**République Algér****ienne Démocratique et PopulaireMinistère de l’enseignement supérieur et de la recherche scientifique**

**Université 8Mai 1945 – Guelma**

**Faculté des sciences et de la Technologie**

**Département d’Electronique et Télécommunications**

****

**Mémoire de fin d'étude**

**Pour l’obtention du diplôme de Master Académique**

**Domaine : Sciences et Technologie**

**Filière : Electronique**

**Spécialité : Instrumentation**

**DETECTION PRECOCE D’ALZHEIMER A L’AIDE DE DEEP LEARNING**

**Présenté par :**

**------------------------------------------**

**BELGUIDOUM Fakhreddine**

**----------------------------------------------**

**Sous la direction de :**

**Pr. BOUKAACHE Abdelnour**

JUIN 2024

**Remerciements**

Après avoir remercié Dieu, le Seul et Unique Éternel, je trouve les mots de gratitude insuffisants pour exprimer à tous ceux qui ont contribué à l'achèvement de cet humble travail que je remets entre vos mains, avec une mention spéciale au professeur encadreur M. Boukaache Abdelnour. Sans oublier nos estimés professeurs membres du jury, pour nous avoir accordé de leur temps, de leur patience et de leur généreuse attention.

Sans oublier tous les professeurs qui se sont occupés de notre formation jusqu'à ce jour et tous les professeurs en général, nous remercions nos collègues étudiants pour toute aide de leur part.

Je remercie mes grandes et petites familles pour leur patience face à notre négligence de leurs droits pour le bien de ce travail, en particulier ma chère mère, que Dieu lui accorde la santé et la miséricorde pour mon père.

**Résumé**

La maladie d'Alzheimer est une maladie neurodégénérative qui touche principalement les personnes âgées. En raison de la spécificité de cette maladie et de ses effets graves, qui impactent non seulement le patient, mais également sa famille et perturbent le cours normal de leur vie, ses répercussions psychologiques et sociales s'étendent à la société dans son ensemble, car la famille est l'élément fondamental de la société. Compte tenu de l'absence de traitement véritablement curatif pour cette maladie, il est crucial de la diagnostiquer à un stade précoce.

C'est pourquoi nous avons opté pour l'utilisation des techniques d'apprentissage profond (**Deep learning**) comme outils modernes, devenus populaires et largement utilisés dans divers domaines. Cette technologie offre une capacité exceptionnelle de classification et de prédiction, particulièrement après les avancées significatives réalisées dans le domaine des ordinateurs en termes de ressources, de puissance des processeurs, de mémoire vive (**RAM**) et des unités de traitement graphique **(GPUs)**.

Les réseaux convolutifs **(CNN)** sont l'un des types les plus importants de réseaux de **Deep Learning**. En raison de leur capacité à extraire des caractéristiques discriminantes des images, nous avons concentré nos efforts sur leur utilisation. Nous avons utilisé un jeu de données d'images de résonance magnétique **(IRM)** pour entraîner notre modèle et évaluer ses performances.

Nous avons également expérimenté un modèle pré-entraîné **SqueezeNet** en utilisant la méthode Transfer Learning. Enfin, nous avons comparé les résultats obtenus avec ceux de travaux similaires dans le même domaine.

**Abstract**

Alzheimer's disease is a neurodegenerative disease that mainly affects older people. Due to the specificity of this disease and its serious effects, which impact not only the patient but also their family and disrupt the normal course of their lives, its psychological and social repercussions extend to society as a whole, because the family is the fundamental element of society. Given the lack of truly curative treatment for this disease, it is crucial to diagnose it at an early stage.

This is why we opted for the use of **Deep learning** techniques as modern tools, which have become popular and widely used in various fields. This technology offers exceptional classification and prediction capacity, particularly after the significant advances made in the field of computers in terms of resources, processor power, random access memory (**RAM**), and graphics processing units (**GPUs**).

Convolutional networks (**CNN**) are one of the most important types of deep learning networks. Due to their ability to extract discriminative features from images, we have focused our efforts on their use. We used a magnetic resonance image (**MRI**) dataset to train our model and evaluate its performance.

We also experimented with a pre-trained **SqueezeNet** model using the transfer learning method. Finally, we compared the results obtained with those of similar works in the same field.

**TABLE DE MATIER**

[INTRODUCTION GENERALE 1](#_Toc169694050)

**CHAPITRE I : DIAGNOSTIC DE LA MALADIE D'ALZHEIMER**

[I. LA MALADIE D'ALZHEIMER 3](#_Toc169694051)

[I.1 Historique : 3](#_Toc169694052)

[I.2 Méthodes traditionnelles de diagnostic : 3](#_Toc169694053)

[I.2.1 Examen neurologique et physique : 3](#_Toc169694054)

[I.2.2 Tests de laboratoire 4](#_Toc169694055)

[I.2.3 Tests neuropsychologiques et de l'état mental 4](#_Toc169694056)

[I.2.4 Imagerie cérébrale 4](#_Toc169694057)

[I.3 Les techniques d’imagerie de la structure cérébrale : 5](#_Toc169694058)

[I.3.1 Tomodensitométrie 5](#_Toc169694059)

[I.3.2 La tomographie par émission de positons : 5](#_Toc169694060)

[I.3.3 Imagerie par résonance magnétique : 6](#_Toc169694061)

[I.4 Applications de deep learning dans le domaine médical 6](#_Toc169694062)

[I.4.1 Imagerie médicale : 6](#_Toc169694063)

[I.4.2 Pathologie numérique : 7](#_Toc169694064)

[I.4.3 Diagnostic assisté par ordinateur : 7](#_Toc169694065)

[I.4.4 Prédiction de maladies : 7](#_Toc169694066)

[I.4.5 Personnalisation du traitement : 7](#_Toc169694067)

[I.4.6 Détection de maladies à un stade précoce : 7](#_Toc169694068)

[II. FONDEMENTS THEORIQUES DE LA MALADIE 8](#_Toc169694069)

[II.1 Les marqueurs de l’Alzheimer 8](#_Toc169694070)

[II.2 Symptômes 8](#_Toc169694071)

[II.2.1 Mémoire : 9](#_Toc169694072)

[II.2.2 Penser et raisonner 9](#_Toc169694073)

[II.2.3 Porter des jugements et des décisions 9](#_Toc169694074)

[II.2.4 Causes 10](#_Toc169694075)

**CHAPITRE II: LE DEEP LEARNING POUR DETECTION D'ALZHEIMER**

[I. DEEP LEARNING ET RESEAUX NEURONES 12](#_Toc169694076)

[I.1 Notions fondamentales : 12](#_Toc169694077)

[I.1.1 Types de Réseaux de Neurones en Deep Learning 12](#_Toc169694078)

[I.1.2 Classification par Deep Learning 19](#_Toc169694079)

[I.1.3 Facteurs de Performance et Comment les Choisir 22](#_Toc169694080)

[I.1.4 Conclusion 23](#_Toc169694081)

[II. METHODOLOGIE POUR LA CLASSIFICATION 24](#_Toc169694082)

[II.1 Collecte et prétraitement des données 24](#_Toc169694083)

[II.1.1 Ensembles d'entraînement, de validation et de test : 24](#_Toc169694084)

[II.1.2 Division aléatoire vs stratifiée : 25](#_Toc169694085)

[II.1.3 Validation croisée : 25](#_Toc169694086)

[II.1.4 Validation croisée stratifiée: 26](#_Toc169694087)

[II.1.5 Taille des ensembles : 27](#_Toc169694088)

[II.1.6 Considérations sur la généralisation : 27](#_Toc169694089)

[II.2 Méthodes d'évaluation de la performance 27](#_Toc169694090)

[II.2.1 Matrice de confusion 27](#_Toc169694091)

[II.2.2 Métriques 29](#_Toc169694092)

[II.2.3 Métrique multi classes 31](#_Toc169694093)

[II.2.4 La généralisation peut se faire de 3 façons : 38](#_Toc169694094)

**CHAPITRE III: RESULTATS DU MODELE PROPOSE**

[I. Architecture du modèle de deep learning 40](#_Toc169694095)

