

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Mouloud Mammeri de Tizi-Ouzou

Faculté de : Génie électrique et d'informatique  
Département : Informatique

# Mémoire de fin d'études

En vue de l'obtention du diplôme de  
Master en Informatique

Spécialité : **Systeme Informatique**

Présenté par :  
**Senani SAMY**

Thème : **Réseaux de neurones convolutionnels  
pour la détection précoce de la rétinopathie  
diabétique**

Soutenu le 10 juillet 2019 devant le jury composé de:

---

Mme. Aoudjit Rachida

Président du Jury

Mme. Chérifi Chafia

Membre du Jury

Mr. Ramdani Mohamed

Membre du Jury

Mlle. Yesli Yasmine

Directeur de mémoire

# Table des matières

INTRODUCTION GENERALE	5
<b>Chapitre I :</b>	<b>8</b>
Deep Learning et Reconnaissance de l'imagerie médicale	8
1. Introduction:	9
2. Machine Learning:	9
2.2. Méthodes d'apprentissage:	10
2.2.1. Apprentissage supervisé:	10
2.2.1.1. Classification:	11
2.2.1.2. Régression:	11
2.2.2. Unsupervised Learning:	11
2.2.1.3. Clustering:	11
2.2.2. Différence entre apprentissage supervisé et non supervisé:	12
2.2.3. Apprentissage semi-supervisé :	13
2.2.4. Apprentissage par Renforcement:	14
3. Deep Learning:	14
3.1. Machine Learning vs Deep Learning:	15
3.2. Pourquoi Deep learning:	16
4. Réseau Neurones ou ANN:	17
4.1. Neurone Biologique:	17
4.3. Neurone Formel (Artificiel):	18
4.4. Fonction d'activation:	19
4.5. Topologies des réseaux de neurones:	20
4.5.1. Famille des réseaux à propagation avant:	20
4.5.2. Famille des réseaux de neurones récurrents:	20
4.5.3. Famille des réseaux à résonance:	21
4.6. Les couches d'un Réseau de neurone:	22
4.7. Fonctionnement réseaux de neurone simple type feed forward :	22
4.8. Traitement des données:	23

4.9. Découpage des données:	<b>24</b>
4.10. Fonction perte:	<b>24</b>
4.11. Rétro-propagation :	<b>25</b>
4.12. Gradient Descent :	<b>25</b>
4.13. Algorithme d'optimisation:	<b>27</b>
4.13.1. Algorithmes d'optimisation de premier ordre:	<b>27</b>
4.13.2. Algorithmes d'optimisation du second ordre :	<b>28</b>
5. Réseaux de Neurones Convolutifs:	<b>28</b>
5.1. Historique:	<b>28</b>
5.2. Architecture d'un ConvNet:	<b>29</b>
5.2.1. Couche Convulsive:	<b>30</b>
5.2.2. Couche de Regroupement:	<b>31</b>
5.2.3. Couche Entièrement Connectée:	<b>32</b>
5.2.4. Couche Sortie:	<b>33</b>
5.2.5. Couche de non-linéarité:	<b>33</b>
5.3. Calcule de la perte:	<b>34</b>
5.4. ConvNet vs ANN:	<b>35</b>
6. Impact de L'IA sur l'imagerie médicale:	<b>35</b>
7. Évaluation d'un modèle:	<b>36</b>
7.1. Matrice de Confusion:	<b>36</b>
7.2. Critères d'évaluation:	<b>37</b>
7.2.1. Sensibilité :	<b>38</b>
7.2.5. Spécificité:	<b>38</b>
7.2.6. Sensibilité vs Spécificité:	<b>39</b>
7.2.6.1. La courbe ROC:	<b>39</b>
7.2.6.3. Relation entre sensibilité, spécificité, taux de faux positif et seuil:	<b>41</b>
7.2.6.4. Représentation de la Courbe ROC :	<b>41</b>
8. Conclusion:	<b>42</b>
Chapitre II :	<b>43</b>
Organe Concerné et Pathologie	<b>43</b>

2. Anatomie de l'œil:	44
2.1. Tunique Externe de l'œil:	44
2.2. Tunique Moyenne de l'œil :	45
2.3. Tunique Interne de l'œil:	46
2.3.1. Zones particulières de la Rétine:	46
2.3.1.1. Macula:	46
2.3.1.3. la pupille:	46
3. Diabète:	47
3.1. Définition :	47
3.2. Types de Diabètes:	48
3.2.1. Type 1:	48
3.2.2. Type 2:	48
3.3. Complications liées au Diabète:	49
4.1. Rétinopathie :	49
4.2. Causes:	50
6. État de l'art	55
6.1 Travaux de Tomi Kauppi et al , 2007 [10]	55
6.2 Travaux de Abràmoff MD, 2010 [14]	55
6.3 Travaux de Maria Valeria Jimenez-Baez, 2015 [15]	56
6.4 Travaux de Kele Xu, 2017 [12]	56
6.5 Travaux de Rishab Gargeyal, 2017 [13]	57
6.6 Travaux de Carson Lam, 2018 [11]	57
7. Synthèse:	58
8. Conclusion:	59
Chapitre III :	60
Conception et Réalisation	60
3.2.1.4. Modifications apporté aux modèles :	68
3.2.2. Dataset:	69
3.2.4. Prétraitement:	70
3.2.6. Couche Dropout:	73

3.3. Réalisation et Implémentation:	73
Chapitre IV :	79
Test et Évaluation des Résultats	79
3. Comparaison des résultats:	99

## **INTRODUCTION GENERALE**

- **Contexte**

Selon le Centre américain de contrôle et de prévention des maladies, on estime à 29,1 millions le nombre de personnes atteintes de diabète aux États-Unis, et l'Organisation mondiale de la Santé estime à plus de 347 millions de personnes sont atteintes par la maladie dans le monde.

La Rétinopathie diabétique (RD) est une complication liée à un diabète de longue durée non traité. Les personnes atteintes de cette pathologie se voient généralement subir des pertes visuelles graves à cause du dépistage tardif. Ces déficiences visuelles peuvent plus ou moins grave selon le stade de la pathologie, conduisant dans le pire des cas à une cécité chez les malades.

- **Problématique**

La progression vers une déficience visuelle peut être ralentie ou évitée, si une RD est détectée à temps. Toutefois, cela peut être difficile car la maladie présente souvent peu de symptômes jusqu'à ce qu'il soit trop tard pour fournir un traitement efficace.

Le processus de détection de la maladie est manuel et fastidieux, cela nécessite un ophtalmologiste qualifié pour évaluer la photographie numérique de fond scopie, mais

souvent lorsque la maladie est diagnostiquée il est souvent trop tard, le patient a subi des pertes irréversibles.

On a estimé le nombre de personnes adultes atteintes de cécité dû à une rétinopathie tardivement diagnostiquée à plus de 93 millions d'où l'importance d'une méthode complète et automatisée, reconnue par la communauté scientifique, pour aider à détecter une cette pathologie de façon précoce.

- **Direction des recherches actuelles**

La reconnaissance d'images médicales dans le but de détecter la présence d'anomalies ou de diagnostiquer des maladies a beaucoup progressé ces dernières années, particulièrement via le Deep Learning ou apprentissage profond est un sous domaine de l'intelligence artificielle qui repose principalement dans son approche de résolution sur l'utilisation des réseaux de neurones.

Une approche reposant sur les réseaux de neurones avec une architecture ConvNet sera proposé afin de Dans le cas du traitement de l'imagerie on va s'intéressé à un type de réseau en particulier les réseaux de neurones à architecture ConvNet qui actuellement donne des résultats fabuleux dans le domaine de l'imagerie depuis son introduction.

C'est une approche qui est approuvé de la communauté scientifique, un ensemble d'étude dans le domaine médicale en l'utilisant comme méthode automatique de détection.

- **Solution proposée**

Nous proposons dans cette optique la réalisation d'un outil permettant de détecter la présence de Rétinopathie diabétique à un stade précoce sur des images médicales type fond scopes (fond d'œil), en utilisant 3 différents algorithmes d'apprentissage supervisé à basé sur les réseaux de neurones à architecture type ConvNet, fournis par l'organisme de compétition imagenet qui regroupe chaque années des équipes qui s'affronte dans le but de faire avancer l'imagerie dans le problème de détection d'objet.

Nous apporterons des améliorations à ces 3 algorithmes qui se caractériserons à des ajouts au niveau de l'architecture, l'entraînement sur un ensemble de données de cliché médicale type fond scope, enfin on utilisera la sensibilité et spécificité comme métriques, ainsi que la courbe roc pour visualisé la performance des algorithmes.

- **Structure du mémoire**

Afin de bien mener ce projet, nous avons structuré le présent mémoire de la manière suivante:

**Chapitre 1: “Deep Learning et Reconnaissance de l’imagerie Médicale”**, définit essentiellement le Deep Learning, les réseaux convolutionnels, ainsi que l’impact de l’IA dans l’imagerie médicale.

**Chapitre 2: ”Organe Concerné et Pathologie”**, décrit la pathologie, ainsi que l’organe touché par cette dernière, incluant les études effectuées à but de diagnostiquer la pathologie.

**Chapitre3: “Conception et Réalisation”**, présente la solution apportée à la problématique posée.

**Chapitre 4: “Tests et Évaluation des Résultats”**, décrit les résultats des tests d’évaluation effectués par rapports aux critères d’évaluation fixés.

## **Chapitre I :**

### **Deep Learning et Reconnaissance de l'imagerie médicale**

## **1. Introduction:**

Dans ce chapitre nous allons parler de l'émergence du Machine Learning et du Deep Learning, les raisons de leur apparition ainsi que leur impact, puis on verra un exemple de réseau de neurones tout en illustrant son fonctionnement interne avant de s'attaquer aux réseaux de neurones convolutionnels dans le traitement de l'imagerie médicale.

## **2. Machine Learning:**

### **2.1. Définition et Histoire:**

L'apprentissage automatique ou Machine Learning (**ML**) est une sous-section du domaine de l'**intelligence artificielle** ou **IA** en informatique, qui vise à apprendre aux machines à effectuer une tâche sans être explicitement programmé, et cela en utilisant un des algorithmes qu'on appellera modèles et de données.

Le concept d'intelligence artificielle est apparu dans les années 50 dans une assemblée rassemblant toute une flopée de savants célèbres en informatique et en mathématique dont **Alan Turing**. Ce savant de génie a aussi prédit le développement du Machine Learning tel qu'on le connaît.

Le ML a refait surface entre les années 70 à 80, l'idée derrière le concept était de créer des algorithmes ayant la capacité d'accumuler de l'expérience, et de la connaissance à partir de données sans être explicitement programmé pour effectuer cette tâche, et c'est vers la fin des années 80 qu'on a le retour des réseaux de neurones (créé plutôt 1943 par **Walter Pitts** et

**Warren McCulloch** inventeurs du premier **Perceptron** qu'on appellera par la suite **Neurone artificiel**. L'idée était de reproduire un **neurone biologique** d'un cerveau humain de façon artificielle en utilisant d'opérations mathématiques, malheureusement cette approche fut très limitée dans la résolution des problèmes. [1][2]

Vers la fin des années 80, on a vu le retour des réseaux de neurones, avec une réinvention de l'algorithme de **Rétropropagation (Backpropagation)**, mais sans succès car le domaine sera finalement laissé à l'abandon faute de **capacité de calcul** et de **stockage** suffisantes.

Il fallait attendre le milieu des années 2000 pour le grand retour des réseaux de neurones avec le **Deep Learning (3)**.

## 2.2. Méthodes d'apprentissage:

Il y a de nombreux sous domaines ou sous sections. On dénote 3 principaux, ses dernières dépendent du type de problèmes que l'on voudrait traiter, on retrouve:

- **Supervised Learning**
- **Unsupervised Learning**
- **Reinforcement Learning.**
- **Semi-supervised Learning**

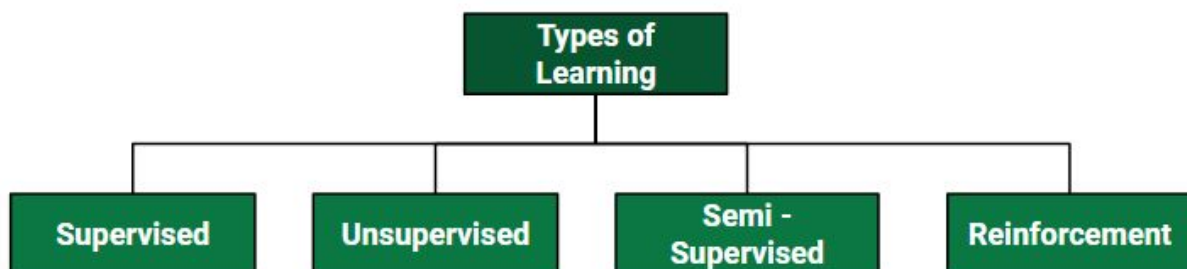


Figure II.1 : Type d'apprentissage en ML

### 2.2.1. Apprentissage supervisé:

Apprentissage supervisé ou Supervised learning prend des données qui sont étiquetées (possédant des étiquettes). Ces étiquettes nous renseignent sur le résultat souhaité, et créent un modèle à l'aide de ses données, afin d'effectuer des prédictions qui pourront généraliser sur de nouvelles données.

Les problèmes dans l'apprentissage supervisé sont 2 types : **Classification**, **Régression**.

### **2.2.1.1. Classification:**

C'est une tâche d'apprentissage supervisé dont la sortie a des étiquettes définies (valeur discrète) ce qu'on appelle classes ou catégories.

On peut avoir des classifications **Binaire** ou alors **Multiclasse**.

La **classification binaire** prédit pour un modèle (0 ou 1) vrai ou faux, dans le cas de **multiclasses** le modèle va prédire plus d'une classe.

Le but ici reste le même quel que soit le nombre de classes, **prédire** si les **valeurs discrètes** appartiennent à une classe particulière.

#### **Exemple:**

Gmail qui est en mesure de prédire si un mail donné est un spam ou mail normal.

### **2.2.1.2. Régression:**

Il s'agit d'une tâche d'apprentissage supervisé dont la sortie a une **valeur continue**.

Le but ici est de prédire une valeur aussi proche de la valeur de sortie réelle que notre modèle le permet, puis l'évaluation est effectuée en calculant la valeur d'erreur. Plus l'erreur est petite, plus la précision de notre modèle de régression est grande.

### **2.2.2. Unsupervised Learning:**

L'apprentissage non supervisé ou unsupervised est un type d'algorithme d'apprentissage automatique utilisé pour tirer des inférences à partir d'ensembles de données constitués de données d'entrée sans réponses étiquetées.

Ce type d'apprentissage est principalement utilisé quand on souhaite étudier un ensemble de données non labellisés.

### 2.2.1.3. Clustering:

Le clustering permet de séparer les données entrées en un ensemble ou groupe de données qui ont des traits similaires et de les affecter à un cluster. Contrairement à la classification dans l'apprentissage supervisé ces différents clusters ou groupe ne sont pas connus à l'avance, c'est l'algorithme lui-même qui va séparer les données aux nombres de clusters qu'il faut.

Comme l'on peut le voir pouvez le voir dans l'exemple **Figure II.3**, les points de jeu de données donnés ont été divisés en groupes identifiables par les couleurs rouge, vert et bleu

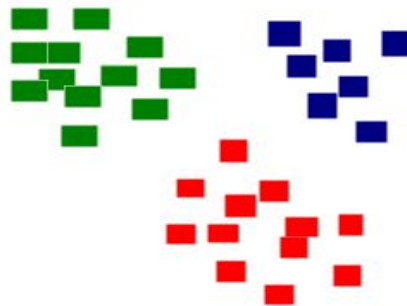


Figure II.3 : Clustering

### 2.2.2. Différence entre apprentissage supervisé et non supervisé:

La majeure différence qu'on peut trouver entre ces 2 types d'apprentissages est la disposition des données d'entrée, mais il existe tout de même d'autres différences qu'on peut trouver dans le tableau ci-dessous.[4]

Ce tableau résume la différence entre apprentissage supervisé et non supervisé:

	Supervised Learning	Unsupervised Learning
Les données d'entrée	Utilise des données connues et étiquetées en tant qu'entrée	Utilise des données inconnues en tant qu'entrée Non étiquetées
Nombre de Classes	Connues	Inconnues

<b>Précision du résultat</b>	Résultats précis et fiables	précision et fiabilité <b>Modéré</b>
------------------------------	-----------------------------	--------------------------------------

**Tableau II.1: Supervising vs unsupervised**

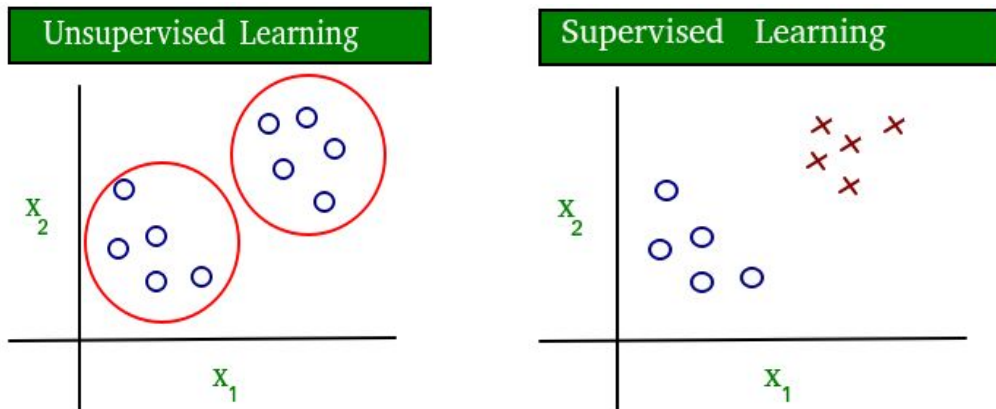


Figure II.2 : **Supervised vs Unsupervised Learning**

### 2.2.3. Apprentissage semi-supervisé :

Nous avons préalablement vue l'apprentissage supervisé et non supervisé, dont la majeure différence réside dans le fait que les données soient étiquetées ou non, et à cela s'ajoute les méthodes adéquates utilisées pour traiter ses données.

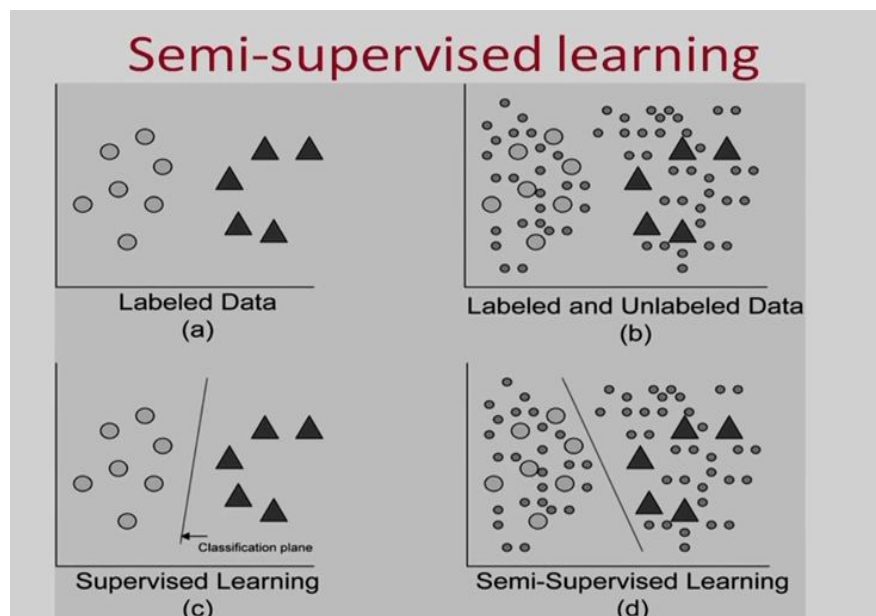


Figure II.4 : **Semi-supervised Learning**

L'apprentissage semi-supervisé regroupe ses deux principes, il prend un ensemble réduit de données étiquetées avec un autre ensemble de données non étiquetées du même types.

L'avantage de ce type d'apprentissage réside principalement dans le processus d'étiquetage des données prend beaucoup de temps et souvent coûteux.

Donc paradoxalement le non étiquetage devient bénéfique pour le processus d'apprentissage, et la construction du modèle et moins coûteuse. [5]

#### 2.2.4. Apprentissage par Renforcement:

Entraînement par renforcement est une technique dans laquelle on immerge un agent dans un environnement où celui-ci interagit avec son environnement dans le but d'apprendre.

Un agent dans un environnement est muni de **sensors** pour capter les informations de son environnement, ainsi que d'**actionneurs** qui lui permettront d'agir dans son environnement.

Les agents observent l'entrée, puis il effectue une action en prenant des décisions. Une fois l'action réalisées, l'agent reçoit des récompenses en conséquence, ce qui renforce le modèle en stockant ses informations dans une base de données. [6][7]

La récompense peut être positive, négative ou nulle selon les actions effectuées.

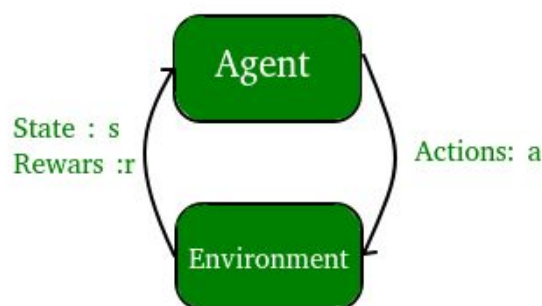


Figure II.4: Apprentissage par renforcement concept

**example:** AlphaGo, robot domestique, ou encors IA dans les jeux vidéos

### 3. Deep Learning:

Deep Learning ou apprentissage en profondeur ou **DL** est une branche du Machine Learning entièrement basée sur des **réseaux de neurones artificiels**. [19]

Le concept d'apprentissage en profondeur existe depuis plusieurs années, mais il a été laissé à l'abandon faute de moyens nécessaires.

Dans le milieu des années 2000 le Machine Learning fait rage dans les compétitions de reconnaissance visuelle, en 2012 **deep mind** une startup dans le domaine de l'IA arrive dans la compétition avec un algorithme de **deep learning** qui bat largement tous les autres compétiteurs, l'année suivante tous les compétiteurs se sont tournés vers le deep learning au vu des résultats obtenus.

L'avancée du DL est dû à l'augmentation en exponentiel qu'ont connu les machines en **capacités de calculs**, et de **stockages**, ainsi que la disponibilité de **données de masses (big data)**, ses 3 ingrédients étaient nécessaires pour exploiter le potentiel du DL qui fût chose impossible dans les années 90.

Les pionniers qui ont soutenus le DL tel que **Geoffrey Hinton**, ou alors **Yoshua Bengio** qui a développé les **réseaux GAN**( generative adversal networks), **Yann leCun** qui est au coeur d'une avancée fulgurante dans le domaine de reconnaissance d'images avec les réseaux Convolutionnels **CNN** voir section **(II.5)**, et son architecture **LetNet**.

**Geoffrey Hinton** a prouvé que l'apprentissage profond pouvait résoudre des problèmes insolubles par d'autres approches. [8]

### **3.1. Machine Learning vs Deep Learning:**

La majeure différence qu'on note entre ses 2 concepts provient de la manière dont les données sont présentées au système (modèle).

- Les algorithmes de ML nécessitent presque toujours des données structurées, alors que les réseaux d'apprentissage approfondis reposent sur des couches de réseaux de neurones artificiels (RNA).
- On voit aussi une différence au sein de l'architecture des modèles qui les composent, on note que les modèles type DL sont plus profond que les modèles type ML.

- Deep learning n'utilise que les réseaux de neurones, alors que pour le ML les réseaux de neurones sont qu'une approche de conception des modèles parmi tant d'autres.

En considérant le fait que le DL est la prochaine étape de l'évolution du ML inculquant aux machines la manière de prendre leurs décisions de façon précise sans l'intervention de l'expert humain.

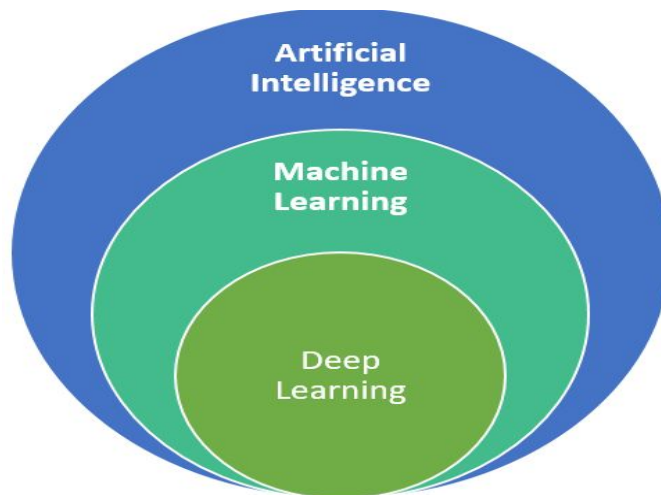


Figure II.5: IA vs ML vs DL

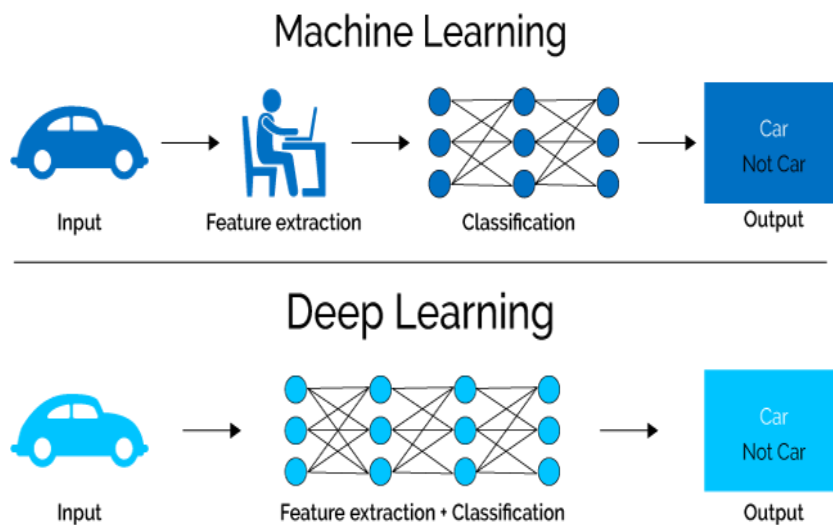


Figure II.6 : ML vs DL

### 3.2. Pourquoi Deep learning:

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

- **L'omniprésence des données:** Nous sommes dans l'ère de l'informatisation (Internet Of Things), et le propre du Deep-Learning est de tirer parti d'une grande quantité de données pour en estimer une représentation abstraite et en tirer parti.
- La **puissance de calcul:** La théorie des réseaux de neurones existe depuis quelques décennies, mais c'est grâce à la puissance de calcul accessible aujourd'hui qui se démocratise, notamment depuis que les GPUs sont devenus la plateforme de choix pour le Deep-Learning.
- Des **besoins croissants dans le domaine de l'IA** : vision par ordinateur, reconnaissance vocale, traitement du langage, ...etc.
- Un **effet de mode.** On a tendance à vouloir appliquer le Deep-Learning partout alors que ça reste un moyen et non une fin. Certains problèmes sont tout à fait solubles par d'autres méthodes d'apprentissage statistique. Cela dit, si beaucoup de gens sont prêts à investir dans le Deep-Learning, il est normal qu'il devienne si populaire.
- **Capacité de Stockage:** qui sont devenus beaucoup plus accessibles à prix raisonnable.
- Apparition de plateformes et communautés fortes encourageant l'évolution de ce domaine, ainsi que sa démocratisation.

