



## Mémoire de Master

Filière Électronique  
Spécialité Électronique des Systèmes Embarqués

présenté par

DJERIOU Faiza

&

BRAHMI Fatma Zohra Radia

# Deep Learning pour la Détection de rétinopathie diabétique

Proposé par : MmeBOUGHERIRA Hamida

Et Co-encadré par : Mme.NACEUR Djamilia

Année Universitaire 2018-2019

*Nous tenons à remercier tout d'abord notre encadreur :  
Mme BOUGHERIRA Hamida, pour tous ses conseils, son  
soutien, son aide, et son orientation.*

*Nous remercions tous les enseignants du département  
d'électronique, et notre promotion 2018-2019 Master  
électronique des systèmes embarqués.*

*Nos sincères remerciements vont aussi à toutes les  
personnes qui nous ont soutenues de près ou de loin dans  
la réalisation de ce travail. En particulier nos parents et  
amis qui nous ont aidés dans ce projet.*

---

## ملخص:

لقد أنشأنا خوارزمية تعلم عميقة من أجل تنفيذ شبكة عصبية قادرة على اكتشاف اعتلال الشبكية السكري ، والذي يبدأ بتصنيف المرضى وغير المرضى وفقاً للتعلم ، ثم ننتقل إلى اكتشاف الحالات الشاذة التي يمكن العثور عليها في قاع العين فيما يتعلق باعتلال الشبكية السكري ، ولهذا استخدمنا MATLAB وشبكة العصب CNN ، ثم جربنا هذه الخوارزمية مع العديد من المعلمات وحصلنا على نتائج مختلفة للتعلم والتحقق قمنا بتصنيف الصور على أنها مريضة وليست مريضة .

**كلمات المفتاحية:** التعلم العميق CNN,, اعتلال الشبكية السكري, والشبكات العصبية, ماتلاب.

---

## Résumé :

Nous avons conçu un algorithme de deep Learning afin d'implémenter un réseau neurone capable de détecter la rétinopathie diabétique, celui-ci commence par classifier les malades et non malades d'après l'apprentissage, ensuite on passe à la détection des anomalies qu'on peut trouver dans un fond d'œil en relation avec la rétinopathie diabétique, pour cela nous avons utilisé MATLAB et un réseau de neurone de type CNN, ensuite on a essayé cet algorithme avec plusieurs paramètres et obtenue différents résultats d'apprentissage et de validation, nous avons classé les images en malade et non malade .

**Mots clés :** Deep Learning; CNN; Rétinopathies diabétiques, Réseaux de neurones, Matlab.

---

## Abstract :

We have created a deep learning algorithm in order to implement a neural network capable of detecting diabetic retinopathy, which begins by classifying patients and non-patients according to learning, then we proceed to the detection of anomalies that it can be found in a fundus in relation to diabetic retinopathy, for this we used MATLAB and a CNN neuron network, then we tried this algorithm with several parameters and obtained different learning and validation results we have classified the images as sick and not sick.

**Keywords :** Deep Learning ; CNN ; Diabetic retinopathy, Neural network, Matlab.

---

## Listes des acronymes et abréviations

VeGF: VascularEndothelialGrowthFactor.

RD : Rétinopathie Diabétique.

DMLA : Dégénérescence Maculaire Liée à l'Age.

RDNP : Rétinopathie DiabétiqueNon Proliférante .

RDP : Rétinopathie DiabétiqueProliférante .

PPR : PhotocoagulationPanRétinienne.

RNNS : Réseaux de Neurones Récurrents .

MLP : Multi Layer Perceptron .

ReLu : RectifiedLinear Unit.

IA :Intelligence Artificiel.

ML :Machine Learning .

NN :Neural Network .

SVM: Support Vector Machine .

DL: Deep Learning.

FC: Fully Connected.

CNN: Convolutional Neural Network .

R-CNN : Region-based Convolutional Neural Networks.

GPU:GraphicsProcessing Unit.

DDA : date du dernier accès .

## Table des matières :

|                                                                   |    |
|-------------------------------------------------------------------|----|
| Introduction générale .....                                       | 1  |
| Chapitre 1 Généralités .....                                      | 3  |
| 1.1 Introduction .....                                            | 3  |
| 1.2 Partie A Aspect médical .....                                 | 3  |
| 1.2.1 L'anatomie de l'œil .....                                   | 3  |
| 1.2.2 La rétine .....                                             | 4  |
| a Définition de la rétine .....                                   | 4  |
| b Constitution de la rétine .....                                 | 4  |
| c Fonction .....                                                  | 5  |
| d des différentes couches de la rétine .....                      | 6  |
| 1.2.3 Pathologies de la rétine .....                              | 7  |
| 1.2.4 Rétinopathie diabétique .....                               | 8  |
| 1.2.5 Classification de la rétinopathie diabétique (RD) .....     | 9  |
| a Rétinopathie diabétique non proliférante (RDNP) .....           | 10 |
| b Rétinopathie diabétique proliférante (RDP) .....                | 11 |
| 1.2.5 Traitement .....                                            | 11 |
| 1.2.6 Photocoagulation au laser .....                             | 11 |
| 1.3 Partie B les réseaux de neurones artificiels .....            | 12 |
| 1.3.1 Réseaux de neurone biologique .....                         | 12 |
| 1.3.2 Réseaux de neurone artificiel .....                         | 13 |
| 1.3.3 Quelques topologies de réseaux de neurones artificiel ..... | 17 |
| a Le neurone Formel .....                                         | 17 |
| b Le réseau de neurone multicouche .....                          | 17 |
| c Le réseau de neurone récurrent .....                            | 17 |
| d Réseau de Hopfield .....                                        | 17 |
| e Le réseau de neurones Convolutionnels .....                     | 18 |
| 1.4 Conclusion .....                                              | 19 |
| Chapitre 2 Deep Learning .....                                    | 20 |
| 2.1 Introduction .....                                            | 20 |
| 2.2 L'intelligence artificielle .....                             | 21 |
| 2.3 Machine learning .....                                        | 21 |
| 2.3.1 Les types d'apprentissage automatique .....                 | 22 |

|                                                              |    |
|--------------------------------------------------------------|----|
| 2.4 Deep Learning .....                                      | 24 |
| 2.4.1 Définition .....                                       | 24 |
| 2.4.2 Fonctionnement .....                                   | 24 |
| 2.5 Les CNNs .....                                           | 25 |
| 2.5.1 Bloc de construction .....                             | 26 |
| a La couche de convolution .....                             | 27 |
| b La couche de pooling .....                                 | 29 |
| c La couche de Relu .....                                    | 32 |
| d La couche entièrement connectée FC (fully connected) ..... | 32 |
| e La couche de perte ( loss ) .....                          | 33 |
| 2.5.2 Les hyper paramètres .....                             | 33 |
| a Nombre de filtres .....                                    | 34 |
| b Forme du filtre .....                                      | 34 |
| c Taille des filtres .....                                   | 34 |
| 2.6 RCNN .....                                               | 34 |
| 2.6.1 Fast-RCNN .....                                        | 35 |
| 2.6.2 Faster-RCNN .....                                      | 36 |
| 2.7 Conclusion .....                                         | 38 |
| Chapitre 3 Expérimentations et résultats .....               | 39 |
| 3.1 Introduction .....                                       | 38 |
| 3.2 Présentation des outils de développement .....           | 40 |
| 3.2.1 Matériel .....                                         | 40 |
| 3.2.2 Deep learning et Matlab. ....                          | 40 |
| 3.3 Prétraitement .....                                      | 41 |
| 3.3.1 Télécharger la base de données .....                   | 41 |
| 3.3.2 Encadrer les images .....                              | 41 |
| 3.3.3 Transformer les images en niveau de gris .....         | 41 |
| 3.4 Apprentissage .....                                      | 44 |
| 3.5 Résultats d'apprentissage .....                          | 47 |
| 3.6 Validation .....                                         | 55 |
| 3.7 Test .....                                               | 58 |
| 3.8 Conclusion .....                                         | 59 |
| Conclusion générale .....                                    | 60 |

|                     |    |
|---------------------|----|
| Bibliographie ..... | 61 |
|---------------------|----|

## Liste des figures :

### Chapitre 1 Généralités

|                                                            |    |
|------------------------------------------------------------|----|
| Figure 1-1: la rétine.....                                 | 4  |
| Figure 1-2:la structure de la rétine. ....                 | 5  |
| Figure 1-3:les couches de la rétine. ....                  | 7  |
| Figure 1-4:classification de rétinopathie diabétique. .... | 9  |
| Figure 1-5: neurone biologique /neurone artificiel. ....   | 13 |
| Figure 1-6:neurone artificiel.....                         | 14 |

### Chapitre 2 Deep Learning

|                                                                                                      |    |
|------------------------------------------------------------------------------------------------------|----|
| Figure 2-1:les implémentations de l'intelligence artificielle. ....                                  | 20 |
| Figure 2-2: les types d'apprentissage automatique.....                                               | 23 |
| Figure 2-3:un exemple du fonctionnement ....                                                         | 25 |
| Figure 2-4: architecture d'un reseau de neurone CNN. ....                                            | 26 |
| Figure 2-5: neurone d'une couche convolution (bleu)connecté a leurs champs<br>recepteur(rouge) ..... | 26 |
| Figure 2-6: exemple de stride.....                                                                   | 28 |
| Figure2-7: Tailles entrée et sortie.....                                                             | 29 |
| Figure 2-8: Pooling maximum avec un filtre 2x2. ....                                                 | 30 |
| Figure2-9: deux types de pooling. ....                                                               | 30 |
| Figure 2-10:Tailles entrée et sortie de la couche pooling. ....                                      | 31 |
| Figure 2-11:la fonction d'activation Relu. ....                                                      | 32 |
| Figure2-12: les différentes couches CNN. ....                                                        | 33 |
| Figure 2-13:Procédure de détection et de localisation d'objets dans R-CNN. ....                      | 35 |
| Figure2-14:Fast R-CNN Architecture.....                                                              | 35 |
| Figure 2-15: Faster R-CNN Architecture.....                                                          | 37 |

### Chapitre 3 Expérimentations et résultats

|                                                                |    |
|----------------------------------------------------------------|----|
| Figure 3-1:plan de travail. ....                               | 39 |
| Figure 3-2:exemple de prétraitement.....                       | 41 |
| Figure 3-3:load images.....                                    | 42 |
| Figure 3-4:organigramme de programme de prétraitement ....     | 43 |
| Figure 3-5:organigramme de prétraitement.....                  | 44 |
| Figure 3-6:architecture de CNN.....                            | 45 |
| Figure3-7:la 1 ère couche.....                                 | 45 |
| Figure 3-8:la couche moyenne.. ....                            | 46 |
| Figure 3-9:déclaration des couches. ....                       | 47 |
| Figure 3-10:Programme pour afficher la couche.....             | 47 |
| Figure 3-11:couche fullyconnected.....                         | 47 |
| Figure 3-12 a:Precision filtre 5*5 nombre de filtre 20.....    | 48 |
| Figure 3-12 b:lossfiltre 5*5 nombre de filtre 20.....          | 48 |
| Figure 3-13 a:Precisionfiltre 5*5 et nombre de filtre 30. .... | 48 |
| Figure 3-13 b:lossfiltre 5*5 et nombre de filtre 30. ....      | 48 |

|                                                                         |    |
|-------------------------------------------------------------------------|----|
| Figure 3-14 a: Precision filtre 3*3 et nombre de filtre 20. ....        | 49 |
| Figure 3-14 b: loss filtre 3*3 et nombre de filtre 20.....              | 49 |
| Figure 3-15 a: Precision filtre 5*5 et nombre de filtre 20.....         | 49 |
| Figure 3-15 b: loss filtre 5*5 et nombre de filtre 20.....              | 50 |
| Figure 3-16 a: Precision 4 couche de convolution . ....                 | 50 |
| Figure 3-16 b: loss 4 couche de convolution . ....                      | 50 |
| Figure 3-17 a: Precision 5 couche de convolution . ....                 | 51 |
| Figure 3-17 b: loss 5 couche de convolution . ....                      | 51 |
| Figure 3-18 a: Precision minibatchsize(64).. ....                       | 51 |
| Figure 3-18 b: loss minibatchsize(64).. ....                            | 52 |
| Figure 3-19 a: Precision filtre 5*5 et nombre de filtre 20(Adam). ....  | 52 |
| Figure 3-19 b: loss filtre 5*5 et nombre de filtre 20(Adam). ....       | 52 |
| Figure 3-20 a: Precision filtre 5*5 et nombre de filtre 30(Adam).. .... | 53 |
| Figure 3-20 b: loss filtre 5*5 et nombre de filtre 30(Adam).. ....      | 53 |
| Figure 3-21 a: Précision filtre 3*3 et nombre de filtre 20(Adam).....   | 53 |
| Figure 3-21 b: filtre 3*3 et nombre de filtre 20(Adam).....             | 53 |
| Figure 3-22 a: Precision validation avec filtrage (5 20) .....          | 56 |
| Figure 3-22 b: loss validation avec filtrage (5 20) .....               | 56 |
| Figure 3-23 a: Precision validation avec filtrage (5 30). ....          | 57 |
| Figure 3-23 b: loss validation avec filtrage (5 30). ....               | 57 |
| Figure 3-24 : organigramme du test. ....                                | 58 |
| Figure 3-25 : affichage de classification . ....                        | 58 |

## Liste des tableaux :

|                                                          |    |
|----------------------------------------------------------|----|
| <i>Tableau 1-1</i> :les signes de RDNP.....              | 10 |
| <i>Tableau 1-2</i> :les signes de RDP.....               | 11 |
| Tableau 1-3: les fonctions d'activation fréquentes ..... | 16 |
| Tableau 1-4: les types de réseaux de neurones. ....      | 18 |
| Tableau 3-1: les différents essais d'apprentissage. .... | 56 |
| Tableau 3-2:apprentissage et validation.....             | 57 |

# Introduction générale

---

Les domaines d'application et usages potentiels de l'intelligence artificielle (IA) sont de plus en plus nombreux et variés : compréhension du langage naturel, reconnaissance visuelle, robotique, système autonome, Machine Learning ...etc.

En particulier dans le domaine médical, l'IA et l'apprentissage profond ont déjà prouvé qu'ils étaient des outils précieux pour améliorer la détection précoce, le diagnostic et le traitement de nombreuses maladies graves.

L'imagerie médicale est certainement l'un des domaines de la médecine qui a le plus progressé ces dernières années. En effet les récentes découvertes permettent non seulement un meilleur diagnostic mais offrent aussi de nouveaux espoirs de traitement pour de nombreuses maladies. Avec l'introduction massive de l'intelligence artificielle (IA) et de l'apprentissage automatique (*Machine Learning*) et (*Deep Learning*), nous entrons dans une ère nouvelle. De telles techniques permettent également de mieux comprendre le fonctionnement de certains organes encore mystérieux, comme le cerveau. Elle permet d'accélérer le diagnostic, mais aussi de le rendre plus sûr et plus fiable, car les systèmes d'auto-apprentissage prennent mieux en compte les nouvelles situations que les systèmes statiques.

En effet, les algorithmes, en particulier les réseaux à convolution utilisés en Deep Learning, s'avèrent plus efficaces que les radiologues les plus compétents pour détecter une tumeur ou tout autre type de lésion.

L'objectif de notre travail est la détection d'une maladie rétinienne qui est la rétinopathie diabétique, en utilisant le deep Learning. Ce projet touche le domaine médical ainsi que celui du traitement d'image. Il entre dans le cadre d'un projet de recherche de l'équipe Bio rétine du laboratoire LATSI (département d'électronique, université de Blida 1), le circuit du Deep Convolutional Neural Network obtenu sera embarqué sur un système de photocoagulation par laser miniaturisé et connecté (IOT), pour l'analyse, le diagnostic la prévention, et le traitement en temps réel des rétinopathies diabétiques.

Dans ce mémoire de fin d'étude, nous nous intéresserons donc à l'imagerie rétinienne. En effet il est possible désormais d'analyser de façon automatique les

images de fonds d'œil (images rétinienne) dans le but de réduire le temps de traitement et de diagnostic de la rétinopathie diabétique pour les ophtalmologues.

Nous élaborons un algorithme de DL que nous entraînons avec une base d'apprentissage disponible sur le net. Nous avons choisis la base de données Kaggle (lien :[kaggle.com/c/diabetic-retinopathy-contest/data](https://kaggle.com/c/diabetic-retinopathy-contest/data) )car elle contient un nombre(1300)important d'imagesde rétines.

Nous utilisons de nombreux outils pour le prétraitement des données (niveau de gris, labellisation...).Le but de notre travail étant la détection de pathologies rétinienne, nous utilisons l'algorithme FasterR-CNN(adapté aux taches de détection )sous Matlab pour implémenter notre DCNN.

Le mémoire s'articule autour de trois chapitres :

Dans le premier chapitre, nous découvrirons dans une première partie l'anatomie générale de l'œil, aussi une description succincte de la rétine. Nous présenterons brièvement les pathologies touchant la rétine en particulier la rétinopathie diabétique ; dans une deuxième partie nous décrivons les réseaux de neurones artificiels qui sont à la base de notre réseau CNN.

Dans le second chapitre, nous étudionsle concept de l'IA (intelligence artificielle) et du Deep Learning, nous parlerons ainsi de l'architecture CNN et R-CNN utilisées dans notre projet.

Dans le troisième chapitre nous décrivons les outils utilisés et les étapes d'implémentation de notre DCNN. Et présenterons les résultats obtenus pour implémentation de notre réseau de neurones capable de détecter la rétinopathie diabétique.

# Chapitre 1 Généralités sur la rétine et les réseaux neurones

---

## 1.1 Introduction :

Le but de notre projet est d'implémenter un algorithme de réseaux neurones permettant de détecter la rétinopathie diabétique. Nous allons d'abord dans une première partie donner un aperçu général de l'œil son anatomie, le globe oculaire, et un aperçu plus détaillé de la rétine.

Ensuite, une description de diverses maladies associées à la rétine (pathologie) est donnée. Enfin, une attention particulière est accordée à la rétinopathie diabétique, ses types, et traitement. Dans la deuxième partie, nous expliquerons les réseaux neurones, nous parlerons aussi des fonctions utilisées ainsi que des types de ces réseaux.

## 1.2 Partie A

### 1.2.1 L'anatomie de l'œil

L'anatomie de l'œil se divise en deux : celle du globe oculaire et celle de ses annexes (les muscles extra-oculaires, les nerfs, la paupière, le système lacrymal et l'orbite)[1].

- **Les annexes du globe oculaires**, on appelle "annexes", les structures situées autour du globe oculaire comme : les paupières, l'appareil lacrymal ; les muscles extra-oculaires ou muscles oculomoteurs.
- **Le globe oculaire peut être décomposé en quatre parties principales** : la couche protectrice ; la couche vascularisée ; le contenu de la cavité interne ; la couche visuelle.

## 1.2.2 La rétine

### a Définition de la rétine

La rétine est un tissu sensoriel multicouche situé à l'arrière de l'œil. Il contient des millions de photorécepteurs qui capturent les rayons lumineux et les convertissent en impulsions électriques. Ces impulsions voyagent le long du nerf optique jusqu'au cerveau où elles sont converties en images[2].

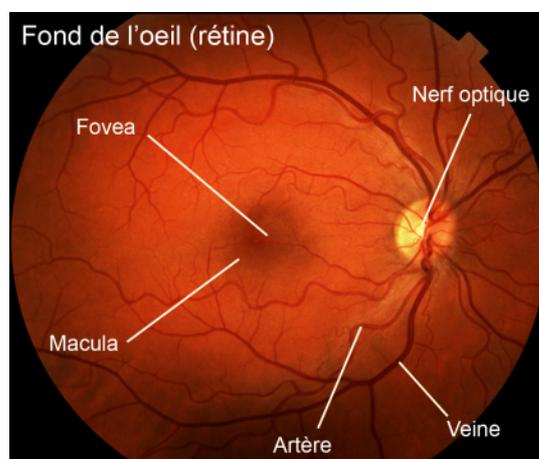


Figure 1-1:la rétine.

### b Constitution de la rétine

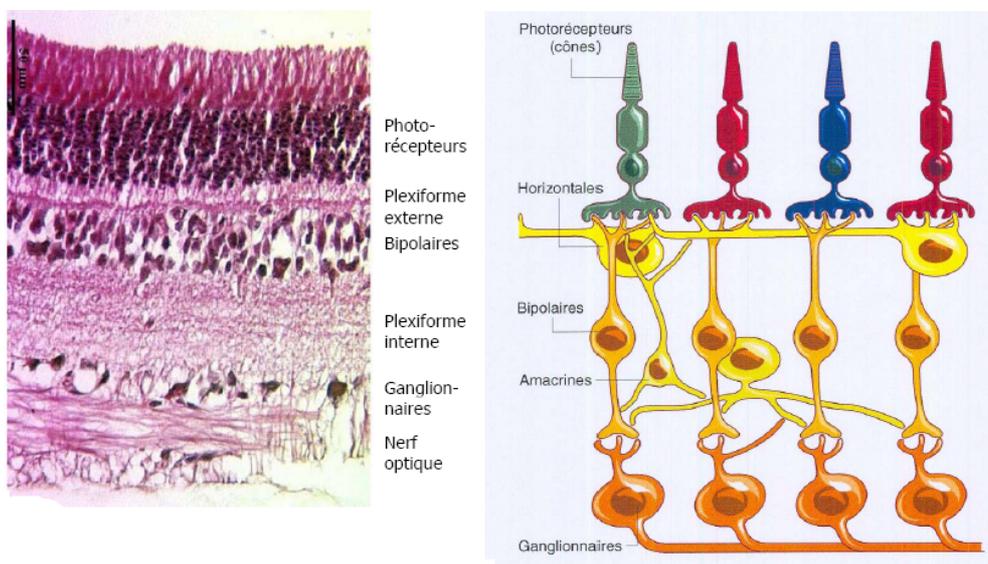
La rétine se caractérise par deux rétines : **centrale** et **périphérique**[2]:

#### ➤ la rétine centrale :

Se trouve dans l'axe de vision, avec plus précisément la **macula** : portion non vascularisée uniquement composée de photorécepteurs appelés « cônes », qui responsables de l'acuité visuelle et de la vision des couleurs, sont majoritaires sur la macula (dépression de la rétine où l'acuité visuelle est maximale) et seuls présents dans la fovéa (centre de la macula).

