la circularité et la compacité.
- **Couleur** : Ces caractéristiques sont basées sur les couleurs RVB de l'image, pour la détection d'exsudats durs par exemple en utilisant des images de fond en couleur[2].
- **Intensité** L'intensité du pixel est exprimée dans les plans R, V et B. Pour par exemple, la détection de taches de coton dans des images DR. De même en calculant les intensités maximale et minimale des pixels pour la détection des exsudats durs et mous[2].
- **Statistique** : Ces caractéristiques sont utilisées pour les mesures statistiques des pixels d'une image de rétinopathie diabétique. Par exemple, Xiao, Yu [14] ont utilisé des caractéristiques statistiques avec des caractéristiques de couleur pour détecter des hémorragies dans des images rétinienne.

Les caractéristiques statistiques utilisées étaient la moyenne, le maximum, le minimum et l'écart-type [2].

- **Texture** : le contraste, la corrélation, l'homogénéité, l'entropie, la variance de la somme, la variance de la différence, l'entropie de la somme, l'entropie de la différence, la moyenne de la somme[2].

Les caractéristiques basées sur **la forme**, **les statistiques** et **la texture** sont les plus discriminantes dans les algorithmes de détection de RD, et peuvent produire des résultats prometteurs pour les chercheurs travaillant dans le domaine de la détection et de la classification de la RD basée sur l'apprentissage automatique[2].

### 3.2.3 classification en apprentissage automatique

La classification est le processus de prédiction de la classe de points de données donnés. Les classes sont parfois appelées cibles/étiquettes ou catégories. La modélisation prédictive de classification est la tâche d'approximation d'une fonction de mappage ( $f$ ) des variables d'entrée ( $X$ ) aux variables de sortie discrètes ( $y$ ) [27].

Il existe actuellement de nombreux algorithmes de classification en apprentissage automatique, classés et schématisés dans la figure suivante :

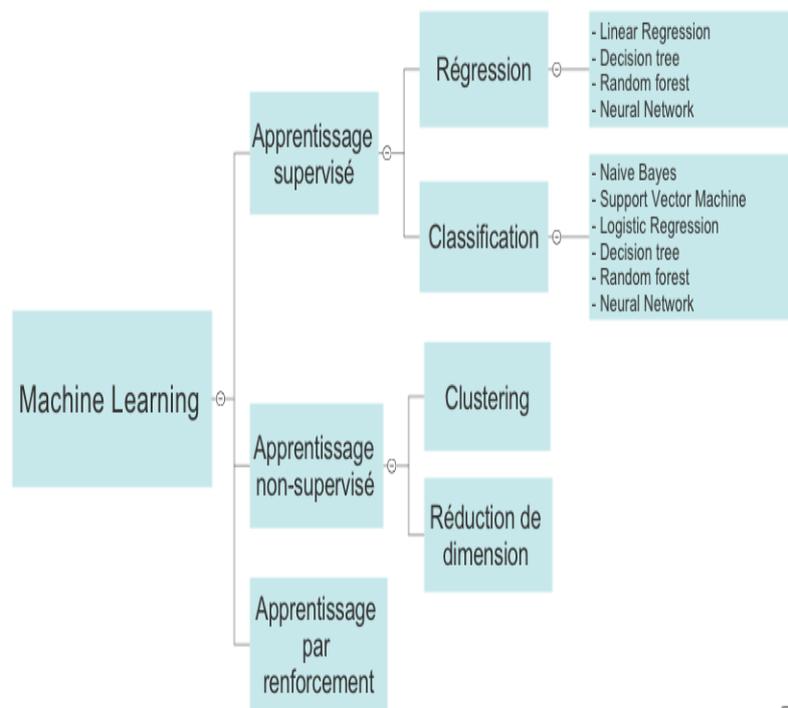


FIGURE 3.5 – Schéma représentant le type et quelques algorithmes de classification en ML

### 3.2.4 Mesures de performances

Les mesures de performance sont utilisées pour évaluer les algorithmes d'apprentissage et constituent un aspect important de l'apprentissage automatique. Dans certains cas, ces mesures sont également utilisées comme heuristiques pour construire des modèles d'apprentissage.

Notations : VP (Vrais Positifs), VN (Vrais Négatifs), FP (Faux Positifs) et FN (Faux Négatifs).

Les vrais positifs (VP) sont le nombre d'images de la maladie classées comme maladie. Le vrai négatif (VN) est le nombre d'images normales classées comme normales tandis que le faux positif (FP) est le nombre d'images normales classées comme maladie. Les faux négatifs (FN) sont le nombre d'images de la maladie classées comme normales [34].

**SP = Spécificité** : (aussi appelée Fraction de Vrais Négatifs) : proportion d'individus négatifs effectivement bien détectés par le test [31]. C'est donc le rapport :  $VN/(VN+FP)$

**Acc = Exactitude** : C'est le rapport  $(VP+VN)/(VP+VN+FP+FN)$ . Plus elle est proche de 1, meilleur est le test [31].

**AUC** l'aire sous la courbe (ou Area Under the Curve – AUC ) est un indice synthétique calculé pour les courbes ROC. L'AUC correspond à la probabilité pour qu'un événement positif ait une probabilité donnée par le modèle plus élevée qu'un événement négatif. Pour un modèle idéal, on a  $AUC = 1$ , pour un modèle aléatoire, on a  $AUC = 0,5$  [31].

**F-Sc = F-Score** : aussi appelée la F-mesure ou score-F1 peut être interprétée comme une moyenne pondérée de la précision et du rappel ou sensibilité. Sa valeur est comprise entre 0 et 1 [31].

**K-Sc = Score Kappa** : indique essentiellement à quel point votre classificateur fonctionne mieux par rapport aux performances d'un classificateur qui devine simplement au hasard en fonction de la fréquence de chaque classe .

Le kappa de Cohen est toujours inférieur ou égal à 1. Les valeurs de 0 ou moins indiquent que le classificateur est inutile [31].

### 3.2.5 Problèmes rencontrés en apprentissage automatique

Il y a deux problèmes principaux rencontrés dans l'apprentissage automatique et ils sont :

— **Surapprentissage**

Au fur et à mesure que le modèle apprend les détails et le bruit des données de formation, il y a débordement parce que cela aura une incidence négative sur le

rendement du modèle sur les nouvelles données. Cela signifie que le bruit ou les fluctuations aléatoires des données de formation seront absorbées par le modèle et apprises comme concept. Le problème est que ces concepts ne sont pas applicables aux nouvelles données et auront un impact négatif sur la généralisabilité du modèle. Les modèles non paramétriques et non linéaires sont plus susceptibles de déborder, et ils ont plus de flexibilité lors de l'apprentissage de la fonction objective. De cette façon, beaucoup d'algorithmes d'apprentissage automatique non paramétriques comprennent également des paramètres ou des techniques pour limiter la quantité de détails dans la formation des modèles. Par exemple, un arbre de décision est un algorithme d'apprentissage automatique non paramétrique très flexible, et il est facile de dépasser les données de formation. Ce problème peut être résolu en élaguant l'arbre après l'apprentissage pour supprimer certains des détails qu'il recueille [9].

- **Sous-apprentissage** Désigne les modèles qui ne peuvent pas être modélisés à partir de données de formation ou qui ne peuvent pas être généralisés en de nouvelles données. Un modèle d'apprentissage automatique sous-ajusté n'est pas un modèle approprié parce qu'il agit sur les données de formation, c'est donc évident. Le problème de sous-apprentissage n'est généralement pas abordé parce qu'il est facile à détecter avec de bons indicateurs de performance. La solution est de continuer à essayer d'autres algorithmes d'apprentissage automatique. Cependant, cela contraste fortement avec le problème de sur-apprentissage [9].

**Remarque** Malgré l'efficacité du ML, mais cette forme de l'IA a des limites (Par exemple : Lorsque tous les aspects des objets à traiter ne peuvent pas être catégorisés en amont), et donc n'est pas adaptée pour les tâches complexes. C'est de cela qu'à été créé le sous ensemble du machine learning, appelé : L'apprentissage profond, dans lequel les réseaux de neurones multicouches apprennent à partir d'une grande quantité de données.

### 3.3 Apprentissage Profond

L'apprentissage profond est un sous-domaine de l'apprentissage automatique traitant des algorithmes inspirés par la structure du cerveau humain. En d'autres termes, il reflète la façon dont notre cerveau fonctionne. Les algorithmes de l'apprentissage profond sont

similaires à la façon dont le système nerveux se structure où chaque neurone se connecte et transmet des informations. L'apprentissage profond est un progrès relativement nouveau dans la programmation de réseaux neuronaux et représente un façon de former des réseaux neuronaux profonds essentiellement, tout réseau neuronal avec plus de deux couches est profonde.

### 3.3.1 Différentes architectures de l'apprentissage profond

Il existe principalement trois types d'architectures d'apprentissage profond, chaque architecture a son propre domaine d'utilisation et ses caractéristiques.

- **DNN (Deep Neural Networks)** Les nœuds sont de petites parties du système, et ils sont comme les neurones du cerveau humain. Lorsqu'un stimulus les frappe, un processus se déroule dans ces nœuds. Certains d'entre eux sont connectés et marqués, d'autres non, mais en général, les nœuds sont regroupés en couches.

Le système doit traiter des couches de données entre l'entrée et la sortie pour résoudre une tâche. Plus il doit traiter de couches pour obtenir le résultat, plus le réseau est considéré en profondeur. Il existe un concept de chemin d'attribution de crédit (CAP) qui signifie le nombre de ces couches nécessaires au système pour accomplir la tâche [32].

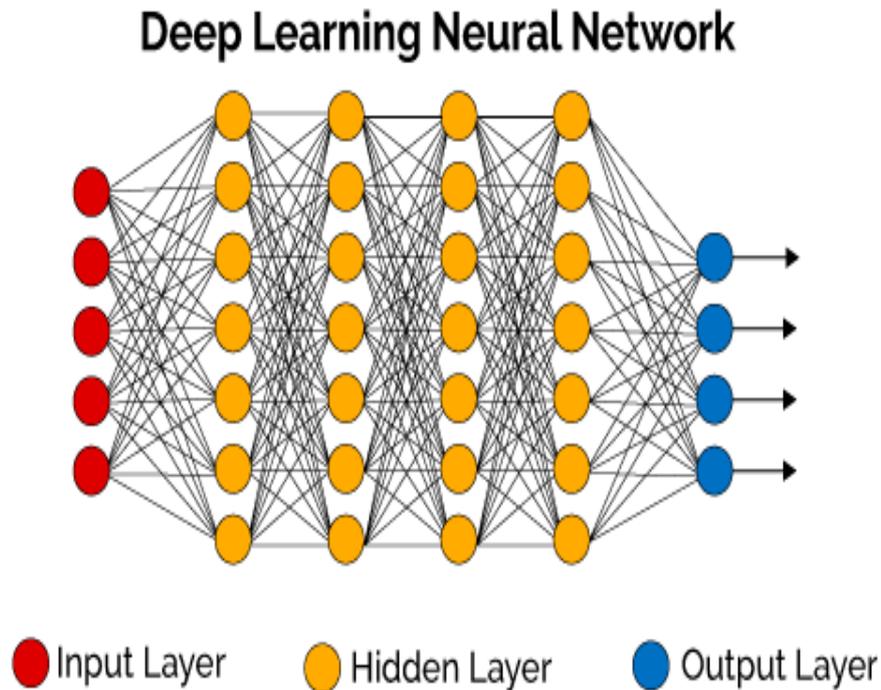


FIGURE 3.6 – Architecture du DNN [52]

— **RNN (Recurrent neural networks)**

Les RNN sont un algorithme couramment utilisé et familier dans la discipline de l'apprentissage profond. RNN est principalement appliqué dans le domaine du traitement de la parole. Contrairement aux réseaux conventionnels, RNN utilise des données séquentielles dans le réseau. Étant donné que la structure intégrée dans la séquence des données fournit des informations précieuses, cette fonctionnalité est fondamentale pour une gamme d'applications différentes. Pour une séquence d'entrée donnée, un diagramme RNN est illustré sur la figure(3.7) [33].