## 4. Réseau Neurones ou ANN:

### 4.1. Neurone Biologique:

Un **Neurone** ou une **Cellule nerveuse** est l'unité fonctionnelle de base qui constitue un système nerveux, il assure la transmission d'un signal **bioélectrique** appelé **influx nerveux**.

Le neurone biologique reçoit en entrées des **signaux** transmis par d'autres neurones, grâce à une interaction qui est faite entre les **dendrites**, et les **synapses** Figure (II.7).

Le signal reçu est analysé et traité en effectuant une sommation des signaux en entrée, le résultat est par la suite comparé à un **seuil d'activation**: s'il est supérieur, une décharge est alors envoyée le long de son axone vers d'autres neurones.

## 4.2. Perceptron:

En 1943 **Walter Pitts** et **Warren McCulloch** [10] ont inventé le tout premier neurone artificiel qu'ils ont nommé **Perceptron** à l'époque. Ces deux chercheurs en neuroscience et science cognitive se sont inspirés du fonctionnement du cortex visuel des mammifères [11].

Le perceptron était en mesure de résoudre certains problèmes, mais il était limité.

C'est alors qu'ils eurent l'idée de mettre plusieurs perceptrons ce qui donne un **multi perceptron**. Communément appelé **réseau de neurones artificiels**.

## 4.3. Neurone Formel (Artificiel):

On retrouve **Poids Synaptiques**, **fonction d'activation** ainsi que **éléments de sortie** qui sont similaires aux fonctions occupées par (Synapses, corps cellulaire et Axone) chez le neurone biologique.

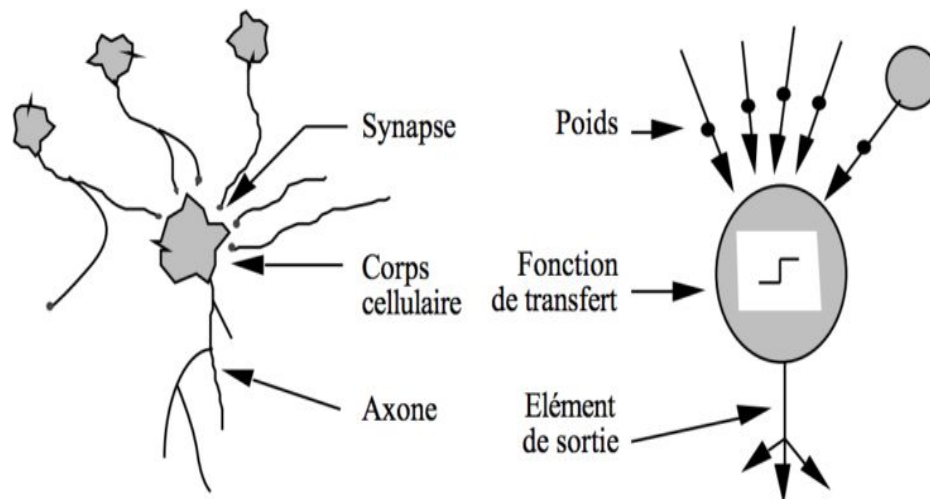


Figure II.7 : Neurone biologique / Neurone artificiel

Le neurone artificiel reçoit lui aussi des stimuli (des **entrées  $X_i$  Vecteur d'entrées**) via ses **poids synaptiques** qu'on appellera entrée la notation la plus répondu est  **$W_{ij}$  (Matrices de poids)** Figure II.8.

Chaque stimuli subit une opération mathématique  $X_i * W_{ij}$ , il est par la suite analysé et traité par le neurone, qui le résultat vers une fonction d'activation s'il le résultat est supérieur, un 1 est renvoyé.

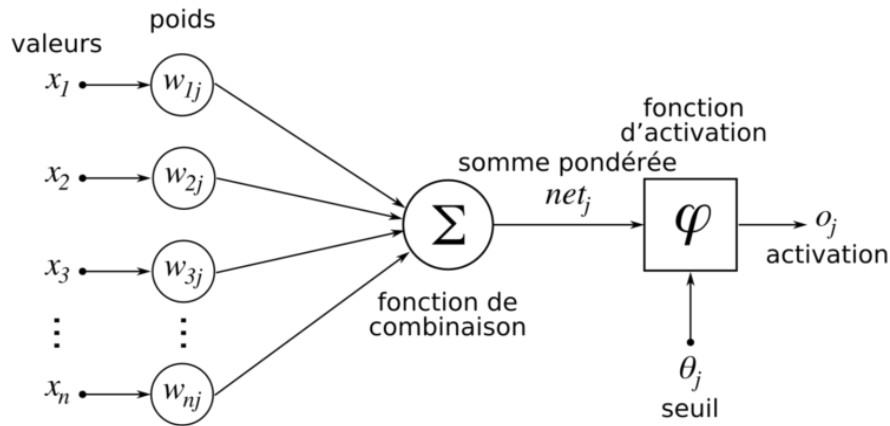


Figure II.8 : Structure d'un neurone artificiel

#### 4.4. Fonction d'activation:

Il existe plusieurs fonction d'activation, le choix de cette dernière dépend de la fonction du modèle qu'on souhaite modéliser.

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parametric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$

Figure II.9 : Fonctions d'activation

La fonction d'activation à utiliser sont généralement non-linéaire comme fonction **relu** ou **Tangente Hyperbolique**, le but de ses fonctions c'est de faire varier le résultat entre un interval précis. **ex:** pour Sigmoid le résultat varie entre 0 et 1. [12]

#### 4.5. Topologies des réseaux de neurones:

Un réseau de neurones n'est rien d'autre que l'agencement de plusieurs neurones artificiels ensemble, c'est alors qu'on peut parler d'architecture ou famille d'architectures.

Cette dernière peut varier selon le type de problème qu'on souhaite traiter.

##### 4.5.1. Famille des réseaux à propagation avant:

Communément appelé Feed-Forward c'est lorsque le réseau propage l'information reçut de la **couche entrées** vers la **couche sortie** sans retour en arrière.

- **Perceptron Simple:** C'est un réseau mono-couche qui ne comporte pas de boucle, et dont la dynamique est déclenché par la réception d'une entrée, il est dit simple car il se compose d'une couche entrée, et d'une autre sortie, l'ensemble des noeux des 2 couches est connectés les uns autres.
- **Perceptron multi couche:** C'est une réseau qui se constitue d'une couche entré, une couche sortie et d'une ou plusieurs couches cachées. il est doté d'un approximateur universel, il suffit d'ajouter un nombre de neurones suffisant au niveau de la couche caché afin d'approximer n'importe quel résultat non-linéaire.
- **Deep learning pour les réseaux profond:** Ce type peut être considéré comme un MLP avec plusieurs couche cachées qu'on retrouve dans des domaines tel que :traduction, reconnaissance du langage, traitement d'image.

##### 4.5.2. Famille des réseaux de neurones récurrents:

**Recurrent neural network** ou **RNN** ces réseaux sont dit récurrent car il comporte des cycle dans leurs graphes de connectivités, ce qui change la dynamique du réseau l'amenant à s'auto-entretenir.

- **Modèle Hopfield:** C'est un réseau qui se compose d'une seule est unique couche représentant à la fois les entrées ainsi que les sortie du réseau, où toutes les unités le constituant sont interconnectées entre elles.

- **Réseau de neurones récurrent à couche:** C'est une catégorie particulière des réseaux récurrents, où l'entrée globale se propage vers l'avant du réseau de couche en couche, mais contient aussi une information sur le passé.

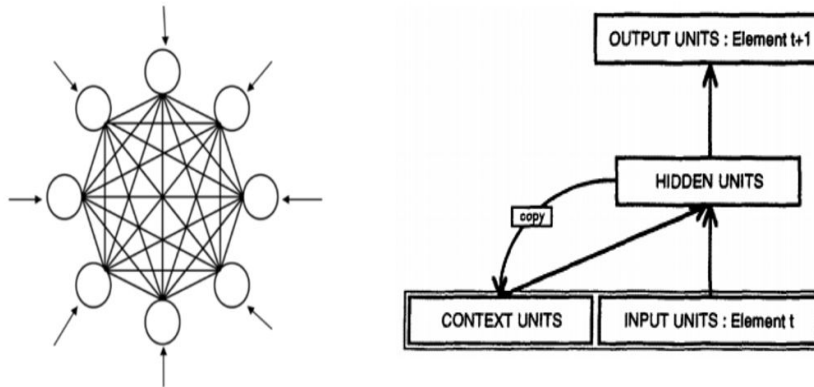
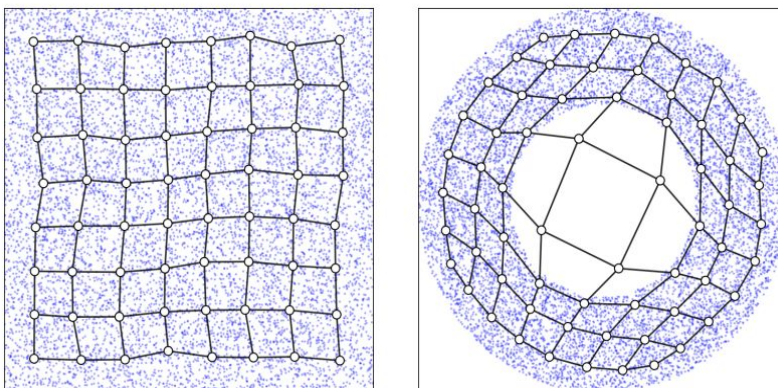


Figure II.10 : Modèle Hopfield

#### 4.5.3. Famille des réseaux à résonance:

- **Machine de Boltzmann:** c'est un réseau de neurones où toutes les couches cachées sont interconnectées, cette machine apprend un comportement désirer, c'est une extension probabiliste du modèle **Hopfield**.
- **Réseaux auto-organisés :**

Ce type de réseau utilise une méthode d'apprentissage **non supervisées**, ce type de réseaux se distingue par une connectivité locale, adapté pour le traitement de l'information en spirales, le modèle le plus connue la **carte auto-organisatrices** de **Kohonen**.[18]



**Figure II.11 : Exemple d'utilisation carte auto-organisation.**

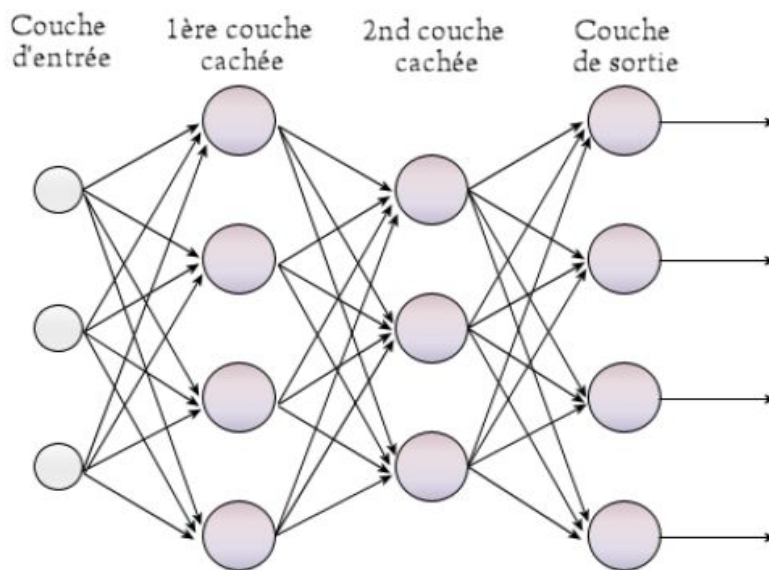
#### **4.6. Les couches d'un Réseau de neurone:**

Le perceptron est organisé en trois couches:

**Couche Entrée: (Input Layer)** c'est l'ensemble de neurones qui porte le signal d'entrée du réseau, et par la suite tous les neurones de cette couche sont reliés à la couche suivante.

**Couche cachée: (Hidden layers)** elles peuvent être une ou plusieurs, c'est ici où les relations entre les variables vont être mise en exergue. Le choix du nombre de couches et de neurones est intuitifs et nécessite de l'expérience venant de l'expert.

**Couche sortie: (Output Layer)** elle représente le résultat du réseau de neurones c'est ce qu'on appelle la **prédiction**.



**Figure II.12 : Perceptron Multicouche**

#### **4.7. Fonctionnement réseaux de neurone simple type feed forward :**

Chaque neurone effectuera une série d'opération section (I.4.3), qui à son tour renvoie les résultats vers les autres neurones situés sur la **couche intermédiaire**. Ainsi cette opération est effectuée successivement sur chaque couche donne une **Somme pondérée de signaux**

$(\sum w_{ij} \cdot x_i) + b$  auquel on ajoute un **biais b**, le résultat est par la suite affecté à une fonction d'activation.

Les poids ne sont pas forcément tous liés entre eux cela peut différer selon la topologie du réseau, et le biais peut être vu comme un neurone externe auquel on affecte toujours 1, grâce à lui la fonction d'activation va être décalée ce qui augmente l'opportunité d'apprentissage.

Les poids ainsi que les biais sont tous 2 des paramètres d'apprentissage dont la valeur va changer tout au long du processus d'apprentissage, on dit qu'ils seront (mis à jour).

#### 4.8. Traitement des données:

Les **données** sont le nerf de la guerre surtout quand il s'agit de Machine Learning, car ces dernières occupent une importance capitale pour construire un bon modèle.

Avant de découper ou fragmenter les données afin de procéder à l'apprentissage, une étape préalable est nécessaire c'est le **prétraitement de données** ou **Data preprocessing**.

Durant cette étape on va traiter les données en effectuant des transformations, le but est de transformer des données d'un état brut à un ensemble de données vierges exploitables par un modèle.

Il y a certains aspects à prendre en compte comme le fait que les données prises en entrées de chaque modèle sont différentes, mais aussi la nécessité de généraliser afin qu'on puisse tester plusieurs algorithmes pour la comparaison.

Nous allons voir 3 méthodes de prétraitement des données pour le machine learning:

- **Rescale Data:** remettre à l'échelle la donnée, car il arrive que certaines données au sein même du dataset aient différentes échelles.

le redimensionnement des données en entrées est chose utile pour l'utilisation d'**algorithme d'optimisation** (ex: gradient descent).

- **Binarize data:** ou **Binarisation des données** c'est une méthode qui permet de transformer des entrées quelconque en valeur binaire, et cela est chose possible grâce au **seuil binaire (binary threshold)** à partir duquel on va comparer si la valeur de nos données sont supérieures au seuil alors on marque d'un 1, et dans le cas contraire on marque d'un 0.

- **Standardize data:** Standardisation ou normalisation des données, c'est une méthode qui s'avère utile pour transformer des attributs en distribution gaussienne et des moyennes et écart-types différents en une distribution gaussienne standard avec une moyenne de 0 et un écart-type de 1.

#### 4.9. Découpage des données:

L'ensemble des données ou **Dataset** à disposition est découpé selon 2 voire 3 parties:

- **Données d'apprentissage:** ou **training set** qui représente environ **80%** des données mise à disposition, elles serviront à entraîner notre modèle.
- **Données de test:** ou **Test set** représente **20%** des données globales elles serviront durant la phase de test du modèle.
- **Données de Validation :** représente une portion (15 à 20%) des données qui vont servir à tester le modèle, pour valider les résultats de l'entraînement. ses données ne servent pas à évaluer l'efficacité d'un modèle, mais plutôt des algorithmes entre eux.

**Remarque :** La validation de l'entraînement n'est pas obligatoire.

#### 4.10. Fonction perte:

Également appelé **fonction d'erreur (Loss Function)**, dont le rôle de base est simple, c'est l'évaluation de nos prédictions effectuées sur le dataset, si elles sont **bonnes** ou **mauvaises**, dans le cas où les prédictions sont mauvaises voir même erronées alors elle générera une valeur supérieure, et dans le cas contraire elle produira une valeur inférieure.

Il y a de nombreuses fonctions permettant de calculer la perte (erreur), voici quelques-unes pour la **classification**. [16]

- **Mean Square Error: MSE** c'est l'une des fonctions les plus utilisées, et plus simple à mettre en oeuvre.

On prend la différence entre vos prédictions, et la vérité sur le terrain, vous la corrigez et vous la calculez en moyenne sur l'ensemble du jeu de données.

- **Log Loss (Cross Entropy Loss):** C'est aussi une fonction de perte populaire, et son utilisation principale réside dans les problèmes de **classifications** employant les **logarithmes**. Section (II. 5.4)

#### 4.11. Rétro-propagation :

C'est ici que toute la magie d'un modèle opère, il est l'algorithme qui rend l'apprentissage d'un modèle possible, aussi appelé **Backpropagation algorithm** [13].

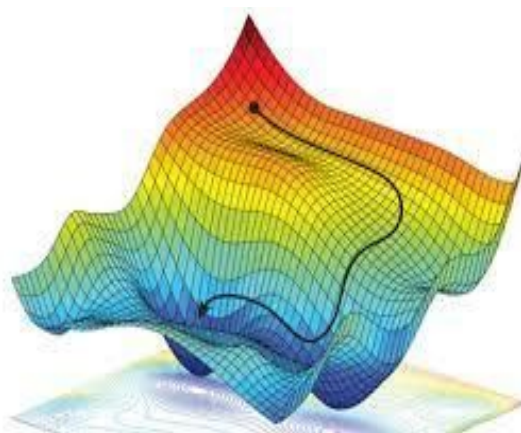
Son fonctionnement est assez simple, il va effectuer une rétro propagation de l'erreur calculé au niveau de la dernière couche (sortie), qu'il va par la suite distribué couche par couche dans le sens contraire d'avancement (en arrière). au sein de la rétro propagation un autre algorithme va prendre le relais appeler **Descente de gradient** ou **gradient descent** il agit au coeur même de la rétro-propagation, c'est lui qui va s'occuper du calcul des **dérivées partielles** de l'erreurs au niveau de chaque couche [14].

L'**erreur** est obtenue en prenant le résultat obtenu après calcul (la prédiction) auquel on soustrait au résultat attendu (cas de l'apprentissage supervisé), cette opération est effectuée grâce à la **fonction de perte (Loss Function)**. [16]

Pour à la fin obtenir un **coefficient d'ajustement** qui nous permettra de mettre à jour nos paramètres d'apprentissages.

#### 4.12. Gradient Descent :

Est par définition un **algorithme d'optimisation** du **premier ordre** (cela signifie qu'on effectue une dérivation partielle du premier de degré de la fonction), dont le but est de trouver un minimum globale pour une fonction (le **minimum globale** c'est la plus petite valeur, qu'une fonction puisse atteindre sur tout son domaine, à la différence d'un **minimum local**, qui quant à lui est n'est que la plus petite valeur pour un intervalle donné) [15].



**Figure II.13 : Schématisation de la descente de gradient**

Dérivé l'erreur du système permet de nous diriger vers le minimum de ce dernier, donc en calculant la dérivée partielle de l'erreur par rapport à un poids, on obtient la pente de la

dépendance de notre système en cette variable. En d'autre terme si le résultat de la dérivée partielle est **positive**, cela signifie qu'il faut **diminuer** notre **poids** pour « **descendre la courbe d'erreur** », et dans le cas contraire **négatif**, il faudrait l'**augmenter** toujours avec le même coefficient calculé.

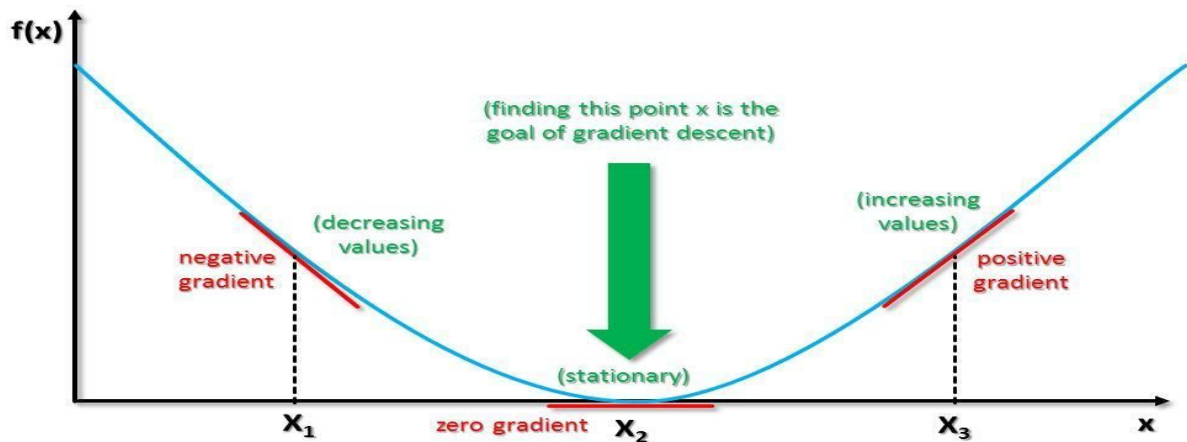


Figure II.14 : Local minima avec Gradient Descent

La mise à jour d'un poids  $W_i$  est donc, pour  $\eta$  le **taux d'apprentissage (Learning rate)** et  $\alpha$  la pente trouvée de l'erreur par rapport à ce poids :  $W_i := W_i - \eta \cdot \alpha$ .

L'ajout d'un **taux d'apprentissage** est important car cela nous permet de raccourcir le temps de recherche du minimum, éviter aussi de tomber dans ce qu'on appelle local minima.

Le choix de la valeur taux d'apprentissage à affecter à notre algorithme est importante car elle nous permet de bénéficier de ses avantages, on peut distinguer 2 cas:

- **valeur trop faible**: le processus s'éternise, et prend plus de temps qu'il n'en faut pour converger.
- **valeur trop grande**: on risque dans ce cas de basculer de l'autre côté sans jamais, sans jamais s'en rapprocher.

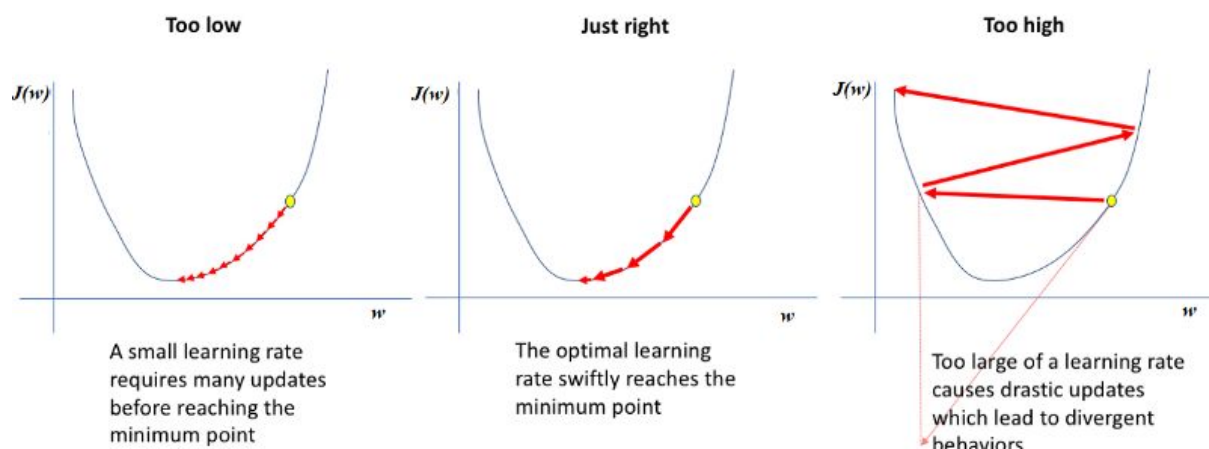


Figure II.15 : Learning rate Petite valeur vs Grande vs valeur idéale

### 4.13. Algorithme d'optimisation:

Le rôle d'un algorithme d'optimisation au sein du modèle durant la phase d'apprentissage est crucial car c'est grâce à ce dernier qu'on sera en mesure de calculer les coefficients d'ajustement nécessaires pour notre modèle.

Il existe de nombreux algorithmes d'optimisation, le bon choix reste une chose importante

Les algorithmes d'optimisation nous aident à **minimiser** (ou à **maximiser**) une **fonction objectif** ou **fonction d'erreur**, son rôle est cruciale durant la phase d'apprentissage car c'est grâce à ce dernier qu'on sera en mesure de calculer les coefficients d'ajustement, pour ajuster les poids ainsi que les biais.

On peut citer plusieurs algorithmes d'optimisation qui sont présents dans 2 principales catégories:

#### 4.13.1. Algorithmes d'optimisation de premier ordre:

Ces algorithmes minimisent ou maximisent une fonction d'erreur  $E(x)$  en utilisant ses valeurs de gradient par rapport aux paramètres.

Les **dérivés** du **premier ordre** nous donnent essentiellement une **ligne tangentielle** (Pente) à un point de sa **surface d'erreur**, la dérivé première nous indique si la **fonction augmente** ou **diminue** à un moment donné. L'algorithme le plus utilisé est la **Descente de gradient**.

Cette méthode est moins coûteuse, et converge vite dans les gros dataset.

Un **gradient** est représenté par une **matrice jacobéennes** - qui est simplement une matrice dont les éléments qui l'a compose sont des **dérivés partiels du premier ordre**.

#### **Descente de gradient classique:**

$\theta = \theta - \eta \cdot \nabla J(\theta)$  C'est la formule de mise à jour des paramètres, où ' $\eta$ ' est le **taux d'apprentissage**, ' $\nabla J(\theta)$ ' est le **Gradient** de la fonction d'erreur,  $J(\theta)$  c'est la **matrice jacobéennes** avec les paramètres ' $\theta$ '.