#### ➤ la rétine périphérique :

Elle est chargée de détecter les informations dans la totalité du champ de vision, les mouvements, la vision est moins précise. Les bâtonnets, sensibles à la quantité de lumière et à son intensité, et responsables de la vision dans des conditions de faible éclairage, se distribuent sur l'ensemble de la rétine et permettent l'élargissement du champ visuel, ou vision périphérique. Les cellules bipolaires mettent en relation les cellules photo-réceptrices et les cellules ganglionnaires. Les cellules ganglionnaires se prolongent par les fibres optiques, qui se réunissent dans la pupille pour former le nerf optique.



**Figure 1-2:** la structure de la rétine.

### ***c* Fonction**

Sa fonction principale la photo transduction, est assurée par les photorécepteurs en synergie avec l'épithélium pigmentaire. La cornée et le cristallin fixent la lumière sur la rétine. Les cônes, produisent des images extrêmement nettes et sont responsables de la vision centrale et de la perception des couleurs. Les bâtonnets, qui sont sensibles à des niveaux d'éclairage plus bas, mais ne sont pas sensibles aux couleurs. Ces bâtonnets sont responsables de la vision périphérique et de la vision nocturne. Le nerf optique transmet les signaux émis par les photorécepteurs (cônes et bâtonnets). Le nerf optique et la rétine ont un réseau vasculaire dense qui leur fournit le sang et l'oxygène. Une partie de ce réseau vasculaire provient de la choroïde, qui est la couche

vasculaire située entre la rétine et l'enveloppe blanche externe de l'œil (sclère). L'artère centrale de la rétine (l'autre source d'apport important de sang vers la rétine) rejoint la rétine près du nerf optique, puis se ramifie en plusieurs branches. Le sang s'écoule de la rétine vers les branches de la veine rétinienne centrale. La veine rétinienne centrale sort de l'œil dans le nerf optique [3].

#### ***d les différentes couches de la rétine***

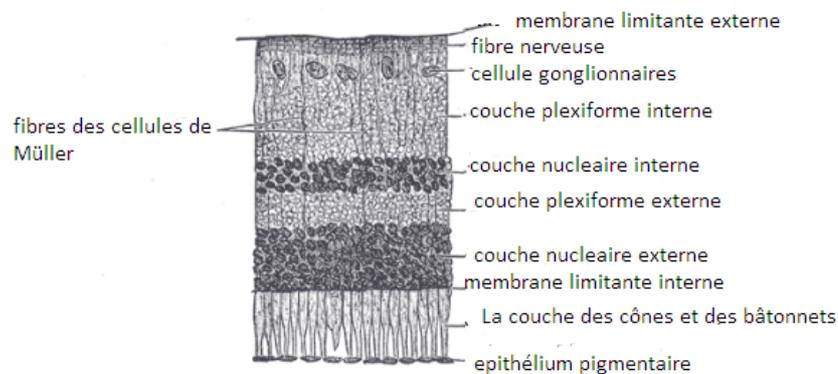
- 1) Membrane limitante interne ou membrane interne.
- 2) La couche de fibres nerveuses contient les axones des cellules ganglionnaires, qui transmettent les signaux au cortex visuel du cerveau.
- 3) La couche des cellules ganglionnaires est faite de grosses cellules nerveuses, dont les cylindraxes, très longs, constitueront le nerf optique. Épaisse de 10 à 20  $\mu\text{m}$ , elle est presque partout faite d'une seule couche, sauf autour de la fovéa, où les noyaux s'empilent sur 7 à 8 rangs expliquant le relief observé à ce niveau.
- 4) La plexi forme interne, où se fait la jonction cellules bipolaires - cellules ganglionnaires, est épaisse de 20 à 30  $\mu\text{m}$ .
- 5) La couche nucléaire interne épaisse d'environ 30  $\mu\text{m}$ , est constituée par les neurones bipolaires qui transmettent l'influx nerveux de la cellule réceptrice à la cellule ganglionnaire, par les cellules d'association et les cellules de soutien : corps cellulaire des fibres de Müller, cellules amacrines.
- 6) La plexi forme externe, épaisse de 20  $\mu\text{m}$ , est formée par la jonction synaptique des cônes et des bâtonnets avec les dendrites des cellules de la couche suivante. Elle est le siège du relais entre rétine sensorielle et rétine cérébrale.
- 7) La couche nucléaire externe est constituée par les corps cellulaires des cônes et des bâtonnets.
- 8) Membrane limitante externe
- 9) La couche des cônes et des bâtonnets est formée par les deux segments externes et internes des cellules visuelles : c'est là que commencent les

premiers phénomènes de la vision. Elle est épaisse de 4  $\mu\text{m}$  et compte environ 130 millions de bâtonnets et 65 millions de cônes.

10) Épithélium pigmentaire. C'est la dernière couche de la rétine. C'est celui qui donne un couleur différente (pigment) pour chaque rétine humaine.

11) Enfin, la limitante externe est formée par la réunion des extrémités internes des fibres des cellules de Müller (rôle de soutien).[4]

Chacun des types cellulaires, cônes et bâtonnets, de ces couches possède une répartition et des caractéristiques physiologiques qui leur sont propres.



**Figure 1-3:**les couches de la rétine.

### 1.2.3 Pathologies de la rétine

La pathologie ophtalmologique est variée car elle concerne tant le globe oculaire lui-même que l'orbite, mais parfois également des structures intracrâniennes (voies optiques et voies de l'oculomotricité). L'exploration des différentes pathologies ophtalmologiques fait appel à de nombreuses investigations paracliniques et, plus spécifiquement, aux diverses techniques d'imagerie (radiographies standard, échographie, tomodensitométrie et imagerie par résonance magnétique).

Il existe différentes variétés de problèmes de la rétine, conditions et maladies. Ceci est une liste courte des problèmes communs de la rétine [5][6].

- le décollement de rétine.
- La dégénérescence maculaire liée l'âge (DMLA).
- Les membranes épirétinennes et trous maculaires.
- les rétinopathies, dont la rétinopathie diabétique.

#### **1.2.4 Rétinopathie diabétique**

L'atteinte de l'œil liée à un diabète déséquilibré reste parmi les premières causes de cécité dans les pays industrialisés. La rétinopathie diabétique est une conséquence de l'hyperglycémie chronique. En cas d'excès de sucre dans le sang, comme au cours d'un diabète, la paroi des capillaires est fragilisée et perd son étanchéité. Il s'ensuit alors un éclatement des vaisseaux de la rétine. On parle de « micro-anévrisme ».

L'hyperglycémie entraîne des dégâts au niveau des fins vaisseaux sanguins de la rétine ce qui provoque des hémorragies et des œdèmes. A des stades plus avancés, la rétine réagit en produisant des vaisseaux anormalement fragiles qui peuvent être à l'origine de complications graves telles que l'hémorragie de la rétine ou du corps vitré, le glaucome ou encore le décollement de rétine. C'est l'hyperglycémie qui est responsable des dommages causés aux petits vaisseaux rétiniens, appelés capillaires, dans la rétinopathie diabétique. Par ailleurs, l'hyperglycémie peut amener des problèmes d'occlusion des capillaires. Lorsque cela survient, le sang ne peut plus circuler librement dans les vaisseaux pour aller fournir à la rétine les éléments dont elle a besoin pour fonctionner. Il survient alors un état d'ischémie, c'est-à-dire un manque de sang dans les régions où les vaisseaux sont bouchés. Pour essayer de pallier ce problème, des facteurs de croissance, dont l'un que l'on nomme VeGF(vascularendothelialgrowth factor), sont sécrétés au niveau de l'œil pour permettre la croissance de nouveaux vaisseaux. Cependant, ces néovaisseaux sont fragiles et donc susceptibles de causer des hémorragies qui peuvent bloquer la vue. Aussi, lorsque les nouveaux vaisseaux régressent, ils peuvent tirer sur la rétine et en occasionner des déchirures et des décollements pouvant avoir une incidence significative sur la vision.[5][6]

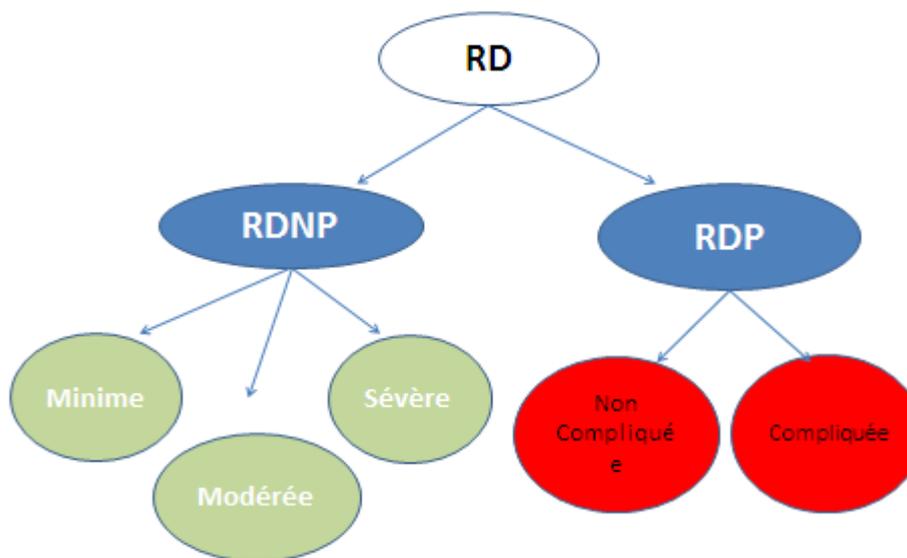
On distingue deux grands types de rétinopathies : la rétinopathie non proliférative caractérisée par quelques microhémorragies et la rétinopathie proliférative avec

apparition de néovaisseaux. La rétinopathie proliférative est caractérisée par des hémorragies plus importantes.

### 1.2.5 Classification de la rétinopathie diabétique (RD)

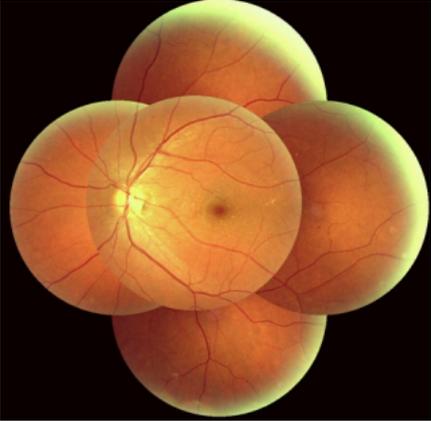
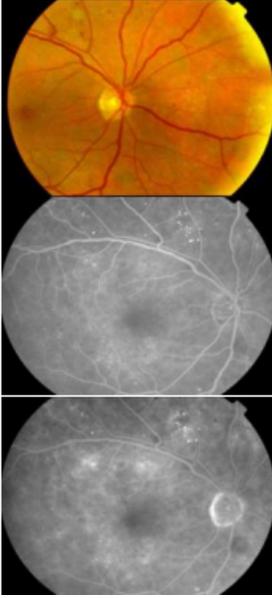
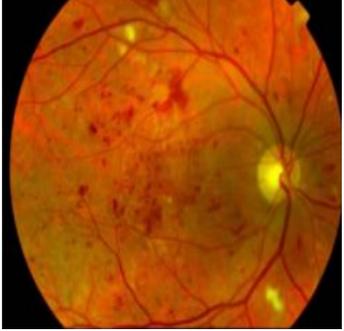
La classification de la RD est établie en fonction du risque visuel lié à la sévérité de l'ischémie rétinienne. Elle débute par un stade de rétinopathie diabétique non proliférante minime puis modérée, et évolue vers la rétinopathie diabétique non proliférante sévère caractérisée par une ischémie rétinienne étendue, puis vers le stade de rétinopathie diabétique proliférante caractérisée par la prolifération de néovaisseaux à la surface de la rétine et/ou sur la papille. L'œdème maculaire peut s'associer à tous les stades de RD. Cependant, son incidence augmente avec la sévérité de la rétinopathie.

Alors On distingue deux grands types de rétinopathies : la rétinopathie non proliférative caractérisée par quelques microhémorragies et la rétinopathie proliférative avec apparition de néo-vaisseaux. La rétinopathie proliférative est caractérisée par des hémorragies plus importantes.[7][8][9]



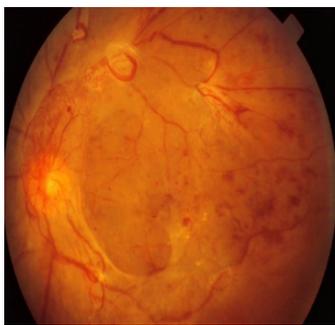
**Figure 1-4:**classification de rétinopathie diabétique.

**a Rétinopathie diabétique non proliférante (RDNP)**

| RDNP minime                                                                                                                                                                                                                                                                     | RDNP modérée                                                                                                                                                                                                                                                                      | RDNP sévère<br>(préproliférante)                                                                                                                                                                                                                                                                                                                             |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>1er stade de la RD, définie par la présence d'au – 1 <math>\mu</math>a et/ou d'1 hémorragie.</p> <ul style="list-style-type: none"> <li>• Pas d'indication au traitement laser</li> </ul>  | <p>µa hémorragie rétinienne punctiformes ou en « flammèches » nombreuses</p> <ul style="list-style-type: none"> <li>• Et/ou hémorragie rétinienne. en « taches » (&lt; 4 quadrants)</li> </ul>  | <ul style="list-style-type: none"> <li>• Stade à haut risque d'évolution vers le néo-vascularisation</li> <li>• Hémorragies rétinienne étendues (4 quadrants)</li> <li>• PPR non obligatoire mais discutée impérative si circonstances d'aggravation rapide</li> </ul>  |

**Tableau 1-1:**les signes de RDNP.

### ***b Rétinopathie diabétique proliférante (RDP)***

| <b>RDP non compliquée</b>                                                                                                                               | <b>RDP compliquée</b>                                                                                                                                                                                                                 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Néovaisseaux pré-rétiniens et/ou prépapillaires de petite taille.<br> | <ul style="list-style-type: none"><li>• Hémorragies intravitréennes.</li><li>• Décollement de rétine.</li><li>• glaucome néovasculaire.</li></ul>  |

**Tableau 1-2:** les signes de RDP.

#### **1.2.5 Traitement**

- équilibre diabète et tension.
- injections intra-vitréennes anti-VEGF/corticoides.
- Traitements spécifiques des complications.
- Photocoagulation au laser.[8]

#### **1.2.6 Photocoagulation au laser**

La photocoagulation panrétinienne (PPR) au laser (est un rayon de lumière qui peut être centré de manière très précise sur un point de la rétine), le PPR est le traitement spécifique de la rétinopathie diabétique proliférante. Elle consiste en une coagulation étendue de toute la rétine périphérique : *elle n'agit pas directement sur les néo vaisseaux mais indirectement par la destruction étendue des territoires d'ischémie rétinienne* : celle-ci permet d'obtenir la régression de la néo vascularisation pré

rétinienne et/ou pré papillaire dans près de 90% des cas et de réduire considérablement le risque de cécité lié à la rétinopathie diabétique proliférante. Elle est réalisée en ambulatoire sous anesthésie de contact, en plusieurs séances.

*La PPR est indiquée :*

- *Dans tous les cas de rétinopathie diabétique proliférante.* La rapidité de sa réalisation dépendra de la sévérité de la RDP : des néo vaisseaux pré rétiniens isolés et peu nombreux seront traités par des séances espacées d'une ou plusieurs semaines ; à l'autre extrême, l'existence d'une néo vascularisation irienne est l'indication d'une PPR urgente, réalisée par des séances très rapprochées.
- *Dans certains cas de RDNP sévère (RDPP),* à titre préventif, chez des sujets à haut risque de passage en rétinopathie proliférante : principalement grossesse, normalisation rapide de la glycémie (notamment par mise sous pompe à insuline), chirurgie de la cataracte. La PPR prévient efficacement dans ces cas la survenue d'une néo vascularisation [8].

## **1.3 Partie B : les réseaux de neurones artificiels**

### **1.3.1 Réseaux de neurone biologique**

Le neurone est une cellule fondamentale du système nerveux des êtres vivants. Le cerveau humain en contient d'environ  $10^{12}$  neurones avec 100000 connexions par l'élément de base du système nerveux central. Chaque neurone est composé :

- D'un corps cellulaire contenant le noyau, de dendrites, nombreuses et ramifiées, qui conduisent l'influx nerveux de leur périphérie jusqu'au corps cellulaire.
- D'un axone, qui conduit le potentiel d'action émis au niveau du corps cellulaire jusqu'aux dendrites d'autres neurones.

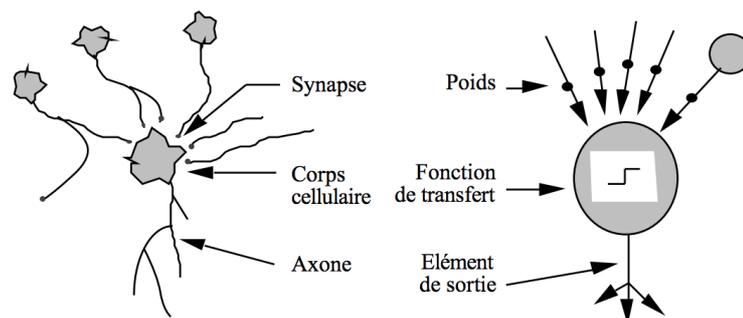
L'influx nerveux y est alors transmis par voie chimique au niveau des synapses.

Les axones peuvent mesurer plusieurs dizaines de centimètres de longueur et sont entourés de gaines de myéline permettant d'optimiser la transmission de l'influx nerveux.

Chaque neurone reçoit donc en entrée des signaux en provenance d'autres neurones, transmis par les dendrites jusqu'au corps cellulaire où ils s'additionnent. L'importance de chaque signal reçu est modulée à la fois par la longueur de la dendrite lui permettant d'atteindre le corps cellulaire et par l'efficacité de la liaison synaptique entre l'axone présynaptique et la dendrite post-synaptique. S'il dépasse un certain seuil, le signal résultant au niveau du corps cellulaire peut alors donner lieu à un potentiel d'action, c'est-à-dire un pic de potentiel électrique qui se propage à travers l'axone jusqu'aux autres neurones dont les dendrites sont connectées à cet axone : on dit que ce neurone décharge. [10]

### 1.3.2 Réseaux de neurones artificiels

Le réseau de neurones artificiels ressemble au réseau de neurones biologique, mais aucune copie parfaite de notre cerveau sous formes mathématiques n'existe pas aujourd'hui. Comme illustre sur la figure 1-5, la représentation de neurone formel est inspirée du principe de fonctionnement de neurone biologique [11] :



**Figure 1-5:** neurone biologique /neurone artificiel.

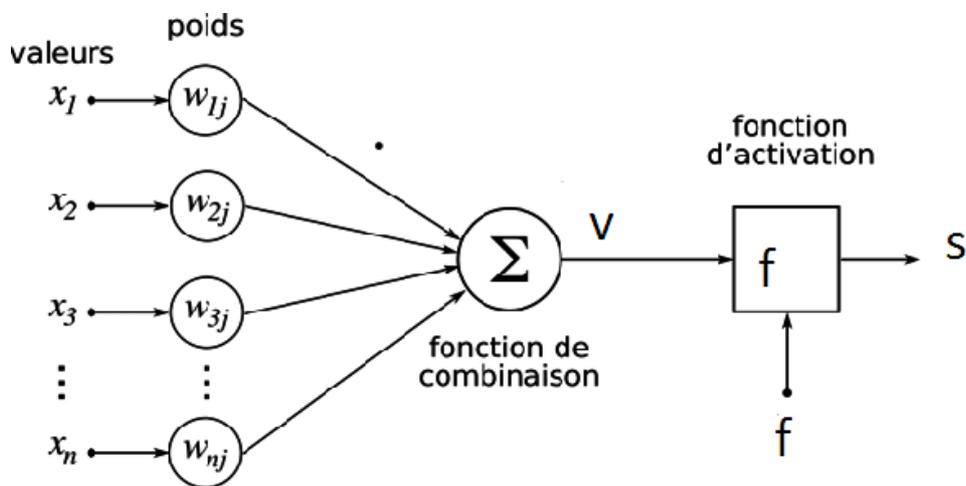
Le réseau de neurones artificiels est un ensemble de neurones formels interconnectés. Il est basé sur un modèle de neurone simplifié et il peut reproduire certaines fonctions du cerveau comme la reconnaissance des formes, le traitement de langage naturel, la mémorisation associative ou bien l'apprentissage. Mais ces neurones formels n'ont pas toutes les capacités des neurones biologiques.

L'évolution des neurones artificiels s'est faite en plusieurs générations sont :

- Le perceptron : première génération.
- Le neurone formel : deuxième génération.
- Le neurone impulsionnel : troisième génération.

Nous décrivons brièvement le neurone formel dans ce qui suit :

➤ **Structure de neurone artificiel ou formel :**



**Figure 1-6: neurone artificiel.**

La figure 1-6 montre la structure d'un neurone artificiel. chaque neurone formel est un processeur élémentaire. il reçoit un nombre variable d'entrées en provenance de neurones amonts. A chacune de ces entrées est associé un poids  $w_{ij}$  représentatif de la force de connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals. Chaque connexion est associée à un poids.