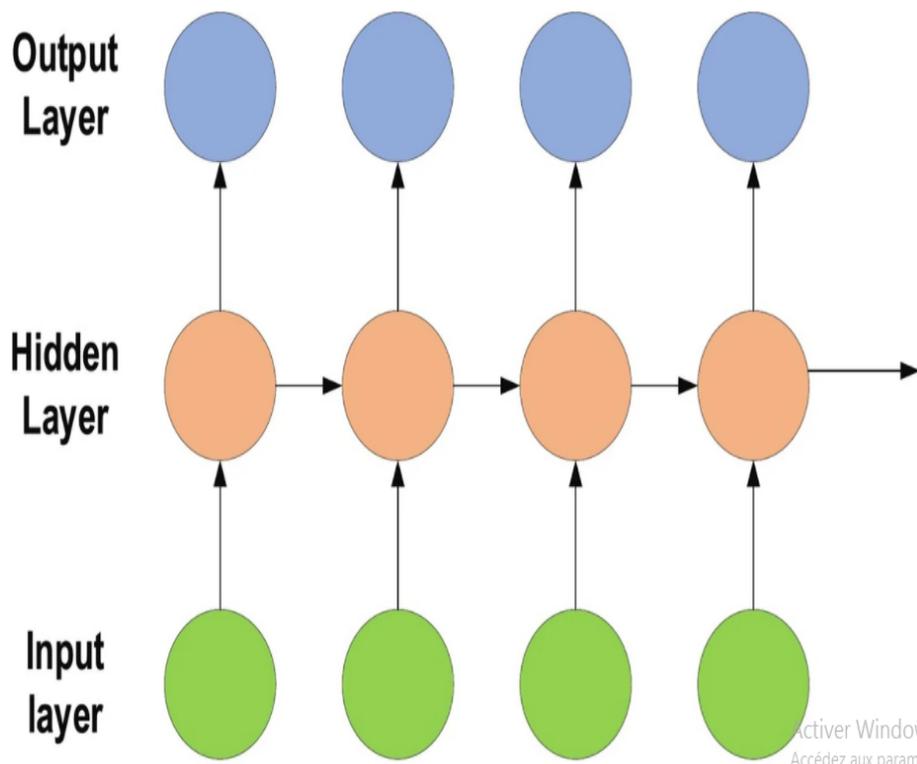


FIGURE 3.7 – Diagramme du RNN [33]

- **CNN (Convolutional Neural Networks)** Il existe différents types de réseaux de neurones et les différences entre eux résident dans leurs principes de travail, le schéma d'actions et les domaines d'application. Les réseaux de neurones convolutifs (CNN) sont principalement utilisés pour la reconnaissance d'images et rarement pour la reconnaissance audio. Il est principalement appliqué aux images car il n'est pas nécessaire de vérifier tous les pixels un par un. CNN vérifie une image par blocs, en commençant par le coin supérieur gauche et en se déplaçant davantage pixel par pixel jusqu'à ce qu'il soit terminé. Ensuite, le résultat de chaque vérification est passé à travers une couche convolutive, où les éléments de données ont des connexions tandis que d'autres n'en ont pas. Sur la base de ces données, le système peut produire le résultat des vérifications et peut conclure ce qui est dans l'image [32].

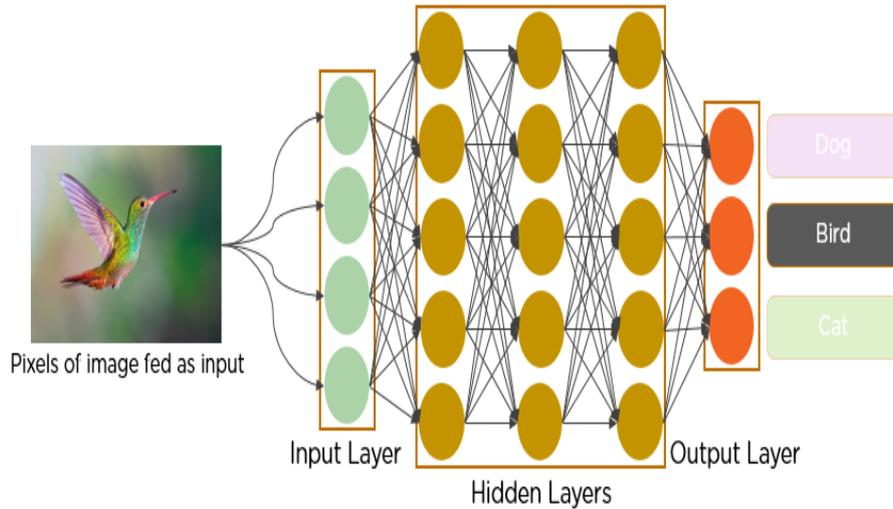


FIGURE 3.8 – Exemple d’architecture CNN [36].

### 3.3.2 Comparaison entre apprentissage automatique et apprentissage profond en traitement d’images

Afin de déterminer quelle méthode utiliser, on établit une comparaison entre l’apprentissage automatique et l’apprentissage profond.

Cette comparaison est illustrée dans le tableau suivant :

Propriétés	Apprentissage Automatique	Apprentissage Profond
Organisation des données	Données structurées	Données non structurées
Base de données	Contrôlable	> 1 million de données
Entraînement	Entraînement par l’humain nécessaire	Système d’apprentissage autonome
Algorithme	Algorithme modifiable	Réseau neuronal d’algorithmes
Champ d’application	Actions simples de routine	Tâches complexes

TABLE 3.1 – Tableau comparatif entre Apprentissage Automatique et Apprentissage Profond [35].

Après la comparaison, on a choisi la méthode de l'apprentissage profond qui correspond au type d'étude qu'on souhaiterait faire, et qui est fortement appliqué pour les tâches complexes.

### 3.4 Méthodes de détection de la RD

Il existe plusieurs méthodes de détection de la RD, on les schématise de la manière suivante :

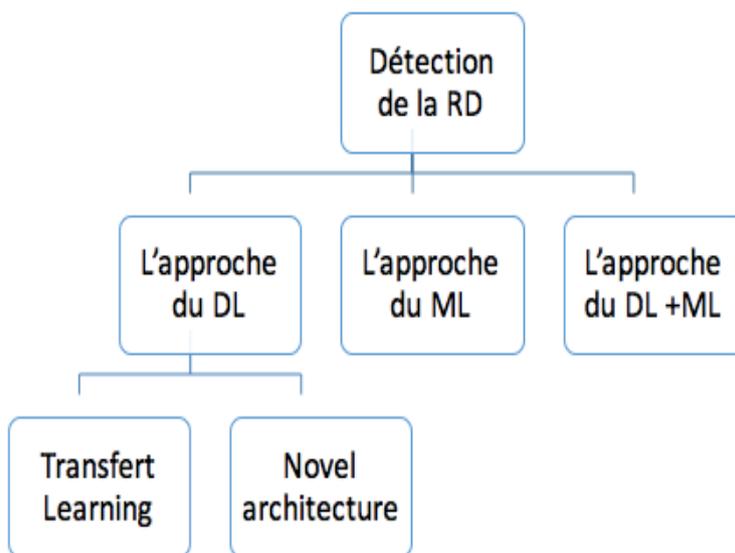


FIGURE 3.9 – Méthodes existantes pour la détection de la rétinopathie diabétique RD

#### 3.4.1 L'approche d'apprentissage automatique

Le processus générique de l'apprentissage automatique pour la détection des DR est illustré à la Figure (3.10). Comme on peut le voir, l'ensemble d'images étiquetées (aussi appelé ensemble d'entraînement) est collecté pour la construction du modèle. Cet ensemble

d'entraînement comprend des images appartenant à différentes classes. Ensuite, plusieurs techniques de prétraitement d'images sont appliquées pour éliminer les caractéristiques inutiles des images collectées [2].

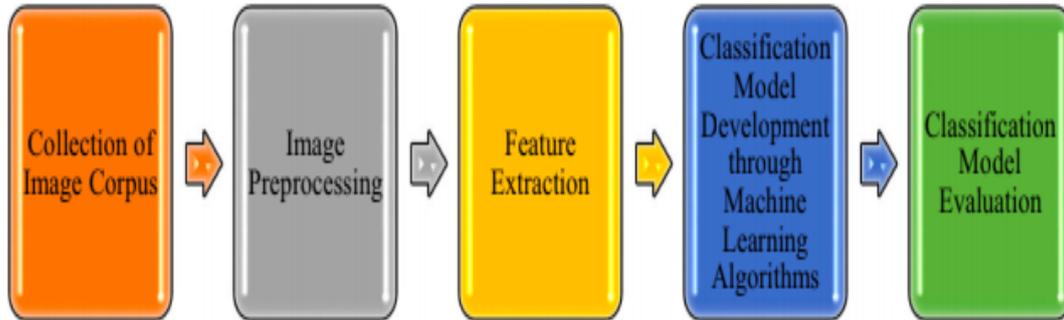


FIGURE 3.10 – Modèle basé sur l'apprentissage automatique [2]

Diverses techniques d'extraction de caractéristiques sont utilisées pour découvrir les caractéristiques les plus discriminantes des images prétraitées. Le résultat de cette étape est le vecteur de caractéristiques principal (MFV). Ce MFV est ensuite utilisé comme entrée de l'algorithme d'apprentissage automatique pour construire le modèle de détection de la RD après avoir appris les règles de classification à partir du MFV. La performance du modèle de détection des DR construit est ensuite évaluée sur de nouvelles images non étiquetées (ou ensemble de test). Plusieurs mesures de performance, y compris la précision, la sensibilité, la spécificité et l'aire sous la courbe (AUC), sont utilisées pour mesurer la performance du modèle de détection des zones mortes. (AUC) sont utilisées pour mesurer la performance de la classification [2].

Par exemple, **Kr Biswas et al** dans [8] ont proposé un système intelligent pour la rétinopathie diabétique qui se base sur la segmentation des vaisseaux sanguins en utilisant le filtre adaptatif séquentiel et la détection des microanévrisme (MA) en utilisant le filtre de canny (CEDA) [31]. Mais avant l'utilisation du système intelligent, un algorithme (CLAHE) Contrast Limited Adaptive Histogram Equalizer [32] est utilisé pour l'amélioration du contraste des images dans le but de distinguer les vaisseaux sanguins et les MA du reste de l'image.

Trois classificateurs sont utilisés par le système proposé, le Support Vector Machine

(SVM), K-Nearest Neighbors (K-NN) avec  $k=5$  et Gaussian Naive Bayes (GNB) pour classer une image en 5 catégories.

2500 images de l'ensemble de données Kaggle sont prises à parts égales dans toutes les classes. Les 80% des données sont utilisées comme ensemble d'apprentissage et 20% comme ensemble de test pour toutes les évaluations.

### 3.4.2 L'approche d'apprentissage profond

Initialement, l'ensemble de données est recueilli et sera utilisé pour la formation et le test de l'architecture d'apprentissage profond. Par la suite, certaines techniques de pré-traitement sont appliquées sur l'ensemble de données afin que le réseau puisse apprendre les fonctionnalités à partir d'une image plus claire. L'image prétraitée est transmise à l'architecture d'apprentissage profond pour l'extraction des caractéristiques et la classification. Les couches de l'architecture d'apprentissage profond utilisent la sortie de la couche précédente comme entrée, traite cette sortie et la passe à la couche suivante. Enfin, la dernière couche produit le résultat du diagnostic [2].



FIGURE 3.11 – Modèle basé sur l'apprentissage profond [2]

### 3.4.3 Approches d'apprentissage profond utilisant l'apprentissage par transfert

Le principe consiste à réutiliser des réseaux pré-entraînés sur d'autres problèmes, par des techniques de transfer learning. La librairie open-source Keras permet de coder cela, avec une interface de programmation d'applications(ou API ) claire et de haut niveau. L'avantage de cette méthode c'est qu'elle n'est pas coûteuse en terme de matériel, de quantité de données annotées nécessaires(images) et de capacités de calcul [17].