[I.1.1 Model pré-entrainé sqeeznet 40](#_Toc169694096)

[I.1.2 Model proposé : 41](#_Toc169694097)

[II. RESULTATS 43](#_Toc169694098)

[II.1 Description des données utilisées 43](#_Toc169694099)

[II.2 Evaluation de modèle (*Squeezent*) qui entrainer sur dataset a 2 classes avec nbr des individués équilibrés : 43](#_Toc169694100)

[II.2.1 Matrice de confusion 44](#_Toc169694101)

[II.2.2 Métriques 44](#_Toc169694102)

[II.3 Evaluation de notre modèle qui entrainer sur dataset a 2 classes avec nbr des individués équilibrés : 45](#_Toc169694103)

[II.3.1 Matrice De Confusion 45](#_Toc169694104)

[II.3.2 Métriques 45](#_Toc169694105)

[II.4 Evaluation de notre modèle qui entrainer sur dataset a 4 classes avec nbr des individués déséquilibrés : 46](#_Toc169694106)

[II.4.1 Matrice de confusion 46](#_Toc169694107)

[II.4.2 Métriques 47](#_Toc169694108)

[II.4.3 Synthèse : 51](#_Toc169694109)

[II.5 Evaluation de model (*Squeezent*) qui entrainer sur dataset a 4 classes avec nbr des individués déséquilibrés : 52](#_Toc169694110)

[II.5.1 Matrice de confusion 52](#_Toc169694111)

[II.5.2 Métriques 53](#_Toc169694112)

[II.5.3 Synthèse : 57](#_Toc169694113)

[II.6 Discussion : 57](#_Toc169694114)

[CONCLUSION GENERALE 59](#_Toc169694115)

**Table Des Figures**

[Figure 1: Images de scanners cérébraux 4](#_Toc169701553)

[Figure 2: image par émission de positon 5](#_Toc169701554)

[Figure 3: Images IRM 6](#_Toc169701555)

[Figure 4:Apparition d'un cerveau atteint d'Alzheimer [3] 10](#_Toc169701556)

[Figure 5: réseaux CNN 13](#_Toc169701557)

[Figure 6:convolution 13](#_Toc169701558)

[Figure 7:stride 14](#_Toc169701559)

[Figure 8: padding 14](#_Toc169701560)

[Figure 9: couche flaten/aplatissement 15](#_Toc169701561)

[Figure 10: Couche fully connected 16](#_Toc169701562)

[Figure 11: Couche dropout 17](#_Toc169701563)

[Figure 12: fonction segmoide 17](#_Toc169701564)

[Figure 13:fonction d’activation tanH 18](#_Toc169701565)

[Figure 14: relU 18](#_Toc169701566)

[Figure 15: overfitting 20](#_Toc169701567)

[Figure 16: under fitting 21](#_Toc169701568)

[Figure 17: fitting 22](#_Toc169701569)

[Figure 18: Detection d’overfitting 22](#_Toc169701570)

[Figure 19: Dataset 24](#_Toc169701571)

[Figure 20:distribution de dataset 24](#_Toc169701572)

[Figure 21: stratification 25](#_Toc169701573)

[Figure 22: Validation Croisée 25](#_Toc169701574)

[Figure 23:Validation croisée et stratifie 26](#_Toc169701575)

[Figure 24: matrice de confusion binaire 28](#_Toc169701576)

[Figure 25: exemple d’une matrice de confusion 29](#_Toc169701577)

[Figure 26: Matrice de confusion multi-classe théorique pour 3 classes 29](#_Toc169701578)

[Figure 27: ROC ET AUC 31](#_Toc169701579)

[Figure 28: passage multi-classes vers bi-classes 32](#_Toc169701580)

[Figure 29:Matrices de confusion des 4 classes séparées 33](#_Toc169701581)

[Figure 30: *Matrice de confusion synthétique* 36](#_Toc169701582)

[Figure 31: *Matrice confusion micro* 37](#_Toc169701583)

[Figure 32: architecture de modèle sqeeznet 40](#_Toc169701584)

[Figure 33: architecture de modèle proposé 41](#_Toc169701585)

[Figure 34: progression d’apprentissage pour sqeeznet a 2 classes 43](#_Toc169701586)

[Figure 35: matrice de confusion pour sqeeznet à 2 classes 44](#_Toc169701587)

[Figure 36: progression d’apprentissage pour notre modèle a 2 classes 45](#_Toc169701588)

[Figure 37: matrice de confusion pour notre modèle à 2 classes 45](#_Toc169701589)

[Figure 38: progression d’apprentissage pour notre modèle a 4 classes 46](#_Toc169701590)

[Figure 39: matrice de confusion pour notre modèle a 4 classes 46](#_Toc169701591)

[Figure 40: matrices binaires pour chaque classe 47](#_Toc169701592)

[Figure 41: matrices a 4 classes pour notre modèle 50](#_Toc169701593)

[Figure 42: matrice synthétique 51](#_Toc169701594)

[Figure 43: progression d’apprentissage de sqeeznet a 4 classes 52](#_Toc169701595)

[Figure 44: matrice de confusion de sqeeznet a 4 classes 52](#_Toc169701596)

[Figure 45: matrice binaire de sqeeznet a 4 classes 53](#_Toc169701597)

[Figure 46: matrices à 4 classes pour sqeeznet 56](#_Toc169701598)

[Figure 47: passage vers matrice synthétique 56](#_Toc169701599)

**Table Des Equations**

[1 13](#_Toc169705284)

[2 14](#_Toc169705285)

[3 14](#_Toc169705286)

[4 15](#_Toc169705287)

[5 13](#_Toc169705288)

[6 15](#_Toc169705289)

[7 26](#_Toc169705290)

[8 26](#_Toc169705291)

[9 26](#_Toc169705292)

[10 27](#_Toc169705293)

[11 27](#_Toc169705294)

[12 27](#_Toc169705295)

[13 28](#_Toc169705296)

[14 28](#_Toc169705297)

[15 28](#_Toc169705298)

[*16* 28](#_Toc169705299)

[17 30](#_Toc169705300)

[18 31](#_Toc169705301)

[19 32](#_Toc169705302)

[20 32](#_Toc169705303)

[..21 34](#_Toc169705304)

**Liste des abréviations**

- **ANN** : Artificial Neural Network

- **RNN**: Recurrent Neural Network

-**LSTM**: Long Short Term Memory

-**GRU**: Gated Recurrent Unit

- **CNN** : Convolutional Neural Network

- **DNN** : Deep Neural Network

- **EC** : entièrement connecté

-**FC:** fully connected

- **GPU** : Graphic Processor Unit

-**TPU**: Tensor Processing Units

- **CPU** : Central Processing Unit

-**RAM**: Random Access Memory

- **IA** : Intelligence Artificielle

- **IRM** : Imagerie par Résonance Magnétique

- **ReLU** : Rectified Linear Unit

-**sgd**: Stochastic gradient descent avec Momentum

-**Adam**: Adaptive Gradient avec Momentum

- **RMSProp**: Root Mean Squared Propagation

-**FPR**: False Positive Rate

-**TPR**: True Positive Rate

-**MSE**: Mean Squared Error

-**RMSE**: Root Mean Squared Error

- **RGB**: Red Green Blue

- **ROC**: Receiver operating characteristic

-**AUC**: Area under Curve

- **TEP** : Tomographie par émission de positrons

**INTRODUCTION GENERALE**

# INTRODUCTION GENERALE

La maladie d'Alzheimer a été décrite pour la première fois en 1906 par le psychiatre et neurologue **Aloïs Alzheimer**, qui a associé les symptômes à des lésions cérébrales spécifiques, les plaques **amyloïdes** et les dégénérescences neurofibrillaires, après l'étude d'une patiente nommée **Augusta Deter**. Ses découvertes ont été confirmées par d'autres chercheurs, et un deuxième cas similaire en 1911 a validé sa théorie [1].