#### **4.13.2. Algorithmes d'optimisation du second ordre :**

Les méthodes de **second ordre** utilisent la **dérivée du second ordre**, ce qui signifie qu'on dérive encore une fois la dérivé du premier ordre pour l'obtenir, elle indique si la première dérivé **augmente** ou **diminue**.

Ce type de dérivé nous fournit une surface quadratique qui touche la courbure de l'erreur en sa surface.

Il est aussi appelé **Hessian** qui est représenté à l'aide d'une **matrice** dont les composantes sont **des dérivées partielle du second ordre**.

Ce type d'algorithme d'optimisation n'est pas très utilisé car il est **très coûteux niveau calcul**, mais il a certe quelques avantages majeurs sur la première méthode,

- il n'ignore pas les courbure d'une courbe donné.
- en terme de performance pas à pas (étape par étape), elles sont bien meilleurs que celles du premier degré.

## **5. Réseaux de Neurones Convolutifs:**

### **5.1. Historique:**

Les systèmes de reconnaissance d'imagerie classiques sont généralement composés de 2 principaux blocs :

- **Extracteur de Caractéristiques:** son rôle est d'effectuer des transformations dans le tableau (matrices) représentatives de l'image en une série de nombres dans un vecteur, ce résultat sera par la suite transmis au classifieur.

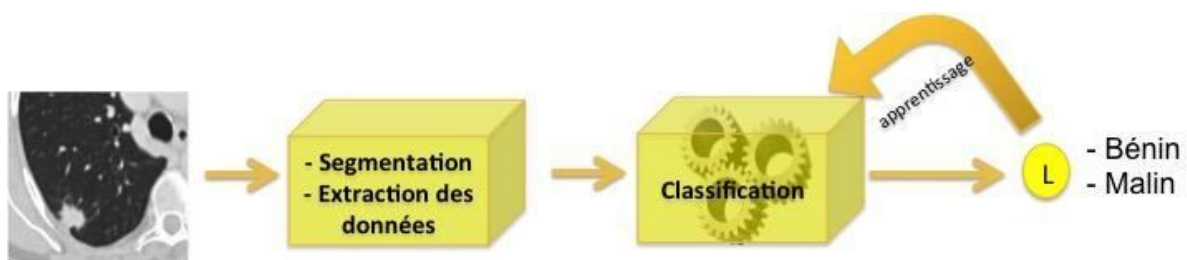
- **Classifieur Entraînable:** aura pour rôle de classifier, en effectuant des calculs de sommes pondérées sur les caractéristiques, et à la fin on obtiendra un résultat qui est comparé à un seuil, où une seule classe sera finalement reconnue.

Ces méthodes **perceptron** ou **logistic regression** sont limitées, et cela à cause de la complexité de créer un tel classifieur adapté au problème, ainsi que les limites de prédiction de ses méthodes.



**Figure II.16: classification des images (Ancienne approche)**

C'est là que le deep Learning intervient avec son approche plus simplifiée, il synthétise ces deux étapes en réalisant à la fois l'extraction des caractéristiques de l'image et la classification. Plus besoin d'extraire les données de l'image pour ensuite les analyser.

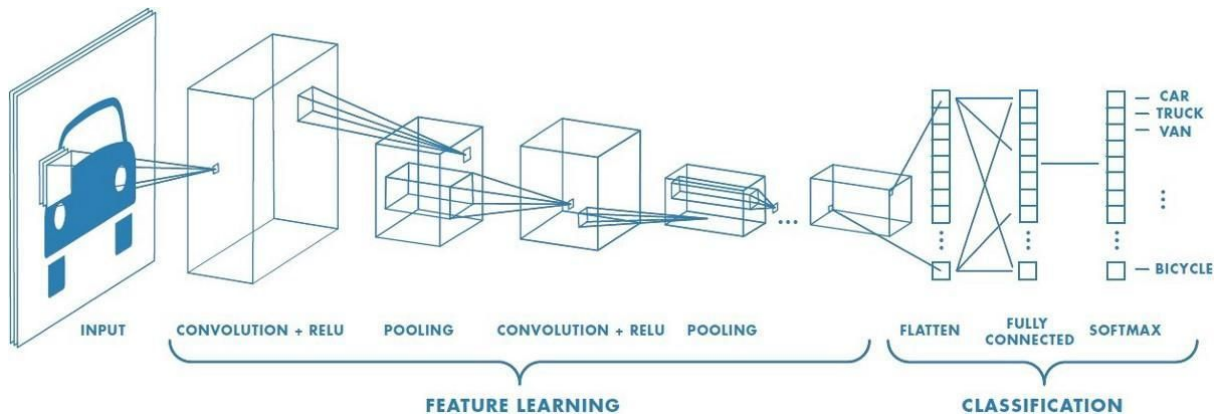


**Figure II.17: classification des images (Nouvelle approche)**

Dans cette partie nous allons voir un algorithme de deep Learning (les réseaux convolutionnels), leurs architectures, ainsi que leur fonctionnement en détails

## 5.2. Architecture d'un ConvNet:

Un CNN comporte généralement trois couches: une **couche convolutionnelle** (Conv Layer) une **couche de regroupement** (pooling layer), et une **couche entièrement connectée** (fully connected layer).



**Figure II.18 : Architecture ConvNet**

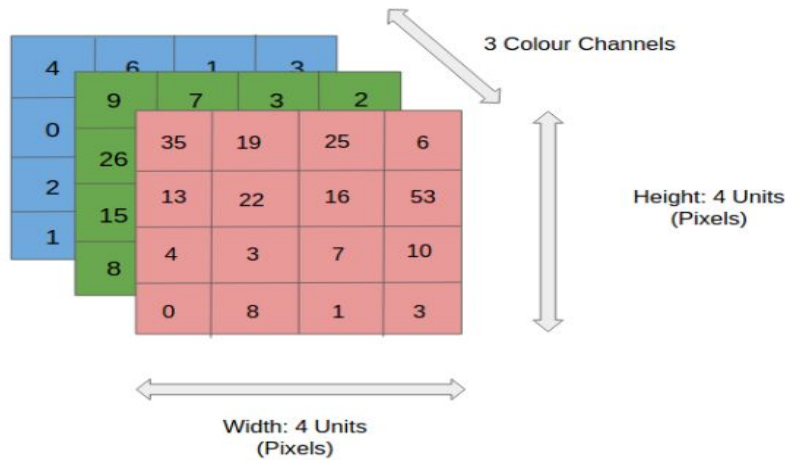
### 5.2.1. Couche Convolutive:

La couche de convolution est la pierre angulaire de la CNN. Il supporte la majeure partie de la charge de calcul du réseau.

Cette couche effectue un produit entre deux matrices, une matrice étant l'ensemble de paramètres pouvant être appris, également appelée noyau ou filtre, et l'autre matrice étant la partie restreinte du champ récepteur.

Le noyau est spatialement plus petit qu'une image, mais plus en profondeur. Cela signifie que, si l'image est composée de trois canaux (RVB), la hauteur et la largeur du noyau seront spatialement petites, mais la profondeur s'étend jusqu'aux trois canaux.

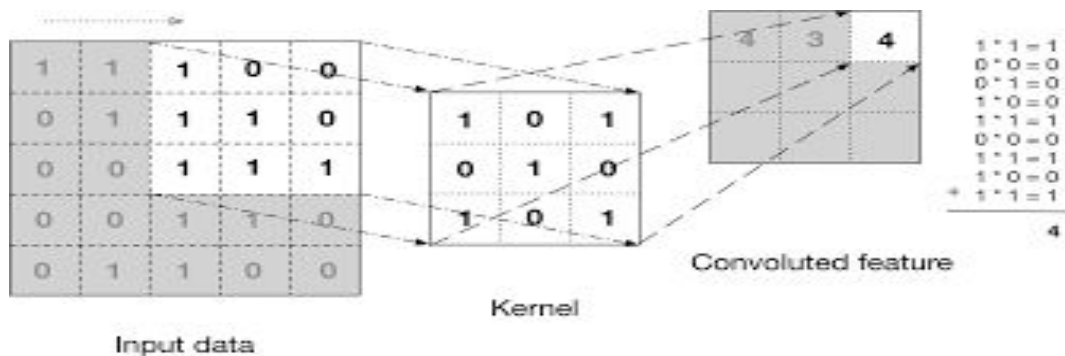
Lors du passage en avant, le noyau glisse sur la hauteur et la largeur de l'image, produisant la représentation de cette région réceptrice. Ceci produit une représentation en deux dimensions de l'image connue sous le nom de carte d'activation qui donne la réponse du noyau à chaque position spatiale de l'image. La taille glissante du noyau s'appelle une foulée (strides).



**Figure II.19 : Représentation d'une image en couleur sous forme matricielle**

Si nous avons une entrée de taille  $W \times W \times D$  et le nombre de noyaux  $D_{out}$  avec une taille spatiale de  $F$  avec la foulée  $S$  et la quantité de remplissage  $P$ , alors la taille du volume de sortie peut être déterminée par la formule suivante:

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

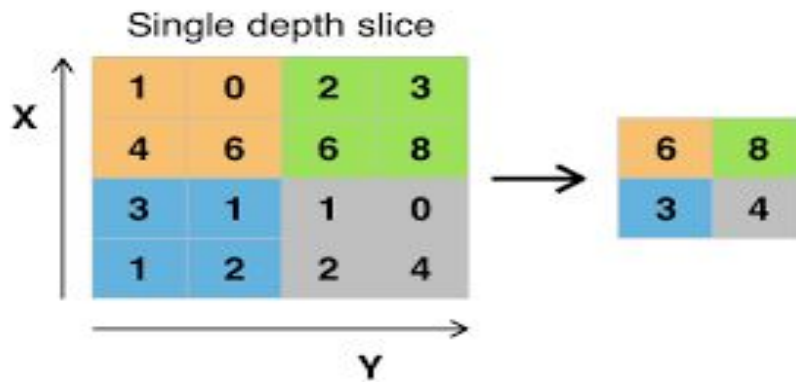


**Figure II.20 : Opération de Convolution**

### 5.2.2. Couche de Regroupement:

La couche de regroupement remplace la sortie du réseau à certains emplacements en générant une statistique résumée des sorties à proximité. Cela aide à réduire la taille spatiale de la représentation, ce qui diminue le nombre de calculs et de poids requis. L'opération de regroupement est traitée sur chaque tranche de la représentation individuellement.

Il existe plusieurs fonctions de regroupement telles que la moyenne du voisinage rectangulaire, la norme L2 du voisinage rectangulaire et une moyenne pondérée basée sur la distance du pixel central. Cependant, le processus le plus populaire est le pooling maximum, qui rapporte la sortie maximale du voisinage.



**Figure II.21: Opération pooling maximum**

Si nous avons une carte d'activation de taille  $W \times W \times D$ , un noyau de regroupement de taille spatiale  $F$  et stride  $S$ , la taille du volume de sortie peut être déterminée à l'aide de la formule suivante:

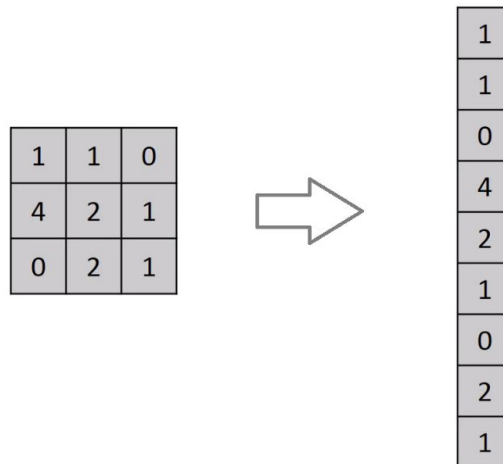
$$W_{out} = \frac{W - F}{S} + 1$$

Cela donnera un volume de sortie de taille  $W_{out} \times W_{out} \times D$ .

Dans tous les cas, la mise en commun fournit une certaine invariance de traduction, ce qui signifie qu'un objet est reconnaissable indépendamment de l'endroit où il apparaît sur le cadre.

### 5.2.3. Couche Entièrement Connectée:

Dans le fonctionnement entièrement connecté d'un réseau de neurones, la représentation d'entrée est aplatie dans un vecteur de caractéristiques et transmise à travers un réseau de neurones afin de prédire les probabilités de sortie. L'image suivante décrit l'opération d'aplatissement:



**Figure II.22: opération d'aplatissement**

Les lignes sont concaténées pour former un vecteur long. Si plusieurs couches d'entrée sont présentes, ses lignes sont également concaténées pour former un vecteur de caractéristiques encore plus long.

Le vecteur de caractéristique est ensuite passé à travers plusieurs couches denses (entièrement connectées). À chaque couche dense, le vecteur de caractéristiques subit les mêmes opérations que dans un réseau de neurones classique.

La couche entièrement connectée aide à mapper la représentation entre l'entrée et la sortie.

#### **5.2.4. Couche Sortie:**

La couche de sortie d'un CNN est chargée de produire la probabilité de chaque classe (chaque chiffre) en fonction de l'image d'entrée. Pour obtenir ces probabilités, nous initialisons notre couche Dense finale afin qu'elle contienne le même nombre de neurones qu'il y a de classes. La sortie de cette couche dense passe ensuite par la **fonction d'activation Softmax**, qui mappe toutes les sorties de la couche dense finale sur un vecteur dont les éléments totalisent un:

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}$$

Ici x totalise chaque élément dans la couche finale.

#### **5.2.5. Couche de non-linéarité:**

Étant donné que la convolution est une opération linéaire et que les images sont loin d'être linéaires, des couches de non-linéarité sont souvent placées directement après la couche de convolution pour introduire une non-linéarité dans la carte d'activation.

Il existe plusieurs types d'opérations non linéaires, les plus populaires étant:

- **Sigmoid:** Prend un nombre réel et le réduit à un intervalle compris entre 0 et 1.

Mathématiquement cela donne  $\sigma(\kappa) = 1 / (1 + e^{-\kappa})$ .

Cependant sigmoid possède certaines propriétés indésirables le gradient peut atteindre des valeurs très petites voir nul ce qui rend l'étape d'apprentissage impossible.

- **Tanh:** écrase un nombre de valeurs réelles dans l'intervalle [-1, 1]. Comme pour le sigmoïde, l'activation sature mais, contrairement aux neurones sigmoïdes, sa sortie est centrée sur le zéro.
- **ReLU:** L'unité linéaire rectifiée (ReLU) est devenue très populaire ces dernières années. Il calcule la fonction  $f(\kappa) = \max(0, \kappa)$ . En d'autres termes, l'activation est simplement un seuil à zéro.

ReLU est plus fiable et accélère la convergence de six fois, en comparaison avec les autres sigmoid ou Tanh.

Un inconvénient de ReLU est qu'elle est fragile pendant la formation, et cela à cause d'une grande valeur de gradient le traverse rendant la mise à jour impossible.

On peut régler ce problème assez facilement en fixant le taux d'apprentissage à une valeur appropriée.

### 5.3. Calcule de la perte:

Pour mesurer la précision des prédictions de notre réseau, en fonction des images d'entrée, nous utilisons une fonction de perte.

Cette dernière attribue un nombre à valeur réelle afin de définir la précision du modèle lors de la prévision du chiffre en sortie. Une fonction de perte courante à utiliser lors de la prédiction de plusieurs classes de sortie est la **fonction Perte catégorique contre entropie (categorical cross entropie)**, définie comme suit:

Ici,  $\hat{y}$  est la **prédiction** du CNN et  $y$  est la **sortie souhaitée**. Lorsque vous faites des prédictions sur plusieurs exemples, nous prenons la moyenne de la perte sur tous les

exemples.

$$H(\mathbf{y}, \hat{\mathbf{y}}) = \sum_i y_i \log \frac{1}{\hat{y}_i} = - \sum_i y_i \log \hat{y}_i$$

#### 5.4. ConvNet vs ANN:

Les réseaux de neurones convolutionnels, **CNN** ou **ConvNet** appartiennent à la famille des réseaux de neurones passage vers l'avant (feed forward), qui ressemblent beaucoup aux réseaux de neurones classiques évoqué dans la partie précédente: ils se constituent de neurones dont les **poids**, et les **biais** peuvent apprendre.

Chaque neurone reçoit des entrées, effectue un produit scalaire, suivi éventuellement d'une fonction de non-linéarité. L'ensemble du réseau n'exprime toujours qu'une seule fonction de score différentiable: des pixels d'image brute d'un côté aux scores de classe de l'autre. Et ils ont toujours une fonction de perte (par exemple, SVM / Softmax) sur la dernière couche (entièrement connectée), et les algorithmes d'optimisations vue fonctionnent toujours.

Ce qui change dans les architectures ConvNet supposent explicitement que les entrées sont des images, ce qui nous permet de coder certaines propriétés dans l'architecture. Ainsi que le traitement effectué sur ses images afin d'extraire les caractéristiques dites importantes (convolution - pooling) qui seront évoqué plus en bas.

Ces opération rendent alors la fonction de transmission directe plus efficace à mettre en œuvre, et réduisent la quantité de paramètres dans le réseau, ce qui réduit considérablement la le nombre de calculs.

### 6. Impact de L'IA sur l'imagerie médicale:

Les techniques d'apprentissage profond permettent d'assister les professionnels de la santé dans l'analyse de l'imagerie médicale, afin de repérer les anomalies avec d'avantage de précision.

C'est là où l'intelligence artificielle peut aider les médecins à détecter plus facilement les anomalies sur les radios des patients un radiologiste ayant à examiner plusieurs clichés radio par jour afin de prendre une décision (perte de précision), alors qu'un algorithme de Machine Learning est en mesure d'en examiner plus en moins de temps et sans pertes de précision.

D'autre part il y a bon nombre de difficulté qui plane dans le domaine du diagnostic à par de cliché médicale, et cela à cause de la taille insignifiante des indicateurs d'une pathologie au sein du cliché rendant la tâche complexe pour les spécialistes même les plus aguerrie. ce qui explique le taux d'erreur de diagnostic médicale.

**ex:** google qui a créé un algorithme pour détecter des tumeur cancéreuses du sein, ce dernier a été en mesure d'atteindre une performance de 92.4% contre 74.6% pour les spécialistes.

Un obstacle majeur pour l'IA c'est la difficulté pour une autorité de valider un système de diagnostic qui représente toutes les caractéristiques d'une boîte noire, contrairement au diagnostic humain dans lequel on peut retracer le cheminement du spécialiste, mais selon les scientifiques du domaine c'est aussi un avantage non négligeable.

Enfin une distinction importante a été apporter par Kimberly Powell (Vice Présidente chez NVIDIA - Intelligence artificielle (Machine & Deep Learning) & Calcul accéléré) sur la **détection**, et **diagnostic** disant: *“Si la détection repose sur des techniques comme la radiologie ou les ultrasons, la pose d'un diagnostic s'appuie sur des technologies plus complexes, comme la tomodensitométrie ou l'IRM, qui riment avec un jeu de données plus complexe”*.

## **7. Évaluation d'un modèle:**

Dans cette partie, on va s'intéresser à comment les modèles de prédictions en Machine Learning sont évalués, et plus précisément dans le cas de la classification.

L'étape d'évaluation s'avère être une étape importante dans la démarche de choix du modèle, d'où la nécessité d'avoir un outil (matrice de confusion) de précision qui nous permet de mener cette étape en bonne et due forme.

### **7.1. Matrice de Confusion:**

**La Matrice de confusion** est généralement utilisée dans le domaine du **Machine Learning** et plus précisément dans les problèmes de type **classification statistique**, également appelée **matrice d'erreur**.

Cette matrice permet de vérifier la **performance** d'un **classifieur**, ainsi que la **confusion** qui règne entre les **différentes classes**, sur un ensemble de **données de test** dont les **vraies valeurs** sont **connues**.

Elle résume et englobe les prédictions effectuées par le modèle de classification, les informations qu'elle nous transmet, nous permettent de voir la confusion entre les classes, et de distinguer les erreurs commises par ses dernières, ainsi que leurs natures.

		Classe réelle	
		-	+
Classe prédite	-	<b>True Negatives</b> <i>(vrais négatifs)</i>	<b>False Negatives</b> <i>(faux négatifs)</i>
	+	<b>False Positives</b> <i>(faux positifs)</i>	<b>True Positives</b> <i>(vrais positifs)</i>

**Figure III.23: Matrice de Confusion**

L'exemple ci-dessus montre une matrice de confusion dans le cas d'une classification binaire (2 classes seulement), mais on peut également représenter à l'aide d'une matrice de confusion dans le cas où le nombre de classes excède 2, section (IV.3).

Au sein de la matrice de confusion, on retrouve les thèmes suivants:

- **Positif (P)**: l'observation est positive (par exemple: c'est une pomme).
- **Négatif (N)**: l'observation n'est pas positive (par exemple: ce n'est pas une pomme).
- **Vrai Positif (TP)**: l'observation est positive, et elle est prédite comme telle.
- **Faux Négatif (FN)**: l'observation est positive, mais a été prédite négative.
- **Vrai Négatif (TN)**: l'observation est négative, et elle est prédite comme telle.
- **Faux Positif (FP)**: l'observation est négative, mais elle est prédite positive.

## 7.2. Critères d'évaluation:

La matrice de confusion contient des valeurs liées aux observations obtenues par le modèle sur le dataset, et à partir de ses dernières on est en mesure de calculer des métriques d'évaluation.

### 7.2.1. Sensibilité :

Le Rappel ou la **sensibilité** est la mesure de la proportion de cas positifs réels prédits positifs (vrais positifs). Elle peut être définie comme le rapport entre le nombre total d'exemples positifs correctement classés et le nombre total d'exemples positifs.

**High Recall (Rappel élevé)** indique que la **classe** est **correctement reconnue** (petit nombre de FN).

Le Rappel est donné par la relation:

$$\text{Rappel} = \frac{TP}{TP + FN}$$

		Actual	
		Positives(1)	Negatives(0)
Predicted	Positives(1)	TP	FP
	Negatives(0)	FN	TN

$$\text{Recall} = \frac{TP}{TP + FN}$$

**Figure III.24: calculer Rappel sur matrice de confusion**

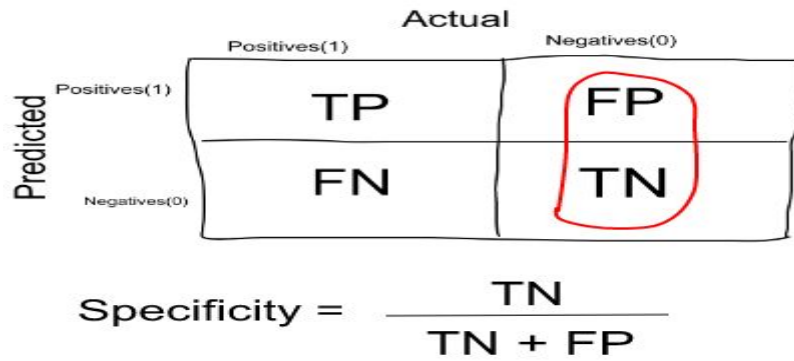
Plus la sensibilité est élevée, plus le vrai positif est élevé, plus le faux négatif est faible. La valeur inférieure de sensibilité signifierait une valeur inférieure de vrai positif, et une valeur supérieure de faux négatif. Pour les domaines de la santé et de la finance, des modèles à haute sensibilité seront souhaités.

### 7.2.5. Spécificité:

La spécificité est définie comme la proportion de négatifs réels, ce qui a été prédit comme négatif (ou vrai négatif). Cette proportion pourrait également être appelée un taux de faux positif. La somme de la spécificité et du taux de faux positifs serait toujours égale à 1.

La mesure de la spécificité se fait grâce à la formule:

$$\text{Spécificité} = \frac{TN}{TN+FP}$$



**Figure III.25: calcule Spécificité sur matrice de confusion**

### 7.2.6. Sensibilité vs Spécificité:

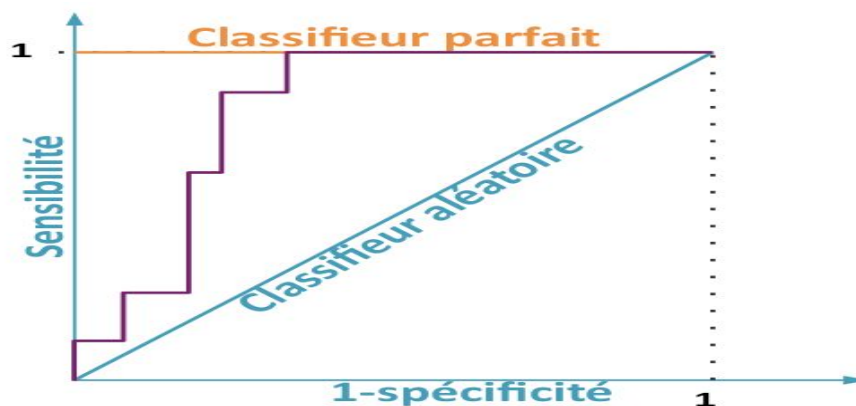
La mesure de sensibilité est utilisée pour déterminer la proportion de cas positifs réels, qui a été correctement prédite, la mesure de spécificité est utilisée pour déterminer la proportion de cas négatifs réels, qui a été correctement prédite.

#### 7.2.6.1. La courbe ROC:

Roc qui signifie Receiver operating characteristic prend des mesures de sensibilité et spécificité afin de tracer la courbe ROC.

Elle prend en abscisse l'**anti spécificité (1-spécificité)**, et en ordonné la **sensibilité**.

L'exemple ci-dessous montre les différents cas de courbes ROC qu'on puisse trouver:



**Figure III.26: Exemple de Courbe ROC pour classifieur aléatoire, normal et parfait**

Grâce à la courbe ROC, on est également en mesure de calculer **AUC ROC** qui est l'aire sous la courbe ROC, qui nous donne une information sur le niveau de séparabilité du modèle.

### 7.2.6.2. Interprétation des Résultats d'une courbe ROC AUC:

Comme nous le savons, ROC est une courbe de probabilité, selon le choix du seuil de la courbe ROC, on tombera dans l'un des cas suivants:

- **Cas 1: Idéal**

C'est une situation **idéale**. Lorsque deux courbes ne se chevauchent pas du tout, cela signifie que le modèle offre une mesure idéale de la séparabilité. Il est parfaitement capable de faire la distinction entre classe positive et classe négative.