Le Transfer Learning (ou apprentissage par transfert) permet de faire de l'apprentissage profond sans avoir besoin d'y passer par des calculs. Le principe est d'utiliser les connaissances acquises par un réseau de neurones lors de la résolution d'un problème afin d'en résoudre un autre plus ou moins similaire. On réalise ainsi un transfert de connaissances, d'où le nom.

Selon le concept de l'apprentissage par transfert(transfer learning), les caractéristiques apprises à l'aide d'une architecture de l'apprentissage profond peuvent être transférées à une architecture similaire de l'apprentissage profond.

Pour cette approche, on a par exemple :

**van Grinsven, van Ginneken [37]** ont détecté des hémorragies à partir d'images de fond d'écran couleur avec l'utilisation de l'échantillonnage sélectif de données. Ils ont appliqué un CNN de 9 couches composé de 5 couches de convolution inspirées d'OxfordNet où, dans chaque couche de convolution ils ont utilisé 32 filtres de taille 3x3. Les couches de convolution ont été suivies par des couches ReLU et spatial max-pooling, une couche entièrement connectée et enfin une couche de classification softmax. Ils ont atteint une sensibilité de 91,90%, une spécificité de 91,40% et une AUC de 0,972 pour la détection d'hémorragie par échantillonnage sélectif sur la dataset Messidor.

**Takahashi, Tampo [38]** ont modifié le GoogLeNet pour détecter les stades de la rétinopathie diabétique. Les modifications appliquées sur le GoogLeNet étaient : les 5 premières couches de précision ont été supprimées, la taille de la récolte était de 1272 pixels, et la taille du lot a été réduite à 4. En utilisant la dataset exclusive par Jichi

Medical University, au Japon, ils ont entraîné 95% des données totales, les ont testées sur les 5% de données restantes, et ont atteint une précision de 81%.

**Shu Wei Ting, Yim-Lui Cheung [39]** ont adapté 8 VGGNet basés sur les architectures CNN afin de développer et valider un système de deep learning (DLS) pour la détection de la rétinopathie diabétique et d'autres maladies oculaires, y compris le glaucome et la dégénérescence maculaire liée à l'âge (DMLA). Une dataset exclusive de multiethnicités a été recueilli de 2010 à 2013 dans le cadre du Programme national de dépistage de la rétinopathie diabétique (SIDRP) de Singapour pour cette recherche. Ils ont atteint une sensibilité de 90,5%, une spécificité de 91,6% et une AUC de 0,936 pour la rétinopathie diabétique référible.

**Perdomo, Arevalo [40]** ont exploré l'architecture LeNet avec 8 couches dont une entrée, couche convolution 1, couche max-pool 1, couche convolution 2, couche max-pool 2, couche entièrement connectée 1, couche entièrement connectée 2 et couche de sortie. Ils ont utilisé un classificateur softmax dans la couche de sortie pour la classification de la rétinopathie diabétique comme image saine ou une image ayant des exsudats. La dataset utilisée était e-optha et a atteint une sensibilité de 99,8%, spécificité de 99,6%, tandis que la précision était de 99,6%.

**Mo et Zhang [41]** ont appliqué le transfer learning avec l'utilisation de l'architecture VGGNet pour la segmentation des vaisseaux rétiniens et la sensibilité atteinte de 77,79%, spécificité de 97,80%, exactitude de 95,21% et leurs résultats sur AUC et Kappa est de 0,9782 et 0,7759 respectivement.

### **3.4.4 Approches d'apprentissage profond utilisant de nouvelles architectures d'apprentissage profond**

Pour la détection de la rétinopathie diabétique grâce à de nouvelles architectures d'apprentissage profond, différents chercheurs ont présenté leurs propres architectures. Il y a un nombre distinct de couches dans l'architecture et différents classificateurs ont été utilisés. La plupart des chercheurs ont utilisé Softmax classifier pour la classification des

images de la rétinopathie diabétique.

Parmi ces études, on a par exemple :

**Yu, Xiao [42]** ont proposé une architecture d'un réseau neuronal convolutionnel de 16 couches pour la détection des exsudats. Ils ont utilisé les CNN pour identifier les exsudats sur la classification pixel-wise par le classificateur softmax et réalisé une précision de 91,92% et sensibilité et spécificité de 88,85% et 96% respectivement.

**Doshi, Shenoy [43]** ont présenté un deep CNN basé sur un diagnostic sévère de rétinopathie diabétique. Ils ont utilisé un CNN de 29 couches , le classificateur softmax et ont détecté les cinq stades de la rétinopathie diabétique. La précision atteinte sur un ensemble de 3 CNN était de 0,3996 sur la matrice kappa.

**Chudzik, Majumdar[44]** ont présenté un CNN de 25 couches pour la détection des microanévrismes. Le résultat obtenu sur la metrique FROC(Free-response Receiver Operating Characteristic) était de 0,562 en utilisant une classification par pixel (le classificateur pixel wise).

### 3.4.5 L'approche d'apprentissage automatique et d'apprentissage profond

Le processus générique d'apprentissage automatique et d'apprentissage profond pour la détection de la rétinopathie diabtétique se caractérise comme suit :

- L'approche d'apprentissage profond sert à faire l'extraction des caractéristiques grâce à la partie convolution des CNN par exemple, en éliminant la partie classification ( Les couches entièrement connectées).

- L'approche d'apprentissage automatique sert à faire la classification en utilisant un algorithme de classification en prenant en entrée les sorties obtenues (les caractéristiques) du modèle d'apprentissage profond utilisé.

Pour cette méthode, on a par exemple :

**Arunkumar et Karthigaikumar [45]** ont utilisé un Réseau de Croyance Profonde

(ou Deep Belief Network DBN) pour la classification des images rétiniennes diabétiques. Tout d'abord, les caractéristiques profondes étaient extraites à l'aide du réseau de croyance profonde (DBN) avec trois couches cachées, puis ces caractéristiques ont été réduites grâce à la technique du réseau neuronal de régression généralisée (GRNN) [47] et, enfin, Le SVM a été utilisé pour la classification des images. Ils ont atteint une sensibilité de 79,32%, spécificité de 97,89% et l'exactitude de 96,73% de l'ensemble de données ARIA accessible au public.

**Orlando, Prokofyeva [46]** ont utilisé un ensemble d'approches de deep learning et de machine learning pour la détection de lésions rouges à l'aide de trois bases de données publiques, à savoir DIARETDB1, e-optha et Messidor. Les caractéristiques qu'ils ont extraites étaient basées sur l'intensité et la forme en appliquant l'architecture d'apprentissage par transfert LeNet [48] avec 10 couches. Ils ont atteint une sensibilité de 97,21% et une AUC de 0,9347.

Le tableau (3.2) récapitule quelques recherches importantes vues en haut, et qui ont appliqué les différentes approches pour la détection de la rétinopathie diabétique.

Classification		Source	Base de données	Méthode	Symptômes détectés	Résultats
ML	/	Biswas et al. [8]	Kaggle DR Dataset	SVM Classifier	Microanévrisme	Acc : 51.8%
DL	Novel Architectures	Yu et al. [6]	/	CNN (16 couches)	Exudates	SE : 88.85%, SP : 96%, Acc : 91.92
		Chudzik et al. [7]	Kaggle DR Dataset	CNN (25 couches)	Microanévrisme	FROC : 0.562
	Transfer Learning	Mo et al. [11]	Kaggle DR Dataset	VGGNet	Vaisseaux rétiniens	SE : 77.79%, SP : 97.80, Acc : 95.21, AUC : 0.9782, Kappa : 0.7759
		Van Grinsven et al. [12]	Messidor	CNN (OxfordNet Architecture)	Hémorragies	SE : 91.90%, SP : 91.40%, AUC : 0.972
ML et DL	/	Arunkumar et Karthigaikumar. [8]	ARIA	SVM Classifier	Microanévrisme	SE : 79.32%, SP : 97.89%, Acc : 96,73%

TABLE 3.2 – Tableau représentatif de l'état de l'art de quelques techniques de détection des symptômes de la R.D.

## Conclusion

Dans ce chapitre, nous avons parlé des bases de l'apprentissage automatique et de l'apprentissage profond ainsi que les différentes approches utilisées pour construire des modèles qui peuvent nous aider pour la résolution de notre problématique. Dans le chapitre suivant, nous allons procéder à la réalisation et la conception de notre modèle.

# Chapitre 4

## Conception et Réalisation

### Sommaire

---

4.1	Conception du modèle . . . . .	46
4.2	Implémentation . . . . .	50
4.3	Entraînement . . . . .	55
4.4	Evaluation . . . . .	58
4.5	Tests et Observations . . . . .	61
4.6	Présentation des interfaces du système . . . . .	63

---

## Introduction

Dans ce chapitre, les différentes parties de notre modèle seront présentées en spécifiant l'environnement et les outils de développements, l'implémentation de la base de données ainsi que les résultats d'expérimentation.

### 4.1 Conception du modèle

Vu que notre travail consiste à traiter des images médicales, le réseau de neurones convolutifs (CNN) est le plus adapté pour modéliser notre architecture.

#### 4.1.1 CNN Architecture

Les CNN comportent deux parties bien distinctes :

— **Une partie convolutive**

Elle fonctionne comme un **extracteur de caractéristiques** des images. Une image est passée à travers 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. Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN [25] .

— **Une partie classification**

Cette partie est constituée de couches entièrement connectées (perceptron multicouche). Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image [25].

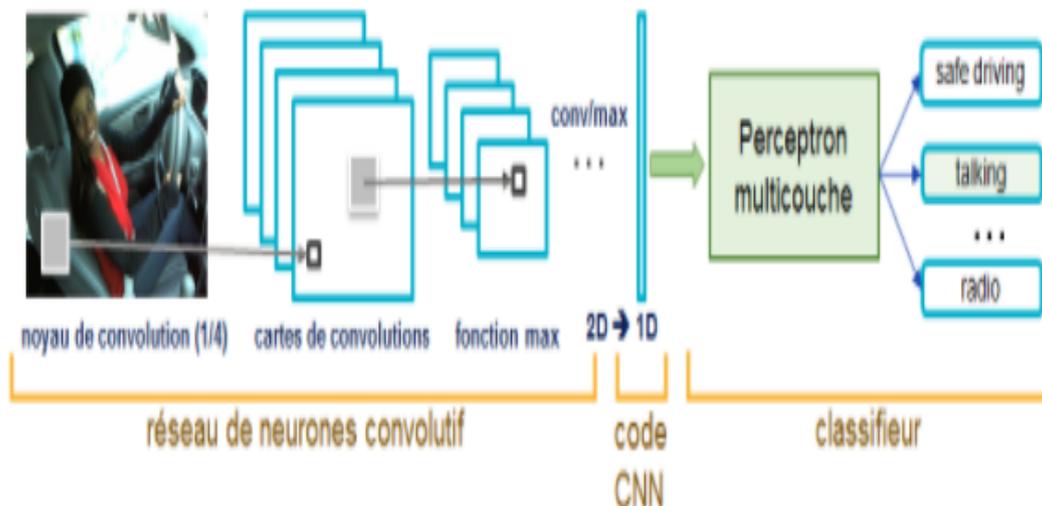


FIGURE 4.1 – Architecture d'un CNN [25].

En utilisant les CNN, nous avons opté pour la méthode du transfer learning avec l'architecture VGG16 qui nous a permis de gagner du temps, car le modèle n'a pas eu à apprendre à partir de zéro étant donné qu'il a déjà été pré-entraîné sur la base de donnée ImageNet.