➤ **Le fonctionnement du neurone formel :**

On distingue deux phases. la première est habituellement le calcul de la somme pondérée des entrées ( $V$ ) selon l'expression suivante :

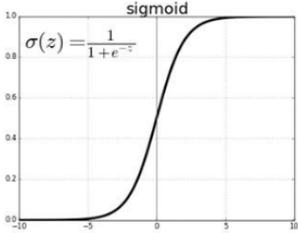
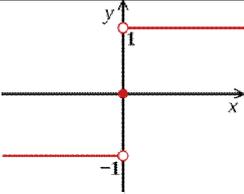
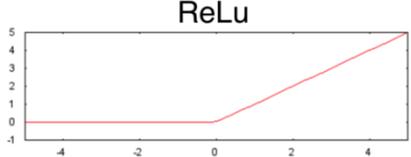
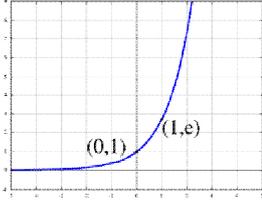
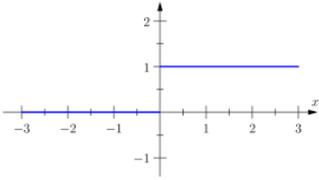
$$V = [ \sum (w_{ij} \cdot x_n) + w_0 ] \quad (\text{eq 1.1})$$

➤ **Les types de fonctions d'activation:**[12]

La deuxième équation (eq1.2) est l'application à cette valeur d'une fonction de transfert ou d'activation. Une fonction de transfert calcule l'état ou la valeur du neurone. C'est cette valeur qui sera transmise aux neurones en aval. Il existe de nombreuses formes possibles pour la fonction de transfert. La plupart de ces fonctions sont continues offrant une infinité de valeurs possibles comprises dans l'intervalle [0, +1] ou [-1,0]. La fonction d'activation (ou de sortie) : elle définit l'état du neurone en fonction de son entrée totale par cette équation :

$$S=f(V_i)=f[\sum (W_{ij} \cdot x_n) + W_0] \text{ (eq 1.2)}$$

Les fonctions d'activation les plus fréquentes et que nous utilisons dans notre réseau de Deep Learning sont présentées dans le tableau 1-3 ci-dessous :

| Fonction                                                  | Equation                                                                                                                    | Graphe                                                                               |
|-----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| La fonction sigmoïde                                      | $F(x) = \frac{1}{1+e^{-x}} \quad (1.3)$                                                                                     |    |
| La fonction signe                                         | $F(x) = \begin{cases} 1 & \text{si } x > 0 \\ -1 & \text{sinon} \end{cases} \quad (1.4)$                                    |   |
| Unité de Rectification Linéaire (ReLU)                    | $F(x) = \begin{cases} 0 & \text{si } x < 0, \\ x & \text{sinon} \end{cases} \quad (1.5)$                                    |  |
| la fonction softmax, ou fonction exponentielle normalisée | $F(x) = \frac{e^{z_j}}{\sum_{k=1}^k e^{z_k}} \quad j \in \{1, \dots, k\} ; \text{ avec } k \text{ nombre réel} \quad (1.6)$ |  |
| La fonction Heaviside                                     | $F(x) = \begin{cases} 1 & \text{si } x > 0, \\ 0 & \text{sinon} \end{cases} \quad (1.7)$                                    |  |

**Tableau 1-3:** les fonctions d'activation fréquentes .

### 1.3.3 Quelques topologies de réseaux de neurone artificiel

Il existe beaucoup de types de réseaux neurones (tableau 1-4), chacun de ces types à une utilisation particulière[13] :

#### **a. Le neurone Formel**

Un neurone formel est une représentation mathématique et informatique d'un neurone biologique. Il possède plusieurs entrées et une sortie qui correspondent respectivement aux dendrites et au cône d'émergence du neurone biologique (point de départ de l'axone). Un neurone formel calcule la somme pondérée des entrées reçues, puis applique à cette valeur une fonction d'activation, généralement non linéaire. La valeur finale obtenue est la sortie du neurone. Le neurone formel est l'unité élémentaire des réseaux de neurones artificiels dans lesquels il est associé à ses semblables pour calculer des fonctions arbitrairement complexes, utilisées pour diverses applications en intelligence artificielle.

#### **b. Le réseau de neurone multicouche**

Le perceptron multicouche (multi layer perceptron MLP) est un classifieur linéaire de type réseau neuronal formel organisé en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement. Chaque couche est constituée d'un nombre variable de neurones, les neurones de la couche de sortie correspondant toujours aux sorties du système.

#### **c. Le réseau de neurone récurrent**

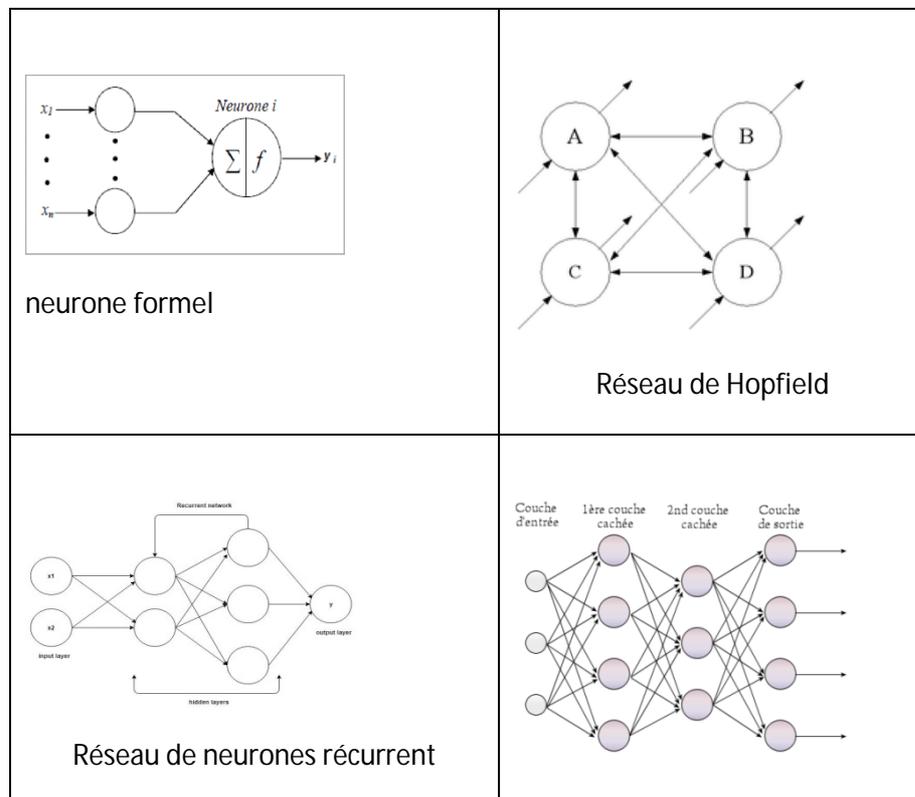
Les réseaux de Neurones récurrents (RNNs) permettent d'analyser les séquences de vecteurs. Le temps entre ici en ligne de compte car les sorties (de la couche de sortie et/ou de la couche cachées) calculées à l'instant  $t$  sont réinjectées en entrée du réseau et/ou en entrée de la couche cachée. On peut en théorie conserver dans le réseau la mémoire de ce qui s'y est passé depuis le début.

#### **d. Réseau de Hopfield**

Le réseau de neurones d'Hopfield est un modèle de réseau de neurones récurrents à temps discret dont la matrice des connexions est symétrique et nulle sur la diagonale et où la dynamique est asynchrone (un seul neurone est mis à jour à chaque unité de temps). Sa découverte a permis de relancer l'intérêt dans les réseaux de neurones.

**e. Le réseau de neurones Convolutionnels**

Un réseau de neurone convolutifs (ou réseau de neurones à convolution, ou CNN ou ConvNet) est un type de réseau de neurones artificiels acycliques dans lequel le motif de connexion entre les neurones est inspiré par le cortex visuel. Les neurones de cette région du cerveau sont arrangés de sorte à ce qu'ils correspondent à des régions qui se chevauchent lors du pavage du champ visuel. Leur fonctionnement est inspiré par les processus biologiques, ils consistent en un empilage de perceptrons multicouches, ces réseaux ont de larges applications dans la reconnaissance d'image et vidéo, les systèmes de recommandation et le traitement du langage naturel (plus détaillé dans le chapitre 2).



|  |                                |
|--|--------------------------------|
|  | réseau de neurones multicouche |
|--|--------------------------------|

**Tableau 1-4:** les types de réseaux de neurones.

- Il existe beaucoup de types de réseaux de neurones, chacun de ces types a une utilisation particulière : Comme nous le verrons dans le chapitre suivant, le deep learning utilise les CNN, les MLP, lesRNN, avec les fonctions d'activation (Softmax, Relu,Sigmoïde ...).

## 1.4 Conclusion

Ce chapitre explique les généralités sur la rétine et les réseaux neurones, en commençant par l'anatomie de l'œil pour mieux comprendre l'emplacement, rôle, et les maladies qui peuvent l'atteindre notamment la rétinopathie diabétique, sa définition, son traitement, ensuite les réseaux neurone, leurs définition, types ainsi que les fonctions utiliser.

Notre travail s'inscrit dans le cadre d'un projet de recherche visant la détection de pathologie rétinienne et son traitement automatique par photocoagulation laser en utilisant le deep learning.

Le Deep Learning et les réseaux neurones ont prouvé leur efficacité dans différents domainesdont domaine médical et le traitement d'images. Dans le second chapitre nous allons approfondir nos connaissances surle deep Learning et le machine Learning, leurs domaines d'utilisation, leurs rôles.

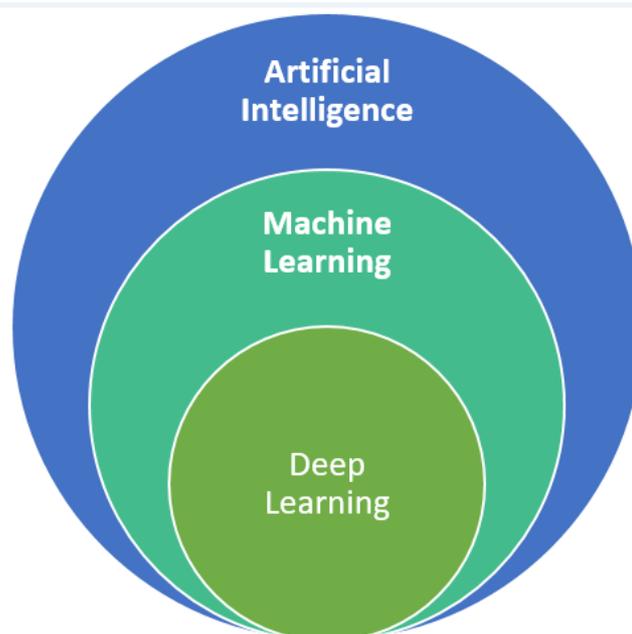
## Chapitre 2 Deep Learning

---

### 2.1 Introduction :

Le but de notre projet est d'implémenter un réseau neurone capable de détecter la rétinopathie diabétique en utilisant le deep learning. Le principe est de détecter les anomalies dans une image de la rétine et cela en déterminant les paramètres du CNN nécessaire.

Dans ce second chapitre, on définit le deep Learning qui est un sous domaine du machine Learning qui est lui aussi un sous domaine de l'intelligence artificielle, et leur fonctionnement et leur utilisation est expliquée.



**Figure2-1** : les implémentations de l'intelligence artificielle.

## 2.2 L'intelligence artificielle

L'intelligence artificielle (IA, ou AI en anglais pour Artificiel Intelligence) est une technique permettant aux ordinateurs d'imiter le comportement humain, notamment l'apprentissage automatique, le traitement du langage naturel, la synthèse de langage, la vision par ordinateur, la robotique, l'analyse de capteurs, l'optimisation et la simulation et autres.

L'IA est implémentée dans un nombre grandissant de domaines d'application. Toutes les grandes entreprises dans le monde de l'informatique planchent aujourd'hui sur les problématiques de l'intelligence artificielle en tentant de l'appliquer à quelques domaines précis. Chacun a ainsi mis en place des réseaux de neurones artificiels constitués de serveurs et permettant d'effectuer de lourds traitements sur des bases de données gigantesques.[14][15]

## 2.3 Machine Learning

Le Machine Learning est une méthode d'intelligence artificielle permettant aux ordinateurs d'apprendre sans avoir été programmés explicitement à cet effet et pour apprendre et se développer, les ordinateurs ont besoin de données à analyser et sur lesquelles s'entraîner, il s'agit d'une science moderne permettant de découvrir des patterns et d'effectuer des prédictions à partir de données en se basant sur des statistiques, sur du forage de données, sur la reconnaissance de patterns et sur les analyses prédictives .

Le ML champ d'étude de l'intelligence artificielle, concerne la conception, l'analyse, le développement et l'implémentation de méthodes permettant à une machine (au sens large) d'évoluer par un processus systématique, et ainsi de remplir des tâches difficiles ou impossibles à remplir par des moyens

algorithmiques plus classiques. L'analyse peut concerner des graphes, arbres, courbes ou images au même titre que de simples grandeurs scalaires.

Cette technique a été créée pour résoudre les problèmes pour lesquels les modèles analytiques sont difficilement disponibles. L'idée principale de Machine Learning est de réaliser un modèle en utilisant les données d'apprentissage lorsque les équations et les lois ne sont pas disponibles (systèmes non-linéaires).

La Machine Learning est très efficace dans les situations où les insights doivent être découverts à partir de larges ensembles de données diverses et changeantes, c'est-à-dire : le Big Data. Pour l'analyse de telles données, la Machine Learning se révèle nettement plus efficace que les méthodes traditionnelles en termes de précision et de vitesse.[14][15]

Parmi les techniques de ML, citons les machines à vecteurs de support (SVM), les arbres de décision, l'apprentissage de Bayes, l'apprentissage des règles d'association, la régression, les réseaux de neurones, ...etc.

### 2.3.1 Les types d'apprentissage automatique

Deux approches majeures sont envisagées afin de réaliser l'apprentissage :

- L'apprentissage supervisé consiste en des variables d'entrée ( $x$ ) et une variable de sortie ( $Y$ ). Vous utilisez un algorithme pour apprendre la fonction de mapping de l'entrée à la sortie.

Le but est d'appréhender si bien la fonction de mapping que, lorsque vous avez de nouvelles données d'entrée ( $x$ ), vous pouvez prédire les variables de sortie ( $Y$ ) pour ces données.

C'est ce qu'on appelle l'apprentissage supervisé, car le processus d'un algorithme tiré de l'ensemble de données de formation (training set) peut être considéré comme un enseignant supervisant le processus d'apprentissage. L'apprentissage supervisé, est un système qui fournit à la fois les données en entrée et les données attendues en sortie. Les données en entrée et en sortie sont étiquetées en vue de leur classification, afin d'établir une base d'apprentissage pour le traitement ultérieur des données.

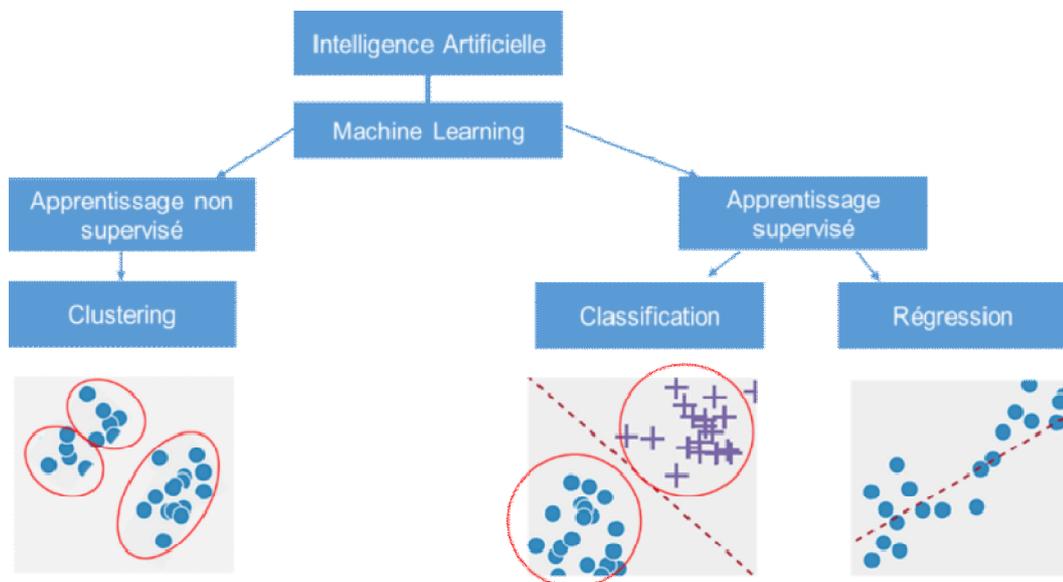
- Par régression : Ce type d'algorithmes supervisés va trouver une valeur continue (un nombre réel) qui est la prédiction de la valeur d'une nouvelle observation donnée. Par exemple la prédiction de prix de maison en fonction de ses caractéristiques.

- Par classification : Ces algorithmes vont classer une donnée dans une catégorie. Par exemple classer un courriel en SPAM ou non, décider si une tumeur est maligne ou bénigne. Il s'agit également d'un algorithme supervisé.
- L'apprentissage non supervisé consiste à ne disposer que des données d'entrée (X) et pas de variables de sortie correspondantes.

L'objectif de l'apprentissage non supervisé est de modéliser la structure ou la distribution sous-jacente dans les données afin d'en apprendre d'avantage sur les données.

Celles-ci sont appelées apprentissage non supervisé car, contrairement à l'apprentissage supervisé ci-dessus, il n'y a pas de réponse correcte ni d'enseignant. Les algorithmes sont laissés à leurs propres mécanismes pour découvrir et présenter la structure intéressante des données.

- clustering: Il s'agit d'une famille d'algorithmes non supervisés. Par conséquent, les données n'ont pas d'étiquettes, il va regrouper les données par similarité. Par exemple, on fournit un ensemble de photos d'animaux (sans qu'on dise de quels animaux il s'agit). Il va regrouper les photos de chats d'un côté et celle des chiens d'un autre etc.[16][17]



**Figure2-2:** les types d'apprentissage automatique.

## 2.4 Deep Learning

### 2.4.1 Définition du deeplearning

Le Deep Learning est une méthode de Machine Learning qui consiste à enseigner à des ordinateurs ce dont les humains sont naturellement capables. Les applications du DL incluent des domaines tels que :

- reconnaissance d'image.
- traduction automatique.
- voiture autonome.
- diagnostic médical.
- recommandations personnalisées.
- modération automatique des réseaux sociaux
- prédiction financière et trading automatisé.
- identification de pièces défectueuses.
- détection de malwares ou de fraudes.[15]

Le Deep Learning apprend à un modèle informatique comment réaliser des tâches de classification directement à partir d'images, de textes ou d'audio. Les modèles de Deep Learning peuvent atteindre un niveau de précision exceptionnel, parfois supérieur aux performances humaines. L'entraînement des modèles s'effectue via un vaste ensemble de données labellisées et d'architectures de réseaux de neurones qui contiennent de nombreuses couches [18].

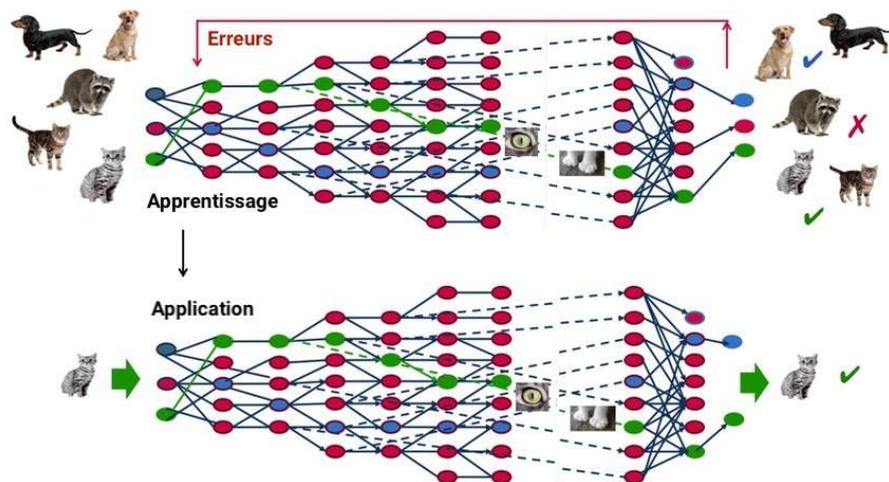
### 2.4.2 Fonctionnement du deeplearning

Le DeepLearning s'appuie sur un réseau de neurones artificiels s'inspirant du cerveau humain. Ce réseau est composé de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente. Le système apprendra par exemple à reconnaître les lettres avant

de s'attaquer aux mots dans un texte, ou détermine s'il y a un visage sur une photo avant de découvrir de quelle personne il s'agit.

À chaque étape, les « mauvaises » réponses sont éliminées et renvoyées vers les niveaux en amont pour ajuster le modèle mathématique. Au fur et à mesure, le programme réorganise les informations en blocs plus complexes. Lorsque ce modèle est par la suite appliqué à d'autres cas, il est normalement capable de reconnaître un chat sans que personne ne lui ait jamais indiqué qu'il n'ait jamais appris le concept de chat. Les données de départ sont essentielles : plus le système accumule d'expériences différentes, plus il sera performant[15].