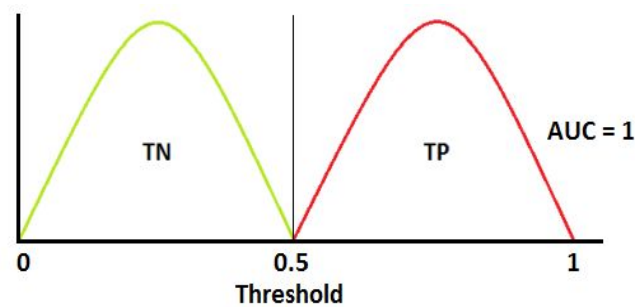


Figure I.27: Cas Idéal

- **Cas 2: Ordinaire**

Lorsque **deux** distributions **se chevauchent**, nous introduisons une **erreur de type 1** et de **type 2**. En fonction du seuil, nous pouvons les minimiser ou les maximiser. Lorsque AUC est égal à 0,7, cela signifie qu'il y a 70% de chances que le modèle puisse faire la distinction entre classe positive et classe négative.

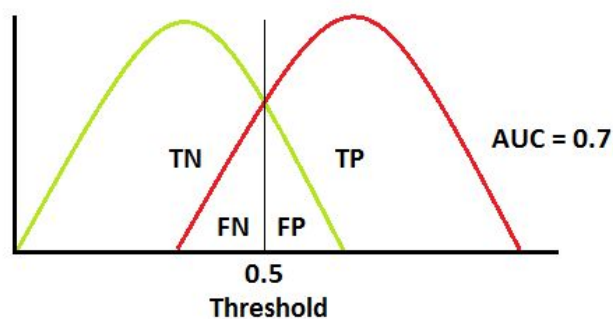


Figure I.28: Cas ordinaire

- **Cas 3: Aléatoire**

C'est la pire situation. Lorsque l'AUC est approximativement égal à 0,5, le modèle ne dispose d'aucune capacité de discrimination permettant de faire la distinction entre classe positive et classe négative.



### Figure I.29: Cas Aléatoire

- **Cas 4: Pire**

Lorsque AUC est approximativement égal à 0, le modèle inverse les classes. Cela signifie que le modèle prédit la classe négative en tant que classe positive et inversement.

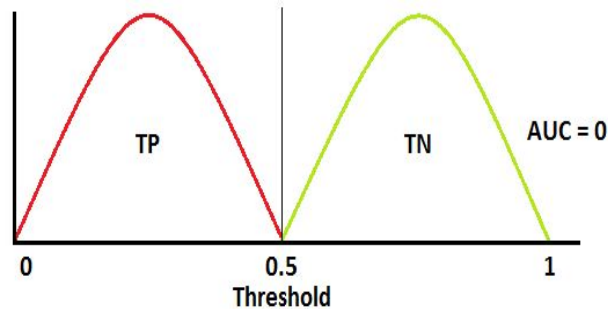


Figure I.30: Pire des cas

#### 7.2.6.3. Relation entre sensibilité, spécificité, taux de faux positif et seuil:

La sensibilité et la spécificité sont inversement proportionnelles. Donc, lorsque nous augmentons la sensibilité, la spécificité diminue et inversement.

La **diminution** du **seuil**, donne **plus de valeurs positives**, ce qui **augmente** la **sensibilité** et **diminue** la **spécificité**.

De même, l'**augmentation** du **seuil**, donne **plus de valeurs négatives**, ce qui **augmente** la **spécificité**, et **diminue** la **sensibilité**.

De même l'augmentation du **TPR** (taux de vrai positive), augmente également **FPR** (taux faux positive) qui se calcul 1-spécificité, et l'inverse reste correcte.

#### 7.2.6.4. Représentation de la Courbe ROC :

- **Cas 1: classification classe binaire:** C'est le cas classique dans lequel on traite une ROC, et on l'évalue.
- **Cas 2: classification multi-classes:** Dans le cas d'une classification multi-classes, on aura une courbe ROC pour chaque classe donnée. Par exemple: 3 classes (X, Y, Z), chaque courbe est comparée aux autres courbes différentes. Il y aura autant de seuil (**threshold**) qu'il y aura de classes. Dans cet exemple, on aura 3 seuils d'activations.

Dans ce mémoire, on est dans le cas multi-classes avec 3 classes.

## **8. Conclusion:**

Ce chapitre a été consacré à la présentation des notions de bases du Deep Learning, et ceci dans le but d'introduire les réseaux de neurones convolutionnels.

En passant par l'explication de concepts plus général tel que le Machine Learning ses type d'apprentissage et problèmes liés, suivit du deep learning avec les réseaux de neurones dans lequel les différences, ainsi que les raisons majeur de l'émergence du deep learning sont misent en avant.

Enfin, nous avons introduit le rôle des réseaux convolutionnel dans le traitement des images, et plus précisément dans le cas de l'imagerie médicale, et l'impact de cette approche dans le domaine médicale.

## **Chapitre II :**

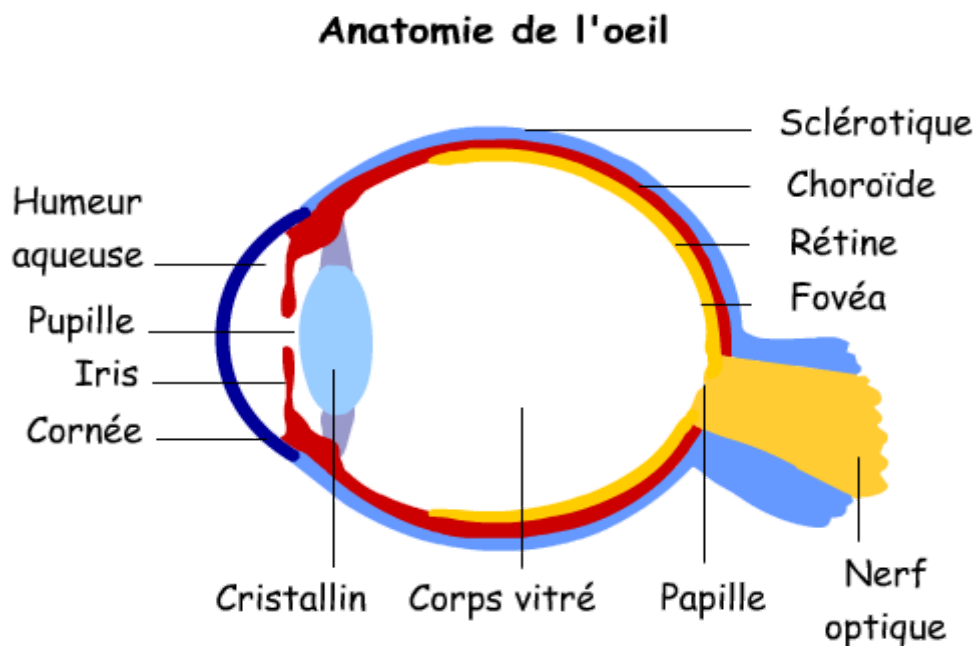
### **Organe Concerné et Pathologie**

## 1. Introduction:

Dans ce chapitre l'accent sera apporté à l'organe oculaire dans lequel nous verrons sa fonction ainsi que son anatomie, puis nous verrons le diabète, et une complication liée à ce dernier la Rétinopathie qui sera le sujet de notre mémoire, enfin un état d'art sur les travaux accomplis par la communauté scientifique dans la détection de la pathologie.

## 2. Anatomie de l'œil:

L'œil humain est l'organe qui octroie la vue, c'est aussi le sens le plus sollicité, il reçoit et transforme les vibrations électromagnétiques de la lumière en influx nerveux qui sont transmis au cerveau via le nerf optique.



**Figure II.1 : Anatomie de l'œil**

L'illustration ci-dessus démontre toute la complexité de cet organe, sa forme sphérique est constituée de 3 enveloppes dites **Tuniques** qui baignent dans un liquide:

### 2.1. Tunique Externe de l'œil:

Aussi appelé **sclérotique** : c'est le blanc de l'œil, elle occupe un rôle important dans la protection contre les dégâts mécaniques, et soutient sa structure, cette dernière est recouverte d'une membrane très fine en sa partie antérieure appelé **conjonctive**.

On peut observer dans la figure ci-dessous la tunique externe en Bleu.

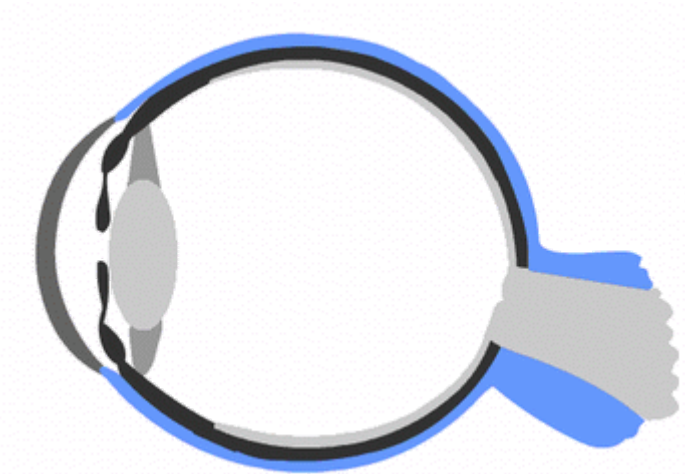


Figure II.2: Sclérotique ou **Tunique Externe**

## 2.2. **Tunique Moyenne de l'œil :**

Aussi appelé la **choroïde** se trouvant entre la **rétine**, et la **membrane externe**, elle est fortement vascularisée et cela veut dire, qu'elle est parcouru par beaucoup de vaisseaux sanguins dont le rôle principale est d'apporter les nutriments nécessaires à l'oeil pour son bon développement, et le bon déroulement de sa fonction.

On peut observer dans la figure ci-dessous la tunique externe en Rouge.

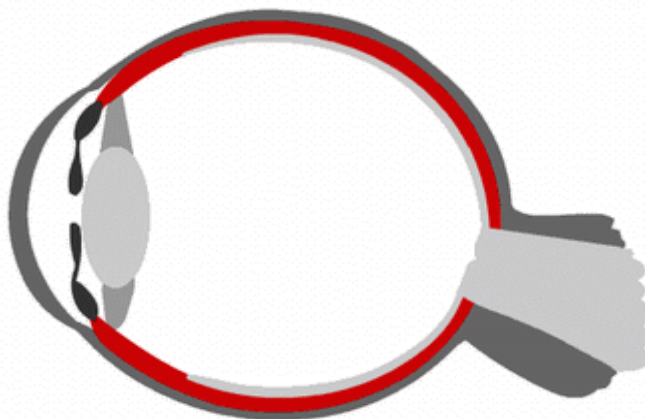


Figure II.3 : **choroïde** ou **Tunique Moyenne**

## 2.3. Tunique Interne de l'œil:

Aussi appelé **Rétine** c'est au niveau de cette partie que la magie opère, elle est connue pour être le siège principale de la vision. Tous les flux lumineux captés sont acheminés jusqu'à cette partie, après avoir été focalisés, et filtrés par le **pupille** et la **cornée**, où les flux lumineux sont transformés en influx nerveux qui sont des impulsions électriques transmises au cerveau via le nerf optique, le cerveau recevra ses impulsions et créera l'image.

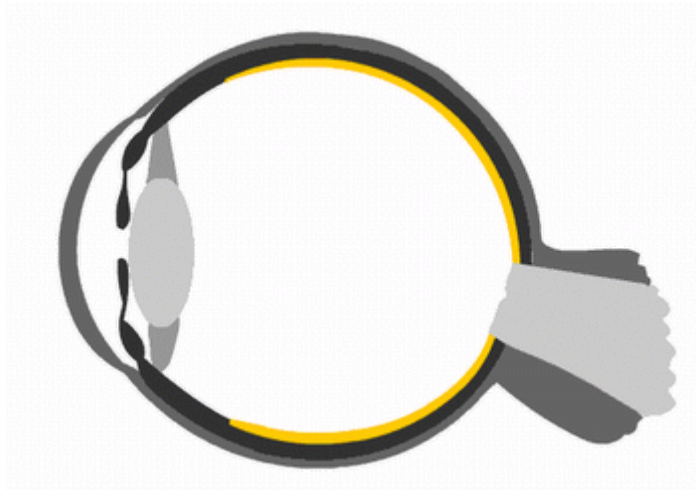


Figure II.4 : **Rétine** ou **Tunique Interne**

### 2.3.1. Zones particulières de la Rétine:

#### 2.3.1.1. Macula:

Tâche jaunâtre, région centrale de la rétine située proche de l'axe optique, c'est au niveau de cette région où la perception visuelle est la plus élevée.

#### 2.3.1.2. Fovéa:

A l'inverse du macula c'est une zone un peu plus sombre. Cette région forme une petite dépression au centre de la rétine, où l'acuité visuelle est à son maximum, et les récepteurs des cônes se trouvent au centre du champ visuel.

#### 2.3.1.3. la pupille:

Région d'émergence du nerf optique, et dépourvue de photorécepteurs, cette partie représente le diaphragme de l'œil, avec la capacité de se contracter ou se dilater selon l'intensité lumineuse.

On peut observer les points ci-dessus sur la figure suivante.

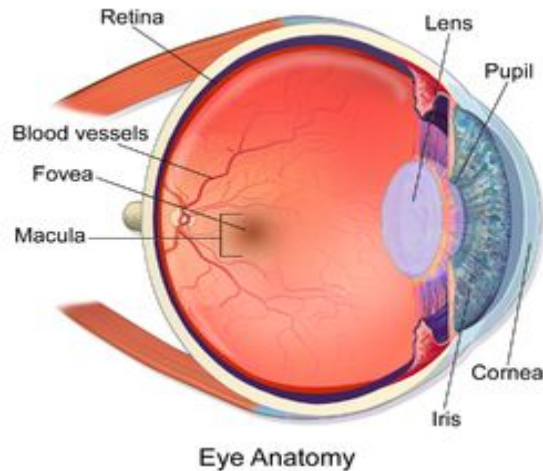


Figure II.5 : coupe transversale de l'œil

### 3. Diabète:

#### 3.1. Définition :

C'est une pathologie empêchant l'organisme d'assimiler le glucose, et cela se traduit par une insuffisance au niveau du **pancréas** ou alors le corps devient résistant à l'**insuline** produite par ce dernier.

Après chaque repas, le corps procède à la digestion des aliments préalablement consommés, et c'est au niveau de l'intestin grêle que les aliments passent dans le sang afin d'être transmis dans tout l'organisme. Durant cette phase, le niveau de **Glycémie** est élevé, cela est dû au niveau élevé du **glucose** dans le sang. On parle alors d'Hyperglycémies.

Afin de fournir les cellules en énergie (Glucose), le corps a besoin de l'insuline sécrétée au niveau du pancréas afin de passer dans le corps et alimenter les cellules, sauf que chez les personnes atteintes de Diabète ce processus ne fonctionne pas.

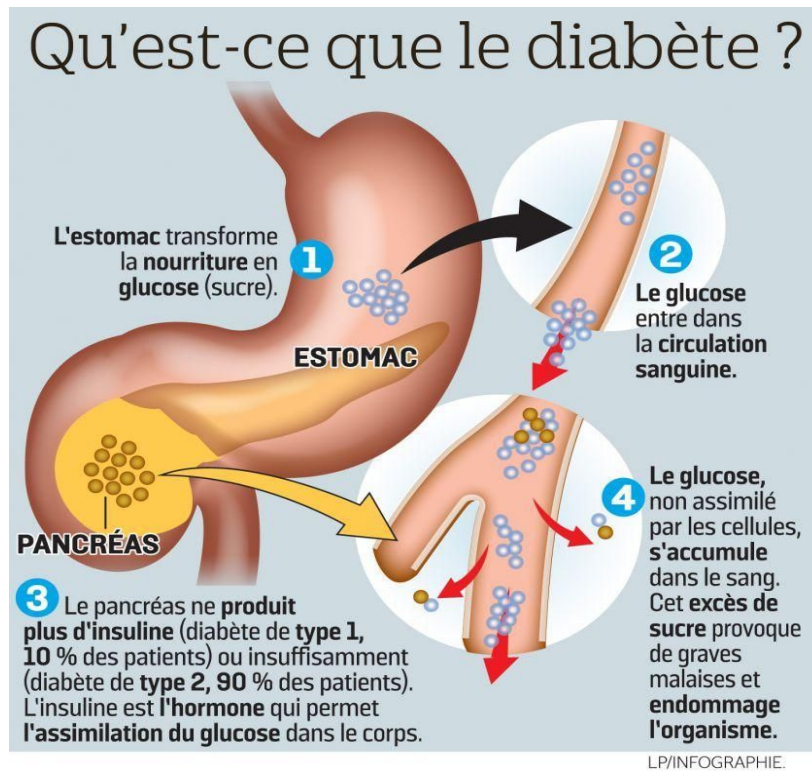


Figure II.6 Qu'est-ce que le Diabète

### 3.2. Types de Diabètes:

#### 3.2.1. Type 1:

Ce type de diabète apparaît généralement chez les personnes jeunes (enfant, adolescent, jeune adulte). Il se caractérise par la destruction des cellules Bêta, produites par le pancréas, par le système immunitaire qui ne les reconnaît plus. Il s'agit ici d'une **maladie auto-immune**.

Dans le diabète de type 1, la vérification du niveau de glycémie dans le sang se fait tout au long de la journée avec des piqûres sur les doigts.

Pour traiter ce type de Diabète, on prescrit des injections d'insuline, pompe d'insuline.

#### 3.2.2. Type 2:

Ce type de Diabète est plus difficilement décelable et cela à cause de l'absence symptômes significatifs. On note que cette pathologie apparaît souvent chez les personnes en surpoids (obèses), en manque d'activité physique régulière ou les quinquagénaires.

Les traitements conseillés sont une activité physique, alimentation saine et structurée ainsi qu'un traitement oral prescrit par un médecin.

### **3.3. Complications liées au Diabète:**

Les complications associées au Diabète ont toutes une cause commune qui est un excès de glucose dans le sang, si ce dernier est à un niveau aussi élevé au fil du temps, il a des effets néfastes sur de nombreux organes du corps, principalement:

- rénaux (Néphropathie)
- Yeux (Rétinopathie)
- Système nerveux (Neuropathie)
- Coeur (Accident cardiaque)
- vaisseaux sanguins (Hypertension artérielle, ...)

Le diabète est aussi associé à d'autres complications comme :

- apnée du sommeil
- capsulite
- dysfonction érectile
- les infections à levures
- infections des voies urinaires
- maladie parodontale

Avec ces complications le patient atteint de la maladie se trouve avec un mode vie tout à fait inédit à gérer au quotidien.

## **4. Rétinopathie Diabétique:**

### **4.1. Rétinopathie :**

Rétinopathie Diabétique est une grave complication liée au diabète qui affecte les yeux, elle est causée par des lésions des vaisseaux sanguins du tissu photosensible situé à l'arrière de l'œil, plus exactement au niveau de la tunique interne ou (rétine).

Au début de la maladie, elle peut causer de légers symptômes ou alors des troubles de la vision. A un stade avancé, elle cause une cécité.

La rétinopathie diabétique est la cause majeure de cécité chez les personnes adultes selon les statistiques.



Figure II.7: **Patient atteint d'une Rétinopathie Diabétique**

Cette Pathologie peut surgir chez les malades atteints du diabète de type 1 ou 2, mais les personnes atteintes de type 1 sont plus favorisées pour contracter la pathologie, en d'autres termes : plus on souffre de la maladie plus on est susceptible de développer cette complication oculaire liée à cette dernière.

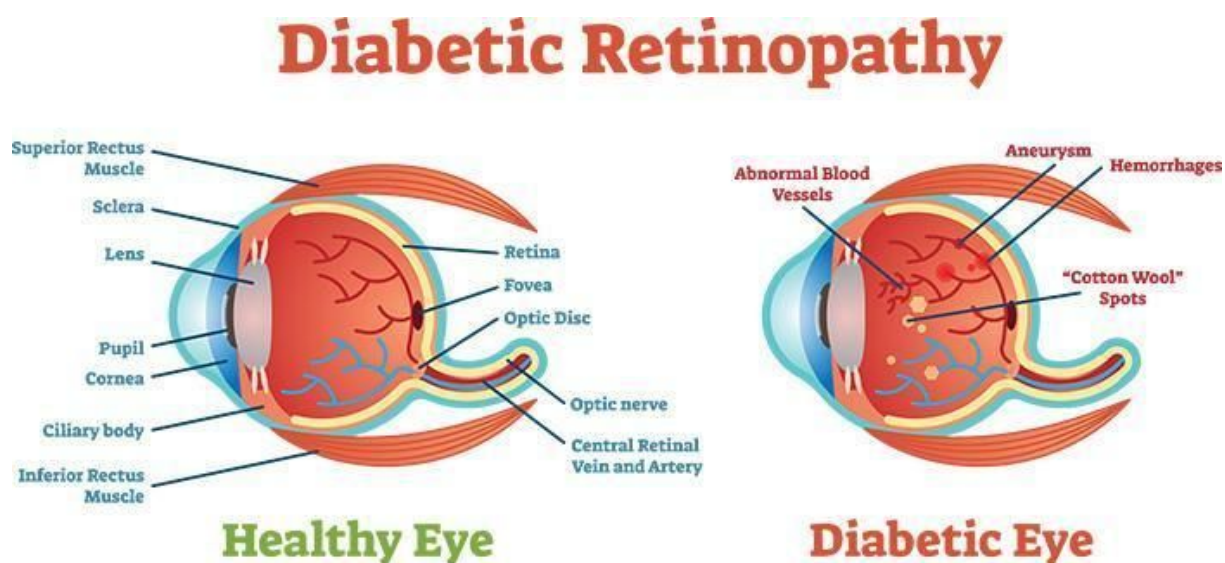


Figure II.8: **Œil sain vs Œil malade**

#### 4.2. Causes:

Le diabète est une pathologie qui se traduit un par une glycémie élevée, cette forte glycémie si elle est persistante peut être en mesure de causer des problèmes graves, comme perforer les vaisseaux sanguins. Dans le cas de la rétinopathie, les vaisseaux touchés sont ceux qui approvisionnent l'oeil.

Ils se retrouvent bouches à cause de l'excès de glucose bloquant ainsi la circulation sanguine, poussant l'oeil à développer de nouveaux vaisseaux, ces derniers se développent mal, et fuient facilement ce qui cause des microanévrismes qui sont des microgouttes de sang qui fuient des vaisseaux.

Ce processus répété du mal développement de nouveaux vaisseaux dans l'oeil qui sont souvent accompagnés de saignement conduit à une diminution de l'acuité visuelle, et à risque de la cécité. [6]

#### **4.3. Facteurs à risques:**

Les facteurs à risques augmentant la probabilité de contracter une rétinopathie diabétique sont:

**Femmes enceintes** en période de gestation, les **consommateurs de cigarettes** atteints de diabète sont à risque de développer une Rétinopathie.

Comme on remarque qu'une pression sanguine élevée porte préjudice aux vaisseaux sanguins de la rétine d'une personne atteinte d'**hypertension**.

La consommation élevée de sucre peut être un facteur déclencheur de beaucoup de complications parmi elles la rétinopathie.

La durée du diabète, cela concerne les personnes qui ont le diabète depuis plusieurs années. Ils sont les plus favorisés à contracter une rétinopathie, on remarque cela plus chez les patients atteints de diabète de type 1. [7]

#### **4.4. Examen de Dépistage:**

Afin de prévenir la rétinopathie diabétique, on effectue des tests de dépistages pour les personnes atteintes de diabète, on procède à des examens ophtalmologiques comme:

- la **mesure de l'acuité visuelle**,

- la **tension de l'oeil**,
- **angiographie** à la fluorescéine: qui consiste en l'injection d'une substance dans le bras avant de prendre des clichés de l'oeil qui facilite la détection de saignements des vaisseaux sanguins de l'oeil.
- **Fondoscopie** ou **Fond d'oeil**.

Tous les diabétiques ne bénéficient pas d'un suivi ophtalmologique annuel comme recommandé.[8]

Le principal contrôle effectué par les ophtalmologistes reste le classique "**examen de fond d'oeil**" obtenu par dilatation de la pupille afin d'observer la rétine (Figure ci-dessous)



Figure II.9: **Examen Fond d'oeil**

Le médecin ophtalmologiste cherche des signes de dilatation des vaisseaux sanguins, et de micro-anévrismes.

Similaire aux autres méthodes, mais ce derniers permet d'obtenir des images de qualité qui seront par la suite examinées par le spécialiste.

Il existe une nouvelle méthode moins contraignante qu'est le **rétinographe non mydriatique**. Il permet de récupérer une photographie numérique de l'oeil sans la dilatation de la pupille, et sans douleur, moins contraignante, et plus efficace que les testes de dépistages classiques, mais reste tout de même très peu utilisée. [9]

#### **4.5. Diagnostic:**

L'ophtalmologiste grâce aux clichés pris par le fond d'œil, il va scruter les vaisseaux sanguins de la rétine à la recherche de dilatations de ses derniers, et des micro-anévrismes (qui sont de minuscules tâches marron vers le rouge sang qui explose en provoquant des troubles de la vision).

## **5. Problématique et Direction de recherches actuelles :**

### *Détection de la rétinopathie diabétique à un stade précoce ?*

La Rétinopathie diabétique étant une complication causant des pertes de la vision, voir une cécité. Cette pathologie n'a pas cessé de se développer et la cause majeure revient au manque de spécialistes avisés pour correctement diagnostiquer la pathologie à un stade précoce.

Dans cette optique nous avons pensé à une solution automatique pouvant détecter cette pathologie à un stade précoce, en utilisant des algorithmes à base de réseaux de neurones architecture type ConvNet, dans le but d'extraire, traiter et enfin classification des clichés.

Dans ce qui suit, nous présenterons un état de l'art sur les travaux réalisés dans le cadre de cette problématique afin d'y voir plus clair.

## 6. État de l'art

### 6.1 Travaux de Tomi Kauppi et al , 2007 [10]

Tomi Kauppi et son équipe du groupe de recherche sur la vision artificiel et reconnaissance des formes de l'université de technologie de Lappeenranta en Finlande, ont travaillé sur un projet dont le thème s'intitule "**Evaluation Database and Methodology for Diabetic Retinopathy Algorithms**", en 2007 dont l'objectif était de concevoir un algorithme suivant une méthodologie médicale, qui soit capable d'évaluer une base de données de Rétinopathie diabétique, et cela en procédant ainsi : travailler avec un ophtalmologiste spécialiste du domaine qui indique que les symptômes liés à la maladie ont des dispositions spatiales au sein de l'image, on peut observer cela au sein des clichés de fond d'oeil.

Une méthode graphique a été utilisée grâce à un logiciel qui a été fournis, afin de prendre en compte les indications du spécialiste, le dataset utilisé DIARETDB0 se constitue de 130 images format JPEG dimension 1152x1500, contenant 4 classes selon le stade de la maladie dont chacune est référencée par des symptômes qui l'a distingue.

Après le traitement, les images ont été transmises à un algorithme de clustérisations (K-means), puis à un modèle type classifieur type (support vector machine/ K plus proches voisins).