### 4.1.2 Modélisation du réseau de neurones

#### Architecture du réseau de neurones

Nous avons opté pour l'architecture VGG16 qui est une architecture de réseau neuronal de convolution (CNN). Ce réseau est un assez grand réseau et il a environ 138 millions de paramètres. [21]

Le VGG16 fonctionne selon les étapes suivantes :

- a) Première couche : une image  $I$  en entrée
- b) Seconde couche : l'image est convoluée avec plusieurs filtres pour extraire différentes caractéristiques (une couche avec plusieurs images).
- c) Troisième couche : on réalise un Pooling (Le Max-Pooling est un processus de discrétisation basé sur des échantillons. Son objectif est de sous-échantillonner une représentation d'entrée (image, matrice de sortie de couche cachée, etc.) en réduisant sa dimension) sur les images convoluées pour réduire la taille des images (voir figure 4.2)

- d) Les étapes b et c se répètent jusqu'à obtention des images qu'on ne peut plus réduire. On remarque que les images ont diminué en taille et gagné en profondeur.
- e) Les images de petites tailles sont aplaties pour former un vecteur qui sera soumis au réseau de neurones pour classification (voir figure 4.2)

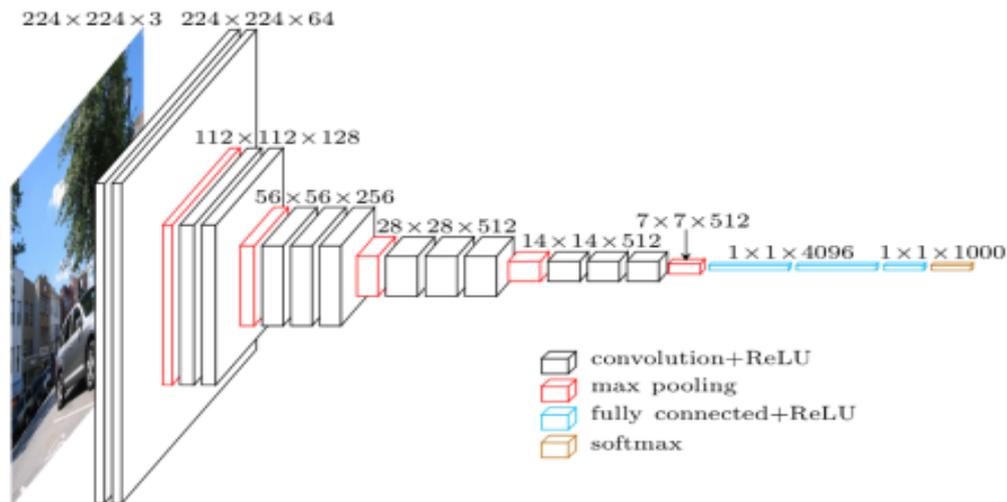


FIGURE 4.2 – Architecture de VGG16[23]

### 4.1.3 Modélisation du VGG16 pour la classification de la RD

Il existe 5 classes différentes pour le diagnostic de la RD à savoir :

- 0 : No-DR
- 1 : Minime
- 2 : Modéré
- 3 : Sévère
- 4 : Proliférante

- Comme le VGG16 possède 1000 classes de sortie, nous l'avons adapté pour donner 5 sorties qui correspondront à la classification de la RD citée ci-dessus, donc la couche de sortie possède 5 neurones.

- Le VGG16 comporte 16 couches, 13 couches de convolution et 3 couches entièrement connectées. - L'image initiale est une image RVB de dimension  $(224 \times 224 \times 3)$ .

- 2 couches convolutives avec des filtres de taille 3\*3, on y trouve 64 filtres différents, ce qui donne un volume de 224\*224\*64 en sortie.

\* Une couche de max-pooling de taille( 2\*2) est utilisée, son but est de réduire la longueur et la largeur de l'image d'un volume de 224\*224\*64 à 112\*112\*64.

\* Suivi de 2 autres couches de convolution avec 128 filtres. Cela donne la nouvelle dimension de 112\*112\*128.

\* Une deuxième couche de max-pooling est appliquée, le volume est réduit à 56\*56\*128.

\* 3 autres couches de convolution sont ajoutées avec 256 filtres, suivies d'une couche de max-pooling qui réduit la taille à 28\*28\*256.

\* 3 couches de convolution de 512 filtres, toujours de taille 3x3, suivies d'une couche de max-pooling qui réduit la taille à 14\*14\*512 \* 3 autres couches de convolution de 512 filtres, sont séparées par une couche max-pool (7\*7\*512)

\* Ensuite le volume 7\*7\*512 est aplati dans une couche entièrement connectée (FC) avec 4096 noeuds et une sortie softmax de 5 classes.

La figure (4.3) résume l'architecture du modèle VGG16 sur l'environnement Jupyter Notebook. Afin d'obtenir ce résumé, il suffit d'utiliser la commande `model.summary()`

Et on aura :

```
Entrée [4] model.summary()

Model: "sequential_1"
-----
Layer (type)                Output Shape              Param #
-----
conv2d_1 (Conv2D)           (None, 224, 224, 64)     1792
conv2d_2 (Conv2D)           (None, 224, 224, 64)     36928
max_pooling2d_1 (MaxPooling2 (None, 112, 112, 64)     0
conv2d_3 (Conv2D)           (None, 112, 112, 128)    73856
conv2d_4 (Conv2D)           (None, 112, 112, 128)    147584
max_pooling2d_2 (MaxPooling2 (None, 56, 56, 128)     0
conv2d_5 (Conv2D)           (None, 56, 56, 256)     295168
conv2d_6 (Conv2D)           (None, 56, 56, 256)     590080
conv2d_7 (Conv2D)           (None, 56, 56, 256)     590080
max_pooling2d_3 (MaxPooling2 (None, 28, 28, 256)     0
conv2d_8 (Conv2D)           (None, 28, 28, 512)     1180160
conv2d_9 (Conv2D)           (None, 28, 28, 512)     2359808
```

```
conv2d_10 (Conv2D)          (None, 28, 28, 512)      2359808
-----
max_pooling2d_4 (MaxPooling2 (None, 14, 14, 512)      0
-----
conv2d_11 (Conv2D)          (None, 14, 14, 512)      2359808
-----
conv2d_12 (Conv2D)          (None, 14, 14, 512)      2359808
-----
conv2d_13 (Conv2D)          (None, 14, 14, 512)      2359808
-----
max_pooling2d_5 (MaxPooling2 (None, 7, 7, 512)      0
-----
flatten_1 (Flatten)         (None, 25088)            0
-----
dense_1 (Dense)             (None, 4096)             102764544
-----
dense_2 (Dense)             (None, 4096)             16781312
-----
dense_3 (Dense)             (None, 5)                20485
=====
Total params: 134,281,029
Trainable params: 134,281,029
Non-trainable params: 0
-----
```

FIGURE 4.3 – Résumé du modèle VGG-16

## 4.2 Implémentation

Dans cette section, nous présenterons les différentes étapes que nous avons réalisé pour implémenter un réseau de neurones

### 4.2.1 Importer VGG16

Ici, dans cette partie, nous avons importé VGG16 de Keras avec des poids pré-entraînés qui ont été entraînés sur imagenet. Ici, comme on peut le voir dans la figure (4.4), le paramètre « include top » est défini sur « true ». Cela signifie que les poids pour l'ensemble de notre modèle seront téléchargés. Si cela est défini sur false, les poids pré-entraînés ne seront téléchargés que pour les couches de convolution et aucun poids ne sera téléchargé pour les couches denses.

```
Entrée [ |  
from keras.applications.vgg16 import VGG16  
  
vggmodel = VGG16(weights='imagenet', include_top=True)
```

FIGURE 4.4 – Importer VGG16

### 4.2.2 Adapter VGG16

Une fois le modèle téléchargé, nous l'avons adapté pour notre énoncé de problème qui consiste à classifier l'image en 5 classes, pour qu'il puisse apprendre davantage sur notre base de donnée. Ici, nous avons défini au debut que les 10 premières couches ne seront pas entraînables, elles garderons les poids déjà acquis lors du pré-entraînement sur ImageNet. En diminuant le nombre de couches non entraînables nous avons remarquer de meilleurs résultats a chaque fois, c'est pour cela que nous avons decider de définir le nombre à 5. Le reste des couches seron entrainer et les poids seront mis à jour à chaque epoch (cycle).

```
Entrée [ |  
  
for layer in vggmodel.layers[:5]:  
    layer.trainable = False
```

FIGURE 4.5 – Adapter VGG-16

Nous avons ajouté à l'avant-dernière couche du modèle (couche dense avec 4096 unités ) une couche (dense) de 5 unités à la fin, et softmax comme fonction d'activation . De cette façon nous avons supprimer la dernière couche du modèle VGG16 qui est faite pour prédire 1000 classes.

```
Entrée [ ] X= vggmodel.layers[-2].output
           predictions = Dense(5, activation="softmax")(X)
           model_final = Model(input = vggmodel.input, output = predictions)
```

FIGURE 4.6 – Adapter VGG-16 -2-

### 4.2.3 Choix de la base de données

Nous avons utilisé la base de données en provenance de la plateforme Kaggle. Dont Les images sont constituées d'images de la rétine sur lesquelles un filtre gaussien a été appliqué pour détecter la rétinopathie diabétique.

Ces images sont redimensionnées en 224x224 pixels afin qu'elles puissent être facilement utilisées avec de nombreux modèles d'apprentissage profond pré-formés.

Toutes les images sont classées en cinq répertoires selon la gravité/stade de la rétinopathie diabétique, à savoir :

- 0 - No\_DR
- 1 - Légère
- 2 - Modérée
- 3 - Grave
- 4 - Proliférate\_DR

```
Entrée [3] dataset
```

	id_code	diagnosis
0	000c1434d8d7	2
1	001639a390f0	4
2	0024cdab0c1e	1
3	002c21358ce6	0
4	005b95c28852	0
...	...	...
3657	ffa47f6a7bf4	2
3658	ffc04fed30e6	0
3659	ffc7b45f213	2
3660	ffd97f8cd5aa	0
3661	ffec9a18a3ce	2

3662 rows × 2 columns

FIGURE 4.7 – Structure de la base de données

Voici quelques exemples d'images que comportent notre base de données :