Le diagnostic précoce de la maladie d'Alzheimer est d'une grande importance, même si la maladie d'Alzheimer ne se guérit pas. On ne peut que ralentir la progression de la maladie et préparer le patient et ses proches à faire face à sa nouvelle situation, à s'y adapter, et atténuer ses mauvaises conséquences.

De même, le processus de diagnostic précoce nous aide à distinguer s'il s'agit de la maladie d'Alzheimer ou d'une autre maladie cérébrale, car il existe plusieurs maladies dont les symptômes sont similaires à ceux de la maladie d'Alzheimer, ce qui nous donne l'avantage d'identifier des médicaments appropriés qui peuvent réduire et ralentir la progression de la maladie [2]

Après avoir pris connaissance de l'importance du diagnostic précoce de la maladie d'Alzheimer et des méthodes traditionnelles utilisées pour diagnostiquer la maladie d'Alzheimer, de leurs avantages et inconvénients, et après avoir pris connaissance du développement en cours dans le domaine de l'apprentissage profond et du développement rapide de ce domaine , en particulier ces dernières années, avec le développement considérable des appareils informatiques et l'augmentation de leurs capacités, ils sont devenus plus réactifs. En plus des besoins de la technologie de Deep Learning, nous avons décidé que nous pourrions bénéficier de la technologie d'apprentissage profond pour aider dans le diagnostic précoce de cette maladie. Par conséquent, l’objectif de nos travail sera d’améliorer nos connaissances dans le domaine de **Deep Learning** et de l’intelligence artificielle, et de travailler au développement de notre propre modèle qui facilite le processus de diagnostic et soit très efficace et fiable.

La technologie du **Deep learning** repose sur le traitement direct de données pré-classifiées sans avoir besoin d’algorithmes préprogrammés pour en extraire les caractéristiques, d’autant plus que nous n’avons pas suffisamment de connaissances sur la maladie d’Alzheimer ou ses caractéristiques.

Tout ce dont nous disposons, ce sont des ensembles de données sur les cas précédemment confirmés de la maladie d'Alzheimer, des enregistrements et des données montrant les stades de développement de la maladie, ainsi que d'autres enregistrements de cas non affectés classés selon l'incidence et le degré de développement de la maladie d'Alzheimer.

Ceci est approprié pour la technologie d'apprentissage profond, qui a besoin de données pré-classifiées pour nous aider à entraîner notre modèle. Plus la quantité de données étiquetées est grande, plus notre modèle devient précis et efficace.

Dans notre recherche, nous avons préféré utiliser une base de données d’images médicales de cas malades à différents degrés et de cas non malades préalablement classés. Parce que nous utiliserons des images médicales comme données d'entrée, il convient d'utiliser des réseaux de neurones convolutifs CNN. Dans notre recherche, nous utiliserons l'environnement de travail de Matlab version R2024 A.

CHAPITRE I

DIAGNOSTIC DE LA MALADIE D'ALZHEIMER

# LA MALADIE D'ALZHEIMER

## Historique :

Un facteur important dans le diagnostic de la maladie d'Alzheimer est la capacité du patient à expliquer les symptômes qu'il ressent. L’avis d’un membre de la famille ou d’un ami proche sur les symptômes et leur impact sur la vie quotidienne peut également être utile. Des tests de mémoire et de réflexion aident également à diagnostiquer la maladie d'Alzheimer[2].

Des analyses de sang et des tests d’imagerie peuvent exclure d’autres causes possibles des symptômes. Cela peut également aider votre professionnel de la santé à mieux identifier la maladie à l’origine des symptômes de la démence.

Dans le passé, la maladie d'Alzheimer n'était diagnostiquée avec certitude qu'après la mort, lorsqu'un examen microscopique du cerveau révélait des plaques et des enchevêtrements. Désormais, les professionnels de la santé et les chercheurs peuvent diagnostiquer la maladie d'Alzheimer avec une plus grande certitude alors que le patient est encore en vie, car les bios marqueurs peuvent détecter la présence de plaques et d'enchevêtrements. Les tests de bios marqueurs comprennent des types spécifiques de tomographie par émission de positons et des tests qui mesurent les protéines amyloïdes et tau dans la partie liquide du sang et du liquide céphalo-rachidien.

## Méthodes traditionnelles de diagnostic :

Les procédures de diagnostic de la maladie d'Alzheimer comprendront probablement les tests suivants [2] [3]:

### Examen neurologique et physique :

Votre médecin procédera à un examen physique. Un examen neurologique peut inclure les tests suivants :

* Réactions involontaires (réflexes).
* Resserrement et force musculaire.
* Capacité à se lever d’une chaise et à traverser une pièce.
* Les sens de la vue et de l'ouïe.
* La coordination motrice.
* Équilibre.

### Tests de laboratoire

Les analyses de sang peuvent aider à exclure d’autres causes possibles de perte de mémoire et de confusion, comme un trouble thyroïdien ou des taux de vitamines considérablement faibles. Les analyses de sang peuvent également mesurer les niveaux de protéines bêta-amyloïdes et tau, mais ces tests ne sont pas largement disponibles et peuvent ne pas être couverts par une assurance.

### Tests neuropsychologiques et de l'état mental

Votre médecin peut vous faire passer un simple examen de votre état mental pour évaluer votre mémoire et vos autres capacités de réflexion. Des formes plus longues de ce type de test peuvent fournir plus de détails sur le fonctionnement mental qui peuvent être comparés à ceux de personnes du même âge et du même niveau d'éducation. Ces tests peuvent aider à déterminer le diagnostic et devenir un point de départ pour suivre les symptômes à l’avenir.

### Imagerie cérébrale

L'imagerie cérébrale est souvent utilisée pour identifier les changements visuels associés à des affections autres que la maladie d'Alzheimer pouvant provoquer des symptômes similaires, tels que des accidents vasculaires cérébraux, des blessures physiques ou des tumeurs. Les nouvelles techniques d'imagerie peuvent aider à détecter des modifications cérébrales spécifiques provoquées par la maladie d'Alzheimer, mais elles sont principalement utilisées dans les grands centres médicaux ou dans le cadre d'essais cliniques.

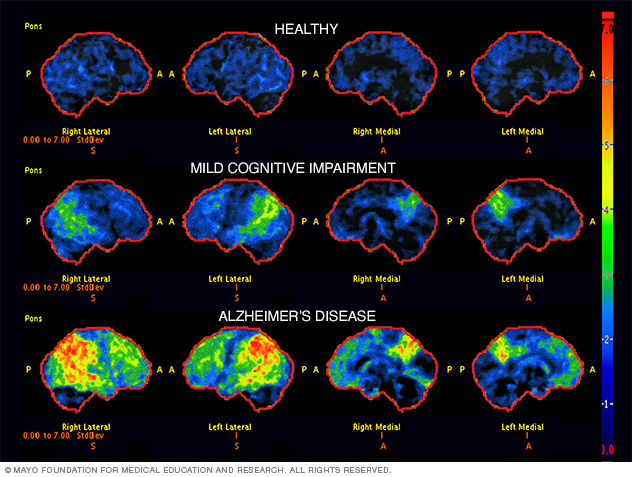
****

Figure 1: Images de scanners cérébraux

Images de scanners cérébraux utilisés pour diagnostiquer la maladie d'Alzheimer Scanners cérébraux (tomographie par émission de positons au fluorodésoxyglucose) utilisés pour diagnostiquer la maladie d'Alzheimer. Les scanners montrent un cerveau normal, un cerveau présentant de légers troubles cognitifs et un cerveau atteint de la maladie d'Alzheimer. Les zones noires et bleues représentent un métabolisme cérébral sain. Tandis que les zones vertes, jaunes et rouges représentent la détérioration du métabolisme cérébral à mesure que la maladie progresse [2]

## Les techniques d’imagerie de la structure cérébrale :

### Tomodensitométrie

 La tomodensitométrie est une technique de radiographie spécialisée qui produit des images transversales du cerveau. Il est généralement utilisé pour exclure les tumeurs, les accidents vasculaires cérébraux et les traumatismes crâniens.