Voici un exemple pour illustrer le fonctionnement du deeplearning :



**Figure2-3 : un exemple du fonctionnement.**

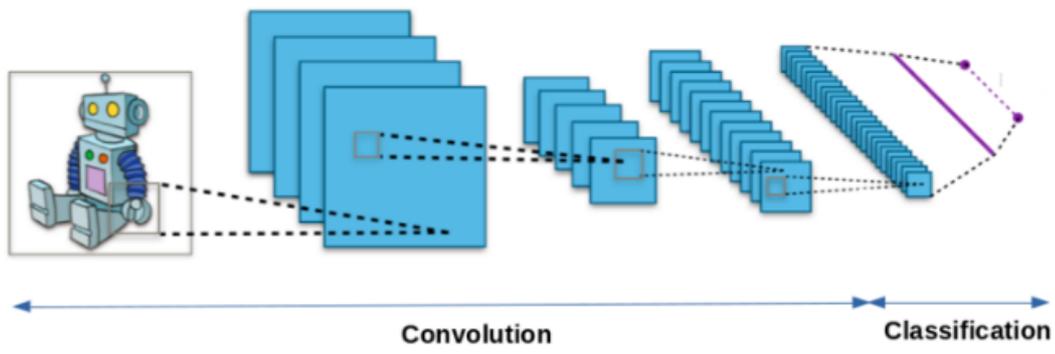
- La particularité du Deep Learning réside surtout dans la capacité du réseau de neurones à apprendre de ces erreurs (même lorsque le résultat est négatif)[18].

## 2.5 Les CNN

Un réseau de neurones à convolution (CNN ou ConvNet) est l'un des algorithmes les plus populaires pour l'apprentissage en profondeur des images et de la vidéo.

Ils sont directement inspirés du cortex visuel des vertébrés. Comme les autres réseaux de neurones, un CNN est composé d'une couche d'entrée, d'une couche de sortie et de nombreuses couches cachées entre celles-ci.

CNN se représente en général sous la forme suivante (figure 2-4):

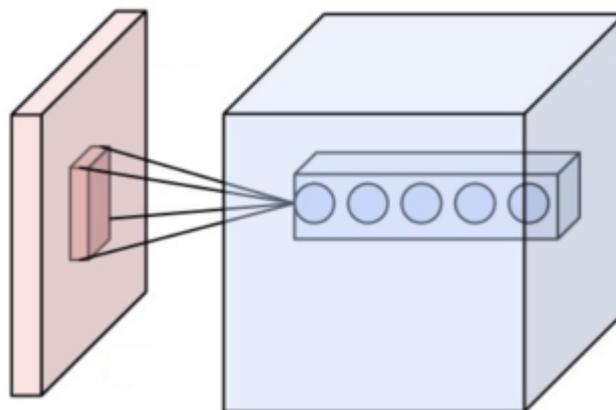


**Figure2-4:**Architecture d'un réseau de neurones CNN.

On distingue deux parties, une première partie que l'on appelle la partie convolutive du modèle et la seconde partie, que l'on va appeler la partie classification du modèle qui correspond à un modèle **MLP (Multi Layers Perceptron)**[21].

### 2.5.1 Bloc de construction

Une architecture CNN est formée par une pile de couches distinctes qui transforment le volume d'entrée en un volume de sortie (par exemple, en tenant les scores de classe) par le biais d'une fonction différentiable. Quelques types distincts de couches sont couramment utilisés. Nous en discutons plus en détail ci-dessous [20] :



**Figure2-5:** Neurones d'une couche convolutive (bleu) connectés à leur champ récepteur (rouge).

### **a La couche de convolution**

La couche convolutive est le bloc de construction central d'un CNN. Les paramètres de la couche sont constitués d'un ensemble de filtres pouvant être appris (ou noyaux), qui ont un petit champ récepteur mais s'étendent sur toute la profondeur du volume d'entrée. En conséquence, le réseau apprend les filtres qui s'activent lorsqu'il détecte un type spécifique d'entité à une certaine position spatiale dans l'entrée.

Chaque entrée du volume de sortie peut être interprétée comme une sortie d'un neurone qui examine une petite région de l'entrée et partage les paramètres avec les neurones de la même carte d'activation.[20]

- **Connectivité locale :**

Lorsqu'il s'agit d'entrées de grande dimension telles que des images, il est peu pratique de connecter des neurones à tous les neurones du volume précédent car une telle architecture de réseau ne prend pas en compte la structure spatiale des données. Les réseaux convolutifs exploitent une corrélation spatiale locale en appliquant un modèle de connectivité locale entre les neurones de couches adjacentes : chaque neurone est connecté à une petite région du volume d'entrée. L'étendue de cette connectivité est un hyperparamètre (plus détaillé dans le paragraphe 2-4-2) appelé champ récepteur du neurone. Les connexions sont locales dans l'espace (sur la largeur et la hauteur), mais s'étendent toujours sur toute la profondeur du volume d'entrée. Une telle architecture garantit que les filtres appris produisent la réponse la plus forte à un modèle d'entrée spatialement local.

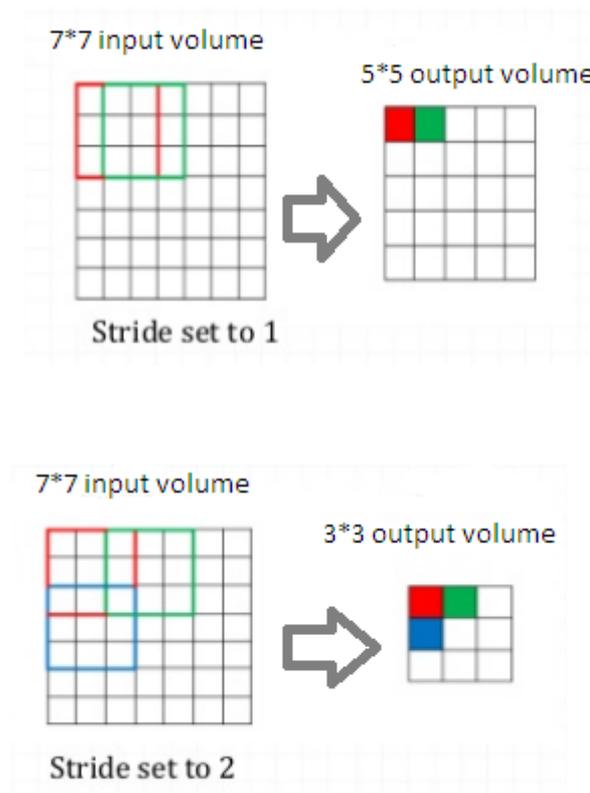
- **Aménagement d'espace**

Trois hyperparamètres contrôlent la taille du volume de sortie de la couche convolutive: la profondeur, la foulée et le remplissage à zéro.

**1.** La profondeur du volume de sortie : contrôle le nombre de neurones de la couche qui se connectent à la même région du volume d'entrée. Tous ces neurones apprendront à s'activer pour différentes caractéristiques de l'entrée. Par exemple, si la première couche de convolution prend l'image brute en

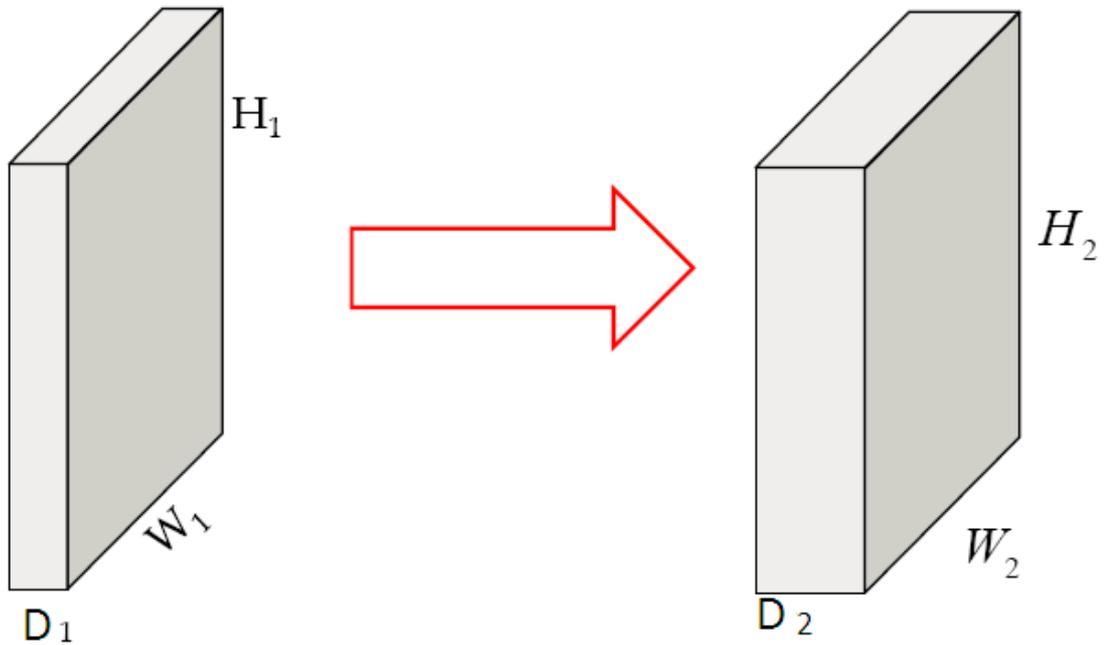
entrée, différents neurones situés le long de la dimension de profondeur peuvent s'activer en présence de divers bords orientés ou de gouttes de couleur.

**2. Stride (foulée) :** contrôle la manière dont les colonnes de profondeur autour des dimensions spatiales (largeur et hauteur) sont allouées. Lorsque la foulée est égale à 1, nous déplaçons les filtres d'un pixel à la fois. Lorsque la foulée est 2 (ou inhabituellement 3 ou plus, bien que cela soit rare en pratique), les filtres sautent de 2 pixels à la fois lorsque nous les faisons glisser. Cela produira de plus petits volumes de sortie dans l'espace.



**Figure 2-6:**exemple de stride .

**3.**Le remplissage à zéro permet de contrôler la taille spatiale du volume de sortie. En particulier, il est parfois souhaitable de préserver exactement la taille spatiale du volume d'entrée.



**Figure2-6:** Tailles entrée et sortie.

A l'entrée un volume de taille  $W_1 \times H_1 \times D_1$  :

Nécessite quatre hyperparamètres: Nombre de filtres  $K$  et leur étendue spatiale  $F$ , et la foulée  $S$ , la quantité de remplissage zéro  $P$ .

Produit un volume de taille  $W_2 \times H_2 \times D_2$  où:

$$H_2 = \frac{H_1 - F + 2P}{S} + 1 \text{ (eq2.1)}$$

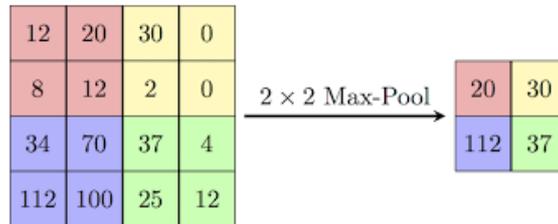
$$W_2 = \frac{W_1 - F + 2P}{S} + 1 \text{ (eq2.2)}$$

$$D_2 = K \text{ (eq2.3)}$$

### ***b La couche de pooling***

Il est courant d'insérer périodiquement une couche de regroupement entre des couches de convolution successives dans une architecture CNN. Sa fonction est de réduire progressivement la taille spatiale de la représentation afin de réduire le nombre de paramètres et de calculs dans le réseau, et donc de contrôler également le rajustement. La couche de pooling fonctionne indépendamment sur chaque tranche de profondeur d'opération MAX. La forme la plus courante est une couche de regroupement avec des filtres de taille  $2 \times 2$  appliqués avec une foulée de 2 échantillons inférieurs pour chaque tranche de profondeur de l'entrée de 2 par la

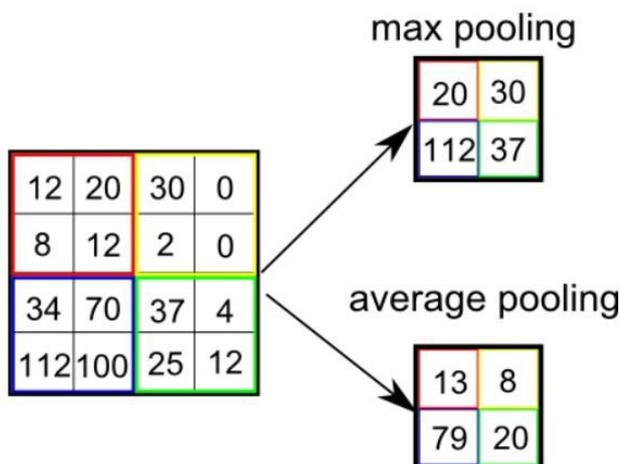
largeur et par la hauteur, éliminant 75% des activations. Chaque opération MAX prend dans ce cas un maximum sur 4 nombres (petite région 2 × 2 dans une tranche de profondeur). La dimension de profondeur reste inchangée [21].



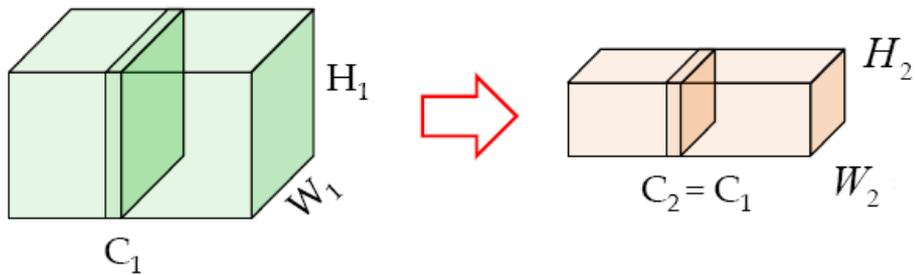
**Figure 2-7:** Pooling maximum avec un filtre 2x2.

Il existe deux types de pooling : pooling maximum et pooling moyen. Max Pooling renvoie la valeur maximale de la partie de l'image couverte par le noyau. Le pooling moyen renvoie la moyenne de toutes les valeurs de la partie de l'image couverte par le noyau.

La mise en commun maximale fonctionne également en tant que supprimeur de bruit. Il élimine complètement les activations bruyantes et effectue également le dé bruitage avec réduction de la dimensionnalité. D'autre part, le pooling moyen effectue simplement une réduction de la dimensionnalité en tant que mécanisme de suppression de bruit. Par conséquent, nous pouvons dire que Max Pooling fonctionne beaucoup mieux que le Pooling moyen [20].



**Figure2-8:** deux types de pooling.



**Figure 2-9:** Tailles entrée et sortie de la couche pooling.

- Accepte un volume de taille  $W_1 \times H_1 \times D_1$  :

Nécessite deux hyperparamètres: leur étendue spatiale  $F$ , et la foulée  $S$ .

- Produit un volume de taille  $2 \times 2 \times 2$  où:

$$W_2 = \frac{(W_1 - F)}{S} + 1$$

(eq 2.4)

$$H_2 = \frac{(H_1 - F)}{S} + 1 \text{ (eq 2.5)}$$

$$D_2 = D_1 \text{ (eq 2.6)}$$

Introduit les paramètres zéro, car il calcule une fonction fixe de l'entrée. Notez qu'il n'est pas courant d'utiliser le remplissage avec des zéros pour les couches de pooling, il est intéressant de noter qu'il n'y a que deux variantes couramment observées de la couche de pooling maximale trouvée dans la pratique:

- Une couche de regroupement avec  $F = 3, S = 2$  (également appelée regroupement par recouvrement).
- Et plus communément  $F = 2, S = 2$ . Les tailles de regroupement avec des champs récepteurs plus grands sont trop destructives.[21]

### c La couche de Relu

ReLU est l'abréviation de Unités Linéaires Rectifiées. C'est une couche de neurones qui applique la fonction d'activation non saturante :

$$f(x) = \max(0, x) \text{ (eq2.7)}$$

Il augmente les propriétés non linéaires de la fonction de décision et du réseau global sans affecter les champs récepteurs de la couche de convolution. D'autres fonctions sont également utilisées pour augmenter la non-linéarité, par exemple la tangente hyperbolique saturante, et la fonction sigmoïde (voir tableau 1-3). Par rapport à d'autres fonctions, l'utilisation de ReLU est préférable, car elle permet au réseau de neurones de s'entraîner plusieurs fois plus rapidement, sans perte significative de précision[20].

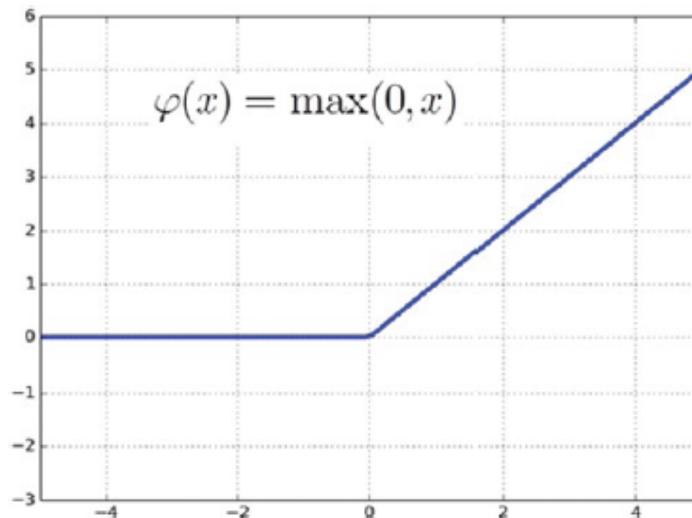


Figure 2-10: la fonction d'activation Relu.

### d La couche entièrement connectée FC(fullyconnected)

Enfin, après plusieurs couches de pool convolutionnels et maximum, le raisonnement de haut niveau dans le réseau de neurones est effectué via des couches entièrement connectées. Les neurones d'une couche entièrement connectée ont des connexions complètes avec toutes les activations de la couche précédente, comme dans les réseaux de neurones classiques (fig2-12)[20].

### e La couche de perte(loss)

La couche de perte spécifie comment l'entraînement du réseau pénalise l'écart entre le signal prévu et réel. Elle est normalement la dernière couche dans le réseau. Diverses fonctions de perte adaptées à différentes tâches peuvent y être utilisées. La perte « Soft max » est utilisée pour prédire une seule classe parmi K classes mutuellement exclusives. La perte par entropie croisée sigmoïde est utilisée pour prédire K valeurs de probabilité indépendante dans [0,1]. La perte euclidienne est utilisée pour régresser vers des valeurs réelles [20]

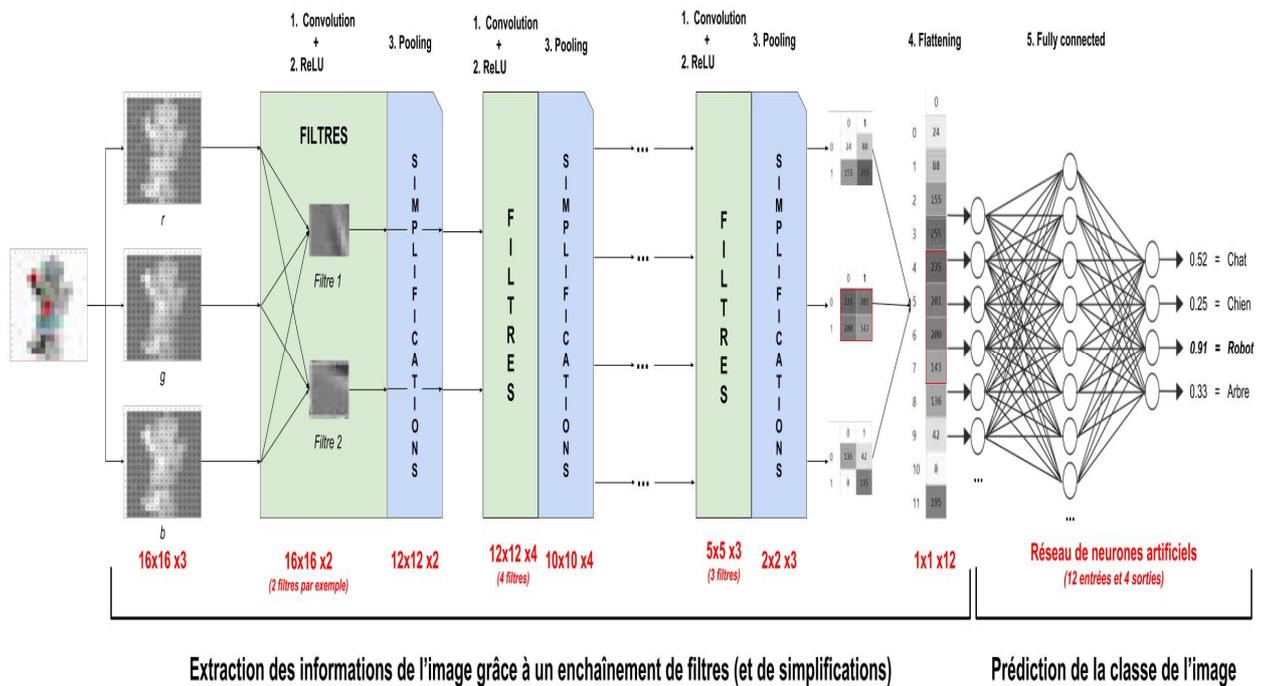


Figure2-11: les différentes couches CNN.

### 2.5.2 Les hyper paramètres

Les CNN utilisent plus d'hyperparamètres qu'un MLP standard. Bien que les règles habituelles en matière de vitesse d'apprentissage et de constantes de régularisation s'appliquent, il convient de garder à l'esprit les points suivants lors de l'optimisation des réseaux de convolution[20].