Concernant l'évaluation du modèle spécificité, ainsi que la sensibilité sont les métriques qui ont été choisis, les meilleurs résultats ont été obtenus sur les 3 classes (exudates (soft and hard) haemorrhages, red small dots (microaneurysm)).

Les résultats obtenus sont: Spécificité (58,70,75,50%), sensibilité (79, 73, 92, 72%) respectivement selon les classes à traiter, Les points forts de leurs approches sont: utilisation d'une méthode automatique pour l'extraction des caractéristiques de la maladie, un logiciel graphique qui a aidé à la location des positions spatiales des symptômes. et de l'autre les points faibles de leurs approches le dataset utilisé est difficile à traiter cela causant un mauvais diagnostic médicale des spéciales, de plus les résultats obtenuq sont loins d'être suffisants pour utiliser cet algorithme à des fins médicales. [10]

Amélioration des critères d'évaluation (spécificité, sensibilité).

### 6.2 Travaux de Abràmoff MD, 2010 [14]

Abràmoff MD et son équipe du Département d'ophtalmologie et des sciences visuelles, Hôpitaux et cliniques de l'Université de l'Iowa, Iowa City, Iowa 52242, Etats-Unis, ont travaillé sur un projet '**Automated early detection of diabetic retinopathy**', en 2010 dont l'objectif est de comparer les performances de la détection automatique de la RD, en utilisant les algorithmes Challenge2009 (a remporté le concours de détection de RD en 2009), et EyeCheck qui est d'actualité, sur des dataset formés grâce à la collaboration de 16670 participants atteints de la RD. Les résultats des 2 détecteurs algorithmiques ont été appliqués séparément afin de comparer à la fin les résultats statistiques obtenus.

AUC de la courbe ROC, sensibilité, et spécificité sont les métriques sélectionnées pour cette étude. Les résultats obtenus: 0.839, 0.821 AUC EyeCheck et Challenge2009 respectivement, ainsi qu'une sensibilité 90%, spécificité 47.7%, 43.6% Challenge2009, et EyeCheck.

Cette approche n'as rien à envier aux pratiques cliniques au vue des performances, une amélioration serait de recourir à plus de données diversifiées.[14]

### **6.3 Travaux de Maria Valeria Jimenez-Baez, 2015 [15]**

Maria Valeria Jimenez-Baez et son équipe ont travaillé sur un projet '**Early diagnosis of retinopathy in primary case**', en 2015 dont l'objectif est d'évaluer l'impact d'une stratégie de détection d'une RD à un stade précoce chez des patients atteints d'un diabète de type 2 à Quintana Roo, au Mexique, en utilisant un programme de dépistage de RD à un stade précoce qui a été mis au point sur 376 169 adultes, dont 683 cas ont été diagnostiqué, un hôpital secondaire a été attribué pour déterminer le degré de RD, et un œdème maculaire a été réalisée.

Les résultats obtenus: 23.8% de RD, dont 28% RD légère (non proliférative), 48% modérées, 16% grave, enfin 8% proliférative (dernier stade). les personnes de plus de 30 ans risquent jusqu'à 2.8 fois plus de développer la pathologie, une grande majorité (55,2%) des personnes étudiées étaient des femmes.

La mise en œuvre du programme de dépistage à un stade précoce a un impact positif sur la santé du bénéficiaire.[15]

### **6.4 Travaux de Kele Xu, 2017 [12]**

Kele Xu et son équipe de l'École de l'information, et de la communication, Université nationale de technologie de défense, Wuhan, ainsi que de École d'informatique, Université

nationale de technologie de défense, Changsha 410073, Chine, ont travaillé sur un projet en commun s'intitulant "**Deep Convolutional Neural Network-Based Early Automated Detection of Diabetic Retinopathy Using Fundus Image**", en 2017 dont l'objectif principal est d'améliorer la classification des image de rétinopathie usant du deep Learning ConvNet.

Ils ont utilisé un modèle type ConvNet de leurs création (8 couches de Conv, 4 couches maxPooling, 1 couche Entièrement connectée), ainsi qu'un type de classifieur (Gradient boosting method/ ConvNet), utilisant les caractéristiques liés aux symptômes de la pathologie, un dataset récupéré de Kaggle/Messidor-1, ainsi que l'utilisation de data augmentation pour augmenter les données d'entraînement.

Les résultats obtenus sont en faveur du ConvNet avec 94.5% d'accuracy.

Le modèle effectue des prédictions très rapide au moins d'une seconde ce qui est très positif pour le domaine médicale.[12]

### **6.5 Travaux de Rishab Gargeya, 2017 [13]**

Rishab Gargeya<sup>1</sup>, Theodore Leng ont collaboré à la réalisation d'un projet dont le thème s'intitule "**Automated Identification of Diabetic Retinopathy Using Deep Learning**", en 2017 dont l'objectif de cette étude était de développer une technologie de diagnostic robuste pour automatiser le dépistage de la RD. Leur modèle est basé sur une architecture ConvNet qui a été testée sur plus 75000 clichés, dont Messidor-1 et E-Ophtha.

Ils ont utilisé comme métriques d'évaluation AUC de la courbe Précision et Rappel, spécificité ainsi que sensibilité, et les résultats obtenus sont: 97, 94, 98% AUC, sensibilité et spécificité respectivement.

Cette approche est en mesure d'être utilisée comme outil de diagnostic auquel les ophtalmologistes peuvent se référer.[13]

### **6.6 Travaux de Carson Lam, 2018 [11]**

Carson Lam et son équipe du Département d'informatique biomédicale, Université de Stanford, ainsi que Palo Alto, Californie; écoles de médecine, Université Stanford, Palo Alto, Californie; 3 Centres de recherches Ames de la NASA, Mountain View, Californie, ont travaillé sur un projet qui s'intitule "**Automated Detection of Diabetic Retinopathy using Deep Learning**", en 2018 qui vise à améliorer la reconnaissance des caractéristiques subtiles au sein du cliché médicale, afin de pallier aux problèmes liés à l'utilisation d'architecture de

ConvNet classique, en utilisant GoogleNet, et ses différentes versions 2/3/4, ainsi que des algorithmes de prétraitements qui visent à améliorer les reconnaissances des caractéristiques subtiles basées sur l'égalisation des histogrammes adaptatifs par contraste (clahe).

Pour ce faire, ils ont eu recours au dataset kaggle (35000 images jpeg qualité moyen/médiocre), ainsi qu'aux données Messidor-1 du laboratoire français (1200 image tif, qualité élevé), on comptait 5 et 4 classes respectivement par dataset selon le stage de la pathologie associée.

Ils ont aussi bénéficié du transfert d'apprentissage dû à l'utilisation de modèle d'imageNet.

Les résultats obtenus 74.5% d'accuracy sur data augmented + clahe. Concernant les critères spécificité / sensibilité pas une grande amélioration au niveau des stages précoces (1,2 classes) car GoogleNet n'a pas été en mesure (détecter les caractéristique subtiles) d'obtenir de bon résultat (Sensibilité) sur le dataset de Kaggle, dû à la **mauvaise labellisation des données, mauvaise qualité d'image**, ainsi que l'espace minime occupé par les symptômes au sein de l'image (moins 1%.)

Points forts l'étude nette amélioration de spécificité, et sensibilité en classe (normale, sévère), ainsi accuracy élevé en ses classes.

Points faibles les classes (middle, ) ,n'ont signalé aucune améliorations.

les améliorations pouvant être apportées est de passer à la détection de classe précoce, une classification multigrade plus bénéfiques, mais plus difficiles. [11]

## 7. Synthèse:

Dans le point qui a précédé (État d'art) on a évoqué certains travaux entrepris, afin de diagnostiquer ou détecter la pathologie, et cela grâce à de nombreuses approches, mais il se trouve que l'approche la plus abouti, jusqu'à maintenant réside dans l'utilisation d'algorithmes de deep learning avec les ConvNet profond, et cela se prouve tout d'abord par l'ensemble des études qui emploie des modèles inspirés d'une architecture ConvNet à partir de 2010, et à cela on ajoute les résultats qui sont plus qu'indicatif sur le potentiel de ses algorithmes, dont certains ont atteint state-of-art et sont même employés dans le domaine médicale comme outils d'aide au diagnostic[12] ou alors outils de diagnostic [15], [13].

De nos jours, l'efficacité des ConvNet n'est plus à démontrer, et leurs emplois dans le domaine de l'imagerie a permis de franchir beaucoup de barrière, qui étaient jusque là infranchissables avec anciennes méthodes.

### **Solution à proposer**

La solution que nous proposons va jumeler l'utilisation de l'architecture ConvNet tirée de Imagenet, data augmentation, Équilibrage de classes, transfert d'apprentissage, traitement de l'image avec Clache section (III. 3.2), Normalisation des données sur le dataset de Kaggle, afin de dresser un test benchmarking, et de comparer nos algorithmes avec les autres approches.

### **8. Conclusion:**

Dans ce chapitre, nous avons vu le diabète et l'une de ses complications la Rétinopathie diabétiques qui est une pathologie qui affecte la rétine des diabétiques, puis on a enchaîné avec les risques liés à la pathologie, les tests de dépistages, ainsi que la méthode de diagnostic, pour finir avec un état de l'art sur les contributions dans le domaine de la reconnaissance de la pathologie au sein de clichés médicaux et sur ce nous avons une solution à la problématique que nous aurons à développer dans les prochains chapitres.

Dans le prochain chapitre, nous allons introduire les concepts fondamentaux qui seront traités dans ce mémoire.

## **Chapitre III :**

### **Conception et Réalisation**

## 1. Introduction:

Nous allons présenter dans ce chapitre la solution que nous proposons pour répondre à la problématique qu'est la détection précoce de la rétinopathie diabétique.

Enfin, on terminera avec la partie théorique, pour aborder la partie technique dans laquelle on introduira l'environnement, langage ainsi que la plateforme utilisé, et on terminera par la présentation de l'application du système de détection.

## 2. Problématique et Contraintes:

L'ensemble du travail qui sera présenté consiste en la mise en œuvre d'un système d'aide au diagnostic médical dont le rôle est de détecter la présence de symptômes liés à la présence ou non de la pathologie **Rétinopathie diabétique**, et selon les différents stades de la pathologie d'un stade Normal (sain) au stade terminale (prolifératif) correspondant à des classes distinctes de la pathologie.

Pour le bon fonctionnement du système, la prise en compte de certains critères de performances sont nécessaires:

- un système fiable.
- un système performant à moindre coût (Rapidité du temps d'exécution).
- un système en mesure de continuer à apprendre au cours de sa vie.

Le but ici est de faire en sorte d'avoir un système qui a les meilleurs taux de performance selon les contraintes fixées (capacité de calcul et stockage limité).

## 3.2. Méthode:

L'ensemble des modèles proposés dans notre solution reposent sur une architecture ConvNet.

Notre choix s'est orienté vers les ConvNet en raison de la nature du problème (apprentissage supervisé type classification), ainsi que du type de données à traiter (Images).

On dispose d'un jeu de données de 35152 échantillons (dataset) fournis exclusivement par la plateforme kaggle, contenant 5 types de classes réparties par stade de la pathologie par un

spécialiste (ophtalmologue), la dimension de ses images varie de 2400x3400 à 2500x3550 faibles qualités. Le dataset a été découpé en 2 80% et 20% pour la phase d'entraînement et de test respectivement. Les données subiront une phase de prétraitement dans laquelle on transforme la donnée brute en données exploitable par le modèle.

Tous les modèles seront entraînés dans les mêmes conditions et nous effectuerons un changement au niveau de la fonction d'optimisation à but comparatif, enfin on pourra recueillir le modèle qui aura obtenus les meilleurs résultats afin de le comparer avec la précédente étude de 2018 [II.10].

Notre choix s'est porté dans cette étude vers Xception, InceptionResnetV2 ainsi que Densenet201 tous issu de imagenet champion de leurs catégories, que nous présenterons en détails dans ce qui suit.

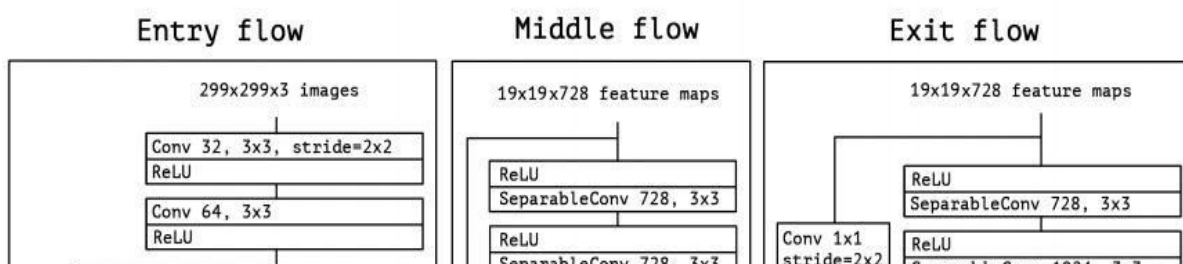
### 3.2.1. Architecture:

Le modèle utilisé dans ce projet vient de la communauté imageNet (une organisation indépendante qui organise des compétitions dans le domaine Computer Vision qui regroupent les meilleurs du domaine).

#### 3.2.1.1. Xception:

Xception [1] est une architecture de réseaux de neurones à convolution profond qui implique des **convolutions séparables en profondeur**. Il a été développé par des chercheurs de Google, qui a présenté une interprétation des **modules Inception** dans les réseaux de neurones à convolution comme une étape intermédiaire entre la convolution régulière et l'opération de convolution séparable en profondeur. Dans cette optique, une convolution séparable en profondeur peut être comprise comme un module **Inception** avec un nombre maximal de tours. Cette observation les a conduits à proposer une nouvelle architecture de réseau de neurones à convolution profonde inspirée d'Inception où les modules Inception ont été remplacés par des convolutions séparables en profondeur.

- **Aperçu de l'architecture:**



### Figure III.1 : Architecture Xception

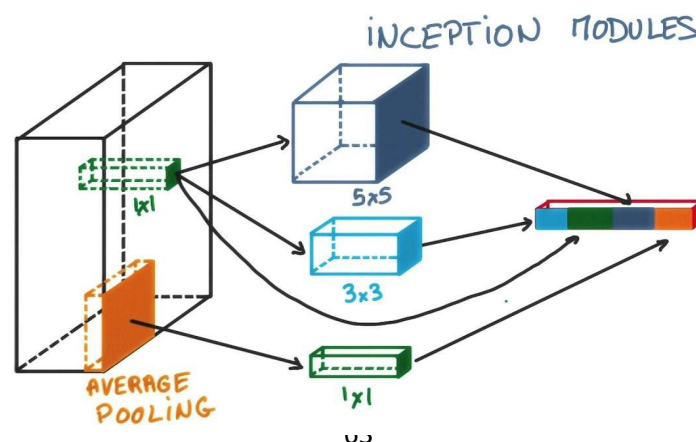
Xception est une architecture efficace qui repose sur deux points principaux:

- **Convolution séparable en profondeur:** sont des alternatives aux convolutions classiques, censées être beaucoup plus efficaces en termes de temps de calcul, en d'autre terme certains noyaux ou filtres peuvent être séparé en autant de vecteur unidirectionnel que de dimensions (axe abscisse et ordonné cas de la 2D), concernant la 3D (on rajoute les vecteurs concernant la profondeur). palliant ainsi au problème des Convolutions classiques.
- **Shortcuts between Convolution blocks:** c'est lorsque l'activation d'une couche est transférée rapidement vers une couche plus profonde du réseau de neurones on l'appel bloc résiduel ou identité.
- **Inception Module:**

Dans le module inception, on effectue une série d'opérations en parallèle au lieu d'en effectuer qu'une seule comme les ConvNet classique. Ces opérations peuvent être des convolutions avec différents formats de filtres (5x5/3x3/1x1), ainsi qu'un pooling moyen (average pooling) comme on peut le voir dans la figure ci-dessous:

**Average Pooling, 1x1 Convolution, 1x1 suivie de 5x5 Convolution**, cette combinaison peut être utilisée avec un choix de paramètres qui rendent le résultat final lié à ce module plus petit que celui calculer. Une concaténation des résultats des différentes opérations pour être envoyer vers le prochain module inception.

Sur la même perspective, on peut créer des modules d'Inception de plus en plus complexe.

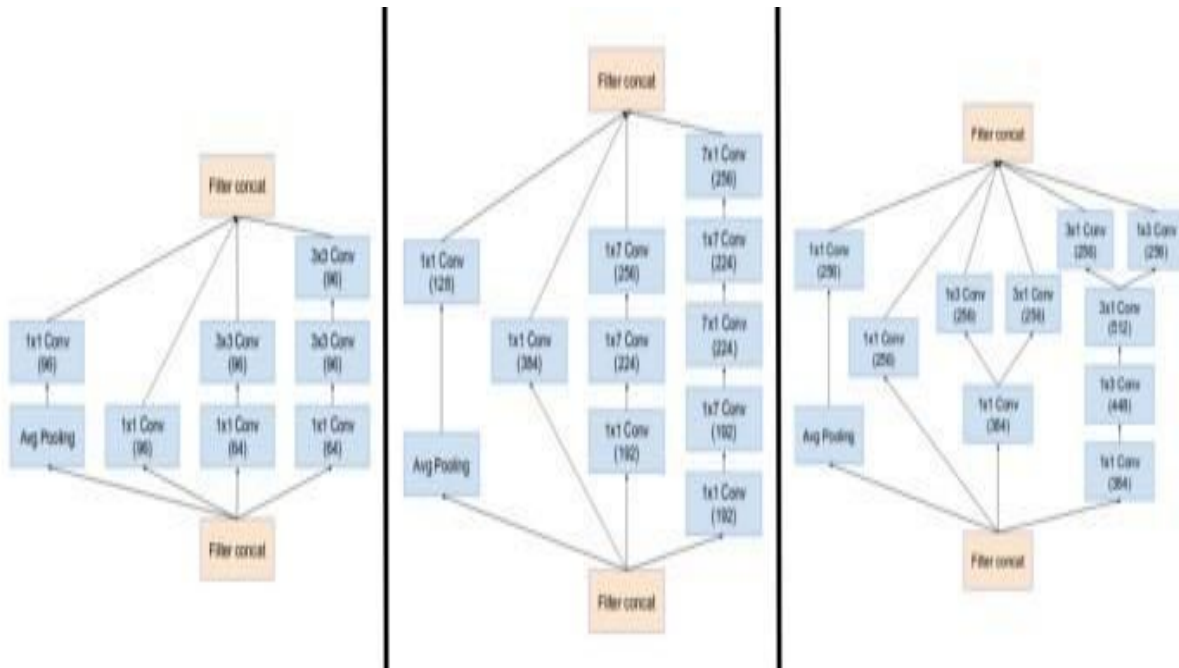


**Figure III.2: Inception Module**

**3.2.1.2. InceptionResnetV2:**

Inception-ResNet-v2 [2] est un réseau de neurones à convolution formé sur plus d'un million d'images de la base de données ImageNet. Le réseau a une profondeur de 164 couches et permet de classer les images en 1 000 catégories d'objets. En conséquence, le réseau a appris des représentations de fonctions riches pour une large gamme d'images. Le réseau a une taille d'entrée d'image de 299 sur 299.

- **InceptionResnet module** : équivalent au module **inception** décrit précédemment, il existe 3 formats différents pour ce module utilisés dans cette architecture, ces différences se caractérisent par une agention différentes des opérations au sein du module.



**Figure III.3 : Bloc InceptionResnet version A, B et C (gauche à droite)**

- **Blocs d'initiation résiduels**

Chaque bloc Inception est suivi d'une couche d'expansion de filtre. (Convolution  $1 \times 1$  sans activation) qui est utilisé pour augmenter la dimensionnalité de la banque de filtres avant l'ajout afin de l'adapter à la profondeur de l'entrée.

Dans le cas d'Inception-ResNet, la normalisation par lots est utilisée uniquement par-dessus les couches traditionnelles, mais pas par-dessus les sommations.

- **Mise à l'échelle des résidus:**

Dans le cas où, le nombre de filtres dépassait 1 000, les variantes résiduelles commençaient à présenter des instabilités, ce qui signifie que la dernière couche avant le pooling moyen ne commençait à produire que des zéros après quelques minutes. Des dizaines de milliers d'itérations. Cela n'a pu être évité, ni en abaissant le taux d'apprentissage, ni en ajoutant une anomalie batch supplémentaire à cette couche.

L'opération de mise à l'échelle des résidus avant de les ajouter à l'activation de la couche précédente semblait stabiliser le problème lors de l'entraînement.

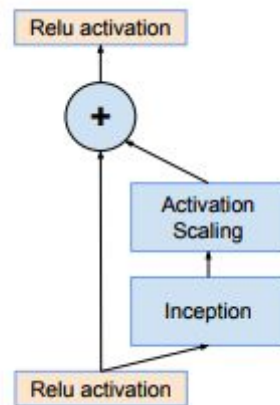
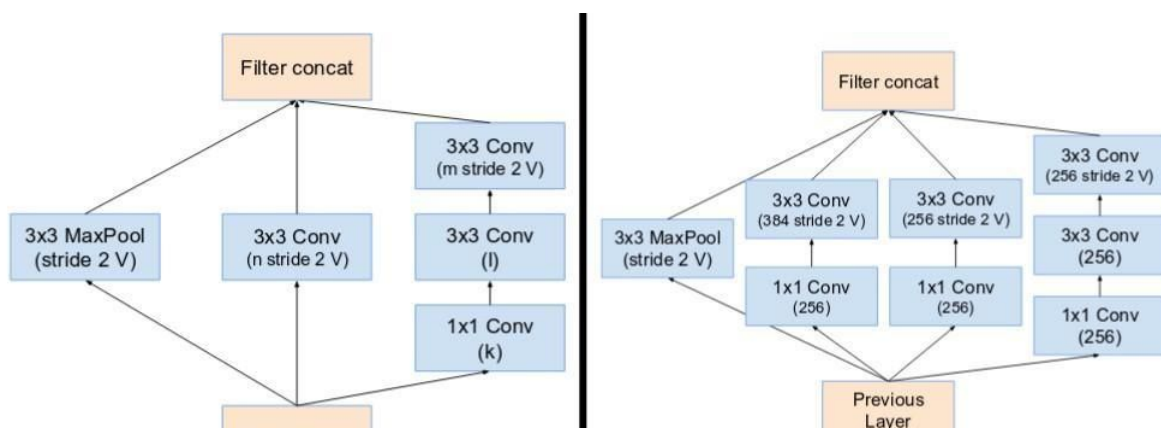


Figure III.3: Mise à l'échelle des résidus

- **Bloc de Réduction:**

La mise en commun à l'intérieur des modules de démarrage principaux a été remplacée au profit des connexions résiduelles afin de procéder à la réduction du bloc, qui va s'occuper de réduire le volume les résultats provenant des blocs d'inceptionResnet.

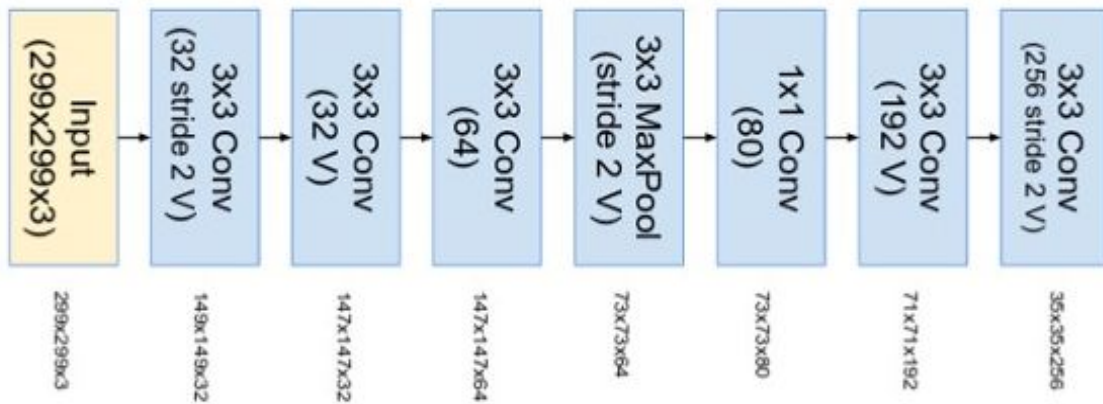
On trouve 2 formats possibles pour le bloc de réduction dont la majeure différence réside dans l'agention des opérations.



**Figure III.4 : Réduction Bloc A et Bloc B (de gauche à droite)**

- **Stem :**

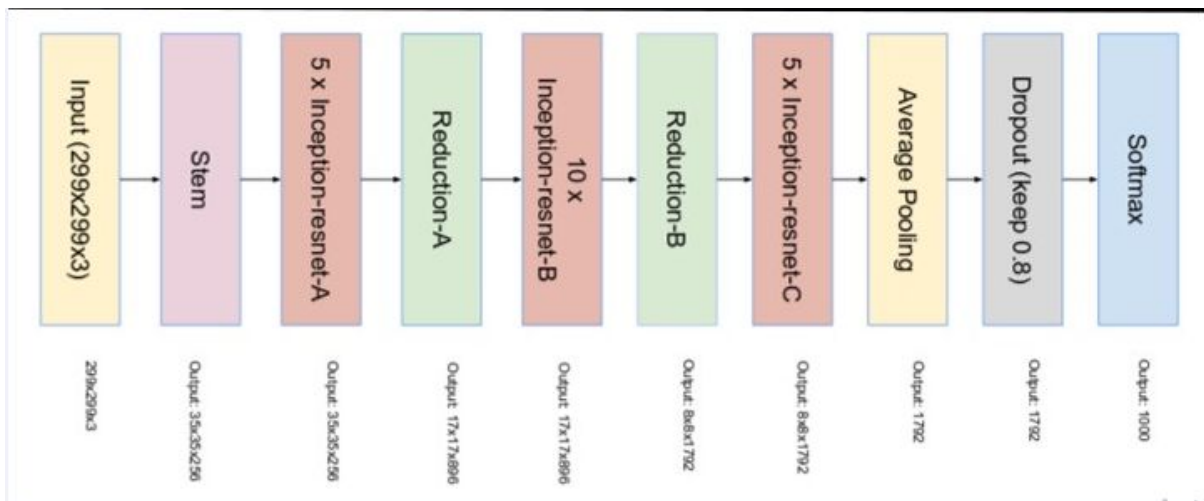
C'est le premier bloc d'opération de cette architecture, il se compose d'une série d'opérations (des convolutions différent format de filtres, ainsi que pooling) séquentielles.