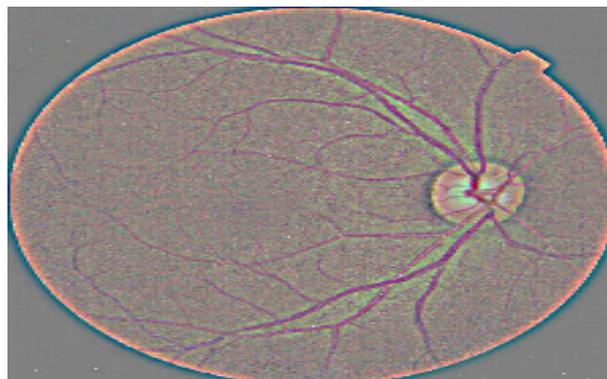


FIGURE 4.8 – NO\_DR

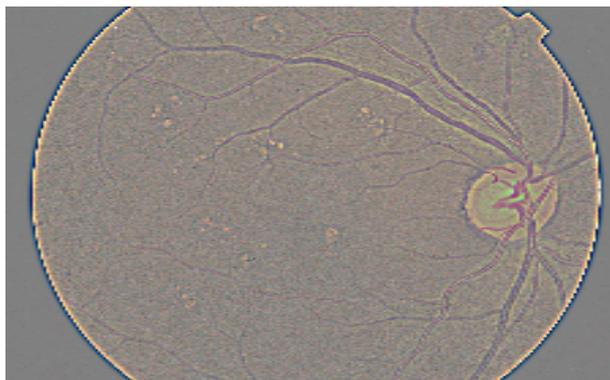


FIGURE 4.9 – Rétinopathie non proliférante minimale



FIGURE 4.10 – Rétinopathie non proliférante modérée

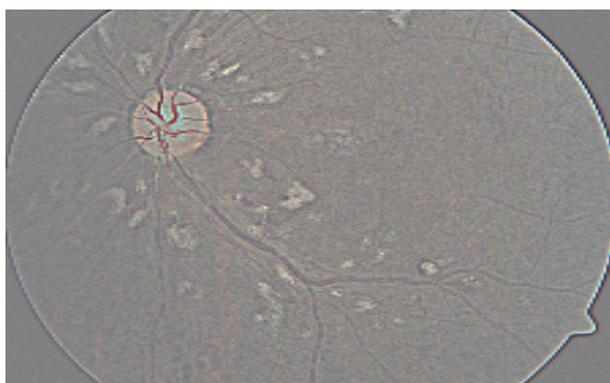


FIGURE 4.11 – Rétinopathie non proliférante sévère

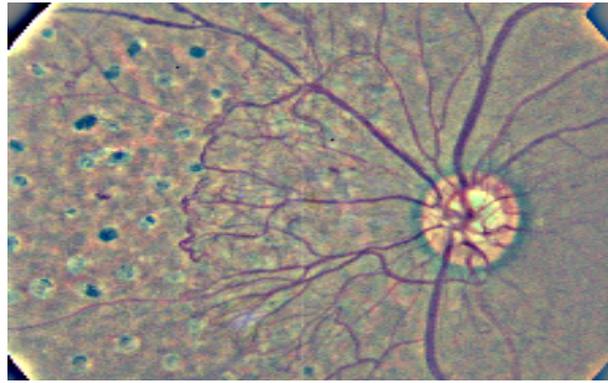


FIGURE 4.12 – Rétinopathie poliférante

Nous avons commencé par réduire notre base de donnée a 100 images de validation et 200 pour l'apprentissage. Une fois que nous avons procédé à l'apprentissage du modèle nous avons réalisé que le modèle arrive pas a généraliser correctement ( la précision des données de validation n'était pas satisfaisante).

Nous avons alors augmenté le nombre de données en ajoutant des images dans notre base de données. - 1917 images pour l'apprentissage - 896 images de validation (pour tester la capacité de généralisation de notre modèle sur de nouvelles données.)

## 4.3 Entraînement

L'entraînement permet au modèle d'apprendre la tâche qui lui est incombée, à savoir détecter la classe correspondante dans notre exemple. Le modèle est entraîné de façon supervisée, il va prendre le jeu de donnée comme entrée.

Afin d'entraîner le réseau de neurones en utilisant la méthode du d'apprentissage par transfert avec l'architecture VGG16, on passe par plusieurs étapes.

### 4.3.1 Propagation vers l'avant

Propagation avant signifie que la donnée traverse le réseau d'entrée à la sortie sans retour en arrière de l'information.

Dans la famille des réseaux à propagation avant, on distingue les réseaux monocouches (perceptron simple) et les réseaux multicouches (perceptron multicouche). Dans notre

cas, on s'intéresse au perceptron multicouche. Dans ce dernier, l'information entre par une couche d'entrée et sort par une couche de sortie. À la différence du perceptron simple, le perceptron multicouche dispose entre la couche en entrée et la couche en sortie une ou plusieurs couches dites « cachées ».

Pour effectuer tout ça, nous avons connecté notre base de données au modèle, en utilisant ces lignes de code.

```
Entrée [?] trdata = ImageDataGenerator()  
          traindata = trdata.flow_from_directory(directory="images_DR/Train",target_size=(224,224))  
          tsdata = ImageDataGenerator()  
          testdata = tsdata.flow_from_directory(directory="images_DR/val", target_size=(224,224))
```

FIGURE 4.13 – Code permettant de relier la base de données au programme

L'entrée est introduite dans le réseau. Dans chaque couche, il y a une fonction d'activation spécifique et entre les couches, il y a des poids qui représentent la force de connexion des neurones. L'entrée traverse les différentes couches du réseau, ce qui génère finalement une sortie.

Les poids sont des variables du modèle qui sont mises à jour pour améliorer la précision du réseau. Un poids est appliqué à l'entrée de chacun des neurones pour calculer une donnée de sortie. Les réseaux de neurones mettent à jour ces poids de manière continue [50].

### 4.3.2 Descente de Gradient

C'est un algorithme d'optimisation qui vise à minimiser la fonction de coût ou à minimiser une erreur. Son objectif principal est de trouver les minima locaux ou globaux d'une fonction en fonction de sa convexité. Cela détermine dans quelle direction le modèle doit aller pour réduire l'erreur.

Il existe plusieurs types, par exemple l'optimiseur de la descente de gradient stochastique (SGD) qui effectue une mise à jour des paramètres pour chaque exemple d'apprentissage. Il existe aussi l'optimiseur Adam qui est un algorithme d'optimisation qui peut être utilisé à la place de la procédure classique de descente de gradient stochastique pour mettre à jour les poids du réseau de manière itérative en fonction des données d'apprentissage.

En fonction du déroulement de la fonction de perte, la fonction optimiser tentera de diminuer la perte. Ce mécanisme sera répété en un certain nombre d'epochs qu'on précisera lors de l'apprentissage.

Dans notre cas, nous avons d'abord opté pour le SGD mais les résultats n'étaient pas satisfaisants. Ce qui a fait qu'on s'est dirigé par la suite vers l'optimiseur Adam.

### 4.3.3 Fonctions d'activation

Nous avons mis en place un réseau neuronal convolutionnel (CNN) qui avait deux fonctions de classification : softmax, et ReLU

Nous avons opté pour ReLU

— **Rectified Linear Units (ReLU) :**

Dans l'architecture VGG16, la fonction d'activation ReLU (Unité linéaire rectifiée) est ajoutée à chaque couche de convolution afin que toutes les valeurs négatives ne soient pas transmises à la couche suivante [23].

— **Softmax :**

Nous avons utilisé la fonction mathématique softmax à la sortie de la nouvelle couche entièrement connectée ajoutée avec 5 classes de sorties, pour convertir le score de chacune des 5 sorties en probabilité d'appartenance de l'image donnée en entrée à chacune des 5 classes. La valeur la plus élevée correspond donc à la classe à laquelle appartient l'image donnée, prédite par le modèle.

### 4.3.4 La fonction de perte

La fonction de perte est utilisée comme mesure de précision pour identifier si notre réseau de neurones a appris les modèles avec précision ou non à l'aide des données d'apprentissage. Ceci est complété en comparant les données d'apprentissage avec les données

de test.

Dans cette étape la fonction de perte va comparer et mesurer la distance entre la valeur estimée et sa vraie valeur.

Comme on a une couche dense de 5 unités à la fin (la sortie du modèle est catégorique), on a opté pour la fonction de perte (`categorical_crossentropy`), il s'agit d'une activation Softmax plus une perte d'Entropie croisée . Si nous utilisons cette perte, nous entraînerons un CNN pour produire une probabilité pour chaque image. Il est utilisé pour la classification multi-classes [49].

### 4.3.5 Propagation vers l'arrière

La méthode de rétropropagation est utilisé pour déterminer comment les poids doivent être ajustés. Les poids sont ajustés au cours de la méthode de descente de gradient.

## 4.4 Evaluation

Dans notre cas, nous avons opté pour l'apprentissage supervisé. Durant le processus d'entraînement du réseau, le modèle apprend la relation entre les entrées (les images) et les sorties (la catégorie de chaque image donnée en entrée).

Au debut nous avons essayé l'approche du fine-tuning total, le but c'est d'entraîner tout le réseau, donc il faut rendre toutes les couches "entraînables" grâce a cette ligne de code (`layer.trainable = True` ).

Mais par la suite nous avons réalisé que pour ça il faut avoir une base de donnée qui comporte un grand nombre d'images et qu'il faut l'exécuter sur une machine assez puissante, ce qui n'est pas le cas de la notre. Donc le modèle a commencé l'apprentissage et au bout de l'epoch 3, il s'est arrêté en raison du manque d'espace et de capacité mémoire. Cette méthode aussi comporte le risque de sur-apprentissage a cause de la limite des données( le modèle ne sera pas capable de généraliser sur les données de test par la suite).

Les paramètres d'apprentissage du modèle utilisé sont mit au nombre maximal supporté par nos machines respectives, et grace a la fonction early stop, le modèle a arrêté de s'entraîner dès que le taux de généralisation "Val\_accuracy" n'augmente pas ou diminue en valeur.

```
Entrée | from keras.callbacks import ModelCheckpoint, EarlyStopping
        checkpoint = ModelCheckpoint("vgg16_1.h5", monitor='val_acc', verbose=1,
                                    save_best_only=True, save_weights_only=False, mode='auto', period=1)
        early = EarlyStopping(monitor='val_acc', min_delta=0, patience=20, verbose=1, mode='auto')
        hist = model.fit_generator(steps_per_epoch=100, generator=traindata, validation_data=testdata,
                                   validation_steps=10, epochs=100, callbacks={checkpoint, early})

Epoch 1/100
3/100 [ ..... ] - ETA: 0:25:03 - loss: 2265.5818 - accuracy: 0.1618
```

FIGURE 4.14 – Figure illustrant le résultat obtenu pour l'apprentissage avec la première méthode

Pour remédier a ça nous avons tester une autre méthode de l'apprentissage par transfert (extraction de features), dans ce cas l'entraînement se fera uniquement sur la dernière couche et on ne ré-entraîne pas les autres couches et c'est cette ligne de code ((layer.trainable = False ) qui nous a permis de faire ça. Nous avons remarqué que l'apprentissage se fais lentement et on a quand même un résultat satisfaisant (Accuracy = 70%) mais on a pu l'améliorer en remplaçant cette méthode par celle du fine-tuning partiel, qui est la plus adapté pour notre cas vu que notre dataset ne comporte pas un grand nombre d'images et de plus les images que nous avons utilisé sont très différentes des images du pré-entraînement qui s'est fait sur la dataset ImageNet où nous avons obtenu de bien meilleurs résultats (Accuracy = 99%, val\_accuracy = 79%).

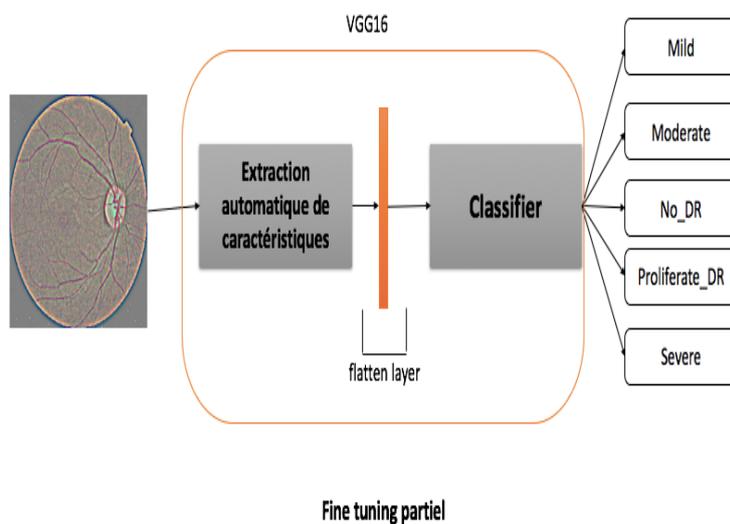


FIGURE 4.15 – Processus du fine tuning partiel sur le modèle utilisé

#### 4.4.1 Résultats d'apprentissage

Afin d'entraîner correctement notre modèle et avoir des résultats satisfaisants, nous avons essayé plusieurs méthodes, et paramètres énoncés ci dessous et résumées dans le tableau (4.1)

- Dans le premier test on a rencontré un des problèmes les plus communs dans l'apprentissage profond, le sur-apprentissage : le modèle est aléatoire, il n'est pas capable de généraliser sur les nouvelles données.
- Le deuxième test nous montre que le modèle n'est pas assez précis avec (accuracy=65%), le modèle n'arrive pas à classifier convenablement les images.
- Dans le troisième, en rajoutant le nombre d'epochs, on a pu obtenir de meilleurs résultats que le précédent test mais toujours pas assez, le modèle n'arrive pas à différencier entre les classes (modéré, sévère et proliférante), et même quand une bonne classification est faite de la part du modèle, le pourcentage de certitude reste très faible.
- Nous avons obtenu de meilleurs résultats en changeant l'optimiseur et le taux d'apprentissage dans le quatrième test, et encore beaucoup mieux dans le dernier test

en modifiant les hyper-paramètres (epoch et pas/epoch).