### ***a* Nombre de filtres**

Étant donné que la taille de la carte des caractéristiques diminue avec la profondeur, les couches proches de la couche en entrée ont tendance à comporter moins de filtres, tandis que les couches les plus hautes peuvent en contenir davantage. Pour égaliser le calcul à chaque couche, le produit du nombre d'entités et du nombre de positions de pixels est généralement choisi pour être à peu près constant sur toutes les couches. Conserver les informations sur l'entrée nécessiterait de garder le nombre total d'activations (nombre de cartes de caractéristiques multiplié par le nombre de positions de pixels) afin qu'il ne diminue pas d'une couche à l'autre. Le nombre de cartes de caractéristiques plus contrôle directement la capacité et dépend du nombre d'exemples disponibles et de la complexité de la tâche.

### ***b* Forme du filtre**

Les formes de champ communes trouvées dans la littérature varient considérablement, Le défi consiste donc à trouver le bon niveau de granularité de manière à créer des abstractions à la bonne échelle, en fonction d'un ensemble de données particulier.

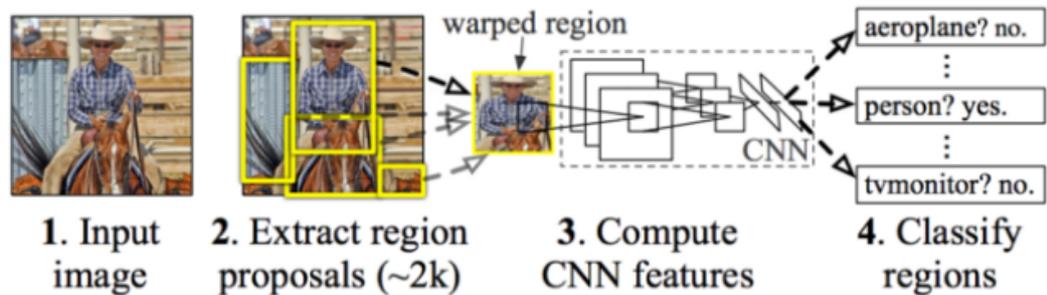
### ***c* Taille des filtres**

Les valeurs typiques sont de taille 2x2. Des volumes d'entrée très importants peuvent justifier un regroupement de taille 4x4 dans les couches inférieures. Cependant, le choix de formes plus grandes réduira considérablement la dimension du signal et risque d'abandonner trop d'informations. Souvent, les fenêtres de pooling ne se chevauchant pas fonctionnent mieux.

## **2.6 RCNN**

Les réseaux de convolution par région (R-CNN) sélectionnent d'abord plusieurs régions proposées à partir d'une image, puis étiquettent leurs catégories et leurs boîtes englobantes. Ensuite, ils utilisent un CNN pour effectuer un calcul en avant afin d'extraire des entités de chaque zone proposée. Ensuite, nous utilisons les caractéristiques de chaque région proposée pour prédire leurs catégories et leurs boîtes englobantes. La méthode du réseau convolutif basé sur la région (RCNN) permet d'obtenir une excellente précision de détection des objets [21][22].

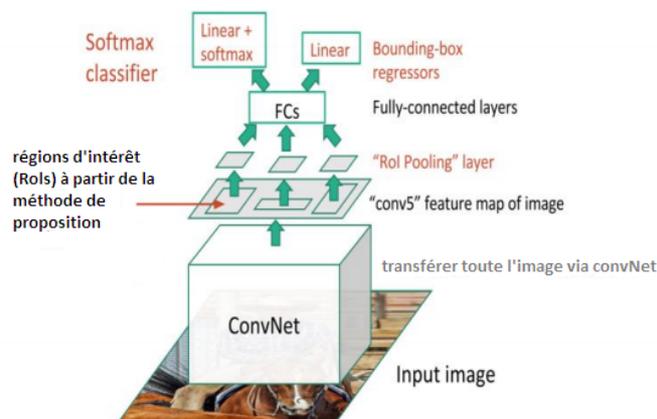
Pour simplifier la tâche de R-CNN, on le réduit aux étapes suivantes [23] :



**Figure 2-12:** Procédure de détection et de localisation d'objets dans R-CNN.

- 1) R-CNN crée ces cadres de sélection, ou propositions de région, en utilisant un processus appelé Recherche sélective. À un niveau élevé, la recherche sélective (indiquée dans fig 2-11 ci-dessus) examine l'image à travers des fenêtres de tailles différentes. Pour chaque taille, essayez de regrouper les pixels adjacents par texture, couleur ou intensité pour identifier les objets.
- 2) Une fois les propositions créées, R-CNN convertit la région en une taille carrée standard et la transmet à une version modifiée d'AlexNet comme indiqué ci-dessus.
- 3) Sur la dernière couche du réseau CNN, R-CNN ajoute une machine à vecteurs de support (SVM) qui classifie simplement s'il s'agit d'un objet et, le cas échéant, quel objet. C'est l'étape 4 de l'image ci-dessus.

## 2.6.1 Fast-RCNN



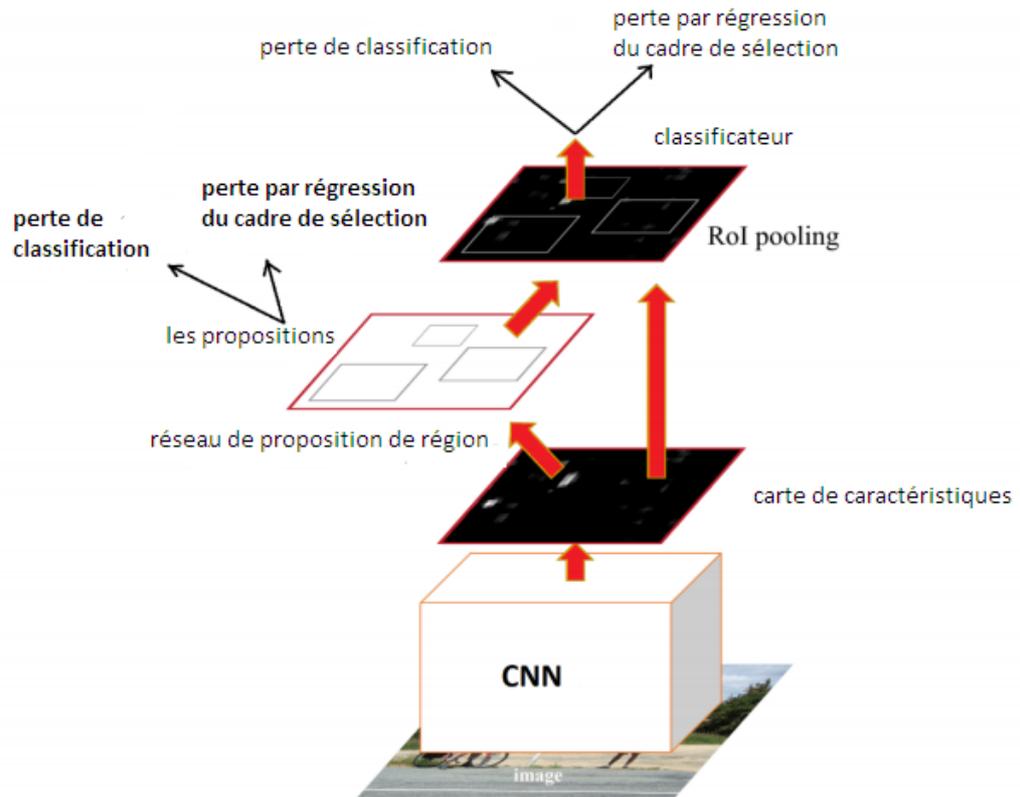
**Figure2-13:** Fast R-CNN Architecture.

En 2015, Ross Girshick, le premier auteur de R-CNN , a résolu ces problèmes et l'architecture résultante s'appelle Fast R-CNN . Dans cette architecture, nous introduisons une image, puis nous la transmettons via un CNN (AlexNet), qui produit une carte de présentation des images à partir de laquelle nous obtenons nos propositions de région qui produit une carte de présentation des images à partir de laquelle nous obtenons nos propositions de région. La prochaine étape est la mise en commun du RoI. RoIPool partage le passage en avant d'un CNN pour une image dans ses sous-régions. Ensuite, les fonctionnalités de chaque région sont regroupées (généralement en utilisant max pooling). Par conséquent, en prenant un seul passage de l'image originale.

Deuxièmement, Fast R-CNN forme conjointement le CNN, le classificateur et le régresseur de boîte englobante dans un seul modèle. Fast R-CNN a remplacé le classificateur SVM par une couche softmax au-dessus du CNN pour générer une classification. Il a également ajouté une couche de régression linéaire parallèle à la couche softmax pour générer les coordonnées du cadre de sélection. De cette manière, toutes les sorties nécessaires provenaient d'un seul réseau [21].

### **2.6.2 Faster-RCNN**

En 2015, une équipe de Microsoft Research composée de ShaoqingRen, de Kaiming He, de Ross Girshick et de Jian Sun a créé collectivement Faster R-CNN. Elle a corrigé le goulet d'étranglement restant dans le processus Fast R-CNN: le proposant régional. Comme nous l'avons vu, la toute première étape de la détection des emplacements d'objets consiste à générer un ensemble de boîtes englobantes potentielles ou de régions d'intérêt à tester. Dans Fast R-CNN, ces propositions ont été créées à l'aide de la recherche sélective, un processus assez lent qui s'est avéré être le goulot d'étranglement de l'ensemble du processus.



**Figure 2-14:** Faster R-CNN Architecture.

Semblable à Fast R-CNN dans cette architecture, une image est transmise à un CNN à partir duquel nous obtenons une carte de caractéristiques convolutionnelles. Elle passe ensuite par un réseau de proposition de région formé à la base données de vérité pour produire les propositions de région.

Le réseau de proposition de région fonctionne en faisant passer une fenêtre glissante sur la carte des caractéristiques de CNN et à chaque fenêtre, en générant  $k$  boîtes de sélections potentielles et des scores indiquant la qualité de chacune de ces boîtes.

Intuitivement, nous savons que les objets d'une image doivent correspondre à certains formats et tailles courants. De cette manière, nous créons  $k$  ratios d'aspect communs tels que nous appelons des boîtes d'ancrage. Pour chacune de ces boîtes d'ancrage, nous produisons une boîte qui les englobe toutes avec un score par position dans l'image.

Enfin, le regroupement des RoI est effectué pour faire correspondre les entrées du CNN, puis l'image passe par la couche de convolution entièrement connectée [21].

## 2.7 Conclusion

Dans ce chapitre les ébauches importantes du Deep Learning sont citées et expliquées, et le CNN qui est l'un des types les plus utilisés dans le traitement d'image, car il joue un rôle majeur dans les domaines des classifications d'image, et possède la capacité de faire évoluer ses propres paramètres.

Dans le prochain chapitre, nous entamerons la réalisation de l'algorithme de DL permettant de réaliser notre travail sous Matlab, nous expliquerons ainsi les outils et paramètres utilisés pour ce but.

## Chapitre 3 Expérimentations et résultats

---

### 3.1 Introduction

Pour l'implémentation de notre réseau neurone pour la détection de la rétinopathie diabétique nous avons choisis le langage de programmation Matlab. Nous installons d'abords Matlab 2018 car c'est la nouvelle version avec de nouvelles fonctions que nous allons utiliser dans notre programme.

Pour le type de notre architecture de réseau neurone nous utiliserons le réseau CNN ,ainsi que toutes ses fonctions et ses méthodes qui sont nécessaires pour l'entraînement de notre réseau neurone et ils seront expliqués dans ce chapitre.

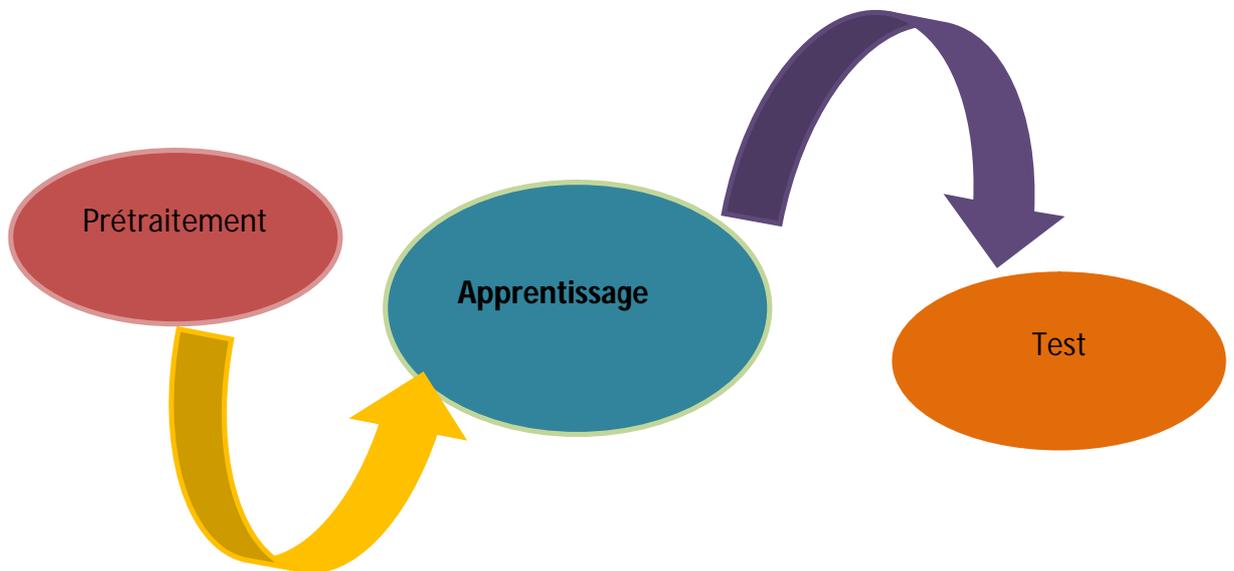


Figure 3-15:plan de travail.

## 3.2 Présentation des outils de développement

### 3.2.1 Matériel

Pour le matériel nous avons essayé avec plusieurs pc (plusieurs CPU) avec différente capacité de mémoire, et cela pour observer la différence dans la durée et la capacité de chaque PC dans l'apprentissage et bien sûr choisir de travailler avec le meilleur.

- PC personnel acer I5 avec un 4GB (on rajouter 2GB) capacité mémoire, et un processeur Intel® coré™ 2 Duo 2.53 GHz, avec Windows 10 édition intégral, service pack 1 64 bit type système.
- PC personnel acer I3 avec un 4GB capacité mémoire, et un processeur Intel® coré™ 2 Duo 2.53 GHz, avec Windows 7 édition intégral, service pack 1 64 bit type système.
- PC personnel DELL, avec Windows 8.1 service 64-bit , et processeur Intel® Core™ i7-4790 CPU @ 3.60GHz (8CPUs) avec RAM du 16 GB.

### 3.2.2 Deep learning et Matlab

MATLAB\_ (Matrix LABoratory) est un logiciel interactif basé sur le calcul matriciel. Il est utilisé dans les calculs scientifiques et les problèmes d'ingénierie parce qu'il permet de résoudre des problèmes numériques complexes en moins de temps requis par les langages de programmation courant, et ce grâce à une multitude de fonctions intégrées et à plusieurs programmes outils testés et regroupés selon l'usage dans des dossiers appelés boites à outils ou "toolbox".

Son avantage, par rapport aux autres langages, est de simplifier au maximum la transcription en langage informatique d'un problème mathématique, en utilisant une écriture la plus proche possible du langage naturel scientifique. Bien qu'il ne soit pas le logiciel ni le langage de programmation le plus utilisé ni le plus rapide dans le deep learning néanmoins ce logiciel a prouvé son efficacité que ce soit dans ce domaine ou dans celui de l'intelligence artificiel grâce à son outil « neural network toolbox » celui-ci simplifier et aide beaucoup dans l'implémentation et l'entraînement des réseaux neurones.

### 3.3 Prétraitement

Le prétraitement convertit l'image en une forme appropriée pour la segmentation et l'extraction de caractéristiques avec moins de complexité. L'élimination du bruit, le lissage, l'affûtage et la normalisation sont des opérations de prétraitement courantes. Le filtrage d'images est utilisé pour lisser, affiner, éliminer le bruit et détecter des contours. Les images avant et après le prétraitement sont respectivement représentées sur la figure 3-2 et l'organigramme de prétraitement (figure 3-3).

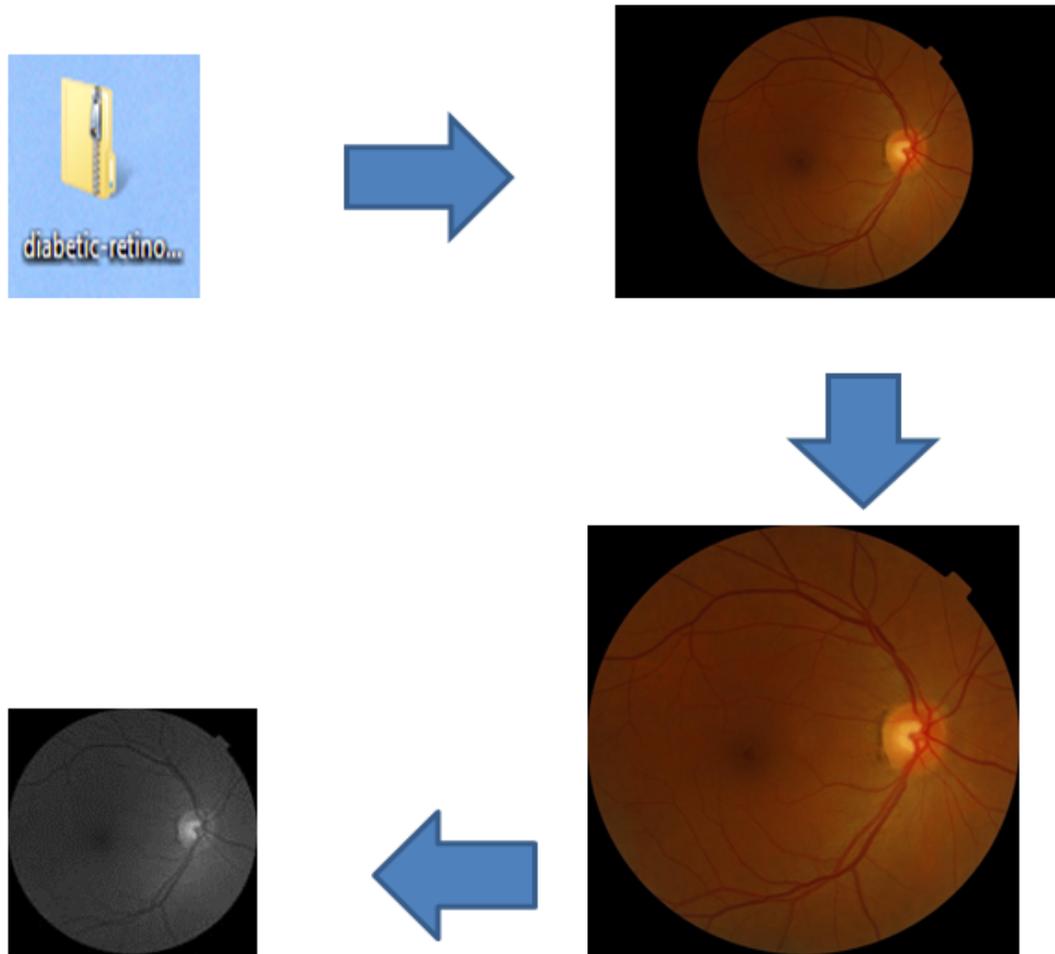


Figure 3-2: exemple de prétraitement.

#### 3.3.1 Télécharger la base de données

Tout d'abords pour nos données nous avons utilisé une base de données kaggle :c'est une plateforme web organisant des compétitions en science des données. Sur cette plateforme, les entreprises proposent des problèmes en science des données et offrent un prix aux *datalogistes* obtenant les meilleures performances. Que nous avons

téléchargée du site (voici le lien :[kaggle.com/c/diabetic-retinopathy-contest/data](https://www.kaggle.com/c/diabetic-retinopathy-contest/data)) celle-ci comporte plus de 1300 images.

### 3.3.2 Encadrer les images :

Nous avons encadré les images pour réduire la taille et enlever les bordures inutiles grâce au logiciel JPEGCROPS qui est une application PC qui permet de recadrer des images et photos JPG.

### 3.3.1 Transformer les images en niveau de gris

Enfin nous devons transformer les images en niveau de gris pour réduire leurs tailles puis on fait l'uniformisation des images à la taille 145x145 (pour rendre l'apprentissage plus rapide) tous en gardant les caractéristiques de celles-ci.