### La tomographie par émission de positons :

Peut prendre des photos du processus de progression de la maladie. Lors d'une tomographie par émission de positons (TEP), un traceur radioactif de faible niveau est injecté dans le sang pour détecter certaines caractéristiques du cerveau. Une tomographie par émission de positons peut inclure les éléments suivants :

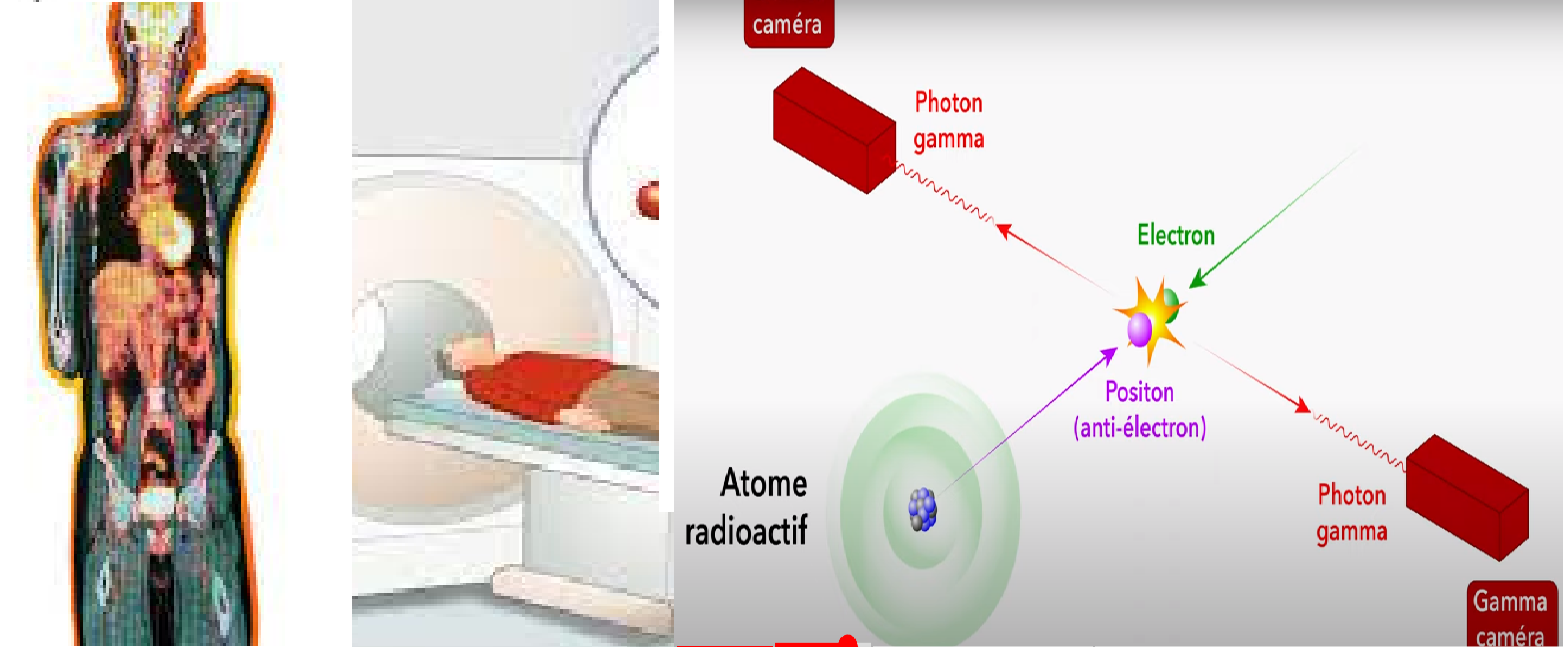


Figure 2: image par émission de positon

* + - * 1. ***Tomographie par émission de positons utilisant du fluorodésoxyglucose*** qui montre les zones du cerveau où le métabolisme des nutriments est altéré. La détection de modèles dans les zones de métabolisme réduit peut aider à différencier la maladie d'Alzheimer des autres types de démence.
        2. ***Tomographie par émission de positons amyloïd***:

Qui permet de mesurer la charge des dépôts amyloïdes dans le cerveau. Ce test est principalement utilisé en recherche, mais il peut également être utilisé si une personne présente des symptômes inhabituels ou une apparition très précoce de démence.

* + - * 1. **Tomographie par émission de positons utilisant la protéine tau**:

Qui mesure les enchevêtrements dans le cerveau et est couramment utilisée en recherche.

### Imagerie par résonance magnétique :

Une IRM utilise des ondes radio et un champ magnétique puissant pour créer des images détaillées du cerveau. Les examens IRM peuvent montrer un rétrécissement de certaines zones du cerveau associées à la maladie d'Alzheimer, mais ils excluent également d'autres conditions. En général, une IRM est préférable à une tomodensitométrie pour évaluer la démence.

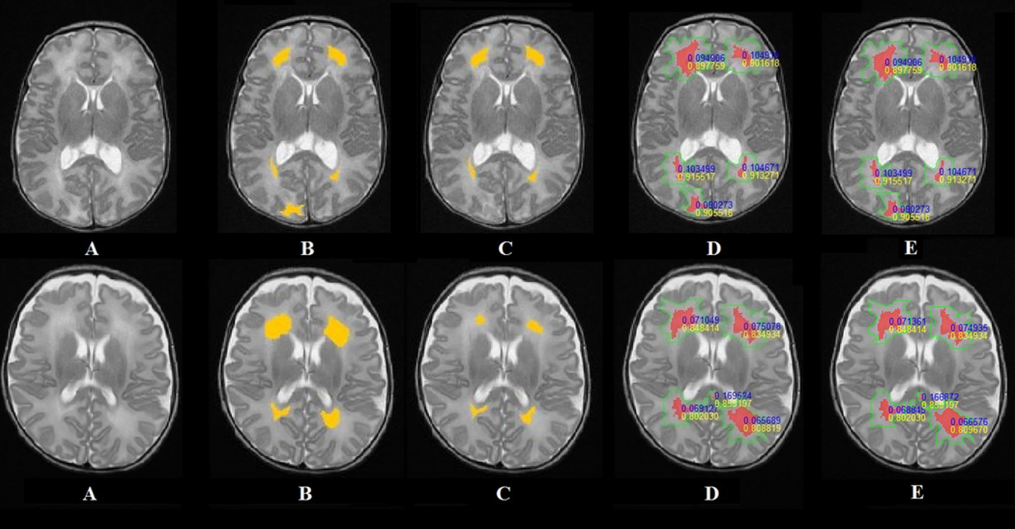


Figure 3: Images IRM [19]

## Applications de Deep learning dans le domaine médical

L'application du deep learning dans le diagnostic médical a révolutionné la manière dont les professionnels de la santé traitent et diagnostiquent les maladies. Voici quelques domaines clés où le deep learning est largement utilisé :

### Imagerie médicale :

Le deep learning est utilisé pour interpréter les images médicales telles que les radiographies, les IRM et les scanners CT. Les réseaux de neurones profonds peuvent détecter des anomalies et aider à diagnostiquer des maladies telles que le cancer, les fractures osseuses et les maladies cardiaques.

### Pathologie numérique :

En pathologie, le deep learning peut être utilisé pour analyser les échantillons de tissus et détecter les cellules cancéreuses. Cela peut aider les pathologistes à diagnostiquer plus précisément les cancers et à déterminer le meilleur plan de traitement.

### Diagnostic assisté par ordinateur :

Les systèmes de diagnostic assisté par ordinateur utilisent le deep Learning pour analyser les données médicales et fournir des recommandations aux médecins. Cela peut être utile dans le diagnostic de maladies complexes et rares.

### Prédiction de maladies :

En utilisant des données médicales et des antécédents de patients, le deep learning peut aider à prédire le risque de développer certaines maladies. Cela peut permettre une intervention précoce et une gestion plus efficace des maladies.

### Personnalisation du traitement :

Le deep learning peut être utilisé pour analyser les données génétiques et médicales des patients afin de personnaliser les traitements. Cela peut conduire à des traitements plus efficaces et à des résultats meilleurs pour les patients.