Test	Optimizer	Taux d'apprentissage	Epoques	Pas/epoque	Accuracy	Validation accuracy	Méthode d'apprentissage par transfert
1	SGD	0.0001	100	2	100%	50%	Extraction de caractéristiques
2	SGD	0.0001	30	2	65%	60%	Fine tuning partiel
3	SGD	0.0001	70	2	73%	85%	Fine tuning partiel
4	Adam	0.000001	100	50	70%	63%	Fine tuning partiel
5	Adam	0.000001	30	59	99%	79%	Fine tuning partiel

TABLE 4.1 – Résumé des résultats d'apprentissage sur le modèle

## 4.5 Tests et Observations

Afin de vérifier la validité et l'efficacité de notre système, plusieurs tests ont été effectués pour chacun des différents stades de la RD.

Quelques résultats sont mentionnés ci-dessous :

### Résultat pour une RD Proliférante

Une fois que l'image est sélectionnée, le système a réussi à détecter le stade correspondant (RD Proliferate) avec une précision de 97.98%. La figure (4.16) montre cela.

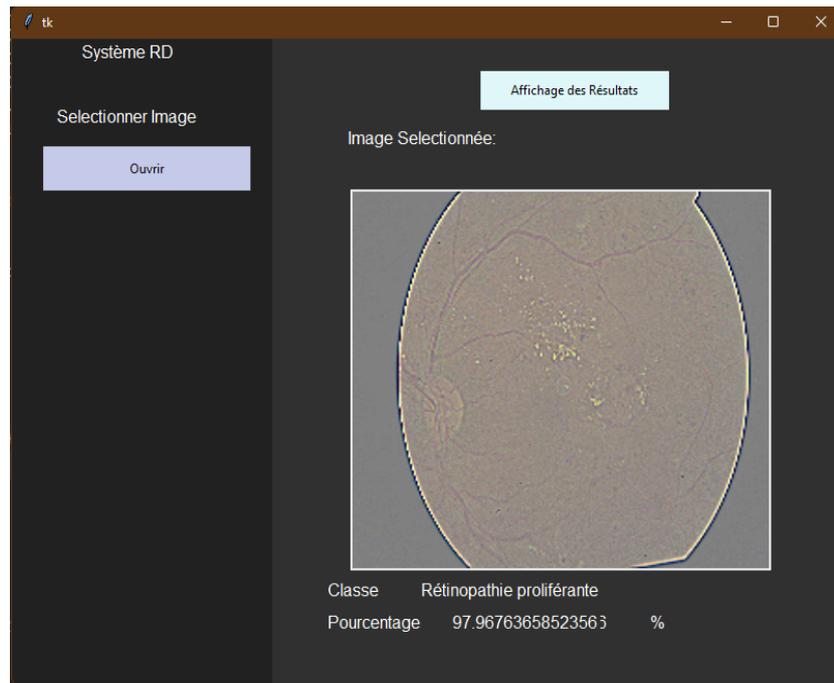


FIGURE 4.16 – Resultats de test pour DR Proliférante

### Résultat pour une RD Modérée

L'exemple ci-dessous montre un taux de reconnaissance de 94.50% pour une image du stade 'Modérée'

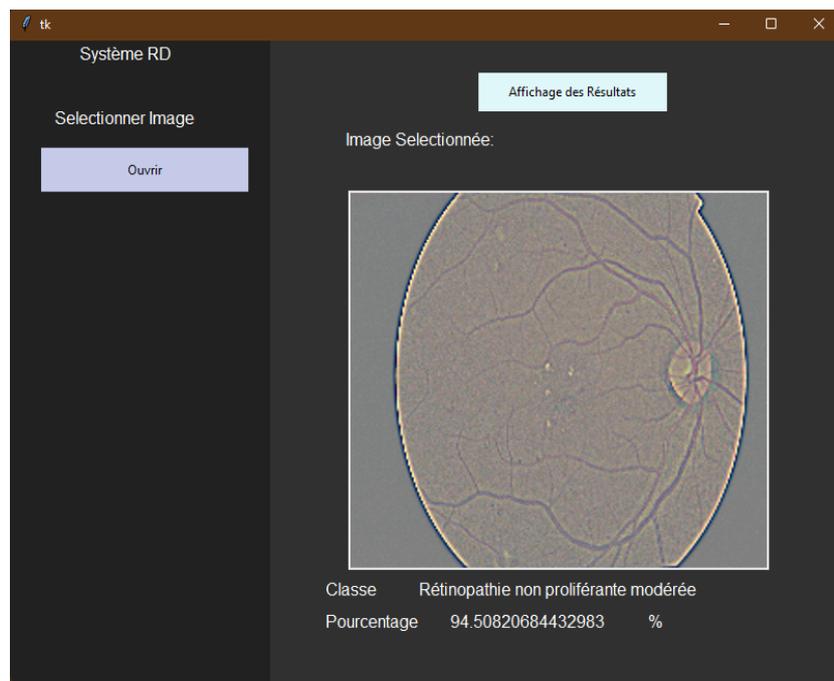


FIGURE 4.17 – Resultats de test pour DR Modérée

### Résultat pour une RD Sévère

Pour une DR Sévère, le système a réussi à obtenir un taux de précision de 84.62% comme le montre la figure (4.18).

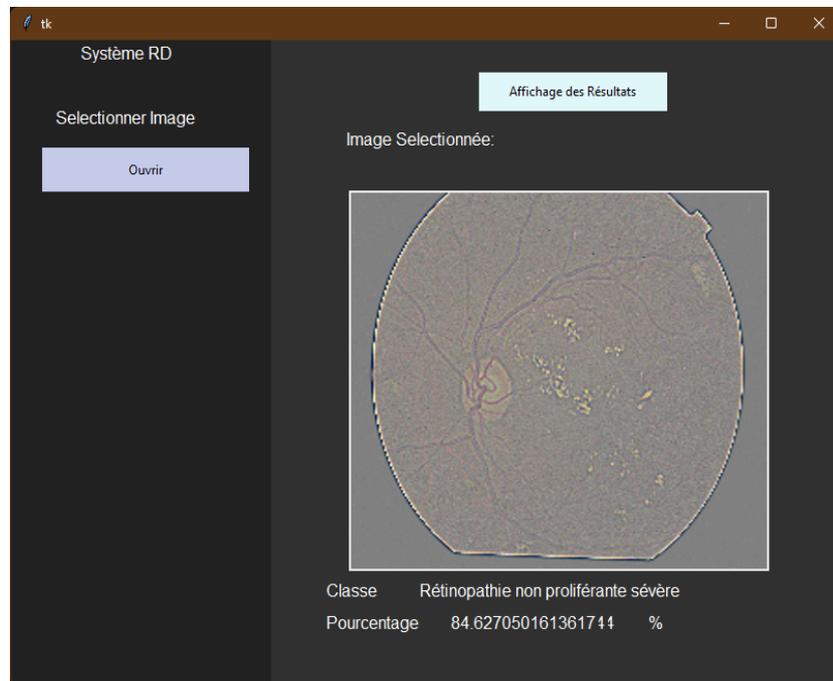


FIGURE 4.18 – Resultats de test pour DR Sévère

## 4.6 Présentation des interfaces du système

L'interface principale se compose d'un bouton (Ouvrir) qui permet de :

- Charger l'image qu'on souhaite analyser.
- Lancer l'analyse et donc le resultat.

En ouvrant l'application, on retrouve cette interface :

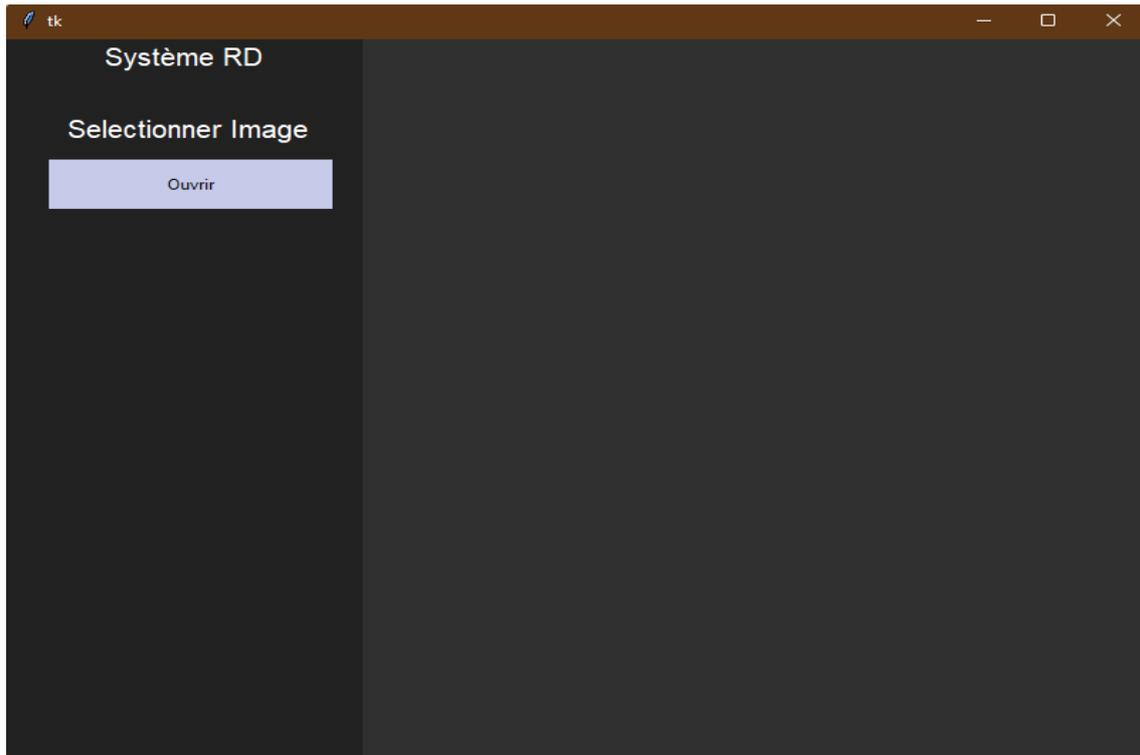


FIGURE 4.19 – Interface d'accueil

En cliquant sur ouvrir, ça nous dirige vers le bureau afin de sélectionner une image à partir de la base de données :

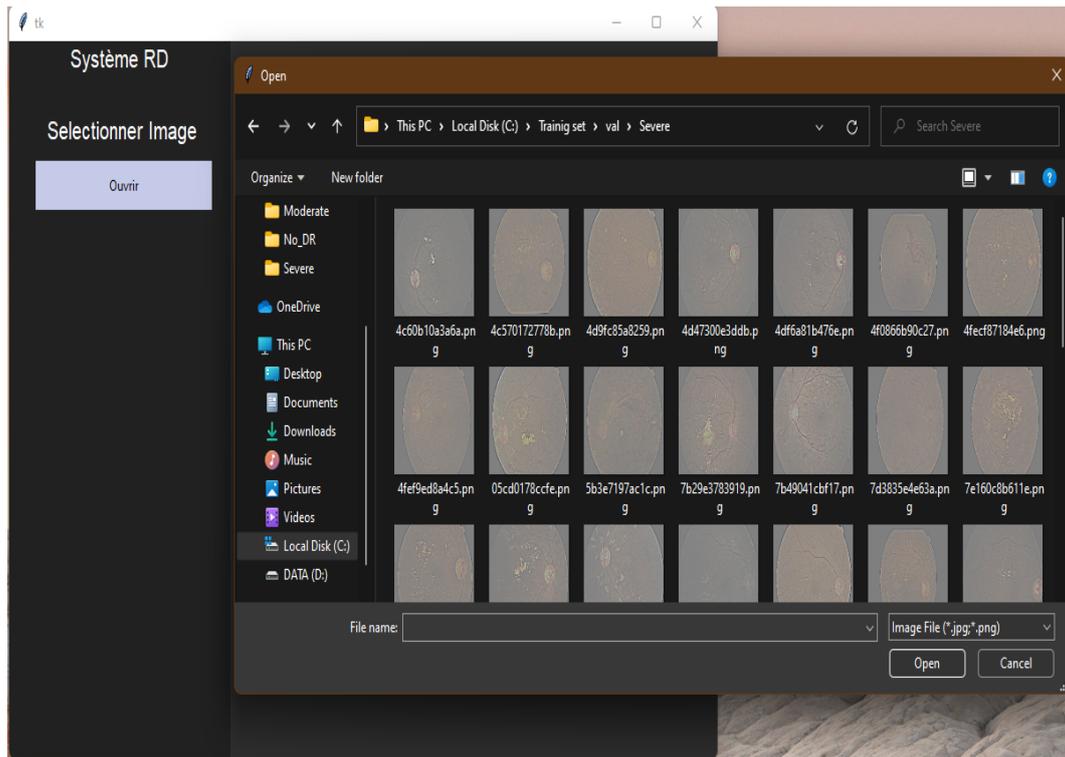


FIGURE 4.20 – Interface pour sélectionner une image de la base de données

Une fois l'image sélectionnée, place à l'analyse et aux résultats. La figure suivante montre le résultat obtenu en sélectionnant une image appartenant à la classe NO-DR.