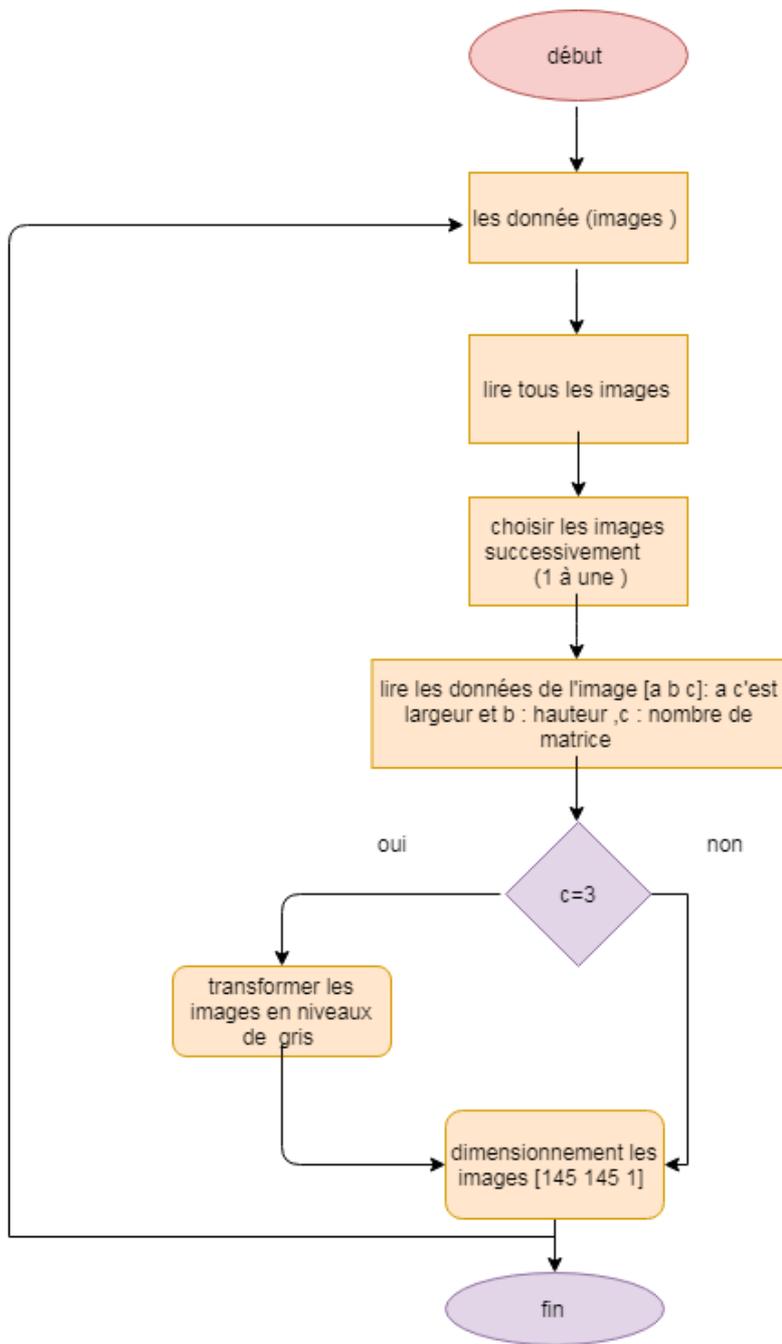
#### Remarque :

- On a travaillé avec la fonction « imageDatastore » c'est une fonction dans matlab pour le deep learning, qui en fait une utilisation efficace avec les grandes collections d'images, et cela s'explique qu'en l'utilisant matlab n'a pas besoin d'importer toutes les données mais utilise seulement le chemin (figure 3-3), (car le deep learning travaille sur larges data, on ne peut pas tous les importer).

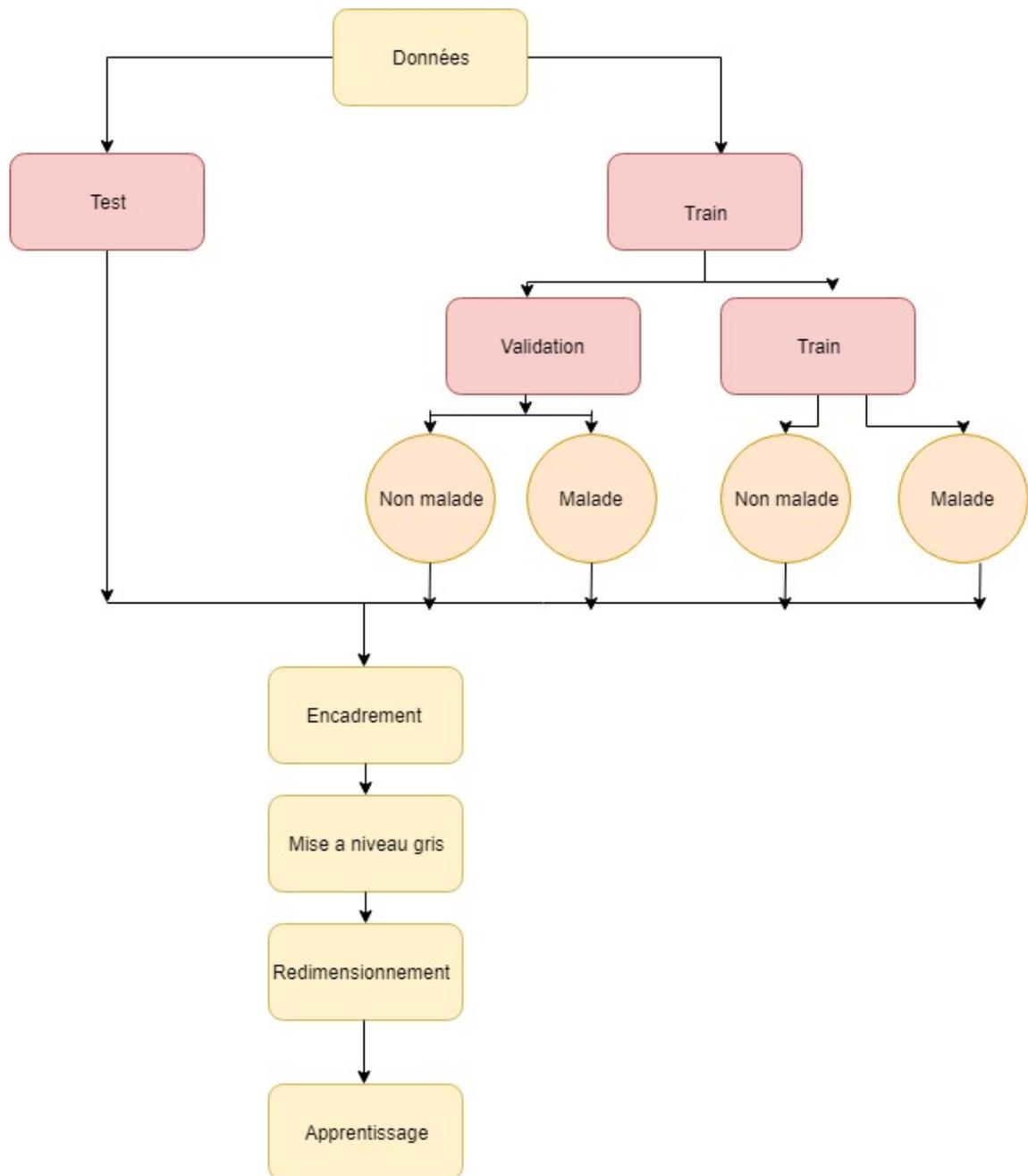
```
digitaldatapath=fullfile('C:\Users\User\Desktop\train')  
imds=imageDatastore(digitaldatapath,'IncludeSubfolders',true,'labelsource','foldernames')
```

Figure 3-3:load images.

- Notre programme de prétraitement : s'il trouve une image ou image noire et blanc (2D et 3D) la transforme en niveau de gris, et si non il la laisse grise et apres s'il ne la trouve pas la taille [145x145] la transforme et la deposee dans le même dossier ( les images seront classées malade et non malade )(voir figure 3-3).
- Nous avons aussi uniformisé les mêmes images à la taille [250x250x1] mais comme nous allons en parler l'apprentissage prend plus de temps et nous donne le même resultat donc nous avons decidée de travailler avec [145x145x1] pour gagner du temps.



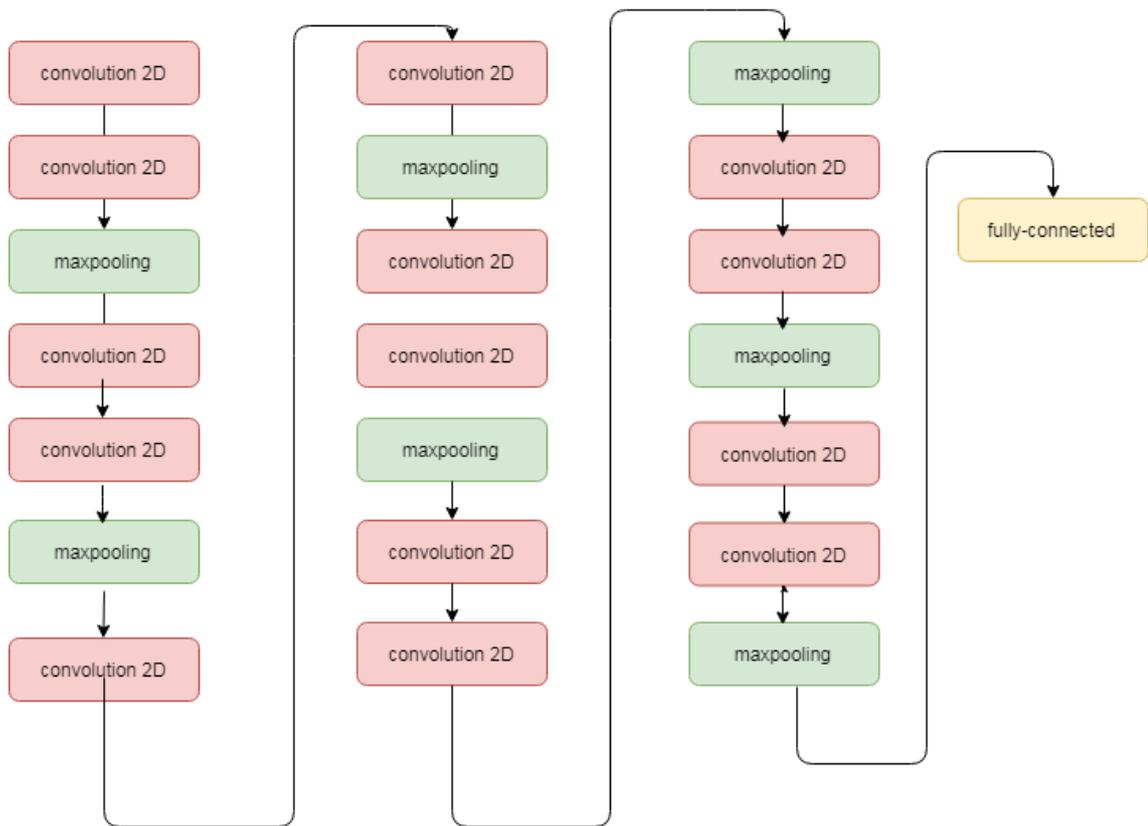
**Figure 3-4:** organigramme de programme de prétraitement.



**Figure 3-5:** organigramme de prétraitement.

### 3.4 Apprentissage

Pour l'apprentissage nous choisirons l'architecture CNN (convolutional neural network) ou réseau de neurones a convolution, car c'est l'architecture la mieux adaptée pour le traitement et la classification d'images, pour avoir le meilleur résultat nous avons essayé avec plusieurs réseaux avec différentes couches, tout d'abord nous avons essayé avec un réseau à 9 couche (filtre 3\*3) suivis par des couches de relu(rectifiedlinear.....) puis des couches de MaxPooling avec 20 Epochs (c'est à dire 200 itérations) et enfin une couche fullyconnectedlayer, mais le résultat n'été pas bien alors nous avons essayé avec 22 couches (filtre 5\*5) .( figure 3-6) :



**Figure 3-6:** architecture de CNN.

Notre réseau de neurone CNN se divise en trois couches principales comme on l'a vu dans la figure 3-6 :

**Input layer** : contient toutes les images utilisées dans l'entraînement malade et non malade nous avons aussi défini les filtres et le nombre de filtrage que nous allons utiliser dans notre apprentissage.

```

inputLayer=imageInputLayer([145 145 1]);
filterSize=[5 5];
numFilters=20;
  
```

**Figure 3-7:** la 1 ère couche.

**Middle layer** : les différentes couches de convolution et max pooling et la fonction d'activation Dans notre programme nous avons utilisé 22couches,à chaque couche de convolution est poursuivie par fonction d'activation (RELU) et la couche maxpooling .

layers = 22x1 Layer array with layers:

```
1 " Image Input      145x145x1 images with 'zero-center' normalization
2 " Convolution      20 5x5 convolutions with stride [1 1] and padding [1 1 1 1]
3 " ReLUReLU
4 " Max Pooling      2x2 max pooling with stride [2 2] and padding [0 0 0 0]
5 " Convolution      8 5x5 convolutions with stride [1 1] and padding [1 1 1 1]
6 " ReLUReLU
7 " Max Pooling      2x2 max pooling with stride [2 2] and padding [0 0 0 0]
8 " Convolution      16 5x5 convolutions with stride [1 1] and padding [1 1 1 1]
9 " ReLUReLU
10 " Max Pooling     2x2 max pooling with stride [2 2] and padding [0 0 0 0]
11 " Convolution     32 5x5 convolutions with stride [1 1] and padding [1 1 1 1]
12 " ReLUReLU
13 " Max Pooling     2x2 max pooling with stride [2 2] and padding [0 0 0 0]
14 " Convolution     20 20x20 convolutions with stride [1 1] and padding 'same'
15 " ReLUReLU
16 " Convolution     20 25x25 convolutions with stride [1 1] and padding 'same'
17 " ReLUReLU
18 " Max Pooling     2x2 max pooling with stride [2 2] and padding [0 0 0 0]
19 " Fully Connected 2 fully connected layer
20 " ReLUReLU
21 " Softmaxsoftmax
22 " Classification Output crossentropyex
```

**Figure 3-8 :** la couche moyenne.

**Final layer :** Dans la dernière couche finale Layers nous avons utilisé fullyconnected layer avec 2 sorties car on a deux catégories d'individus : ceux qui sont malades et d'autres qui ne sont pas.

Ensuite nous avons relié les trois principales couches :

```
layers=[
    inputLayer
    middleLayers
    finalLayers
]
```

**Figure3-9:** déclaration des couches.

- A l'aide de « DeepDreamImage » on a affichée la dernière couche :

```

layer = 21;
name = net.Layers(layer).Name;
channels = 1:56;
I = deepDreamImage(net, layer, channels, 'PyramidLevels', 1);
figure
I = montage(I, 'ThumbnailSize', [64 64]);
imshow(I)
title(['Layer ', name, ' Features'])

```

Figure 3-10 : programme pour afficher la couche .

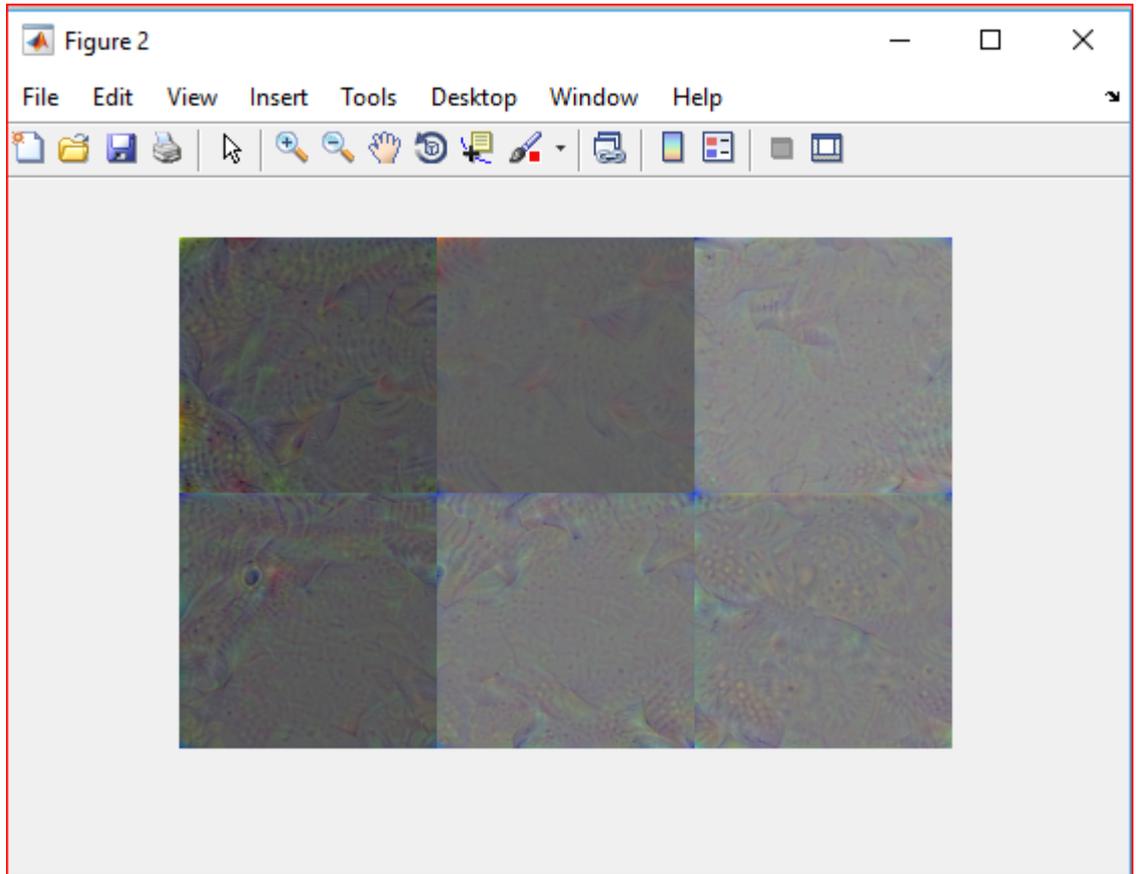
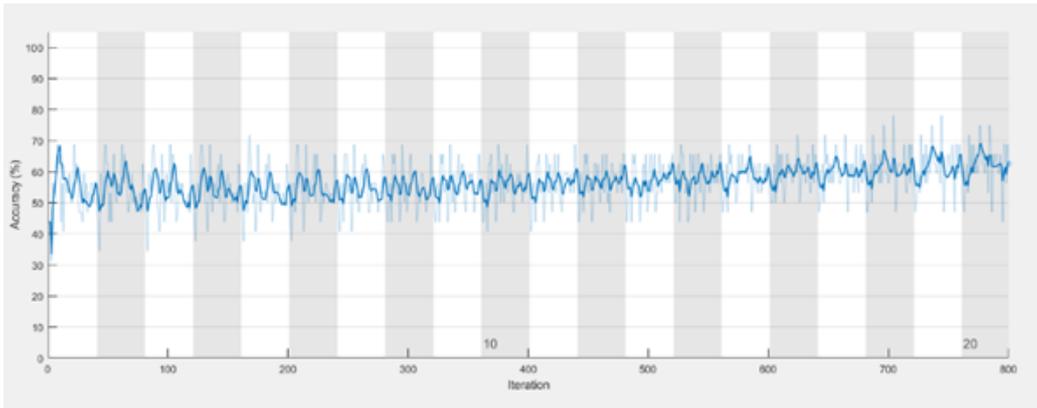


Figure 3-11 : couche fullyconnected .

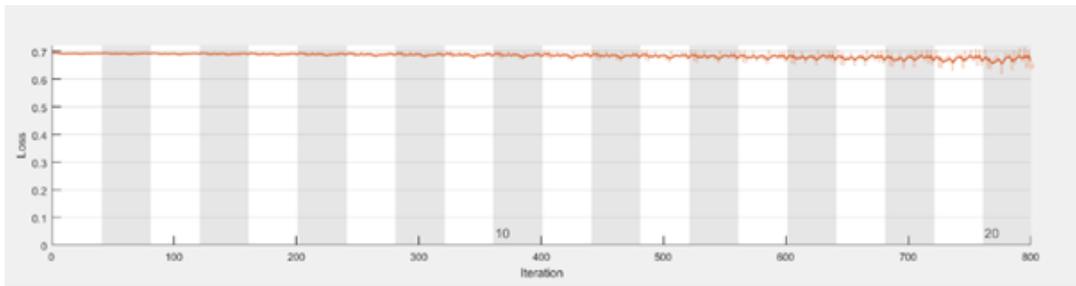
### 3.5 Résultats d'apprentissage :

On a lancé plusieurs apprentissages et on a chaque fois fait des changements :

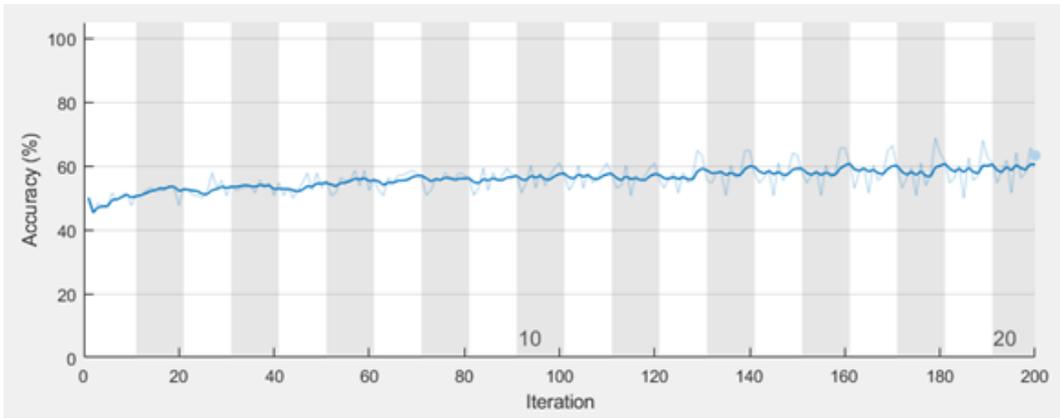
- a Résultat 1 :** dans les premiers essais on a changé les hyperparamètres (numéro de filtre et le nombre de filtre) en gardant le même nombre de couches et la même fonction sigmoïde on a obtenu les résultats suivants :



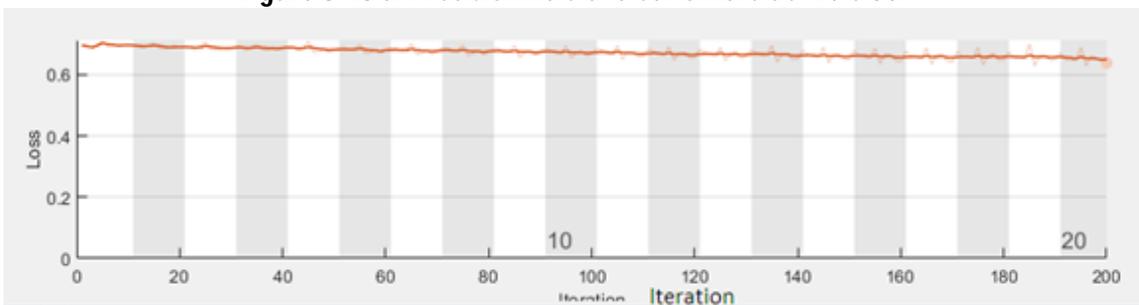
**Figure 3-12a :** Précision filtre 5\*5 nombre de filtre 20.