### Détection de maladies à un stade précoce :

En analysant de grandes quantités de données, le deep learning peut aider à détecter les signes précoces de maladies telles que le cancer, les maladies cardiaques et les maladies neurodégénératives, ce qui peut permettre un traitement plus précoce et plus efficace.

**En résumé**: le deep learning a le potentiel de transformer le diagnostic médical en permettant des diagnostics plus précis, des traitements plus personnalisés et une meilleure gestion des maladies.

# FONDEMENTS THEORIQUES DE LA MALADIE

## Les marqueurs de l’Alzheimer

La maladie d'Alzheimer est un trouble cérébral qui s'aggrave avec le temps. Elle se caractérise par des modifications dans le cerveau qui entraînent des dépôts de certaines protéines. La maladie d'Alzheimer provoque le rétrécissement du cerveau et la mort des cellules cérébrales. La maladie d'Alzheimer est la cause la plus fréquente de démence : un déclin progressif de la mémoire, de la pensée, du comportement et des compétences sociales. Ces changements affectent la capacité d’une personne à fonctionner.

Sur les quelque 55 millions de personnes atteintes de démence dans le monde, 60 à 70 % seraient atteintes de la maladie d'Alzheimer.

Les premiers signes de la maladie comprennent l’oubli d’événements ou de conversations récents. Au fil du temps, cela évolue vers de graves problèmes de mémoire et une perte de la capacité d’effectuer les tâches quotidiennes.

Les médicaments peuvent améliorer ou ralentir la progression des symptômes. Les programmes et services peuvent aider à soutenir les personnes atteintes de la maladie et leurs soignants.

Il n’existe aucun traitement qui guérisse la maladie d’Alzheimer. Aux stades avancés, une perte grave des fonctions cérébrales peut entraîner une déshydratation, une malnutrition ou une infection. Ces complications peuvent entraîner la mort.

## Symptômes

La perte de mémoire est le symptôme central de la maladie d'Alzheimer. Les premiers signes se manifestent par des difficultés à se rappeler des événements ou des conversations récentes. Cependant, la mémoire continue de se détériorer et d'autres symptômes apparaissent à mesure que la maladie progresse [3].

Dans les premiers stades, une personne atteinte d'Alzheimer peut être consciente de ses difficultés à se souvenir des choses et à penser clairement. Au fur et à mesure que les symptômes s'aggravent, ces problèmes deviennent plus évidents pour les membres de la famille ou les amis.

Les changements cérébraux liés à la maladie d'Alzheimer entraînent des problèmes croissants avec :

### Mémoire :

La perte de mémoire associée à la maladie d'Alzheimer persiste et s'aggrave progressivement, affectant la capacité de fonctionner au travail ou à la maison. Les personnes atteintes de la maladie peuvent

* Répéter les mêmes déclarations et questions.
* Oublier des conversations, des rendez-vous ou des événements.
* Égarer des objets dans des endroits inhabituels.
* Se perdre dans des endroits familiers.
* Oublier les noms des membres de la famille et des objets du quotidien***.***

Et avoir du mal à trouver les mots justes pour exprimer leurs pensées ou participer à des conversations.

### Penser et raisonner

La maladie d'Alzheimer entraîne des difficultés de concentration et de réflexion, notamment sur des concepts abstraits comme les nombres. Effectuer plusieurs tâches à la fois devient difficile. Gérer ses finances, équilibrer ses chéquiers et payer ses factures à temps deviennent des défis. Finalement, la personne peut être incapable de reconnaître et de gérer les chiffres.

### Porter des jugements et des décisions

La maladie d'Alzheimer entraîne un déclin de la capacité à prendre des décisions et à porter des jugements judicieux dans les situations quotidiennes. Par exemple, une personne peut faire de mauvais choix dans un contexte social ou porter des vêtements adaptés au mauvais type de temps. Il peut devenir plus difficile pour quelqu’un de réagir aux problèmes quotidiens. Par exemple, la personne peut ne pas savoir comment gérer les aliments qui brûlent sur la cuisinière ou prendre des décisions lorsqu'elle conduit.

#### Planifier et effectuer des tâches familières

Les activités quotidiennes qui exigent une série d'étapes deviennent un défi. Cela peut inclure la planification et la préparation des repas, ou même jouer à un jeu préféré. À un stade avancé, les personnes atteintes d'Alzheimer peuvent oublier comment effectuer des tâches de base telles que s'habiller ou se baigner.

#### Changements de personnalité et de comportement

Les changements cérébraux qui surviennent dans la maladie d'Alzheimer peuvent affecter l'humeur et les comportements. Les problèmes peuvent inclure les éléments suivants :

* Dépression.
* Perte d'intérêt pour les activités.
* Retrait social.
* Sautes d’humeur.
* Méfiance envers les autres.
* Colère ou agressivité.
* Changements dans les habitudes de sommeil.
* Errant.
* Perte des inhibitions.
* Délires, comme croire que quelque chose a été volé.

### Causes

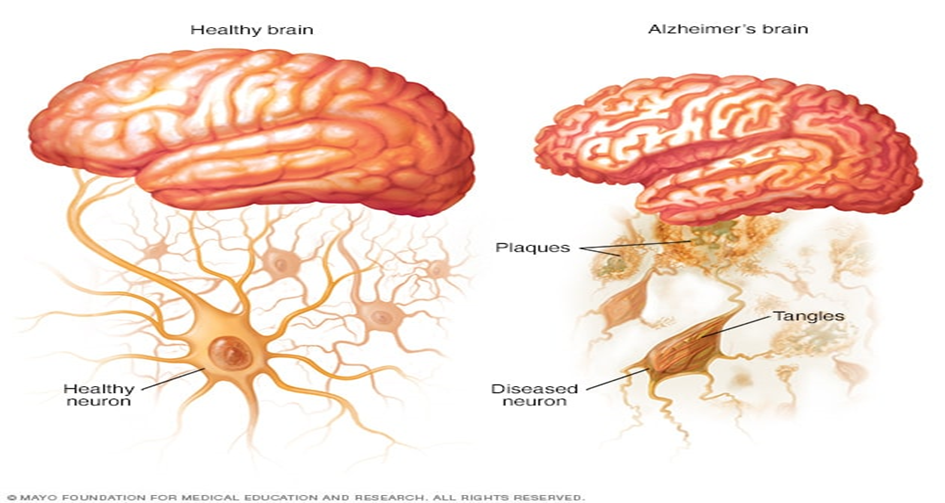
****Les causes exactes de la maladie d’Alzheimer ne sont pas entièrement comprises. Mais à la base, les protéines cérébrales ne fonctionnent pas comme d’habitude. Cela perturbe le travail des cellules cérébrales, également appelées neurones, et déclenche une série d'événements. Les neurones sont endommagés et perdent leurs connexions entre eux. Ils finissent par mourir [3] .

Figure 4:Apparition d'un cerveau atteint d'Alzheimer [3]

Les scientifiques pensent que chez la plupart des gens, la maladie d'Alzheimer est causée par une combinaison de facteurs génétiques, liés au mode de vie et à l'environnement qui affectent le cerveau au fil du temps. Dans moins de 1 % des cas, la maladie d'Alzheimer est causée par des modifications génétiques spécifiques qui garantissent presque qu'une personne développera la maladie. Dans ces cas, la maladie débute généralement à l’âge mûr.

Le développement de la maladie commence des années avant les premiers symptômes. Les dommages commencent le plus souvent dans la région du cerveau qui contrôle la mémoire. La perte de neurones se propage selon un schéma quelque peu prévisible à d’autres régions du cerveau. Au stade avancé de la maladie, le cerveau a considérablement rétréci.

Les chercheurs qui tentent de comprendre la cause de la maladie d’Alzheimer se concentrent sur le rôle de deux protéines :

* + Plaquettes. La bêta-amyloïde est un fragment d’une protéine plus grosse. Lorsque ces fragments s’agglutinent, ils semblent avoir un effet toxique sur les neurones et perturber la communication entre les cellules cérébrales. Ces amas forment des dépôts plus importants appelés plaques amyloïdes, qui comprennent également d'autres débris cellulaires.
  + Des enchevêtrements. Les protéines Tau jouent un rôle dans le système de soutien et de transport interne d'une cellule cérébrale pour transporter les nutriments et autres matières essentielles. Dans la maladie d'Alzheimer, les protéines tau changent de forme et s'organisent en structures appelées enchevêtrements neurofibrillaires. Les enchevêtrements perturbent le système de transport et endommagent les cellules.