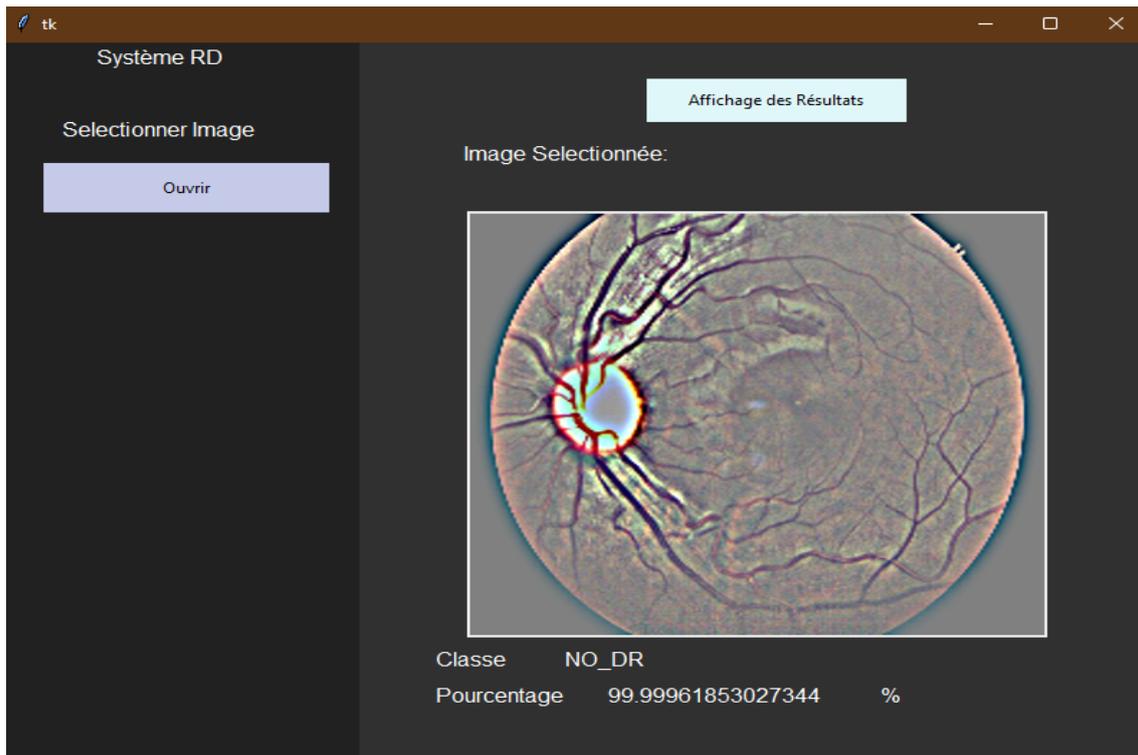


FIGURE 4.21 – Résultat d'une NO-DR

## Conclusion

Dans ce chapitre nous avons présenté en premier lieu la mise en oeuvre des différents étapes de conception et de réalisation de notre système. Nous avons ensuite donné quelques résultats et tests appliqués sur le système. Par la suite nous allons définir l'environnement de développement, les bibliothèques utilisées et le langage de programmation python.

## Environnement de développement

Pour notre projet, nous avons eu à utiliser 2 environnements à savoir ANACONDA pour la gestion des librairies et Jupyter Notebook pour écrire et exécuter le code dans la même interface utilisateur.

- **ANACONDA** : C'est une distribution Python libre qui intègre directement un grand nombre de packages (il n'est donc plus nécessaire de les installer, mais on peut en ajouter d'autres si nécessaire avec le gestionnaire de packages Conda). [python anaconda env pdf]

L'interface ANACONDA se présente ainsi :

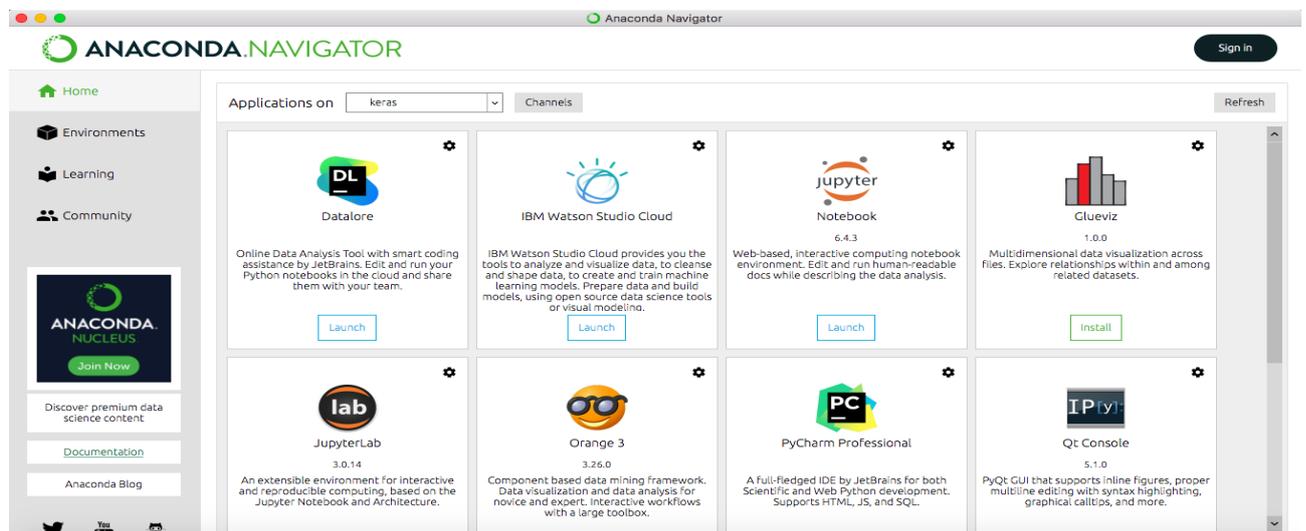


FIGURE 4.22 – Interface de la plateforme ANACONDA

Ce qui nous mènera à notre éditeur, afin de pouvoir saisir notre code. On a choisi Jupyter Notebook, afin d'y accéder on aura juste à cliquer sur Launch sur la plateforme ANACONDA.

- **Jupyter Notebook** : C'est une interface web open source qui permet d'inclure du texte, vidéo, audio, images, ainsi que la possibilité d'exécuter du code à partir de différents langages de programmation. Cette exécution est accomplée par voie de communication à un noyau de calcul (noyau). Par défaut, Jupyter Notebook comprend uniquement le noyau. Cependant, étant un projet open source, il a été

possible d'augmenter nombre de noyaux disponibles. Ceux-ci incluent les noyaux pour Octave, Julia, R, Haskell, Ruby, C/C++, Fortran, Java, SageMath, Scala, Matlab et Mathematica. Ainsi, cette interface peut marquer une étape dans la normalisation des moyens de fournir du contenu scientifique, sans être limitée à une seule langue. Cette polyvalence a permis de répandre son utilisation dans les environnements éducatifs et de recherche. Dans ce dernier domaine, il est de plus en plus courant d'inclure Cahiers Jupyter avec des calculs, des données et des chiffres supplémentaires pour les documents scientifiques. [jupyter notebooks pdf]

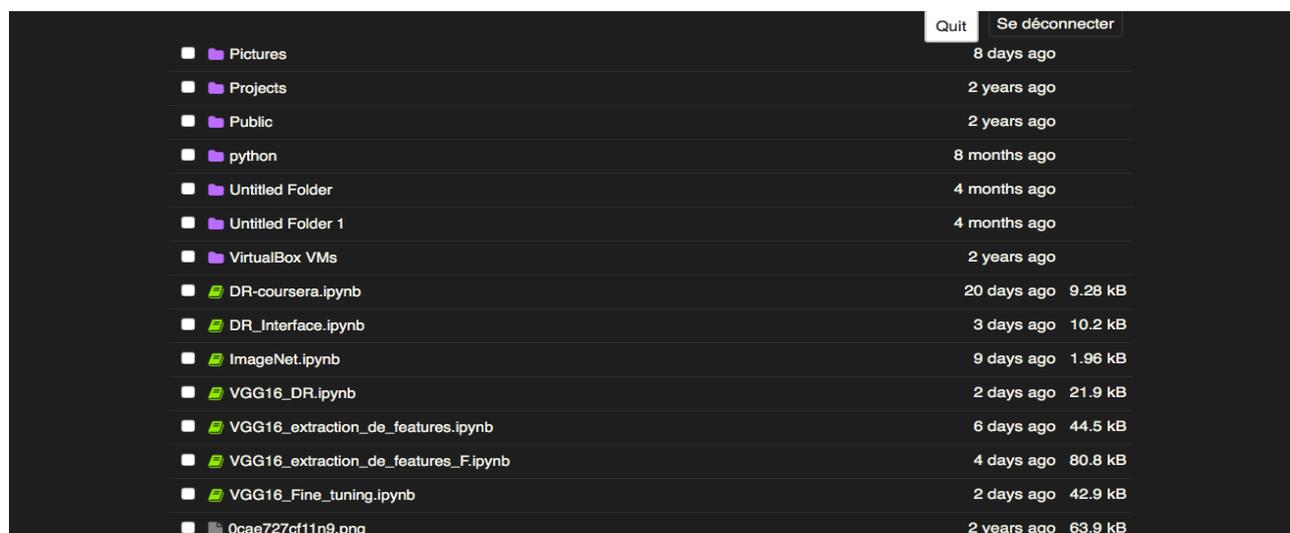


FIGURE 4.23 – L'interface d'accueil de Jupyter Notebook

## Librairies utilisées :

- **Keras** : Keras est une API (Application Programming Interface) du deep learning écrite en Python, fonctionnant sur le dessus de la plate-forme d'apprentissage automatique TensorFlow. Il a été développé en mettant l'accent sur l'expérimentation rapide. Il est essentiel de pouvoir passer de l'idée au résultat le plus rapidement possible pour faire de bonnes recherches. [18]

Nous avons donc utilisé cette librairie afin d'implémenter le modèle VGG-16.