**Figure 3-12 b :** loss (perte )filtre 5\*5 nombre de filtre 20.



**Figure 3-13 a:** Précision filtre 5\*5 et nombre de filtre 30.



**Figure 3-13 b:** loss filtre 5\*5 et nombre de filtre 30.

### Discussion :

- Pour la figure 3-12 le taux d'apprentissage est de 70% et le taux de perte 0.695
- Pour la figure 3-13 le taux d'apprentissage est 60% et le taux de perte est à peu près le même que le premier.
- Nous remarquons que figure 3-12 c'est à dire qu'on utilisant un filtre 5\*5 avec un nombre de filtres 20 on obtient un meilleur résultat qu'on utilisant un filtre 5\*5 avec un nombre de filtre 30.

### Résultat:

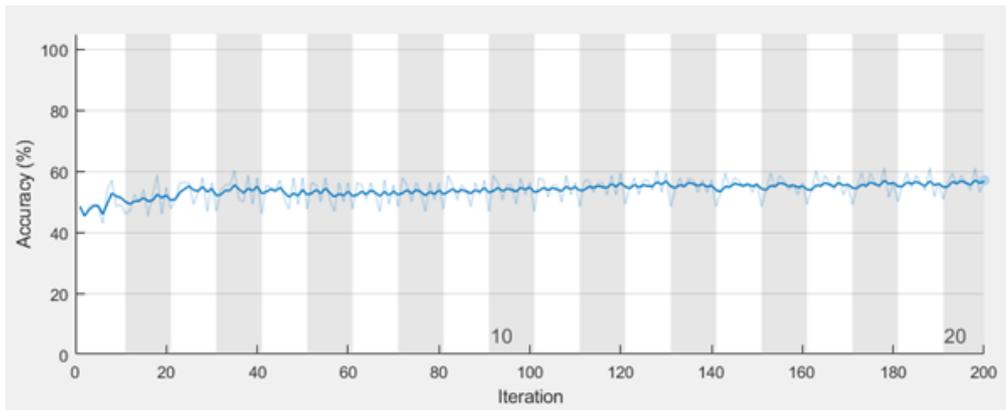


Figure 3-14 a: Précision filtre 3\*3 et nombre de filtre 20.

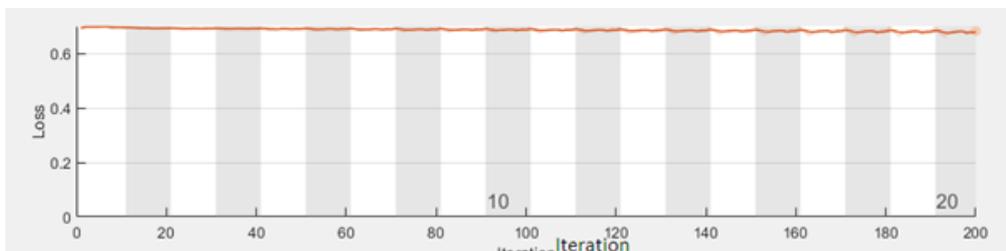


Figure 3-14 b: loss filtre 3\*3 et nombre de filtre 20.

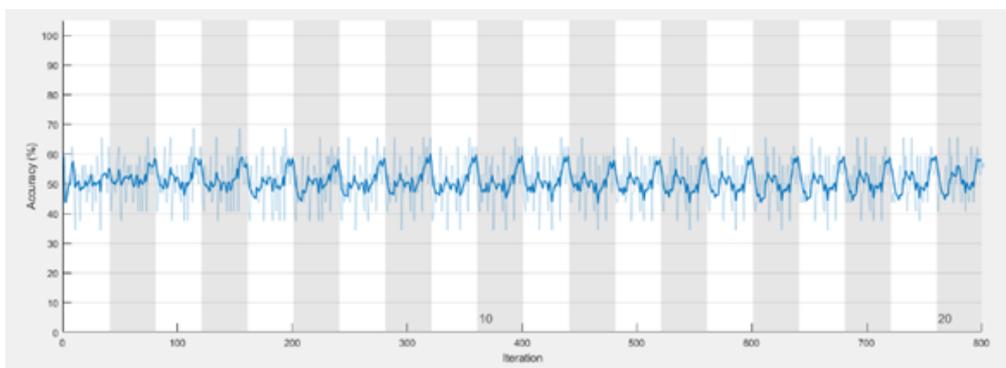
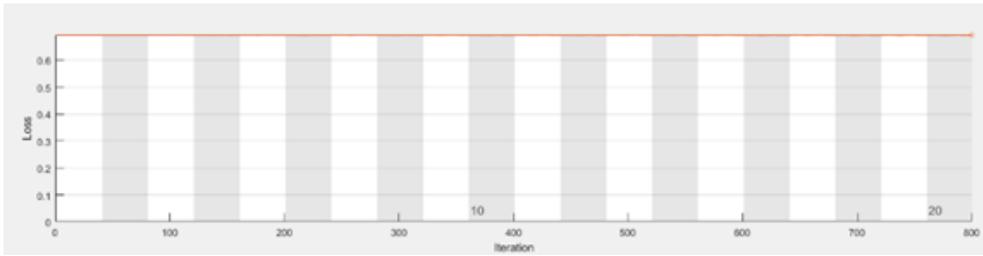


Figure 3-15 a: Précision filtre 5\*5 et nombre de filtre 20.

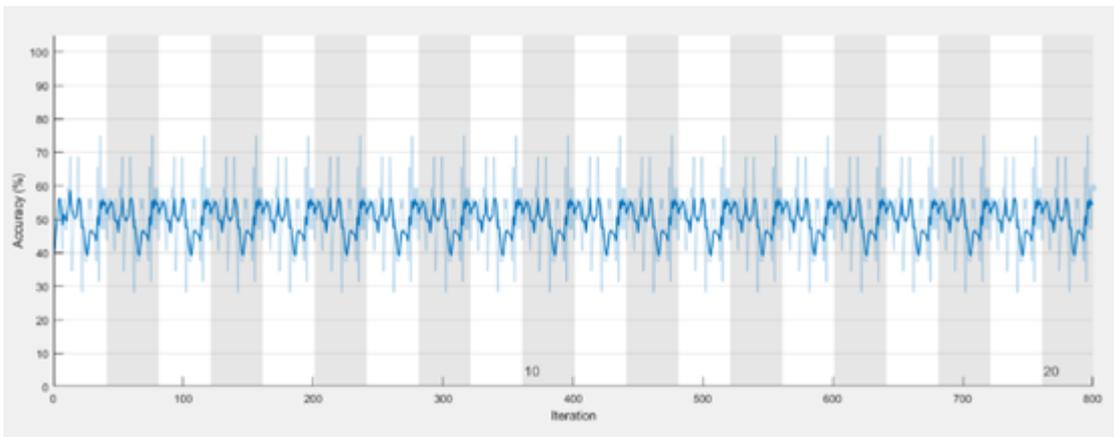


**Figure 3-15 b:**loss filtre 5\*5et nombre de filtre 20.

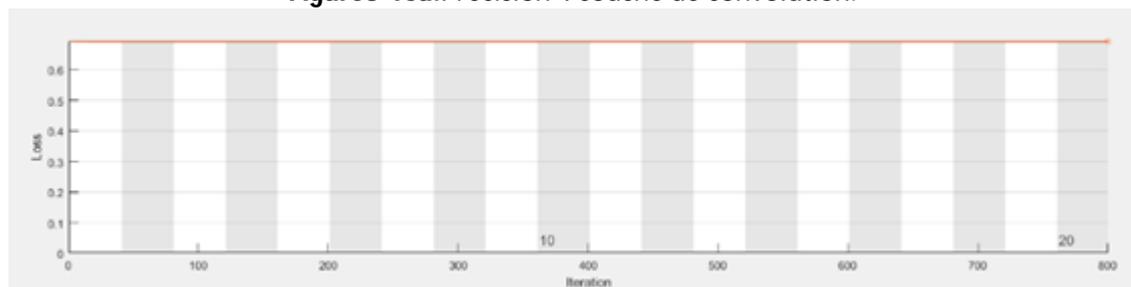
**Discussion :**

- Le taux d'apprentissage pour cet essai (figure 3-14) est de 58% et le taux de perte est 0.699.
- Pour figure (3-15), nous obtenons a peu près les mêmes résultats c'est-à-dire 58% pour la précision et 0.699 pour la perte.
- Nous remarquons que nous avons obtenu un meilleur résultat avec un filtre 5\*5 qu'avec un filtre de 3\*3.

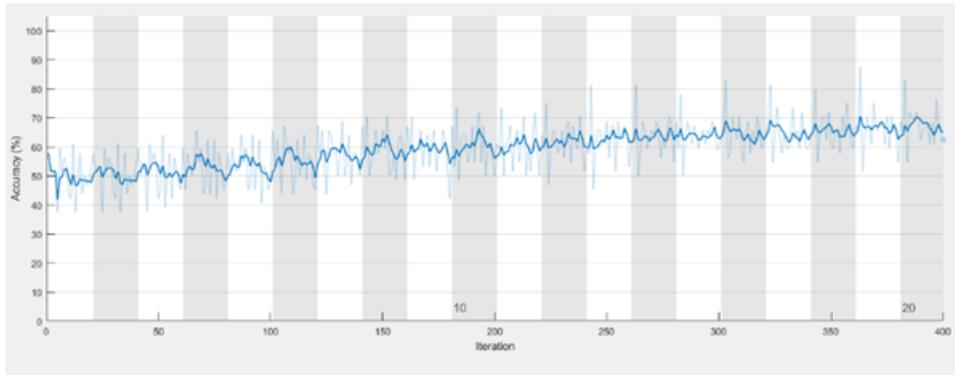
**b Résultats 2 :** dans un deuxième essai on a travaillé avec un filtre 5\*5 et nombre de filtre 30 et avec la fonction d'activation sigmoïde, et on a changé les couches de convolution on a rajouté d'autre couches les résultats sont présentés ci-dessous :



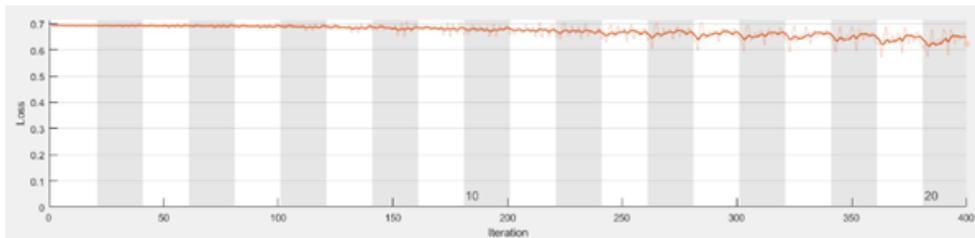
**Figure3-16a:**Précision 4 couche de convolution.



**Figure3-16b:**loss4 couche de convolution.



**Figure 3-17 a** :Précision 5 couche de convolution.

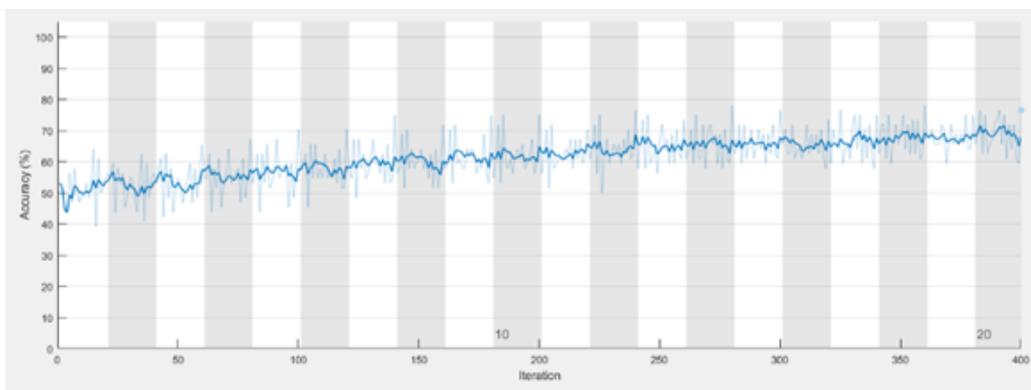


**Figure 3-17 b**:loss 5 couche de convolution.

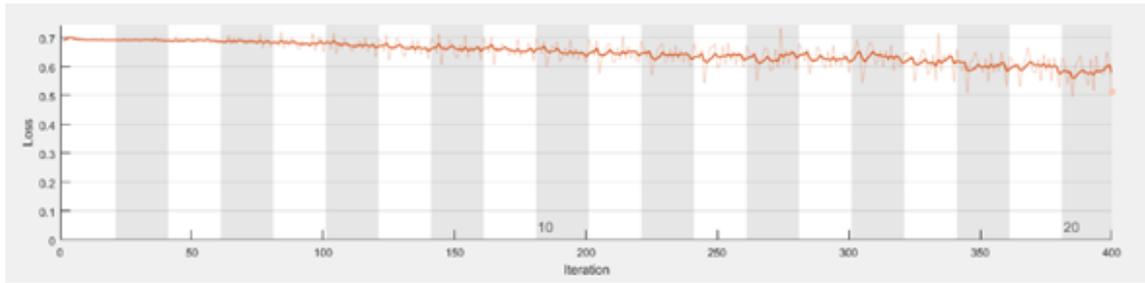
### Discussion :

- Taux d'apprentissage 60% et le taux de perte 0.700 de la figure 3-16 .
- On a rajoutée une autre couche de convolution (figure 3-17) nous remarquons le résultat est meilleur que le premier(figure 3-16) ou nous avons seulement modifié les numéros de filtre de la dernière couche.

**c Résultats 3** : on a aussi changé le minibatchsize qui est par default 128 qui nous donne 200 itérations donc on a essayé de le changer, on a remarqué plus le batchsize est petit plus on a d'itération :



**Figure 3-16 a**: Precisionminibatchsize(64).

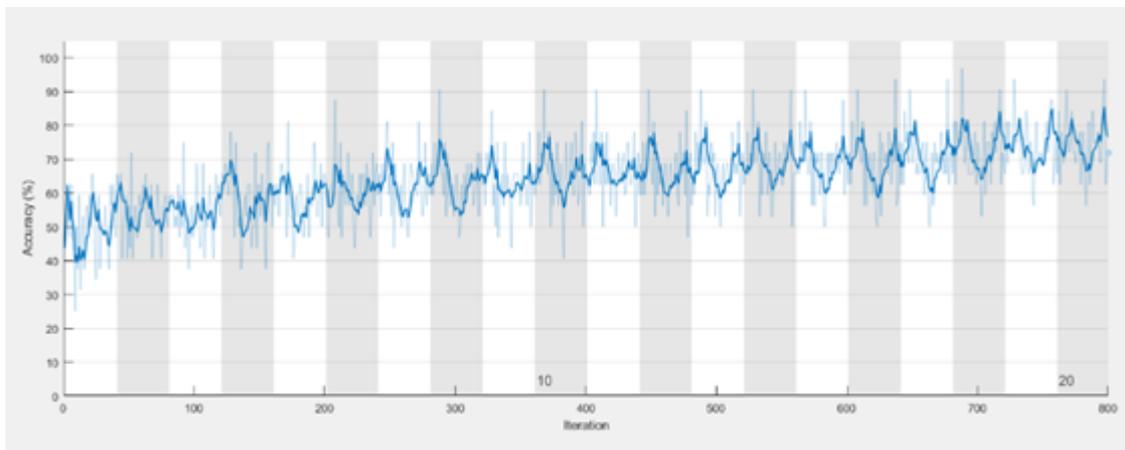


**Figure 3-18 b:** lossminibatchsize(64).

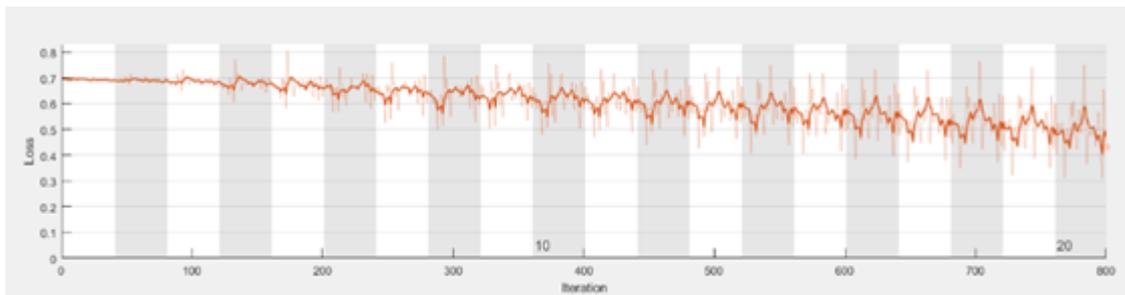
**Discussion:**

- Avec un minibatchsize de 64 on a obtenue 400 itérations.
- Le deuxième on a obtenue 800 itérations avec minibatchsize 32.
- Plus on réduit le nombre de minibatchsize plus on aura d'itérations.

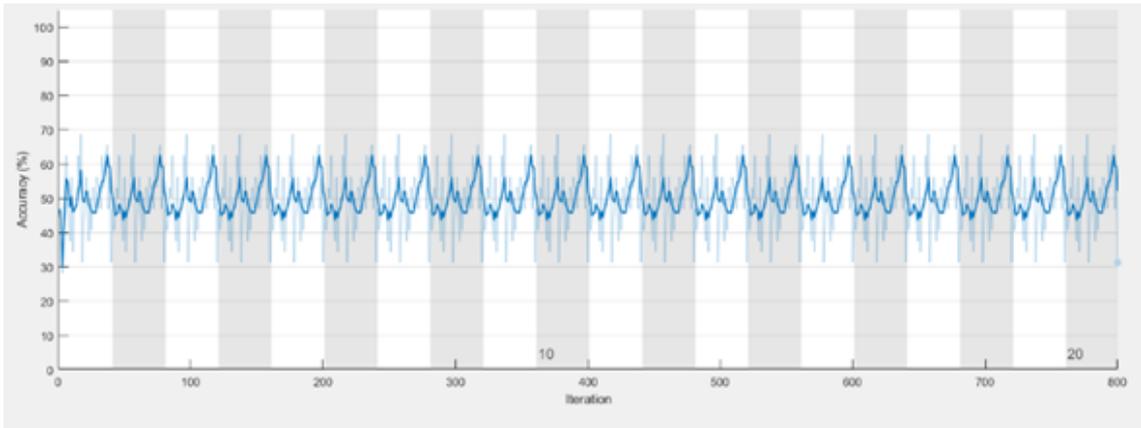
**d Résultats 4 :** dans un autre essai on a changé la fonction sigmoïde en fonction Adam pour essayer d'obtenir de meilleurs résultats voici les résultats obtenus sont montrés ci-dessous :



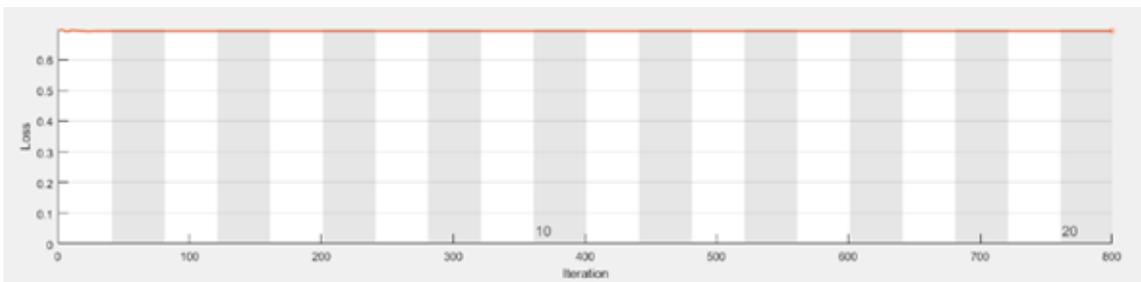
**Figure 3-19 a:** Précision filtre 5\*5 et nombre de filtre 20(Adam).



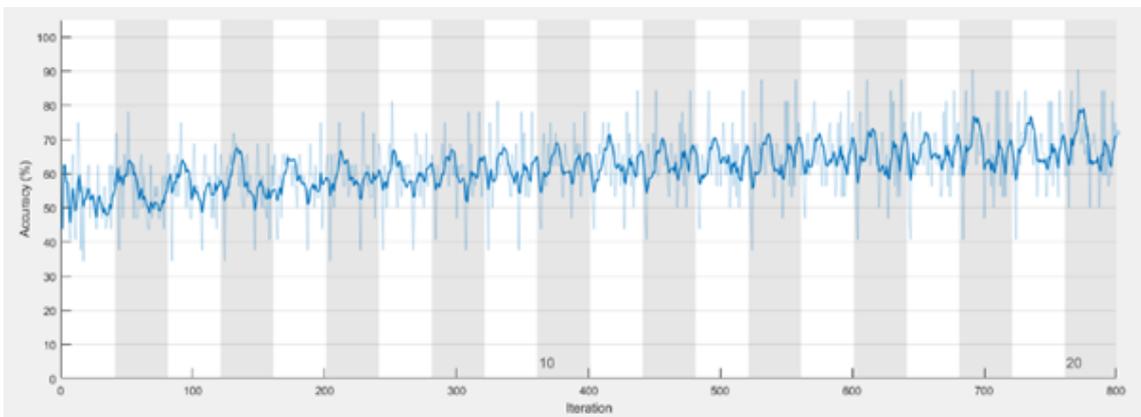
**Figure 3-19 b:** lossfiltre 5\*5 et nombre de filtre 20(Adam).