**CHAPITRE II**

**LE DEEP LEARNING POUR DETECTION D'ALZHEIMER**

# DEEP LEARNING ET RESEAUX NEURONES

## Notions fondamentales :

***Le deep learning, ou apprentissage profond, est une sous-discipline de l'apprentissage automatique (machine learning) qui se concentre sur l'utilisation de réseaux de neurones artificiels pour modéliser et résoudre des problèmes complexes d'apprentissage automatique.***

Les réseaux de neurones profonds sont caractérisés par leur capacité à apprendre des représentations de données à plusieurs niveaux, en utilisant une architecture composée de nombreuses couches de neurones artificiels. Chaque couche extrait des caractéristiques des données d'entrée et les transmet à la couche suivante, permettant au réseau de découvrir des motifs et des abstractions de plus en plus complexes.

Le deep learning a été révolutionnaire dans de nombreux domaines, notamment la vision par ordinateur, la reconnaissance vocale, la traduction automatique et bien d'autres, en permettant aux ordinateurs d'effectuer des tâches qui étaient auparavant considérées comme difficiles voire impossibles pour les machines.

### Types de Réseaux de Neurones en Deep Learning

Les réseaux de neurones utilisés en deep learning peuvent être classés en plusieurs types en fonction de leur architecture et de leur application [12] :

#### Réseaux de Neurones Convolutifs (CNN)

Un réseau neuronal convolutif (CNN) est une architecture de réseau pour l'apprentissage en profondeur qui apprend directement à partir des données. Les CNN sont particulièrement utiles pour trouver des modèles dans les images afin de reconnaître des objets. Ils peuvent également être très efficaces pour classer les données non-image telles que les données audio, les séries chronologiques et les signaux [12] .

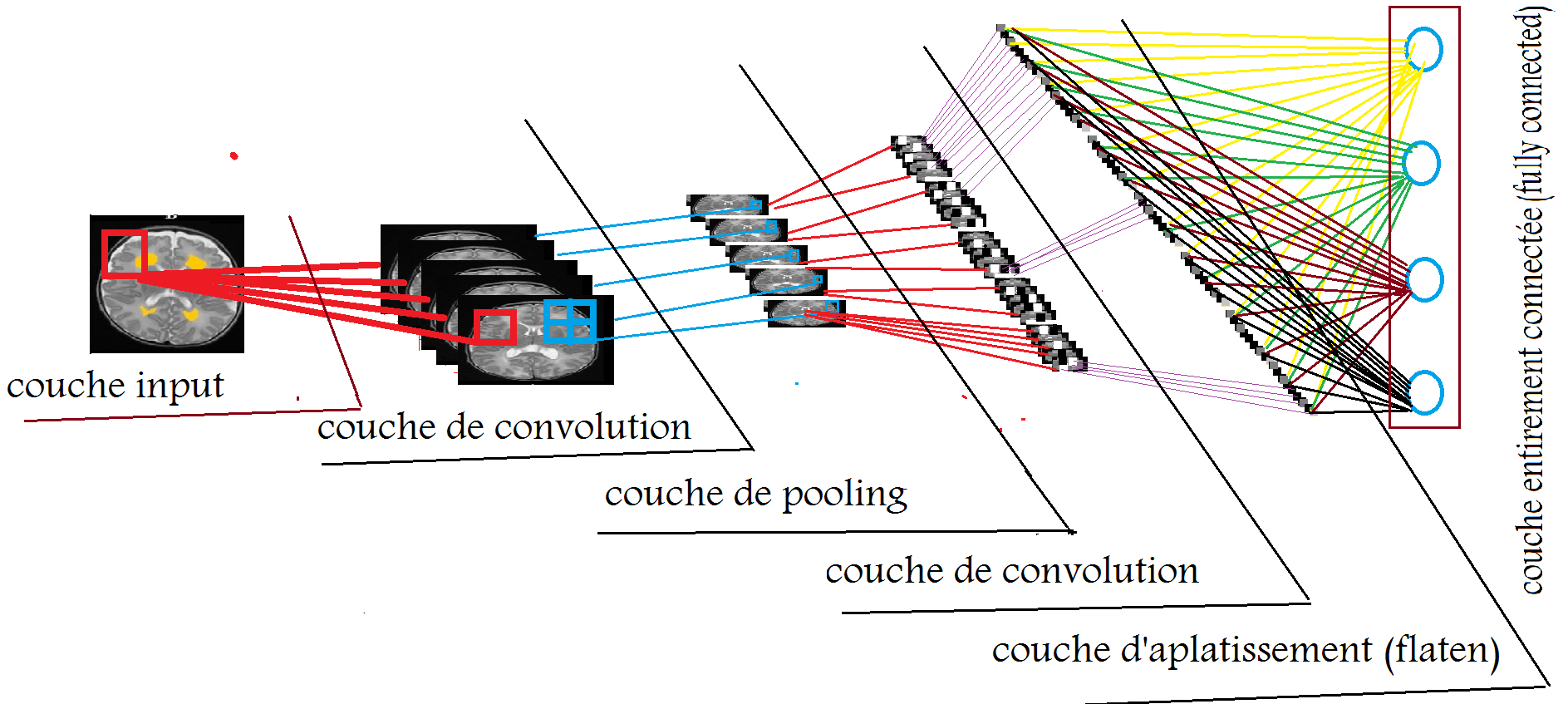


Figure 5: réseaux CNN

##### Kernel ou filtre ou détecteur de caractéristiques :

Dans un réseau de neurones convolutifs, le noyau n'est rien d'autre qu'un filtre utilisé pour extraire les caractéristiques des images**.**

taille de sortie= (i-k) +1 1

(I: Taille de l'entrée , K : Taille du Kernel)

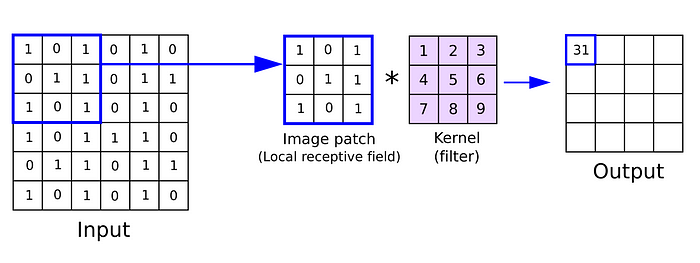


Figure 6:convolution

1. **Stride (Foulée)**

La foulée est un paramètre du filtre du réseau neuronal qui modifie la quantité de mouvement sur l'image ou la vidéo. Nous avions la foulée 1 donc ça va prendre un par un. Si nous donnons la foulée 2 alors elle prendra de la valeur en sautant les 2 pixels suivants.

TAILLE DE SORTIE = +1 2

(i : Taille de l'entrée, K : Taille du kernel, S : Stride)

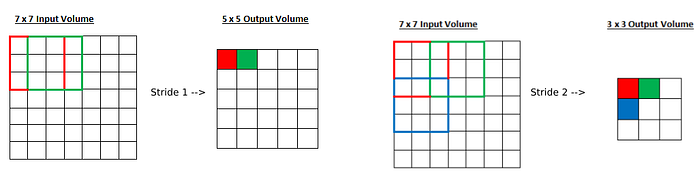


Figure 7:stride

1. **Padding**

Le remplissage est un terme pertinent pour les réseaux de neurones convolutifs car il fait référence au nombre de pixels ajoutés à une image lorsqu'elle est traitée par le noyau d'un CNN. Par exemple, si le remplissage dans un CNN est défini sur zéro, alors chaque valeur de pixel ajoutée aura une valeur nulle. Lorsque nous utilisons le filtre ou le noyau pour numériser l'image, la taille de l'image deviendra plus petite. Nous devons éviter cela car nous voulons conserver la taille d'origine de l'image pour extraire certaines fonctionnalités de bas niveau. Par conséquent, nous ajouterons quelques pixels supplémentaires en dehors de l’image.