**Figure III.5 : Bloc Stem**

- **Architecture Globale d'InceptionResnetV2:**

La structure simplifiée du réseau pour InceptionResnetV2 est la suivante:



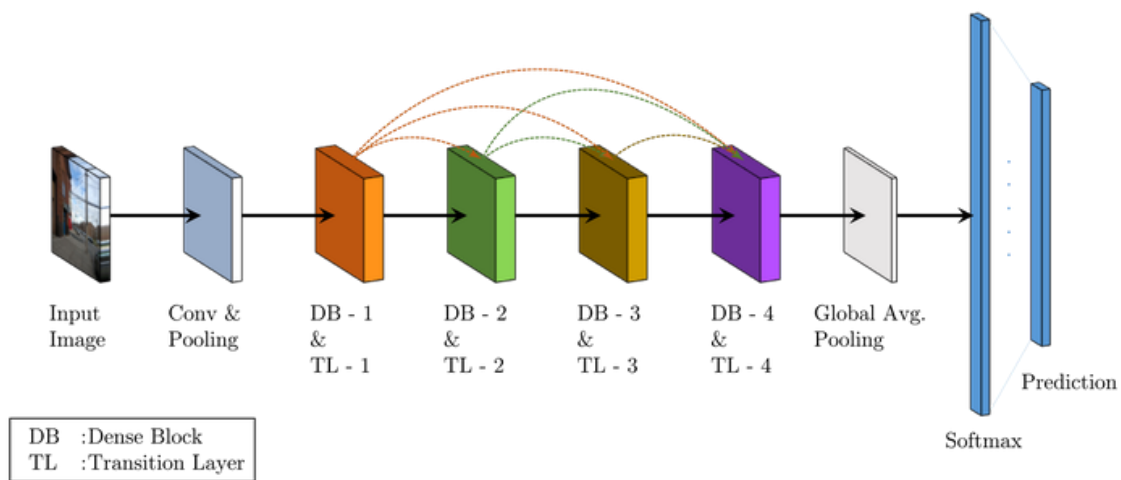
**Figure III.5 : Structure InceptionResnetV2**

### 3.2.1.3. DensetNet:

DenseNet [3] est une nouvelle architecture ConvNet (d'imagenet) qui a obtenu des résultats élevés sur des datasets de classifications (CIFAR, SVHN, ImageNet) en utilisant moins de paramètres que ses prédécesseurs (autres modèles d'imagenet).

- **Architecture DenseNet:**

DenseNet est composé de **blocs denses**. Dans ces blocs, les couches sont étroitement liées entre elles: chaque couche reçoit en entrée toutes les cartes d'entités en sortie des couches précédentes.



**Figure III.6:Example Architecture DenseNet**

- **Bloc Dense:**

DenseNet est composé de blocs denses. Dans ces blocs, les couches sont étroitement liées entre elles: chaque couche reçoit en entrée tous les résultats des convolutions des couches précédentes. L'utilisation extrême des résidus est un mécanisme utilisé dans DenseNet afin de connecter les couches d'un même bloc avec celles des autres blocs suivants. Les couches créent une supervision profonde car chaque couche reçoit une supervision accrue de la fonction de perte grâce aux connexions plus courtes.

On retrouve au sein d'une couche d'un bloc dense les opérations suivantes :

- **Normalisation des lots :** C'est une technique qui permet d'accélérer la convergence d'un modèle et améliorer sa performance.
- **Activation ReLU**
- **Convolution 3x3**

- **Couche de transition:**

Une couche de transition est utilisée pour contrôler la complexité du modèle. Il réduit le nombre de canaux en utilisant la couche convolutionnelle  $1 \times 1$  et divise par deux la hauteur et la largeur de la couche de pooling moyen avec un pas de 2, réduisant ainsi la complexité du modèle. Elle assure aussi une fonction de compression qui aide à améliorer la compacité du modèle.

Une couche de transition est composée de:

- **Normalisation des lots.**
- **Convolution  $1 \times 1$ .**
- **Average pooling.**

- **Bloc BottleNeck:**

Bottleneck est l'une des raisons pour laquelle DenseNet a peu de paramètres malgré la concaténation de nombreux résidus.

Un bloc dense avec bottleneck peut être vu comme :

- **Normalisation des lots**
- **ReLU activation**
- **$1 \times 1$  Convolution bottleneck produisant  $grow\ rate * 4$ . feature maps.**
- **Normalisation des lots**
- **ReLU activation**
- **$3 \times 3$  Convolution**

**Remarque :** Le BottleNeck aide le réseau à disposer de centaines, voire de milliers de couches.

#### **3.2.1.4. Modifications apporté aux modèles :**

Tous les modèles ont subi une transformation au niveau de leur architecture qui se présente comme suit:

**Couche entrée:** ajout de:

- couche entrée (299, 299, 3) pour Xception /InceptionResnetV2.
- couche entrée (224, 224, 3) pour DenseNet201.

**Couche Sortie:** ajout de:

- une couche dense (entièrement connecté de 256 neurones) + fonction activation Relu.
- couche dropout (0.4).
- fonction de perte (categorical cross entropie) calcul de perte avec 3 catégories (pour nos 3 classes).
- couche softmax.

### 3.2.2. Dataset:

Le Dataset (jeux de données) utilisé pour ce projet a été fourni par Kaggle. Kaggle est une communauté regroupant des data scientiste tout niveau confondu, qu'il leurs permet de publier leurs travaux. Kaggle organise aussi des compétitions proposées par des entreprises ou particuliers dont les vainqueurs seront payés.

Notre dataset comprend 35152 images de qualité moyenne, il a été découpé en 12 fichiers zip en raison de la taille volumineuse des données, et accompagné d'un fichier csv contenant les détails sur les données (stade de la pathologie, id patient, info sur l'œil).

Ses images de fond d'œil ont été vérifiées par un médecin ophtalmologiste qui les a classés en 5 catégories (classes), dont chaque classe représente un stade de la maladie:

- 0 - No DR (sain)
- 1 - Mild
- 2 - Moderate (**pré-proliférative**)
- 3 - Severe (sévère)
- 4 - Proliferative DR (stade terminale **Proliférative**)

Avant de fournir le dataset toutes les informations personnelles liées au patient ont été supprimé.

Dans notre étude on a été en mesure d'utiliser que 1134 échantillons de données (en raison des moyens), fragmentées en 3 classes:

- Classe 1: No DR
- Classe 2: regroupe les 2 classes (Mild et Moderate)
- Classe 3: regroupe les 2 classes (Severe et Proliferate DR)

La raison du regroupement est principalement due à la similitude (Symptômes) existante entre les classes regroupées, ainsi qu'au manque de données issu de chaque classe, et mauvaise fiabilité des données du dataset Kaggle.

Le dataset a été fragmenté proportionnellement à 80% et 20% jeu d'entraînement et jeu de test respectivement, avec une disposition aléatoire des catégories au sein du jeu de d'entraînement et de test.

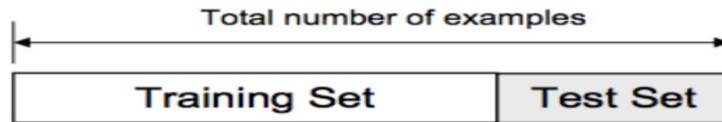


Figure III.3: Fragmenter Dataset

### 3.2.3. Représentation du système:

Notre système de détection de rétinopathie contient les étapes suivantes:

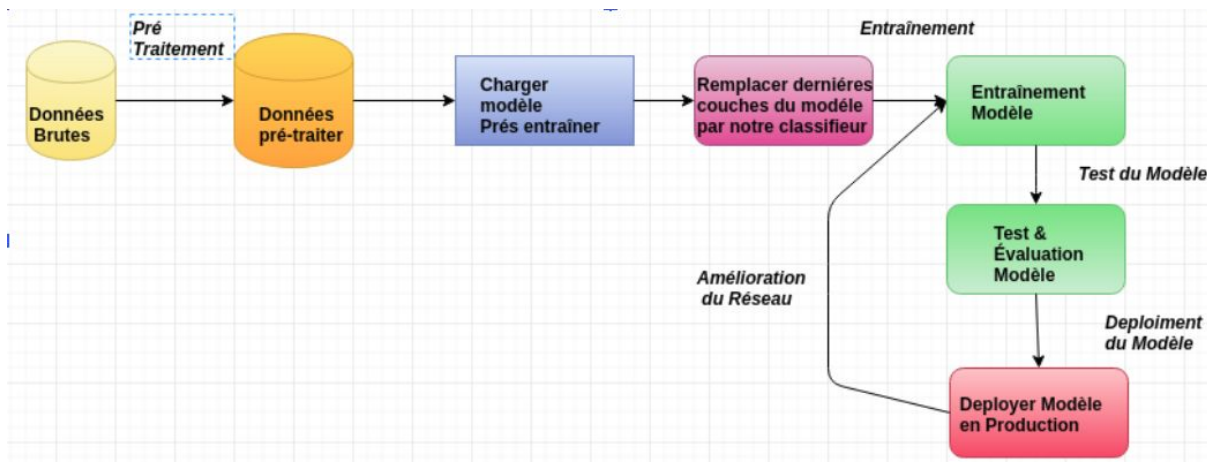


Figure III.8: Diagramme du Système de détection

### 3.2.4. Prétraitement:

Le prétraitement est l'étape durant laquelle les données qui sont à l'état brute subissent un traitement afin de les rendre exploitables par le modèle.

Dans notre cas, nos images sont des clichés de fond scopie, les traitements appliqués dessus sont :

- Changement du format de données en tensor (matrice à dimension supérieur à 3).
- On a effectué un **Under-sampling (Réduction d'échantillon)** sur nos données, afin de palier au problème de déséquilibre au niveau des classes rapportant toutes les classes au nombre d'échantillon de la classe la plus faible (classe 5).
- Regroupement des classes avec faible nombre d'échantillon selon cette dispositions :

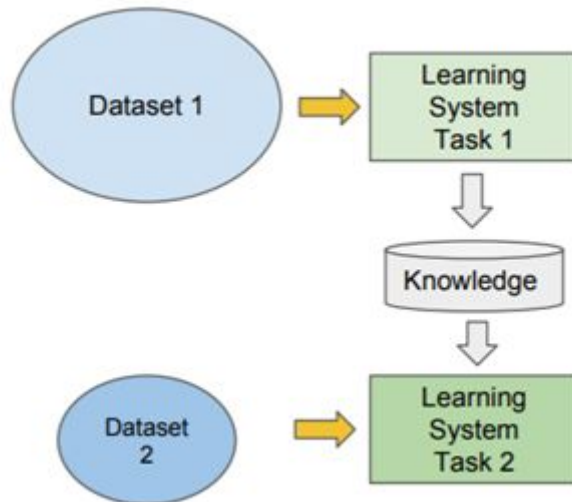
Classe 0 : Non malade (classe0), classe 1 : Middle (classe 1 et 2), classe 2 : Sévère (classe 3 et 4).

- Augmentation du nombre de données disponibles par classe de façon homogène, (Data augmentation)
- Normalisation des images en soustrayant l'intensité minimale de pixel de chaque canal, et en la divisant par l'intensité moyenne de pixel pour représenter les pixels dans une plage de 0 à 1.
- Traitement des images par un **algorithme Clahe** sur des Image en RGB (Contrast-Limited Adaptive Histogram Equalization) dont le rôle est d'améliorer la qualité de l'image. **L'image** est tout d'abord **divisée** en **régions rectangulaires** sous la forme d'une grille, et l'**égalisation d'histogramme standard** sont appliquées à **chaque région**. Après toutes ces **régions** sont **combinées**, afin d'obtenir une **image complète optimisée**.
- Redimensionnement des images sous un format acceptable par les modèles
  - (229, 299, 3) Xception / InceptionResnetV2.
  - (224, 224, 3) DenseNet201.

### 3.2.4. Transfert Learning (transfert d'apprentissage):

On a également utilisé le transfert d'apprentissage (le Transfert Learning est un problème de recherche en Machine Learning qui se focalise sur la sauvegarde de connaissances apprises par un modèle sur un problème donné, puis la réutilisation de ses connaissances sur un autre problème similaire. Il a pour but d'améliorer les performances d'un modèle.

Il existe différentes manières ou techniques pour effectuer un transfert d'apprentissage. Dans notre cas, chaque modèle (Xception / InceptionResnetV2/ Denset201) est pré-entraîné sur un dataset d'imageNet (Tâche détection d'objet), puis on réutilise ce dernier avec les paramètres d'apprentissage déjà près sur notre problème (Tâche détection de la pathologie).



**Figure III.5: Schématisation du Transfert d'apprentissage**

### 3.2.5. Data augmentation:

Data augmentation est une technique très utilisée dont le rôle comme son nom l'indique est de créer de la donnée, et cela pour de nombreuses raisons:

- le modèle nécessite plus de données pour être entraîné, on risquerait un problème de Sur ou Sous-apprentissage (tout dépendra du comportement du modèle).
- le dataset est déséquilibré, ce qui se traduit par un déséquilibre au sein des classes ayant un nombre d'échantillons disproportionnés (ce qui rend la tâche d'évaluation plus complexe).

Pour augmenter nos données, on a généré des données à partir des originaux, avec les spécifications présentes:

- Transposition de l'image.
- Rotation de l'image de différents angles (120°, 72°, 45°).

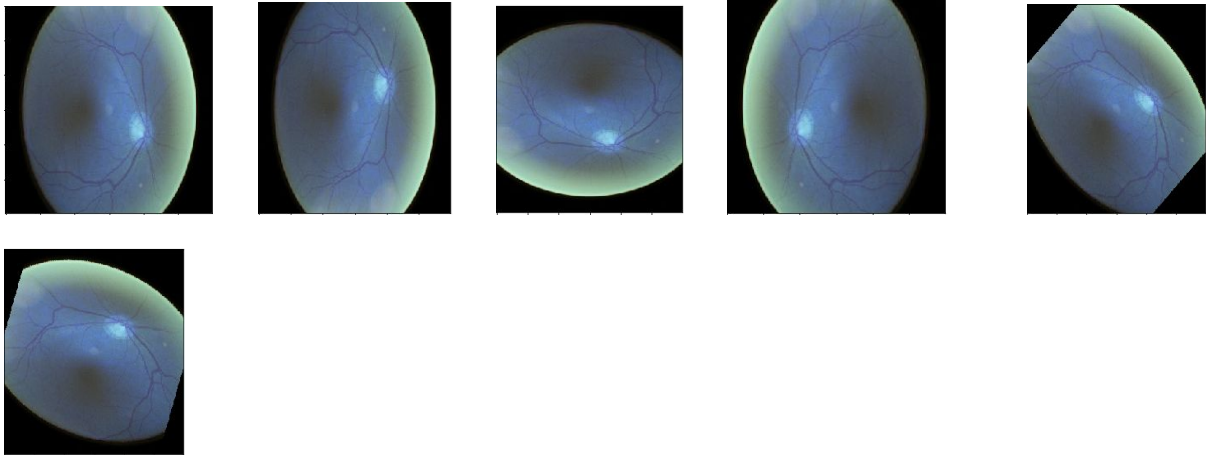


Figure III.6: Exemple de Data augmentation

### 3.2.6. Couche Dropout:

Les modèles contiennent au sein de leur architecture une couche dropout.

- **Dropout:** c'est une technique qui a pour but d'abandonner des unités (à la fois cachés ou visibles) dans un réseau de neurones, ces neurones seront choisis au hasard grâce à une probabilité  $1-p$ , soit conservés avec la probabilité  $p$  de sorte qu'il reste à la fin réseau réduit, et par abandon on entend par là que ces neurones ne seront pas pris en compte durant le passage en avant ou en arrière pour éviter le sur-apprentissage.

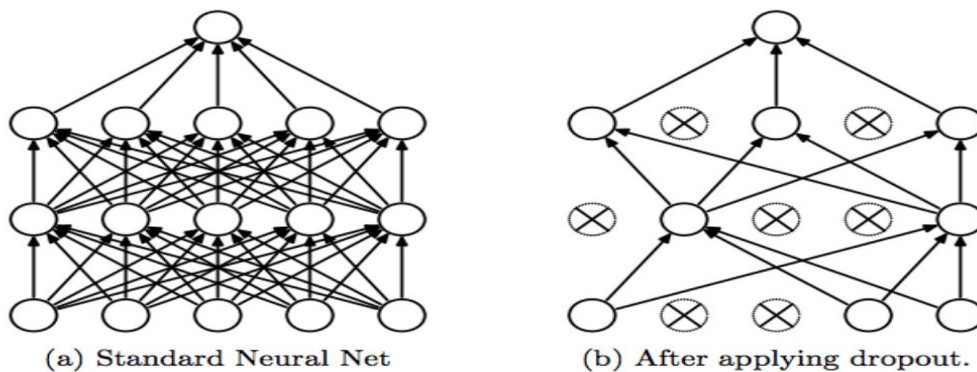


Figure III.7: Réseau standard vs Réseau dropout appliqué

### 3.3. Réalisation et Implémentation:

Dans cette partie nous allons présenter les outils ainsi que technologies (Langage de programmation, framework, interface, bibliothèque) utilisés afin de réaliser notre système, et enfin présenter les différentes interfaces de notre application.

### 3.3.1. Outils utilisés:

Notre choix s'est orienté vers le langage de programmation Python, en utilisant le framework de haut niveau pour le Deep Learning Keras, ainsi que des bibliothèques pour la visualisation, mathématique et exploration de données (Matplotlib, Numpy, Pandas, sklearn) sous python.

### 3.3.2. Python:

Python est un langage interprété de haut niveau, c'est aussi un langage permettant le multi paradigme, ce qui signifie qu'on peut utiliser plusieurs paradigmes au sein même du langage dont l'**impérative** (procédurale, orienté objet), et **déclarative** (Fonctionnel).

Python a connu ses dernières années une hausse fulgurante au seins des sondages des langages les plus utilisés du moment, et l'une des principales raisons est son utilisation par la majorité de la communauté des développeurs pour le machine Learning et le deep Learning.[2]

Python est simple d'utilisation et rapide il offre une large palette de bibliothèques pour le traitement, l'exploration, la visualisation des données, ainsi que l'entraînement et l'évaluation des modèles.

Python dispose d'une communauté forte et active qui le soutient, ainsi que l'existence de certaines plateformes OpenSource (github, stackexchange, ...)



### Figure III.1: Matplotlib, Scipy, et numpy

#### 3.3.3. Keras:

Keras est une bibliothèque open source écrite en Python pour le deep learning. Elle est capable de fonctionner en haut niveau de certains Framework tel que TensorFlow , Microsoft Cognitive Toolkit, Theano ou Plaid ML.

Conçu pour permettre une expérimentation rapide avec des réseaux de neurones profonds, Keras se veut simple d'utilisation, intuitive, et rapide.



Figure III.2: Keras

#### 3.3.3. Modèle sous Keras:

Keras est aussi appelé API de haut niveau, elle permet d'effectuer une classification suivant les étapes suivantes:

1. Importer les modules et dépendances
2. Préparation des données en format acceptable par l'API.
3. Construire le modèle en utilisant tf.keras (Tensorflow-Keras), puis le compiler.
4. Entraîner le modèle avec un Dataset préalablement préparé.
5. Evaluation du modèle.
6. Sauvegarder le modèle prêt à être utilisé en production.

#### 3.3.4. Keras GPU vs CPU:

Enfin Keras dispose du choix de la façon dont on veut entraîner le modèle CPU ou GPU, Contrairement au CPU dont le rôle est d'effectuer des calculs de façon séquentielle, et ne peut paralléliser qu'un nombre réduit de calculs selon le nombre de cœurs dont il dispose, le GPU,

lui est en mesure d'exécuter un nombre important de calculs en parallèle, ce qui fait du GPU un matériel essentiel pour des tâches nécessitant beaucoup de calculs en parallèle.

Pour finir, Keras a la possibilité d'exécuter Tensorflow en haut niveau, il a la possibilité d'utiliser les TPU (Tensor Processor Unit) dont le rôle principal de ce matériel est d'effectuer des calculs parallèle pour des tâches de machine learning (uniquement accessible sur le google plateforme cloud).

### **3.3.5. Cloud:**

Pour mener à bien le projet du moins la partie entraînement et évaluation du modèle nécessaire au système, on a dû utiliser un service Cloud gratuit offert par Kaggle / Google Colabs.

Cloud offre accès à un service cloud Paas (plateforme as a Service) dont l'ensemble des bibliothèques sous python nécessaires sont déjà installées.

La configuration de la machine distante est :

- Linux 64bits.
- 1 CPU intel Xeon(R) 2.20Ghz.
- RAM 12 Go DD.
- GPU K Tesla Free.
- Computing Environment: Python / Keras / Tensorflow.

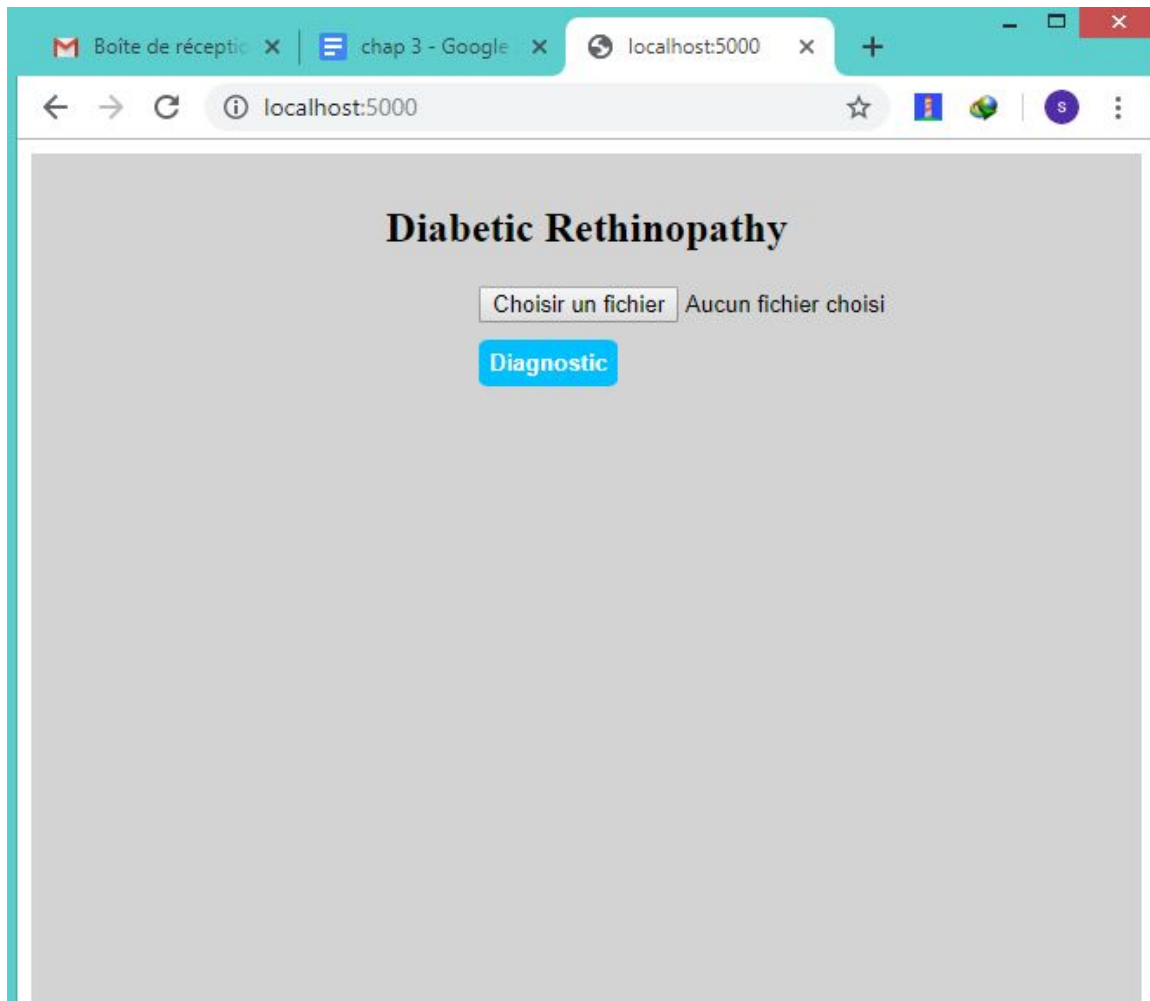
## **4. Interface de l'application :**

C'est une application web 2 tiers dont la partie s'occupe du traitement des clichés envoyer par le client, afin d'emmètre un diagnostic sans effectuer de sauvegarde des clichés médicales, cette application tourne sous Flask qui est un micro Framework écrit sous python qui gérer le côté server de notre application, elle se compose principalement de 2 interfaces :

- Interface soumission du cliché médicale (image fond d'œil)
- Interface résultats dans laquelle on va afficher les résultats lié à la prédiction effectué sur le cliché soumis.

### **4.1. Interface Subissions :**

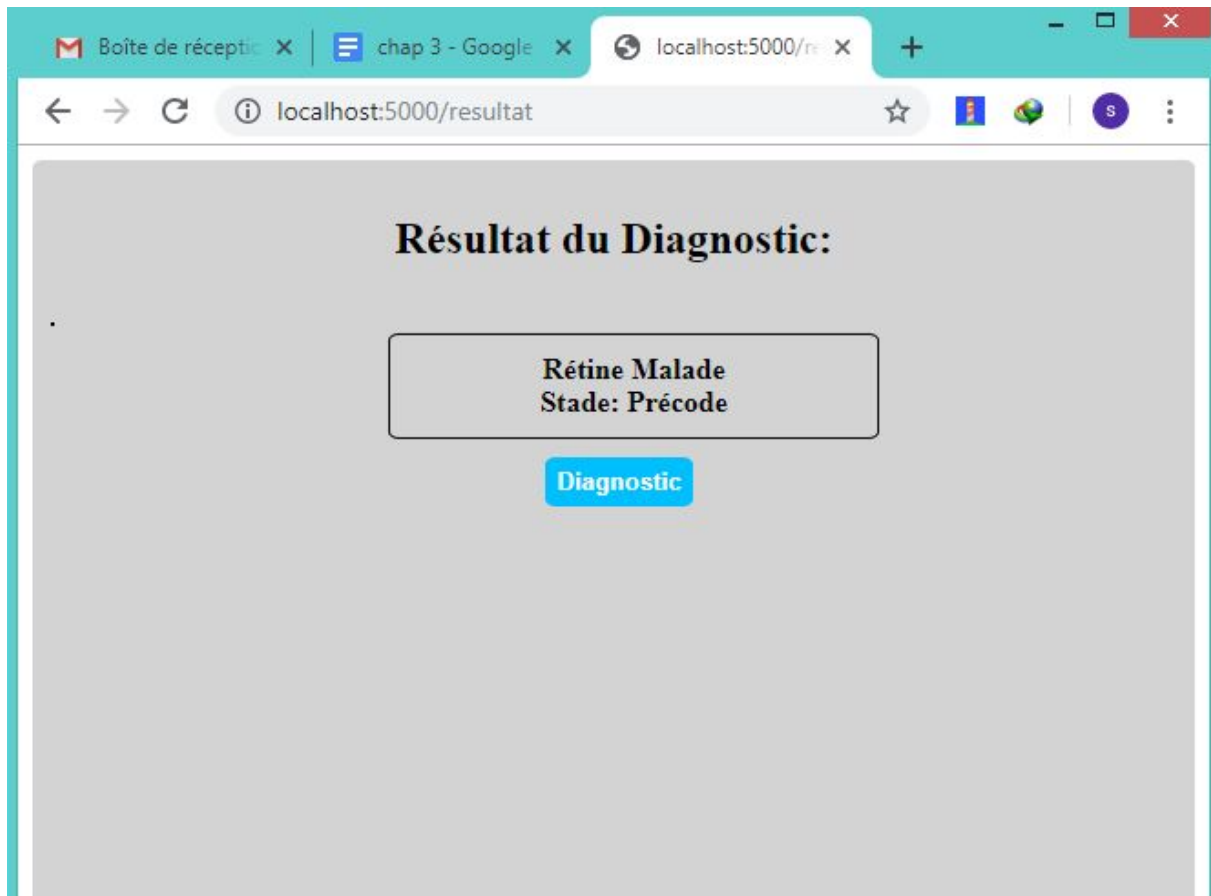
Ici on dispose de 2 boutons, 1 pour sélectionner le cliché à soumettre (Choisir un fichier), et l'autre (Diagnostic) pour effectuer la soumission du cliché, ainsi que le diagnostic.