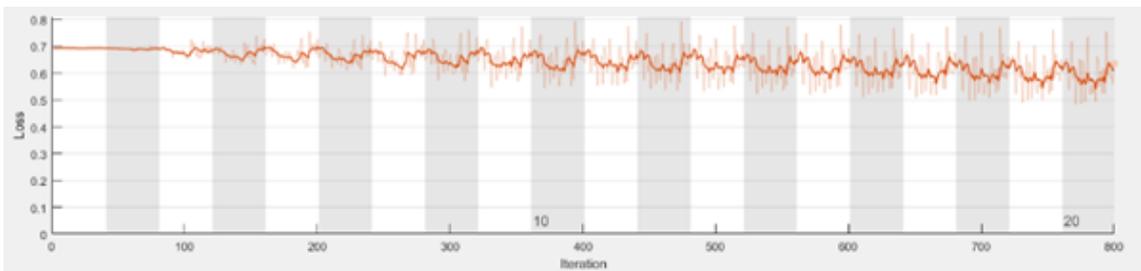
**Figure 3-20 a:** Precision filtre 5\*5 et nombre de filtre 30(Adam).



**Figure 3-20 b:** loss filtre 5\*5 et nombre de filtre 30(Adam).



**Figure 3-21 a:** filtre 3\*3 et nombre de filtre 20(Adam).



**Figure 3-21 b:** filtre 3\*3 et nombre de filtre 20(Adam).

**Discussion :**

- On a obtenu (figure 3-19) un taux de réussite d'apprentissage 80% et le taux de perte 0.501.
- Et pour la figure 3-20 un taux de précision de 53.13% et le taux de perte 0.6931.
- Et pour la figure 3-21 on a obtenu un taux de précision de 70.01% et le taux de perte 0.521.
- Nous remarquons qu'avec la fonction Adam nous avons obtenu un meilleur résultat qu'avec la fonction sigmoïde.

Le tableau ci-dessous résume les résultats obtenus précédemment :

|                                                                             | Filtre | Nombre de filtre | Temps d'apprentissage | Remarque                                                                                             |
|-----------------------------------------------------------------------------|--------|------------------|-----------------------|------------------------------------------------------------------------------------------------------|
| Les résultats suivants sont obtenus à partir d'une machine i5 et de RAM 4GB |        |                  |                       |                                                                                                      |
| Exemple 1<br>Figure 3-14                                                    | 3      | 20               | 227 min               | Nous avons constaté que le résultat n'était pas suffisant avec un taux de 57% et un taux de perte de |
| Exemple 2<br>Figure3-12                                                     | 5      | 20               | 621min                | Taux d'apprentissage 70% et un taux de perte de 0.545 .                                              |
| Exemple 3<br>Figure 3-15                                                    | 5      | 20               | 688min                | Mais nous avons aussi modifié l'architecture Taux d'apprentissage 64.8%, et taux de perte 0.5954     |
| Exemple 4<br>Figure 3-13                                                    | 5      | 30               | 715min                | Taux d'apprentissage 60.1% et de perte 0.624                                                         |
| Les résultats suivants sont obtenus à partir d'un Pc i7 et de RAM 16GB      |        |                  |                       |                                                                                                      |
| Exemple 5                                                                   | 5      | 20               | 20min                 | Taux de réussite de 62.50%                                                                           |

|                                                   |   |    |       |                                                         |
|---------------------------------------------------|---|----|-------|---------------------------------------------------------|
| Figure 3-17                                       |   |    |       | Taux de perte : 0.6472                                  |
| Exemple 6<br>Figure 3-18                          | 5 | 30 | 30min | Taux de réussite de 60.21%<br>Taux de perte : 0.6472    |
| On a changé la fonction sigmoïde en fonction Adam |   |    |       |                                                         |
| Exemple 7<br>Figure 3-19                          | 5 | 20 | 20min | Taux de réussite de 80%<br>Taux de perte : 0.501        |
| Exemple 8<br>Figure 3-20                          | 5 | 30 | 30min | Taux de réussite de 53.13%, et un taux de perte 0.6931  |
| Exemple 9<br>Figure 3-21                          | 3 | 20 | 25min | Taux de réussite de 71.88% et un taux de perte : 0.6352 |

**Tableau 3-1:** les différents essais d'apprentissage.

**Discussion :**

Ce tableau3-1 montre la différence entre tous les essais avec le temps et le taux de réussite et de perte.

Une autre remarque importante l'apprentissage avec les deux différents pc nous donne de différents résultats et aussi le temps d'apprentissage est largement réduit.

### 3.6 Validation :

#### Expérience :

On a divisé le dossier train en deux dossier un pour l'apprentissage et l'autre pour la validation le premier contient 450 pour malade et 450 pour non malade quand a validation il contient le reste des images.

#### Résultat :

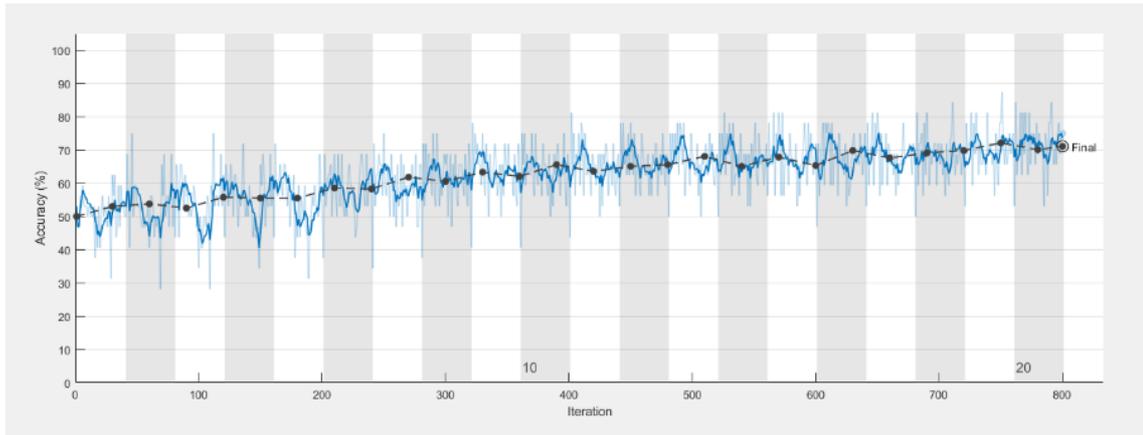


Figure 3-22 a: Précision validation avec filtrage (5 20).

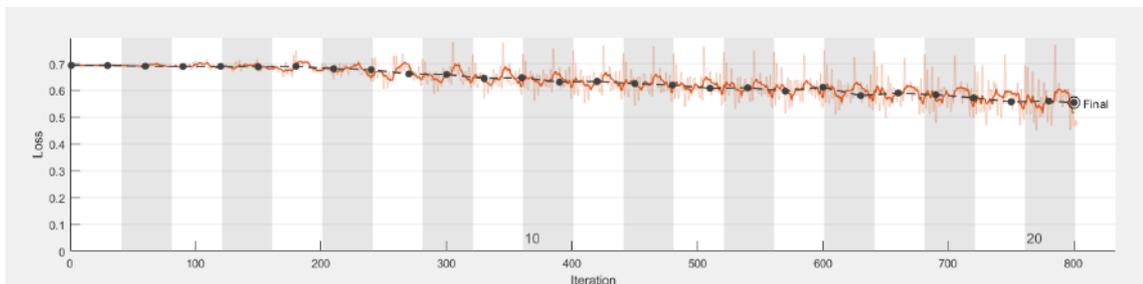


Figure 3-22 b: loss validation avec filtrage (5 20).

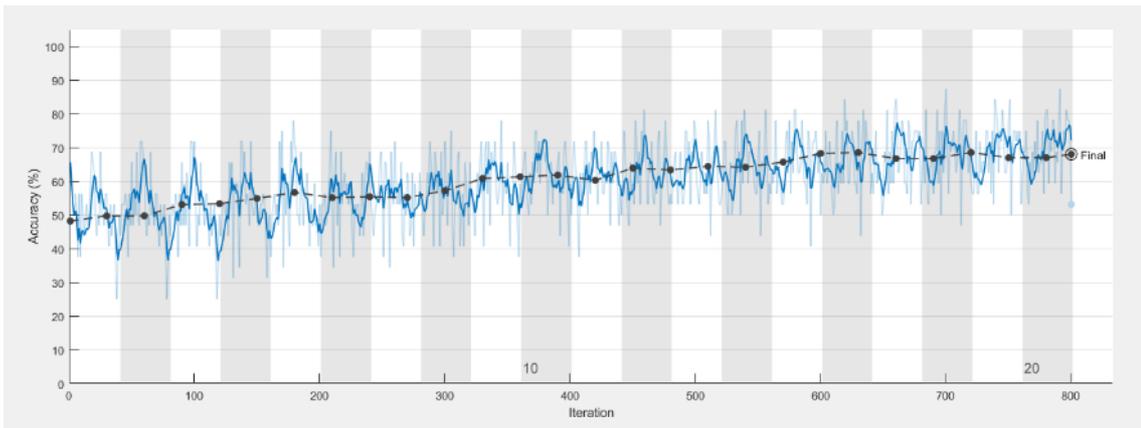
#### Discussion :

On remarque que le taux de précision de validation est de 71% et de perte 0.5542 Le taux d'apprentissage est de 75.00% et de perte 0.4773 .

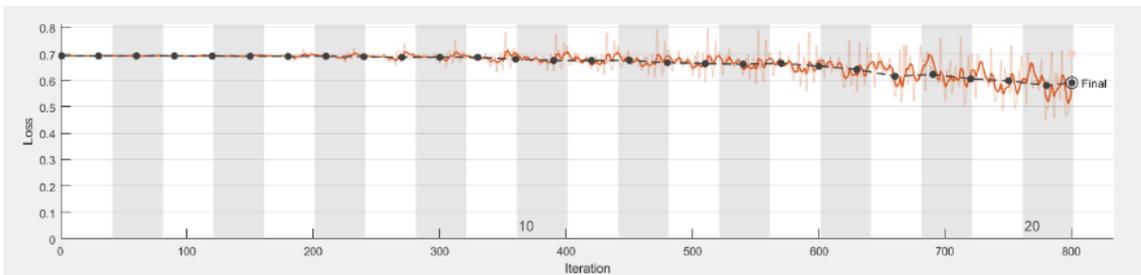
#### Expérience :

On a changé le numéro de filtre (30) et ajouté des couches de convolution (3) et on constate que la précision de validation et d'apprentissage est meilleur que le premier (figure 3-22).

**Résultat :**



**Figure 3-16 a:**Précision validation avec filtrage (5 30).



**Figure 3-21 b:**lossvalidation avec filtrage (5 30).

| Epoch | Iteration | Time Elapsed<br>(hh:mm:ss) | Mini-batch<br>Accuracy | Validation<br>Accuracy | Mini-batch<br>Loss | Validation<br>Loss | Base Learning<br>Rate |
|-------|-----------|----------------------------|------------------------|------------------------|--------------------|--------------------|-----------------------|
| 1     | 1         | 00 :00 :02                 | 56.25%                 | 49.74%                 | 0.6931             | 0.6932             | 1.0000e-04            |
| 10    | 400       | 00 :04 :41                 | 68.75%                 | 63.92%                 | 0.6663             | 0.6677             | 1.0000e-04            |
| 20    | 800       | 00 :09 :15                 | 96.88%                 | 77.58%                 | 0.3325             | 0.4748             | 1.0000e-04            |

**Tableau 3-2:** apprentissage et validation.

**Discussion :**

Le taux de validation de figure (3-23)est de 69.9% et de perte 0.601, et le taux d'apprentissage est de 96% et de perte 0.594.

Celui-ci est le meilleur résultat obtenu jusqu'à présent et celui utiliser pour le test.

### 3.7 Test :

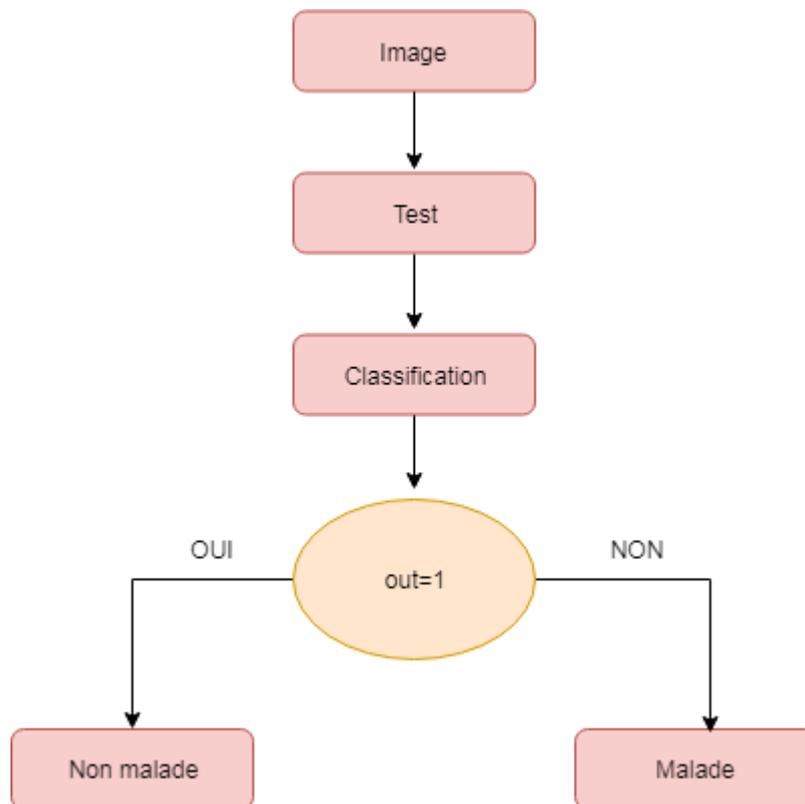


Figure3-24: organigramme du test.

### Résultat :

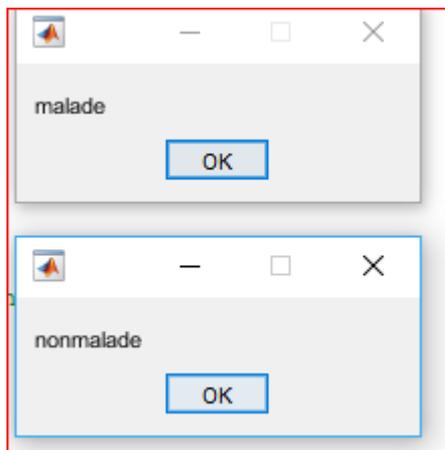


Figure 3-25 :affichage de classification .

Ce programme fait la classification des images test entre malade et non malade, si la rétine est atteinte un petit message affiche malade sinon il affiche non malade.

A l'aide d'un ophtalmologiste « Dr.KARAOUI » nous avons confirmé quelques images mais on a pas eu le temps de les exploiter convenablement pour pouvoir faire la détection.

On a essayé la détection avec R-CNN et fasterR-CNN mais ça n'a pas marché car les images qu'on a ne sont pas claires au point de déterminer exactement l'œdème ou l'hémorragie.

### **3.8 Conclusion**

Dans ce chapitre nous avons présenté les différents résultats obtenus des différents essais réalisés, malgré l'absence du GPU et grâce à l'optimiseur Adam nous avons obtenu une précision et un taux de perte acceptable pour faire notre classification, cet algorithme pourrait être développé puis être intégré plus tard dans un système embarqué qui permettrait d'aider les médecins dans leurs diagnostics.

## Conclusion générale

---

Les pathologies de la rétine sont diverses et importantes. La rétinopathie diabétique est l'une d'entre elles. Cette maladie est généralement causée par le diabète. Un contrôle régulier et un diagnostic précoce chez l'ophtalmologiste peut prévenir des dommages irréversibles.

Notre travail s'inscrit dans le cadre de traitement d'image médicale, l'objectif est de concevoir un algorithme capable de classifier et de détecter la rétinopathie diabétique et de l'implémenter plus tard dans un système embarqué.

Pour la réalisation on a utilisé le Deep Learning, l'une des révolutions de la vision artificielle qui a prouvé son efficacité dans presque tous les domaines. Comme architecture nous avons utilisé les réseaux de neurone CNN et cela pour leurs efficacités dans le traitement et la classification d'images.

Les hyper paramètres pour l'apprentissage sont difficiles à trouver, c'est pour cela que dans le but d'obtenir le meilleur résultat (de précision et de perte), nous avons effectué plusieurs essais avec des hyper paramètres et des modèles d'architecture (les couches) différentes jusqu'à atteindre notre but.

Au cours de l'exécution de l'algorithme nous avons rencontré plusieurs problèmes, en particulier avec le temps d'exécution qui était très long. Pour régler ce problème et avoir de meilleurs résultats utiliser un GPU au lieu d'un CPU serait un choix judicieux, il serait aussi préférable d'utiliser de larges bases de données.

A travers ce projet nous avons réalisé la partie classification et détection, en changeant la base de données et quelques paramètres. L'algorithme serait capable de détecter d'autres anomalies. La phase suivante serait d'implémenter ce réseau dans un système embarqué, une fois réalisé, ce projet aura un impact p dans le domaine médical.

## Bibliographie

---

- [1] :Erika Sutter, Allen Foster et Victoria Francis : 'HANYANE-BIEN VOIR ET MIEUX VIVRE AU VILLAGE',
- [2] : site internet : <https://www.guide-vue.fr/glossaire/retine>, DDA : le 04 Mars 2019.
- [3] :Claude Garceau : ' Guide pour les patients atteints de diabète ', l'Institut Universitaire de Cardiologie et de Pneumologie de Québec, plusieurs spécialistes québécois du diabète ont participé à la publication du *Guide pour les patients atteints de diabète : Comment devenir expert dans la prise en charge de sa maladie*. Le guide est publié aux Presses de l'Université Laval.
- [4] : site internet : <https://www.futura-sciences.com/sante/dossiers/medecine-oeil-vision-dela-vision-667/page/5/> ,DDA : le 09 Mars 2019.
- [5] : lee R,Wong Tien Y Sabanayagam: 'C.Epidemiology of diabetic retinopathy ,diabetic macular edema and related vision loss', *Eye Vis (lond)* 2015/2/17.
- [6] : site internet : <http://www.chu-poitiers.fr/specialites/ophtalmologie/maladie-de-loeil/> , DDA : le 01 Avril 2019.
- [7] : ' Rétinopathie diabétique (RD)', Collège des Ophtalmologistes, Universitaires de France (COUF) ,2013.
- [8] :Sujith Kumar: ' Automatic Detection of Diabetic Retinopathy in Non-dilated RGB Retinal Fundus Images', *International Journal of Computer Applications (0975 – 888) Volume 47– No.19, June 2012* , Department of ECE R.N.S Institute of Technology, Channasandra, Bangalore-61, India.
- [9] : Lynda BEKRI HADJ HAFSI : 'Rétinopathie diabétique (RD) Collège des Ophtalmologistes ',Universitaires de France (COUF) 2013 .
- [10] :Alain Droniou : ' Apprentissage de représentations et robotique développementale', Université Pierre et Marie Curie- Paris VI, 2015. Français, 30 April 2015.

- [11] : YOUS Mohamed ,BENLKAID Ahmed : 'reconnaissance des caractéristiques manuscrites numériques par la transformée en ondelette et le réseaux de neurone multicouche', université Blida ,institut d'électronique ,novembre 2000.
- [12] : Florent Simon : 'Deep learning ,les fonctions d'activation', publie le 04/10/2018 à 00 :3912.
- [13] :BoughabaMohammed ,Boukhris Brahim : 'L'apprentissage profond (Deep Learning) pour la classification et la recherche d'images par le contenu',UNIVERSITE KASDI MERBAH OUARGLA,2016/2017.
- [14]: IA Giang Nguyen , Stefan Dlugolinsky , Martin Bobák , Viet Tran,ÁlvaroLópezGarcía,IgnacioHeredia,PeterMalík,LadislavHluchý : ' Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey', Published online: 19 January 2019.
- [15] : Jean Claude Heudin : 'comprendre le Deep Learning : 'une introduction aux réseaux de neurones ',2 novembre 2016.
- [16] :Pascal SCALART : Cours « Modélisation Paramétrique, Filtrage Optimal et Adaptatif »
- [17] : site internet : <https://prevn.fr/blog/machine-learning/>, DDA : le 14 Juin 2019.
- [18] :livre 'Introducing Deep Learning with MATLAB' ,2018 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.
- [19] : site internet : [https://fr.wikipedia.org/wiki/Réseau\\_neuronal\\_convolutif](https://fr.wikipedia.org/wiki/Réseau_neuronal_convolutif) , DDA : le 30 Juin 2019.
- [20] :SumitSaha : ' deep learning /machine learning / data science / computer vision ', le 15 December 2018 .
- [21]:SACHIT KAUL : ' REGION BASED CONVOLUTIONAL NEURAL NETWORKS FOR OBJECT DETECTION AND RECOGNITION IN ADAS APPLICATION' ,The University of Texas at Arlington in Partial Fulfillment , December 2017.
- [22]: site internet: [https://www.d2l.ai/chapter\\_computer-vision/rcnn.html](https://www.d2l.ai/chapter_computer-vision/rcnn.html) , DDA: le 05 Juillet 2019 .
- [23]: Dhruv Parthasarathy : 'Brief History of CNNs Image Segmentation' :from R-CNN to Mask R-CNN',Athelas ,le 22 Avril 2017