TAILLE DE SORTIE = +1 3

(i : Taille de l'entrée, K : Taille du noyau, S : Stride, p : Padding)

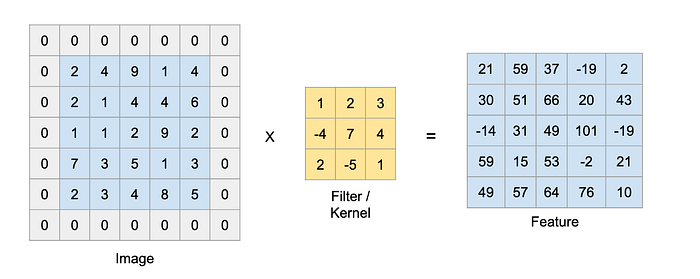


Figure 8: padding

1. **Le pooling**

Le pooling dans des réseaux de neurones convolutifs est une technique pourgénéraliser les fonctionnalités extraites par des filtres convolutifset aider le réseau à reconnaître les caractéristiques indépendamment de leur emplacement dans l'image.

1. **flatten (Aplatir)**

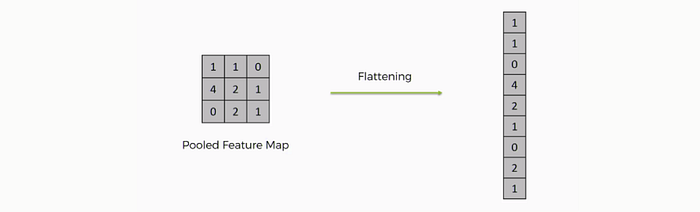
L'aplatissement est utilisé pour convertir tous les tableaux bidimensionnels résultants des cartes de caractéristiques regroupées en un seul long vecteur linéaire continu. La matrice aplatie est introduite en entrée de la couche entièrement connecté pour classer l'image.

Figure 9: couche flaten/aplatissement

##### Couches utilisées pour créer CNN

Les réseaux de neurones convolutifs se distinguent des autres réseaux de neurones par leurs performances supérieures avec les entrées de signaux d'image, de parole ou audio. Ils comportent trois principaux types de couches, à savoir :

* Couche convolutive
* Couche de mise en commun (pooling)
* Couche entièrement connectée (FC)

1. **Couche convolutif**

Cette couche est la première couche utilisée pour extraire les différentes caractéristiques des images d'entrée. Dans cette couche, nous utilisons une méthode de filtre ou de noyau pour extraire les caractéristiques de l'image d'entrée.

|  |
| --- |
|  |

………………………………………….4

1. **Couche pooling**

L'objectif principal de cette couche est de diminuer la taille de la carte de caractéristiques convoluées afin de réduire les coûts de calcul. Ceci est effectué en diminuant les connexions entre les couches et en opérant indépendamment sur chaque carte de caractéristiques. Selon la méthode utilisée, il existe plusieurs types d’opérations de Pooling. Nous avons une mise en commun maximale et une mise en commun moyenne.

|  |
| --- |
|  |

……………………………5

1. **Couche fully connected (entièrement connectée)**

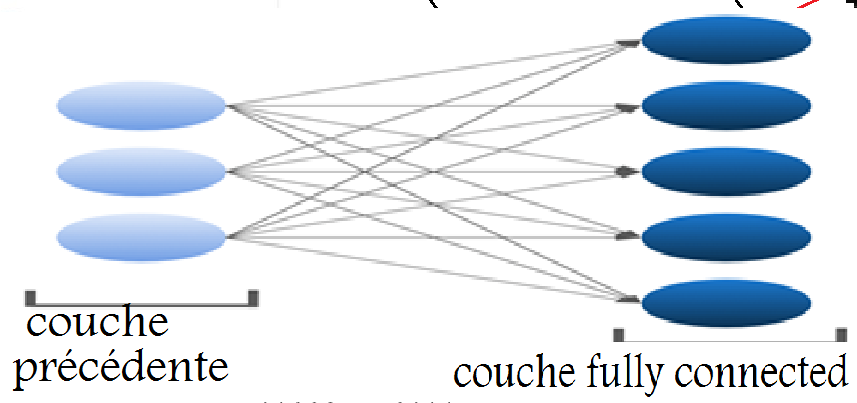


Figure 10: Couche fully connected

La couche entièrement connectée (FC) comprend les poids et les biais ainsi que les neurones et est utilisée pour connecter les neurones entre deux couches différentes. Ces couches sont généralement placées avant la couche de sortie et forment les dernières couches d'une architecture CNN.

1. **dropout**

Une autre caractéristique typique des CNN est une couche Dropout. La couche Dropout est un masque qui annule la contribution de certains neurones vers la couche suivante et laisse inchangés tous les autres.

Nous pouvons développer des outils faciles à utiliser que nous mettons au service des médecins pour les aider à diagnostiquer la maladie et son degré. Pour cela, nous suggérons d'utiliser le Raspberry Pi car il prend en charge le langage Python, qui à son tour prend en charge l'apprentissage profond. .

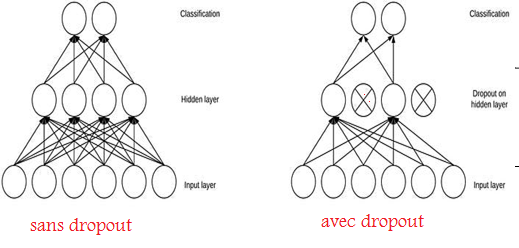


Figure 11: Couche dropout

1. **Fonction d'activation**

Une fonction d'activation décide si un neurone doit être activé ou non. Cela signifie qu'il décidera si l'entrée du neurone dans le réseau est importante ou non dans le processus de prédiction. Il existe plusieurs fonctions d'activation couramment utilisées telles que les fonctions ReLU, Softmax, tanH et Sigmoid. Chacune de ces fonctions à un usage spécifique.

* **Sigmoïde**(10) : Pour une classification binaire dans le modèle CNN ,Lorsque la fonction d'activation d'un neurone est une fonction sigmoïde, c'est une garantie que la sortie de cette unité sera toujours comprise entre 0 et 1.