**Figure III. : Interface Soumission**

#### **4.2. Interface Résultats :**

Ici on a directement les résultats liés à notre prédiction sur le cliché, on dispose de deux principales informations la 1<sup>ère</sup> Nature du cliché (Sain ou malade), la 2<sup>ème</sup> le stade de la pathologie dans le cas où la nature du cliché serait malade.



**Figure III. : Interface Résultats**

## **5. Conclusion:**

À travers ce chapitre, nous avons vu les différentes étapes de conception et de réalisation de notre système de détection de rétinopathie à un stade précoce, ainsi que la présentation des différents modules le constituant, enfin présentation des interfaces de l'application utilisant le système.

Dans le prochain chapitre on entamera la dernière partie de ce mémoire consacré aux résultats, suivit d'une série de tests d'évaluation.

## **Chapitre IV :**

### **Test et Évaluation des Résultats**

## 1. Introduction:

Dans ce chapitre, nous allons présenter les résultats de notre approche basée sur l'utilisation des réseaux convolutionnels avec l'architecture proposée par Imagenet Xception, DenseNet201 et InceptionResnetV2 dans le but de détecter la présence d'une Rétinopathie.

Nous commencerons par exposer les résultats, suivis de l'évaluation des résultats selon les critères de performance liés au problème, et enfin nous terminerons par une comparaison des résultats de notre approche avec celle des travaux préalablement effectués dans le domaine.

## 2. Choix des Critères d'évaluation :

En raison, du domaine dans lequel on souhaite développer notre modèle, on prend en compte des métriques qui respectent au mieux les contraintes liées au domaine médical, nos choix se sont portés sur la **Sensibilité**, et la **Spécificité**.

Nous avons soumis nos modèles à une série de tests dans le but d'évaluer les performances du système global, pour comparer les résultats, et pour ce faire on a relevé à chaque fois les mesures suivantes:

1. **Sensibilité:** ou le **Rappel** représente les taux de vrais positifs (TPR), ce qui signifie la proportion des gens qui ont été prédit positifs au test.

En d'autres termes, si on prend une population de patients qui ont passé le test, la sensibilité fait référence à la proportion de personnes atteintes par la maladie, et qui ont été prédites correctement.

2. **Spécificité:** représente le taux de faux positif (FNR), ce qui signifie la proportion des gens qui ont été prédit négatifs au test.

La spécificité est une mesure qui vise à calculer la proportion de personnes qui ne souffrent pas de la maladie, et qui ont été prédites comme telles.

La prise en compte de ses 2 métriques, ainsi que le bon choix du seuil (c'est une valeur à partir de laquelle on considère les résultats positifs, on fixe soi-même la valeur) permettra de maximiser les valeurs de spécificités, et de sensibilités.

Cette étape s'avère cruciale, car elle nous permettra d'éviter certains cas qui rendraient notre modèle non fiable au diagnostic médical.

- **Cas 1 (Spécificité faible):** Prédire des patients sains malades.
- **Cas 2 (Sensibilité faible):** Prédire des patients malades sains.
- **Cas 3 (Sensibilité et spécificité faibles):** Prédiction erronée, le modèle aurait une valeur de prédiction en dessous d'un modèle aléatoire.

Le meilleurs des cas serait d'avoir une spécificité, ainsi qu'une sensibilité au maximum, mais cela s'avère être une tâche complexe car il existe un compromis entre ses deux métriques, pousser une métrique au max détériorerait l'autre métrique et vice versa. Pour ce faire, il y a un moyen qui nous permettra d'aller contre ce compromis, et cela en **compromettant la spécificité, en faveur de la sensibilité.**

### **Ex: Cas pratique**

Dans un cas pratique, il serait plus judicieux d'avoir un patient dit sain prédit comme malade que l'inverse, car on pourrait toujours lui faire passer des tests complémentaires confirmant ou non la véracité de notre diagnostic.

Prédire le contraire serait désastreux surtout dans un domaine aussi critique que le domaine médical.

**3. ROC / AUC:** une technique graphique utilisée pour visualiser la performance de notre modèle. La courbe ROC prend les 2 métriques **sensibilités** et l'**anti-spécificité (1-spécificité)**. Cette courbe est d'ailleurs très répondu quand il s'agit de traiter des cas dans lesquels les métriques citées précédemment sont incluses, elle nous donne des informations sur le comportement du modèle (la hausse ou la basse des deux métriques, AUC, performance du modèle selon tous les seuils possibles) . L'AUC ou l'aire sous la courbe ROC est la valeur la plus importante qui permet d'estimer la performance d'un modèle à travers la courbe Roc .

Sauf que cette courbe reste correcte du point de vue de ses résultats lorsque nous avons un **dataset équilibré** (un dataset est dit équilibré lorsque le nombre d'échantillon répartis sur

les différentes classes est le même ou plus ou moins proche) ce qui est notre cas, dans le cas contraire la courbe nous donne pas d'informations complémentaires par rapport à ça.

## 2. Etude comparative entre modèles Xception vs DenseNet201 vs InceptionResnetV2:

Dans cette partie on présentera une comparaison entre la classification basée sur différents modèles, ainsi que l'influence du choix des **hyper-paramètres** (cela désigne une série de paramètres qui sont mis avant le processus d'entraînement, le choix de ses derniers à une importance considérable sur les performances du modèle).

### Configuration des paramètres du Système:

- **Paramètres fixes du système:** Nombre de couches, Nombre de neurones de chaque couche, **Données d'entraînement (80%)**, ainsi que **données de test (20%)**.
- **Paramètres variables du système:** Résolution de l'image en entrée, Couche Dropout, algorithme d'optimisation, taux d'apprentissage, nombre d'itérations (**epoch**), Taille du lot de données (**Batch size**) pour **entraînement** et **test**.

### 2.1. Evaluation de Xception:

On a sélectionné une configuration concernant le système basé sur le modèle **Xception**.

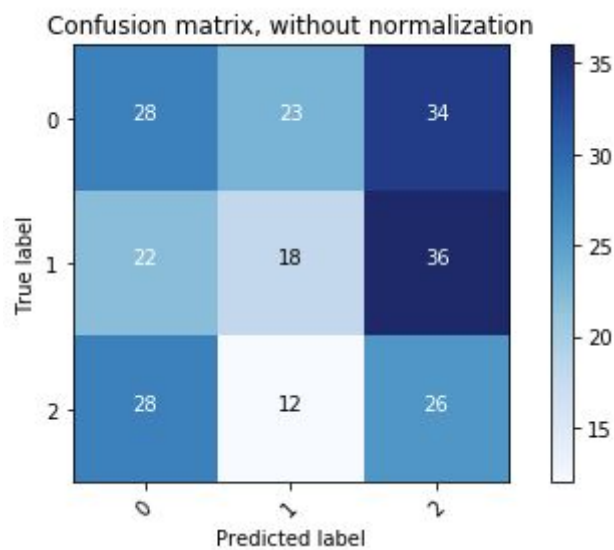
Nom du Modèle	Taux d'apprentissage	Couche Dropout	Résolution d'entrée du modèle	Algorithme d'Optimisation
<b>Xception, InceptionResnetV2, DenseNet (a)</b>	<b>0.001</b>	<b>0.4</b>	<b>(299, 299, 3), (299, 299, 3), (224, 224, 3)</b>	<b>Adam</b>
<b>Xception, InceptionResnetV2, DenseNet (b)</b>	<b>0.001</b>	<b>0.4</b>	<b>(299, 299, 3), (299, 299, 3), (224, 224, 3)</b>	<b>Adagrad</b>

<b>Xception, InceptionResnetV2, DenseNet (c)</b>	<b>0.001</b>	<b>0.4</b>	<b>(299, 299, 3), (299, 299, 3), (224, 224, 3)</b>	<b>Stochastic Gradient Descent (sgd)</b>
--	--------------	------------	--	--

**Tableau VI.1: Configuration des Modèles Xception, InceptionResnetv2 et DenseNet201**

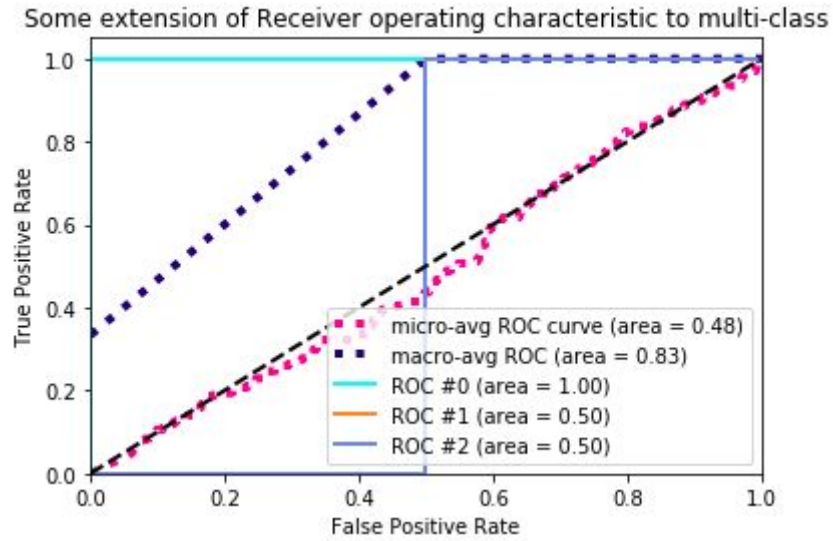
Concernant le Nombre d'itération, il a été fixé à 100 (**epochs=100**), avec un **batch\_size** de 32 pour la phase l'entraînement et la phase de test.

**Résultats obtenus:**



**Figure VI.1: Matrice de Confusion avec Cas a**

Matrice de confusion montre les résultats des prédictions du modèle sur les données de test, on remarque que les taux prédictions valide (diagonale inverse) est inférieur à celui des prédictions erronées (reste de la matrice).



**Figure VI.2: Courbe ROC /AUC pour les 3 classes (Cas a)**

Pour les classes 1 et 2 les résultats de l'AUC indique 0.5 ce qui signifie que le modèle a un comportement aléatoire, concernant la classe 0 AUC = 1 signifie que le modèle est dans un cas idéale pour prédire la classe 0

```

Metrics for classification: #0
F-measure 0.34
Precision 0.36
Recall 0.33
Specificity 0.88

Metrics for classification: #1
F-measure 0.28
Precision 0.34
Recall 0.24
Specificity 1.54

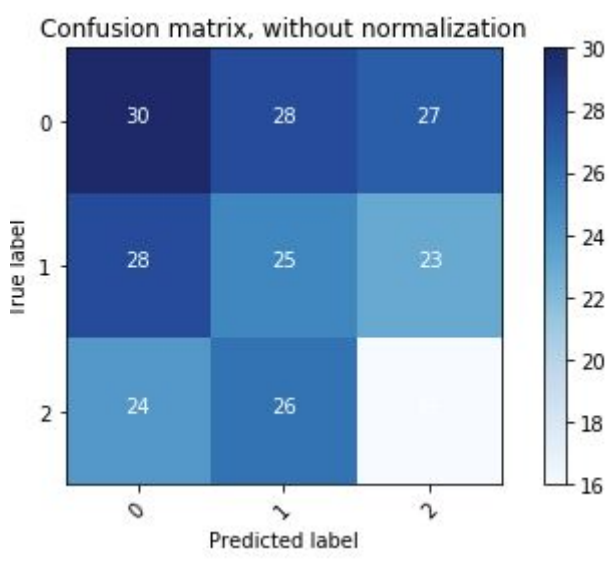
Metrics for classification: #2
F-measure 0.32
Precision 0.27
Recall 0.39
Specificity 0.66

Accuracy: 0.32

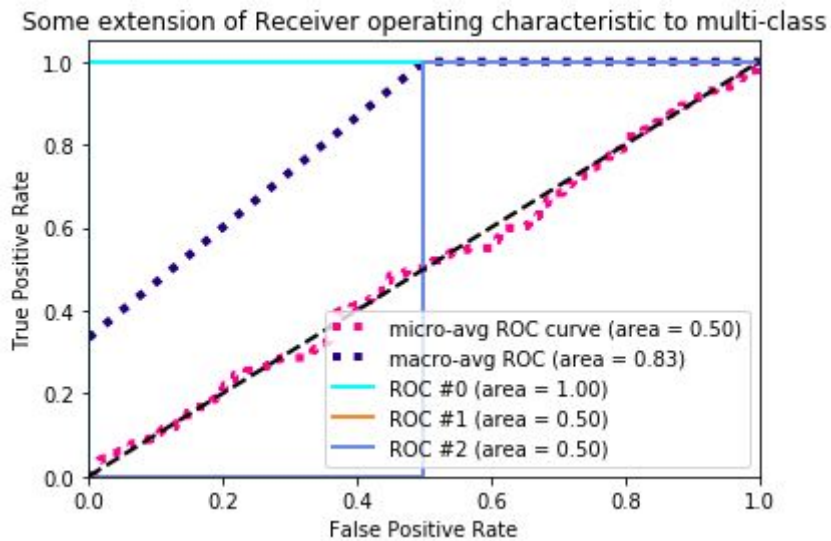
Metrics for classification: ALL CLASSES
F-measure: 0.31
Precision 0.32
Recall 0.32
Specificity 1.03

```

**Figure VI.3: Ensemble des métriques d'évaluation Cas a**



**Figure VI.4: Matrice de Confusion Cas b**



**Figure VI.5: Courbe ROC /AUC pour les 3 classes (Cas b)**

Cette figure VI.5 correspond à exactement à figure VI.2

```

Metrics for classification: #0
F-mesure 0.36
Precision 0.37
Recall 0.35
Specificity 0.79

Metrics for classification: #1
F-mesure 0.32
Precision 0.32
Recall 0.33
Specificity 0.85

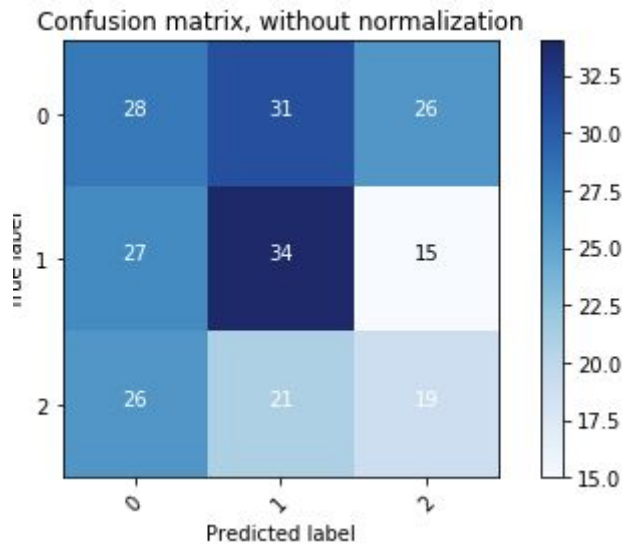
Metrics for classification: #2
F-mesure 0.24
Precision 0.24
Recall 0.24
Specificity 1.10

Accuracy: 0.31

Metrics for classification: ALL CLASSES
F-mesure: 0.31
Precision 0.31
Recall 0.31
Specificity 0.91

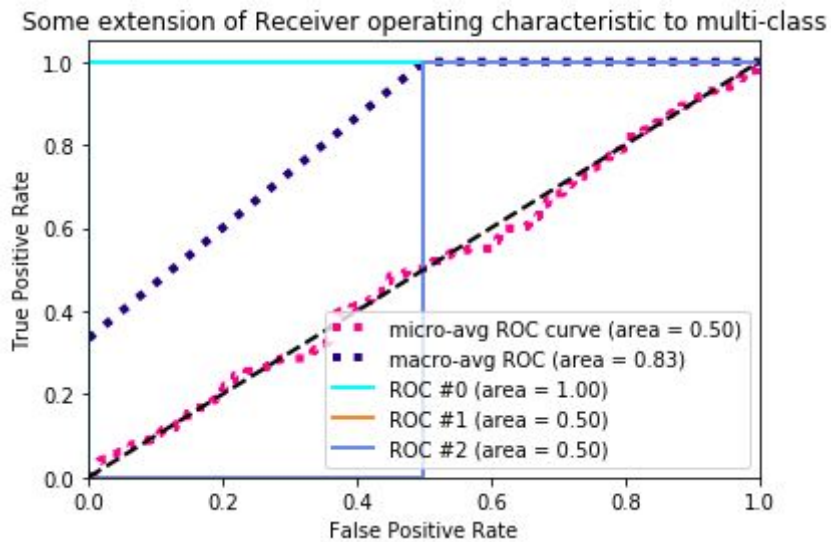
```

**Figure VI.6: Ensemble des métriques d'évaluation Cas b**



**Figure VI.7: Matrice de Confusion Cas c**

Hausse des vraies prédictions au sein de la classe 1 (précoce)



**Figure VI.8: Courbe ROC /AUC pour les 3 classes (Cas c)**

Les classe 0 et 1 ont obtenu un score 0 pour AUC ce qui signifie que le modèle n'est pas en mesure de distinguer entre positive et négative, concernant la classe 2 il a donné un maximum de 1

```
Metrics for classification: #0
F-mesure 0.34
Precision 0.35
Recall 0.33
Specificity 1.00

Metrics for classification: #1
F-mesure 0.42
Precision 0.40
Recall 0.45
Specificity 0.90

Metrics for classification: #2
F-mesure 0.30
Precision 0.32
Recall 0.29
Specificity 1.51

Accuracy: 0.36

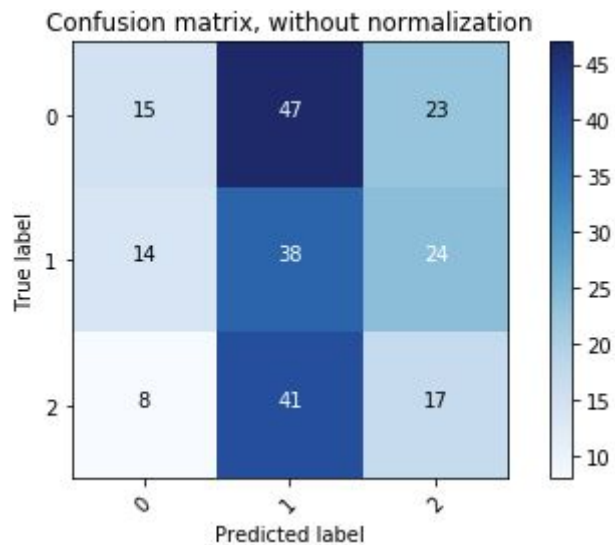
Metrics for classification: ALL CLASSES
F-mesure: 0.35
Precision 0.35
Recall 0.35
Specificity 1.14
```

**Figure VI.9: Ensemble de Métrique Cas c**

## **2.2. Evaluation d’InceptionResnetV2:**

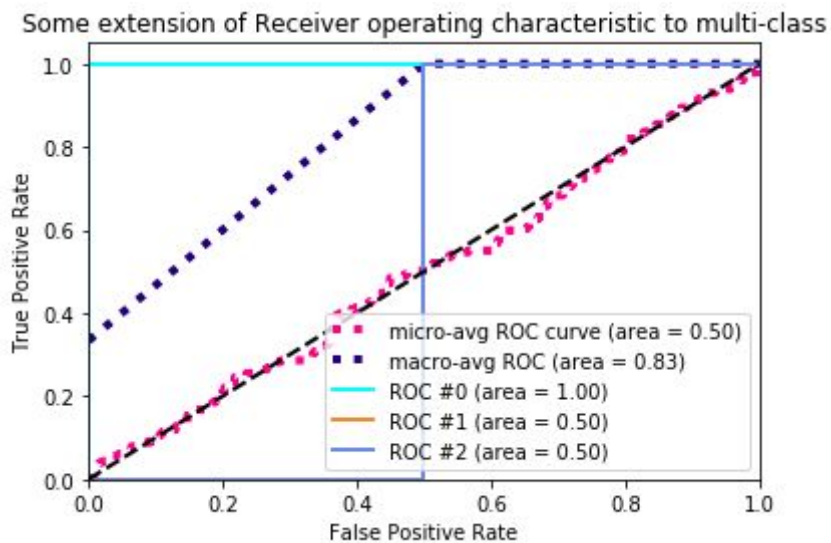
La configuration concernant le système basé sur le modèle **InceptionResnetV2** est définis au niveau du **Tableau IV.1**.

**Résultats obtenus:**



**Figure VI.10: Matrice de Confusion Cas a**

Les prédictions moyennes au niveau de la classe 1 (précoce), et médiocre concernant les autres classes.



**Figure VI.11: Figure VI.5: Courbe ROC /AUC pour les 3 classes (Cas a)**

Les classe 1 et 2 ont obtenu 0.5 en AUC comportement du modèle est aléatoire sur ses classes là, concernant la classe 0 il a obtenu le max de 1.

```

Metrics for classification: #0
F-mesure 0.25
Precision 0.41
Recall 0.18
Specificity 2.50

Metrics for classification: #1
F-mesure 0.38
Precision 0.30
Recall 0.50
Specificity 0.36

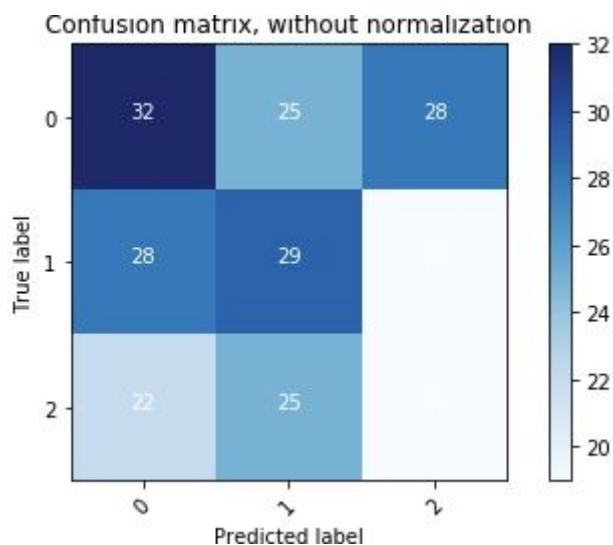
Metrics for classification: #2
F-mesure 0.26
Precision 0.27
Recall 0.26
Specificity 1.13

Accuracy: 0.31

Metrics for classification: ALL CLASSES
F-mesure: 0.29
Precision 0.32
Recall 0.31
Specificity 1.33

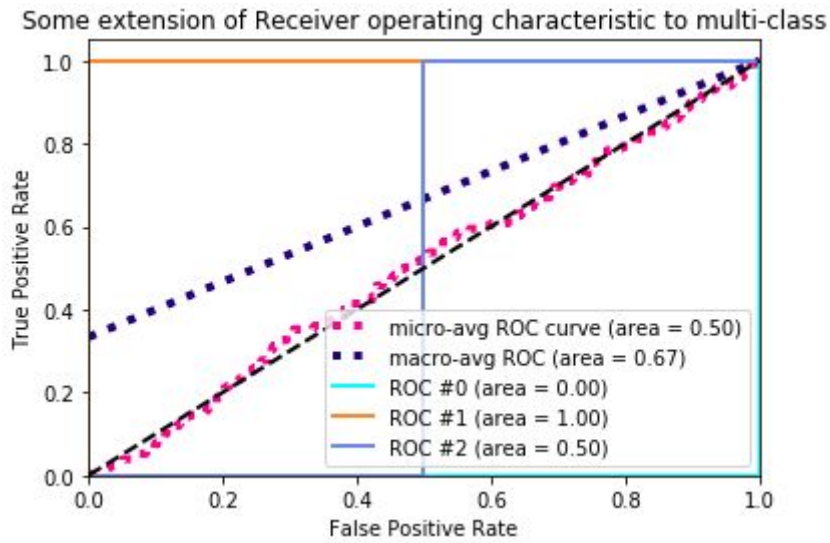
```

**Figure VI.12: Ensemble de Métrique Cas a**



**Figure VI.13: Matrice Confusion Cas b**

Les prédictions de la classe 0 est élevé, suivis des prédictions moyennes pour la classe 1 (précoce).