- **matplotlib** Matplotlib est une bibliothèque complète pour créer des visualisations

statiques, animées et interactives en Python. [19]

Il est par exemple possible de créer des tracés, des histogrammes, des diagrammes à barre et tous types de graphiques à l'aide de quelques lignes de code. Il s'agit d'un outil très complet, permettant de générer des visualisations de données très détaillées. [21]

Cette librairie nous a servi à visualiser nos résultats sous forme de graphe.

## Langage de programmation utilisé

Pour la réalisation de ce travail nous avons opter pour python comme langage de programmation

### 4.6.1 Definition :

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl. [22]

### 4.6.2 Utilisation :

Python est un langage de programmation qui peut s'utiliser dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées. Il est cependant particulièrement utilisé comme langage de script pour automatiser des tâches simples mais fastidieuses, comme un script qui récupérerait la météo sur Internet ou qui s'intégrerait dans un logiciel de conception assistée par ordinateur afin d'automatiser certains enchaînements d'actions répétitives (voir la section Adoption). On l'utilise également comme langage de développement de prototype lorsqu'on a besoin d'une application fonctionnelle avant de l'optimiser avec un langage de plus bas niveau. Il est particulièrement répandu dans le monde scientifique, et possède de nombreuses bibliothèques optimisées destinées au calcul numérique. [22]

# Conclusion générale

La rétinopathie diabétique (RD) ou la maladie oculaire diabétique est une condition médicale dans laquelle les dommages se produisent à la rétine en raison du diabète causant la cécité. La détection de la RD à un stade précoce est très importante pour sauver la vision du patient. Grâce à la puissance de l'IA et l'apprentissage profond, les médecins seront en mesure de prévenir la cécité avant qu'elle ne se produise, ce qui est très puissant. Cette technique peut donc être cruciale, en particulier pour les personnes qui vivent dans des zones rurales où le dépistage médical et les médecins expérimentés sont limités. C'est pour cette raison que nous avons essayé d'élaborer un système capable de prédire l'existence et la gravité de cette maladie. Pour cela nous avons exploité la notion des réseaux de neurones convolutifs en utilisant l'approche de l'apprentissage par transfert pour l'aide au diagnostic médical de la rétinopathie diabétique.

L'apprentissage par transfert nous a permis d'exploiter le réseau VGG16 pré-entraîné sur d'autres problèmes, que nous avons adapté sur la classification de la rétinopathie diabétique, l'avantage de cette méthode c'est qu'elle n'est pas coûteuse en terme de matériel, de quantité de données annotées nécessaires (images) et de capacités de calcul.

L'implémentation du modèle a prouvé son efficacité dans la classification et la prédiction de la rétinopathie diabétique, avec un taux de précision (accuracy) et un taux de généralisation (val\_accuracy) de 79% avoisinant les 99%. Le modèle a été entraîné sur un échantillon de 1917 images du fond de l'oeil, et 896 ont servi pour la validation. On peut dire donc que les résultats obtenus pour ce modèle ont été très satisfaisantes.

### **Perspectives**

- Améliorer le taux de validation de notre modèle.
  
- Comme les patients sont censés avoir un examen régulier de la rétine, il est important de développer une procédure de détection de changement progressif dans la rétine au fil du temps qui pourrait être intéressant pour le dépistage de la RD.

# Bibliographie

- [1] Colin A et al, *Maladies vasculaires rétiniennes : la rétinopathie diabétique, Rétine et vitré*, 2015-2016.
- [2] Uzair Ishtiaq, Sameem Abdul Kareem, Erma Rahayu Mohd Faizal Abdullah<sup>1</sup>, Ghulam Mujtaba, Rashid Jahangir, Hafiz Yasir Ghafoor <sup>1,2</sup>, *Diabetic retinopathy detection through artificial intelligent techniques : a review and open issues*, Springer, *Multimed Tools Appl* 79, 15209–15252 (2020). <https://doi.org/10.1007/s11042-018-7044-8>
- [3] FEROU AMEL, *analyse des images couleur du fond d'oeil pour l'aide au diagnostic en ophtalmologie : application a la détection des pathologie rétiniennes*, Doctorat Électronique Biomédicale, université ABOU—BAKR—BELKAID tlemcen, département de génie électrique et électronique, Mai 2014.
- [4] Zhang, W., Zhong, J., Yang, S., Gao, Z., Hu, J., Chen, Y., Yi, Z. (2019), *Automated identification and grading system of diabetic retinopathy using deep neural networks*, *Knowledge-Based Systems*, volume(175), pages (12-25), 2019.
- [5] [https://www.fmc-tourcoing.org/PROGRAMME\\_FMC/DIABETE1000/DIABETEFMI/alfedia retina.html](https://www.fmc-tourcoing.org/PROGRAMME_FMC/DIABETE1000/DIABETEFMI/alfedia retina.html), consultle(15/05/2021)
- [6] Ambika Shetkar, C. Kiran Mai, and C. Yamini Ka, *Diabetic Symptoms Prediction Through Retinopathy*, *Machine Learning Technologies and Applications : Proceedings of ICACECS*, 13, 2021.
- [7] XU, Kele, FENG, Dawei, et MI, Haibo. Deep convolutional neural network-based early automated detection of diabetic retinopathy using fundus image. *Molecules*, 2017, vol. 22, no 12, p. 2054.
- [8] Saroj Kr. Biswas, Rohit Upadhya, Nipan Das, Dolly Das, Manomita Chakraborty, and Biswajit Purkayastha, *An Intelligent System for Diagnosis of Diabetic Retinopathy*, *Soft Computing for Problem Solving* 2019, pages(97–110), Année(2020).

- [9] 2D/3D medical image segmentation by embedding EfficientNet in Convolutional neural network : Application on BraTS challenge 2020
- [10] Efficient Convolutional Neural Network for 2D Echochardiographic images segmentation : Application on CAMUS DataSet
- [11] Varun Gulshan et al, 2016, Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs
- [12] VAN GRINSVEN, Mark JJP, VAN GINNEKEN, Bram, HOYNG, Carel B., et al. Fast convolutional neural network training using selective data sampling : Application to hemorrhage detection in color fundus images. IEEE transactions on medical imaging, 2016, vol. 35, no 5, p. 1273-1284.
- [13] Zhou W et al (2017) Automatic microaneurysm detection using the sparse principal component analysis- based unsupervised classification method. IEEE Access 5 :2563–2572
- [14] Xiao D, et al (2017) Retinal hemorrhage detection by rule-based and machine learning approach
- [15] Jyotismita Chaki, S. Thillai Ganesh, S.K Cidham, S. Ananda Theertan, Machine learning and artificial intelligence based Diabetes Mellitus detection and self-management : A systematic review, (2020)
- [16] Nacerdine larbi,segmentation d'images avec le deep learning, Mémoire de master, Université Mouloud Mammeri De Tizi-Ouzou, departement d'automatique, (2018).
- [17] [https://blog.octo.com/classification-dimages-les-reseaux-de-neurones-convolutifs-en-toute-simplicité/?fbclid=IwAR2Y-nELSWKKVz5bhOdD4hljc7fU9\\_VOygBbEqf1gBCWJzfkSjBRdW4Caxo](https://blog.octo.com/classification-dimages-les-reseaux-de-neurones-convolutifs-en-toute-simplicité/?fbclid=IwAR2Y-nELSWKKVz5bhOdD4hljc7fU9_VOygBbEqf1gBCWJzfkSjBRdW4Caxo), consulté le (02/09/2021).
- [18] <https://keras.io/about/>, consulté le (05/09/2021)
- [19] <https://matplotlib.org/>, consulté le (05/09/2021).
- [20] <http://dspace.univ-tlemcen.dz/bitstream/112/11251/1/Ms.EBM.Bensenousi%2BAhmed%20Brahim>
- [21] <https://datascientest.com/matplotlib-tout-savoir>, consulté le (06/09/2021).
- [22] <https://www.python.org/doc/essays/blurb/>, consulté le (12/09/2021)
- [23] <https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c>, consulté le (15/08/2021)

- [24] Abien Fred M. Agarap, (2019), Deep Learning using Rectified Linear Units (ReLU).
- [25] <https://datascientest.com/convolutional-neural-network?fbclid=IwAR2LYn4uT5gghyjLKfY8oSBe73xuABmW19O81VcenMwTKQAR4dUtVcnJfUo> , consulté le (22/09/2021)
- [26] Aurélien Garivier, Apprentissage supervisé , (12 et 13 juin 2013).
- [27] Dongare, A. D., Kharde, R. R., Kachare, A. D. (2012). Introduction to artificial neural network. International Journal of Engineering and Innovative Technology (IJEIT), 2(1), 189-194.
- [28] <https://www.aboutkidshealth.ca/FR/Article?contentid=1941language=French> , consulté le (25/09/2021)
- [29] <https://machinelearningmastery.com/softmax-activation-function-with-python/> , consulté le (25/09/2021)
- [30] <https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623> , consulté le (27/09/2021)
- [31] <https://www.xlstat.com/fr/solutions/fonctionnalites/indicateurs-de-performance-de-modeles> , consulté le (29/09/2021)
- [32] <https://www.kdnuggets.com/2020/02/deep-neural-networks.html> , consulté le (23/08/2021)
- [33] <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00444-8> , consulté le (14/08/2021)
- [34] <https://www.sciencedirect.com/science/article/pii/S2352914820302069> , consulté le (07/09/2021)
- [35] <https://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/deep-learning-vs-machine-learning/> , consulté le (8/10/2021)
- [36] <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/> , consulté le (9/10/2021)
- [37] van Grinsven M et al (2016) Fast convolutional neural network training using selective data sampling : application to hemorrhage detection in color fundus images. IEEE Trans Med Imaging 35(5) :1273–1284
- [38] Takahashi H et al (2017) Applying artificial intelligence to disease staging : Deep learning for improved staging of diabetic retinopathy. PLoS One 12(6) :e0179790

- [39] Shu Wei Ting D et al (2017) Development and validation of a deep learning system for diabetic retinopathy and related eye diseases using retinal images from multiethnic populations with diabetes. *JAMA : J Am Med Assoc* 318(22) :2211–2223
- [40] Perdomo O, Arevalo J, Gonzalez FA (2017) Convolutional network to detect exudates in eye fundus images of diabetic subjects
- [41] Mo J, Zhang L (2017) Multi-level deep supervised networks for retinal vessel segmentation. *Int J Comput Assist Radiol Surg* 12(12) :2181–2193
- [42] Yu S, Xiao D, Kanagasingham Y (2017) Exudate detection for diabetic retinopathy with convolutional neural networks
- [43] Doshi D, et al (2016) Diabetic retinopathy detection using deep convolutional neural networks. in 2016 International Conference on Computing, Analytics and Security Trends (CAST)
- [44] Chudzik P et al (2018) Microaneurysm detection using fully convolutional neural networks. *Comput Methods Prog Biomed* 158 :185–192
- [45] Arunkumar R, Karthigaikumar P (2017) Multi-retinal disease classification by reduced deep learning features. *Neural Comput Applic* 28(2) :329–334
- [46] Orlando JI et al (2018) An ensemble deep learning based approach for red lesion detection in fundus images. *Comput Methods Prog Biomed* 153(C) :115–127
- [47] Cigizoglu HK, Alp M (2006) Generalized regression neural network in modelling river sediment yield. *Adv Eng Softw* 37(2) :63–68
- [48] LeCun Y et al (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11) :2278– 2324
- [49] [https://gombru.github.io/2018/05/23/cross\\_entropy\\_loss/](https://gombru.github.io/2018/05/23/cross_entropy_loss/), consulté le (26/09/2021)
- [50] <https://moncoachdata.com/blog/comprendre-les-reseaux-de-neurones/>, consulté le (12/10/2021)
- [51] <https://cacm.acm.org/magazines/2021/7/253464-deep-learning-for-ai/fulltext>, consulté le (17/10/2021)
- [52] [https://ebookreading.net/view/book/EB9781787121423\\_395.html](https://ebookreading.net/view/book/EB9781787121423_395.html), consulté le (19/10/2021)