**Figure VI.14: Courbe Roc 3 classes Cas b**

La classe 0 a obtenu 0 en AUC cela signifie que le modèle ne peut pas distinguer en positif et négatif, pour la classe 1 il a obtenu le max, enfin la classe 2 il est au niveau de l'aléatoire.

```

Metrics for classification: #0
F-mesure 0.38
Precision 0.39
Recall 0.38
Specificity 0.96

Metrics for classification: #1
F-mesure 0.37
Precision 0.37
Recall 0.38
Specificity 1.02

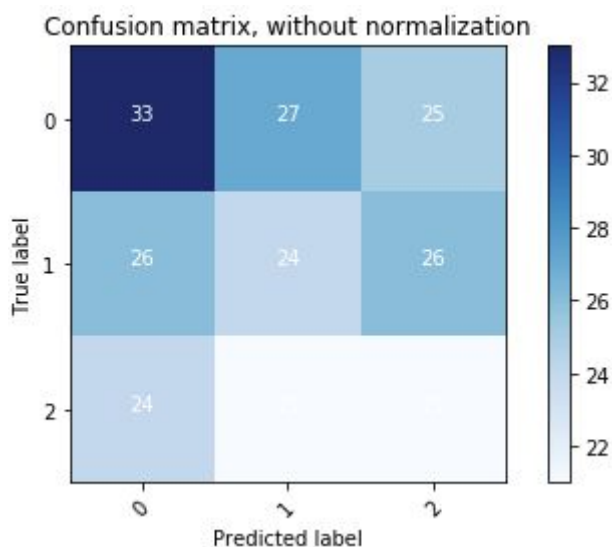
Metrics for classification: #2
F-mesure 0.29
Precision 0.29
Recall 0.29
Specificity 1.30

Accuracy: 0.35

Metrics for classification: ALL CLASSES
F-mesure: 0.35
Precision 0.35
Recall 0.35
Specificity 1.09

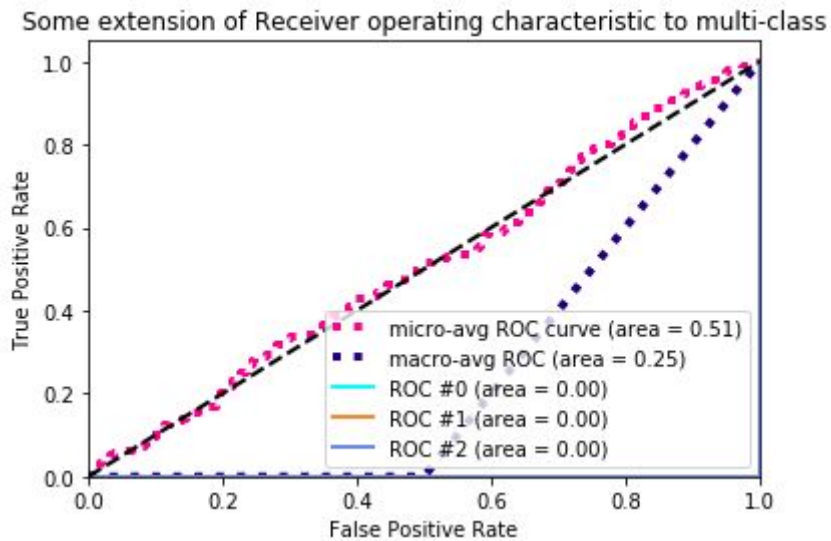
```

**Figure VI.15: Ensemble de Métriques Cas b**



**Figure VI.16:Matrice de Confusion Cas c**

La classe 0 a obtenu des prédictions élevées, pour le reste des classes 1 et 2 elles sont faibles.



**Figure VI.17: Courbe ROC /AUC 3 classes cas c**

Les résultats obtenu en AUC pour les 3 classes confondus sont à 0 le modèle n'est pas en mesure de distinguer entre positif et négatif pour les 3 classes.

```

Metrics for classification: #0
F-mesure 0.39
Precision 0.40
Recall 0.39
Specificity 0.90

Metrics for classification: #1
F-mesure 0.32
Precision 0.33
Recall 0.32
Specificity 1.12

Metrics for classification: #2
F-mesure 0.30
Precision 0.29
Recall 0.32
Specificity 1.12

Accuracy: 0.34

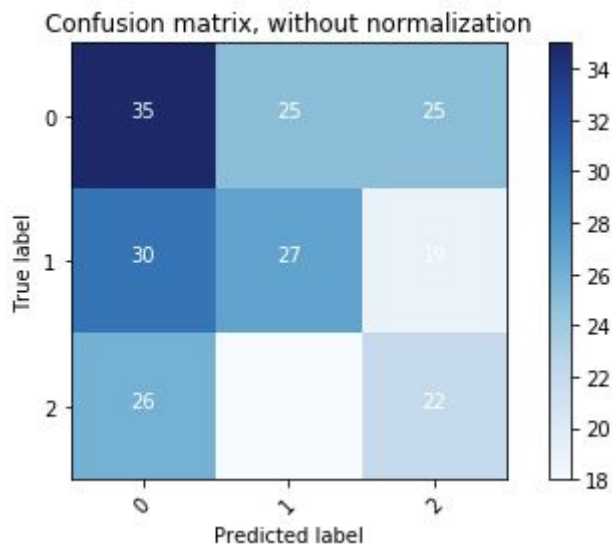
Metrics for classification: ALL CLASSES
F-mesure: 0.34
Precision 0.34
Recall 0.34
Specificity 1.05
  
```

**Figure VI.18: Ensemble de Métriques cas c**

### 2.3. Evaluation de DenseNet201:

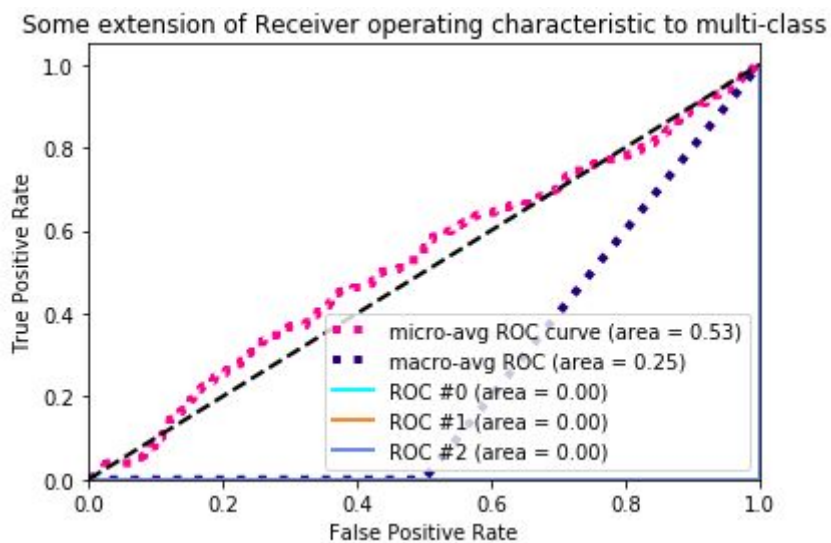
La configuration concernant le système basé sur le modèle **DenseNet201** est définis au niveau du **Tableau IV.1**.

#### Résultats obtenus:



**Figure VI.19: Matrice de Confusion cas a**

Les résultats des prédictions sont similaire au cas figure VI.13.



**Figure VI.20: Courbe ROC /AUC 3 classe cas a**

Concernant cette figure les résultats sont similaires à la figure VI.17

```

Metrics for classification: #0
F-mesure 0.45
Precision 0.45
Recall 0.44
Specificity 0.57

Metrics for classification: #1
F-mesure 0.26
Precision 0.26
Recall 0.25
Specificity 1.54

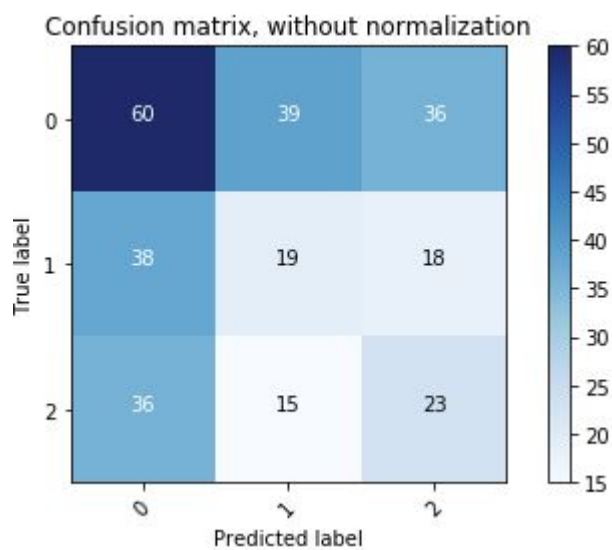
Metrics for classification: #2
F-mesure 0.30
Precision 0.30
Recall 0.31
Specificity 1.46

Accuracy: 0.36

Metrics for classification: ALL CLASSES
F-mesure: 0.34
Precision 0.34
Recall 0.34
Specificity 1.19

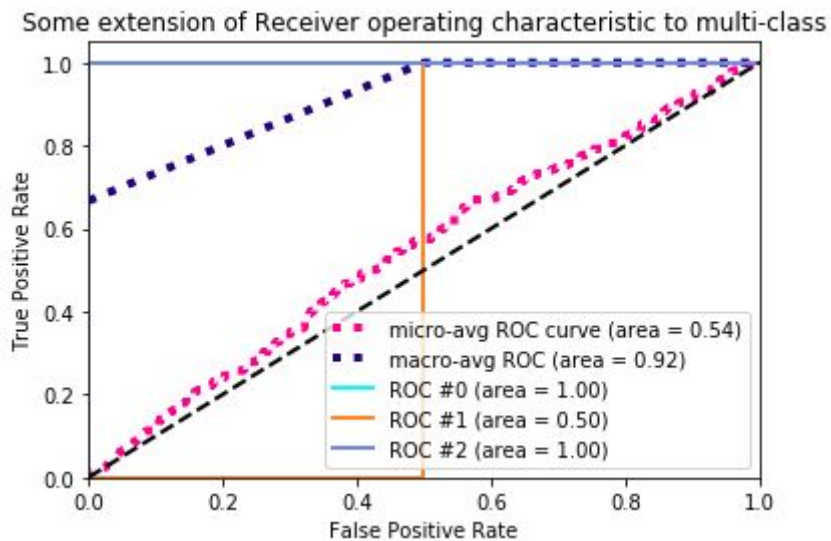
```

**Figure VI.21: Ensemble de Métriques cas a**



**Figure VI.22: Matrice de Confusion cas b**

Les résultats des prédictions sont similaires à la figure VI.16



**Figure VI.23: Courbe Roc / AUC 3 classes cas b**

La classe 0 et 2 ont obtenu un max de 1 en AUC, ainsi qu'un 0.5 en classe 1 ce qui signifie qu'il est aléatoire concernant le cas précoce.

```

Metrics for classification: #0
F-mesure 0.45
Precision 0.45
Recall 0.44
Specificity 0.57

Metrics for classification: #1
F-mesure 0.26
Precision 0.26
Recall 0.25
Specificity 1.54

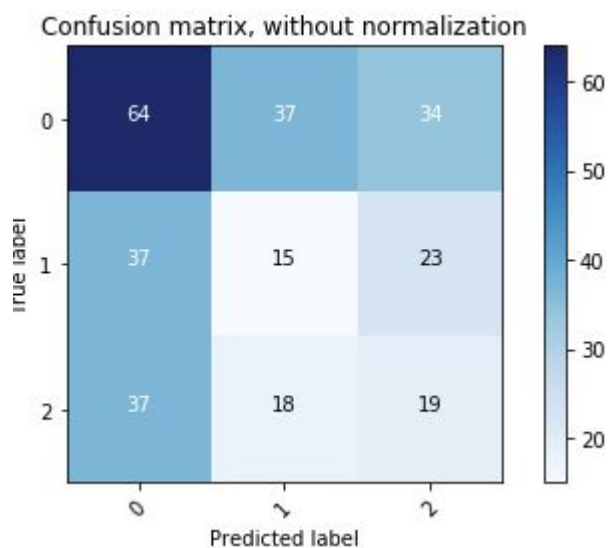
Metrics for classification: #2
F-mesure 0.30
Precision 0.30
Recall 0.31
Specificity 1.46

Accuracy: 0.36

Metrics for classification: ALL CLASSES
F-mesure: 0.34
Precision 0.34
Recall 0.34
Specificity 1.19

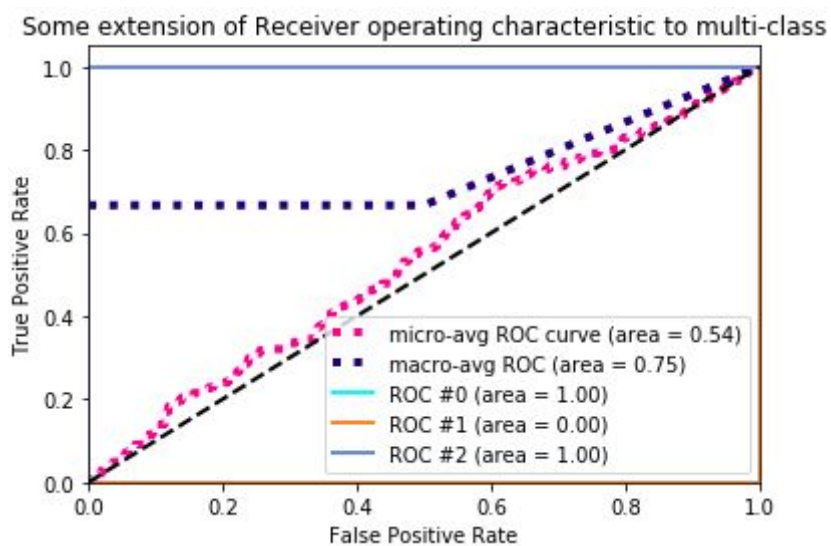
```

**Figure VI.24: Ensemble de Métrique cas b**



**Figure VI.25: Matrice de Confusion cas c**

Résultats des prédictions élevés concernant la classe 0, faible pour les autre classes 1 et 2.



**Figure VI.26: Courbe ROC/ AUC 3 classes cas c**

Les classes 0 et 2 ont obtenu 1 en AUC, mais 0 pour la classe 1 (précoce)

```

Metrics for classification: #0
F-mesure 0.47
Precision 0.46
Recall 0.47
Specificity 0.46

Metrics for classification: #1
F-mesure 0.21
Precision 0.21
Recall 0.20
Specificity 1.51

Metrics for classification: #2
F-mesure 0.25
Precision 0.25
Recall 0.26
Specificity 1.39

Accuracy: 0.35

Metrics for classification: ALL CLASSES
F-mesure: 0.31
Precision 0.31
Recall 0.31
Specificity 1.12

```

**Figure VI.27: Ensemble de Métrique Cas c**

Nom Modèle	Spécificité	Sensibilité	AUC Roc
<b>Xception (a)</b>	#0 : 0.88 #1 : 1.54 #2 : 0.66	#0 : 0.33 #1 : 0.24 #2 : 0.39	#0 : 1 #1 : 0.50 #2 : 0.50
<b>Xception (b)</b>	#0 : 0.79 #1 : 0.85 #2 : 1.10	#0 : 0.35 #1 : 0.33 #2 : 0.24	#0 : 1 #1 : 0.5 #2 : 0.5
<b>Xception (c)</b>	#0 : 1.0 #1 : 0.9 #2 : 1.51	#0 : 0.33 #1 : 0.45 #2 : 0.29	#0 : 0 #1 : 0 #2 : 1

<b>DenseNet201 (a)</b>	#0 : 0.88 #1 : 1.33 #2 : 1.41	#0 : 0.41 #1 : 0.36 #2 : 0.33	#0 : 0 #1 : 0 #2 : 0
<b>DenseNet201 (b)</b>	#0 : 0.57 #1 : 1.54 #2 : 1.46	#0 : 0.44 #1 : 0.25 #2 : 0.31	#0 : 1 #1 : 0.5 #2 : 1
<b>DenseNet201 (c)</b>	#0 : 0.46 #1 : 1.51 #2 : 1.39	#0 : 0.47 #1 : 0.20 #2 : 0.26	#0 : 1 #1 : 0 #2 : 1
<b>IncepResV2 (a)</b>	#0 : 2.50 #1 : 0.36 #2 : 1.13	#0 : 0.18 #1 : 0.50 #2 : 0.26	#0 : 0.50 #1 : 0 #2 : 0.50
<b>IncepResV2 (b)</b>	#0 : 0.96 #1 : 1.02 #2 : 1.30	#0 : 0.38 #1 : 0.38 #2 : 0.29	#0 : 0 #1 : 1 #2 : 0.5
<b>IncepResV2 (c)</b>	#0 : 0.90 #1 : 1.12 #2 : 1.12	#0 : 0.39 #1 : 0.32 #2 : 0.32	#0 : 0 #1 : 0 #2 : 0

**Tableau IV.4 : Tableau comparative des modèles selon les métriques**

### 3. Comparaison des résultats:

Après la série de tests expérimentaux faites sur l'ensemble des configurations citées dans le tableau IV.1, **DenseNet201 cas c** a obtenu les meilleurs résultats en termes de métriques d'évaluation, ainsi que temps de prédiction, ce qui le place en première position devant les 2 autres.

En utilisant le modèle InceptionResnetV2, on a obtenu les pires résultats qu'ils s'agissent de Spécificité ou même de sensibilité, et on peut voir ça très clairement dans le graphe de la courbe ROC quel que soit l'algorithme d'optimisation utilisé.

Concernant le modèle Xception les résultats sont effectivement meilleurs que InceptionResnetV2, mais reste tout de même assez faible compte tenu du temps coûteux

d'entraînement, cependant il reste plus rapide en terme de prédiction que ses prédécesseurs Inception V3 et V4 [II.14].

Le modèle DenseNet201 a fournis les meilleurs résultats de cette étude, surpassant Xception, et InceptionResnetV2 de loin que ce soit en terme de spécificité et rappel, cependant il possède un temps de prédiction trop important en raison de la profondeur du modèle.

Enfin, concernant la méthodologie suivie tout au long de cette étude, notre modèle a obtenu des résultats de classification élevés, on peut aussi noter une amélioration non négligeable au niveau de la classe 2 qui correspond au stade précoce de la pathologie, ce qui était le problème majeur dans les précédentes études tels que dans [II.14].

#### **4. Conclusion:**

Après avoir sélectionné les métriques adéquates pour évaluer un système d'aide au diagnostic médical notamment la spécificité et la sensibilité, nous avons effectué une série de tests d'évaluation sur différents paramètres fixés pour les 3 modèles étudiés Xception, InceptionResnetV2, ainsi que DenseNet201 Les résultats obtenus n'étaient pas à la hauteur et cela principalement à cause du manque de données d'infrastructure pour entrainer le modèle de façon adéquate.

## CONCLUSION GÉNÉRALE:

Dans ce mémoire, nous avons traité une problématique majeure qui vise à détecter la présence d'une Rétinopathie diabétique à un stade précoce sur des clichés rétiniens (Fond d'œil), et cela en utilisant les modèles de réseaux de neurones ConvNet issus d'ImageNet dont Xception, DenseNet201 et InceptionResnetV2 qui ont prouvé leur efficacité sur d'autres tâches telle que la reconnaissance d'objets.

On a exposé la pathologie, ainsi que l'organe concerné suivi d'un état d'art sur la détection de cette pathologie, en discutant les points positifs et négatifs de chaque étude.

On peut dire que l'étude la plus prometteuse en termes de résultats est [II.11] ayant travaillé sur le même dataset Kaggle, et ayant utilisé des modèles d'imagenet.

On a rencontré certaines difficultés durant l'étude, elles résident dans la création d'un système de détection de la pathologie à un stade précoce, et cela pour de nombreuses raisons: Le cruel manque de données d'entraînement pour les classes précoces, la fiabilité du dataset, données de mauvaise qualité, ainsi que l'origine même du système de détection utilisant une approche Deep Learning nécessitant une puissance de calcul non négligeable.

A cela, on a proposé une solution pour pallier aux problèmes rencontrés durant l'étude qui se traduit par l'utilisation d'une plateforme cloud pour le prétraitement des données, ainsi que l'entraînement des modèles. Pour l'évaluation de notre système de détection, on a opté pour l'utilisation de métriques tel sensibilité, spécificité, ainsi que d'une représentation graphique courbe ROC qui représente ces deux métriques au sein d'un graphe. Ses métriques nous permettent de traiter, étudiés, ainsi qu'évaluer les résultats de nos modèles dans le but d'améliorer les performances de détection.

L'étude a démontré que le modèle avec les meilleurs résultats est DenseNet201 avec une configuration (**Learning rate de 0.001, dropout à 0.4 et optimiseur stochastic gradient descent**).

Pour la réalisation de ce système de détection, on a opté pour une solution en Python avec le Framework keras utilisant Tensorflow en background, qui offre les modèles utilisés dans l'étude avec un transfert d'apprentissage inclus.

Une perspective d'amélioration à apporter à ce système serait d'incorporer l'un de ses principes:

- Utiliser un prétraitement de l'image différent de clahe.
- Utiliser le Gradient Synthétique pour accélérer la phase d'entraînement, ainsi qu'augmenter les performances du modèle.

## **Bibliographie:**

### **Chapitre 1:**

- [1] [http://www.inflamoeil.org/l-oeil/anatomie/l-oeil-sain/article/anatomie-de-l-oeil?fbclid=IwAR2P8YjcugxRxJMchRp\\_MiK40b86Xfxn5GVqW9\\_ZwSUXzCEGD5Rt\\_oUF-o](http://www.inflamoeil.org/l-oeil/anatomie/l-oeil-sain/article/anatomie-de-l-oeil?fbclid=IwAR2P8YjcugxRxJMchRp_MiK40b86Xfxn5GVqW9_ZwSUXzCEGD5Rt_oUF-o)
- [2] <https://www.parlonsdiabete.com/parlons-diabete/le-diagnostic>
- [3] <https://www.diabete.qc.ca/en/understand-diabetes/all-about-diabetes/complications/diabetes-complications?fbclid=IwAR1OjctLDXP2f5Lm4L3uYnWGxqgwBcgI2pMrZ2Ek06xIrbhDpxPhJSyXANM>
- [4] [https://www.espace-diabete.com/services/le-diabete/les-consequences-du-diabete?fbclid=IwAR3pBoKsgPnQC4sK3fxqR0EIXUIpZc7EO2IFE5bNTGXYiF8uJ\\_pS7mWhYOO](https://www.espace-diabete.com/services/le-diabete/les-consequences-du-diabete?fbclid=IwAR3pBoKsgPnQC4sK3fxqR0EIXUIpZc7EO2IFE5bNTGXYiF8uJ_pS7mWhYOO)
- [5] <https://www.parlonsdiabete.com/parlons-diabete/le-diagnostic>
- [6] <https://santecheznous.com/condition/getcondition/retinopathie>
- [7] [https://www.news-medical.net/health/Diabetic-Retinopathy-Risk-Factors-\(French\).aspx](https://www.news-medical.net/health/Diabetic-Retinopathy-Risk-Factors-(French).aspx)
- [8] [https://www.has-sante.fr/portail/upload/docs/application/pdf/2011-03/fiche\\_de\\_synthese\\_retinopathie\\_vf.pdf](https://www.has-sante.fr/portail/upload/docs/application/pdf/2011-03/fiche_de_synthese_retinopathie_vf.pdf)
- [9] [https://www.sante-centre.fr/portail\\_v1/gallery\\_files/site/133/3687/3741.pdf](https://www.sante-centre.fr/portail_v1/gallery_files/site/133/3687/3741.pdf)

[10] DIARETDB0: Evaluation Database and Methodology for Diabetic Retinopathy Algorithms

[https://www.researchgate.net/publication/255610068\\_DIARETDB0\\_Evaluation\\_Database\\_and\\_Methodology\\_for\\_Diabetic\\_Retinopathy\\_Algorithms](https://www.researchgate.net/publication/255610068_DIARETDB0_Evaluation_Database_and_Methodology_for_Diabetic_Retinopathy_Algorithms)

[11] Automated Detection of Diabetic Retinopathy using Deep Learning

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5961805/>

[12] Deep Convolutional Neural Network-Based Early Automated Detection of Diabetic Retinopathy Using Fundus Image

<https://www.ncbi.nlm.nih.gov/pubmed/29168750>

[13] Automated Identification of Diabetic Retinopathy Using Deep Learning

<https://www.ncbi.nlm.nih.gov/pubmed/28359545>

[14] Automated early detection of diabetic retinopathy

<https://www.ncbi.nlm.nih.gov/pubmed/20399502>

[15] Early diagnosis of retinopathy in primary case

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4437282/>

[16] Automated Detection and Differentiation of Drusen, Exudates, and Cotton-wool Spots in Digital Color Fundus Photographs for Early Diagnosis of Diabetic Retinopathy IOVS-06-0996 accepted version

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2739583/>

### **Chapitre 2:**

[1] <https://www.import.io/post/history-of-deep-learning/>

[2] <https://www.quantmetry.com/une-petite-histoire-du-machine-learning/>

[3] <https://www.geeksforgeeks.org/getting-started-machine-learning/>

[4] <https://towardsdatascience.com/unsupervised-learning-with-python-173c51dc7f03>

[5] <https://pdfs.semanticscholar.org/b326/2e25bb5ead6159b87990adb40c66ef83b76.pdf>

[6] <https://www.datascience.com/blog/what-is-semi-supervised-learning>

[7] <https://www.geeksforgeeks.org/what-is-reinforcement-learning/>

- [8]<https://www.usine-digitale.fr/article/le-turing-award-recompense-trois-pionniers-du-deep-learning-dont-le-francais-yann-lecun.N824015>
- [9]<http://www.scilogs.fr/intelligence-mecanique/architecture-des-reseaux-de-neurones-reseau-x-de-neurones-artificiels-classiques-2-3/>
- [10][https://en.wikipedia.org/wiki/Walter\\_Pitts](https://en.wikipedia.org/wiki/Walter_Pitts)
- [11][https://fr.wikipedia.org/wiki/Cortex\\_préfrontal](https://fr.wikipedia.org/wiki/Cortex_préfrontal)
- [12][https://fr.wikipedia.org/wiki/Fonction\\_d%27activation](https://fr.wikipedia.org/wiki/Fonction_d%27activation)
- [13]<https://en.wikipedia.org/wiki/Backpropagation>
- [14][https://en.wikipedia.org/wiki/Gradient\\_descent](https://en.wikipedia.org/wiki/Gradient_descent)
- [15]<http://runder.io/optimizing-gradient-descent/>
- [16]<https://blog.algorithmia.com/introduction-to-loss-functions/>
- [17]<https://www.geeksforgeeks.org/gradient-descent-algorithm-and-its-variants/>
- [18]<http://www.scilogs.fr/intelligence-mecanique/architecture-des-reseaux-de-neurones-reseau-x-de-neurones-artificiels-classiques-2-3/>
- [19][https://en.wikipedia.org/wiki/Deep\\_learning](https://en.wikipedia.org/wiki/Deep_learning)

### **Chapitre 3:**

- [1] Xception: Deep Learning with Depthwise Separable Convolutions  
<https://arxiv.org/abs/1610.02357>
- [2] Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning  
<https://arxiv.org/abs/1602.07261>
- [3] Densely Connected Convolutional Networks  
<https://arxiv.org/abs/1608.06993>
- [4] Deep Residual Learning for Image Recognition  
<https://arxiv.org/abs/1512.03385>