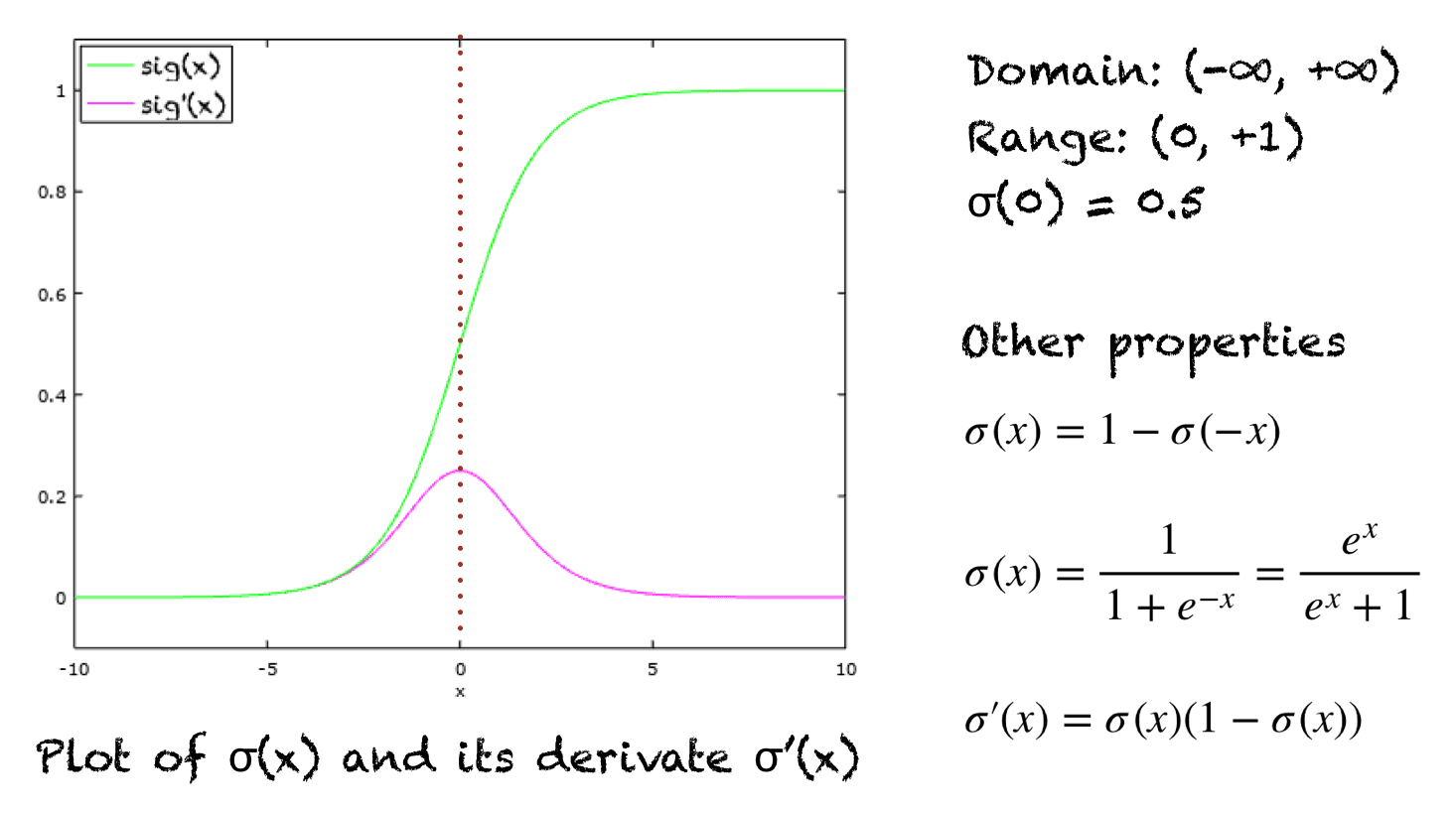


Figure 12: fonction segmoide

* **tanH :** La fonction tanh est très similaire à la fonction sigmoïde. La seule différence est qu’elle est symétrique autour de l’origine. La plage de valeurs, dans ce cas, va de -1 à 1.

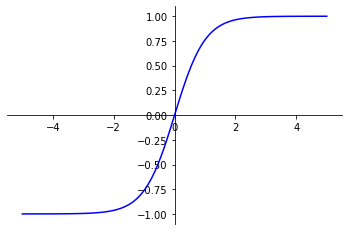


Figure 13:fonction d’activation tanH

* **Softmax** : Il est utilisé dans la régression logistique multinomiale et est souvent utilisé comme dernière fonction d'activation d'un réseau neuronal pour normaliser la sortie d'un réseau selon une distribution de probabilité sur les classes de sortie prédites.

…………….6

* **RelU** : le principal avantage de l'utilisation de la fonction ReLU par rapport aux autres fonctions d'activation est qu'elle n'active pas tous les neurones en même temps.

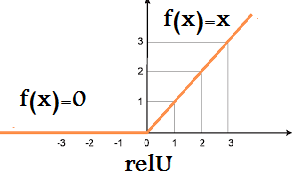


Figure 14: relU

#### Réseaux de Neurones Récurrents (RNN)

Les RNN sont adaptés aux données séquentielles telles que le texte et les séries temporelles. Ils ont des connexions récurrentes qui leur permettent de conserver des informations sur les étapes précédentes.

* **Neurones avec Boucles** : Permettent de traiter des séquences de données.
* **LSTM et GRU** : Variants des RNN qui gèrent mieux le problème de la vanishing gradient en stockant des informations sur de longues séquences.

#### Réseaux de Neurones de Type Transformer

Les modèles Transformer ont révolutionné le traitement du langage naturel. Ils utilisent des mécanismes d'attention pour traiter les données en parallèle et capturer les relations à longue distance.

* **Mécanisme d'Attention** : Permet de concentrer l’attention sur des parties spécifiques de l'entrée.
* **Encoders et Decoders** : Structures utilisées pour comprendre et générer des séquences.

#### Algorithmes d'Optimisation

Ces algorithmes ajustent les poids du réseau pour minimiser la fonction de perte [17] [7] .

* **Descente de Gradient Stochastique (SGD)** : Base de nombreux algorithmes d'optimisation.
* **Adam** : Combine les avantages de deux autres extensions de la descente de gradient, l'Adagrad et le RMSProp.

#### Techniques de Régularisation

Elles sont utilisées pour éviter le surapprentissage (overfitting).

* **Dropout** : Désactive de manière aléatoire des neurones pendant l'entraînement.
* **Batch Normalization** : Normalise les activations de couche pour accélérer l'entraînement et améliorer la stabilité.

### Classification par Deep Learning

#### Transfer learning

Le transfert learning repose sur l'utilisation d'un modèle pré-entraîné sur une base de données spécifique de grande taille et de nombreuses classes, Et le réapprendre partiellement sur d'autres dataset, même si cela diffère en termes de catégories ou de type d'images [4].

Le but est de réduire le coût en termes de temps et de ressources et permet également d'utiliser une petite **dataset**

#### Overfitting et underfitting

L’***overfitting*** et l’***underfitting*** sont deux problèmes courants dans l’apprentissage automatique qui affectent les performances du modèle.

##### Overfitting (Le surajustement) :

*Se produit lorsqu'un modèle apprend trop bien les détails et les bruits de l'ensemble de données d'entraînement, au point qu'il fonctionne exceptionnellement bien sur cet ensemble de données mais ne parvient pas à se généraliser à de nouvelles données. Autrement dit, il devient trop spécifique aux données d'entraînement et perd sa capacité à prédire correctement sur des données qu'il n'a jamais vues.*

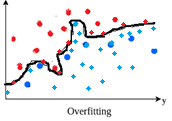


Figure 15: overfitting

1. **Causes courantes de surapprentissage :**

* Modèle trop complexe (trop de paramètres par rapport à la quantité de données)
* Trop peu de données d'entraînement
* Trop d'itérations de formation

1. **Solutions pour éviter le surajustement** :

* Réduire la complexité du modèle (par exemple, moins de couches dans un réseau neuronal, moins de profondeur dans un arbre de décision)
* Utiliser davantage de données d'entraînement
* Utiliser des techniques de régularisation (comme la régularisation L1 ou L2)
* Utiliser le dropout (pour les réseaux de neurones)

Validez le modèle avec des techniques telles que la validation croisée

##### L’underfitting (Le sous-ajustement) :

Se produit lorsqu'un modèle est trop simple pour capturer les tendances sous-jacentes des données. Dans ce cas, le modèle ne fonctionne pas bien ni sur les données d'entraînement ni sur les nouvelles données. Cela signifie qu’il n’a pas suffisamment appris des données pour faire des prédictions précises.



Figure 16: under fitting

1. **Causes courantes de sous-ajustement :**

* Modèle trop simple (pas assez de paramètres pour capturer la complexité des données)
* Données de formation insuffisantes ou mal préparées
* Trop peu d'itérations de formation

1. **Solutions pour éviter le sous-ajustement :**

* Augmenter la complexité du modèle (ajouter plus de couches ou de neurones dans un réseau neuronal, utiliser un modèle plus complexe)
* Utiliser des fonctionnalités plus pertinentes ou nombreuses
* Entraîner le modèle plus longtemps
* Préparer et nettoyer correctement les données d'entraînement

*Pour mieux comprendre, imaginez que vous essayez de prédire les prix des logements en fonction de certaines caractéristiques comme la taille, le nombre de chambres, etc.*

Si vous disposez d’un modèle très simple qui ne prend en compte que la taille de la maison (et ignore d’autres facteurs importants), vous risquez d’être sous-évalué car le modèle est trop simple pour capturer toutes les nuances qui influencent le prix.

Si vous disposez d'un modèle très complexe qui prend en compte chaque petit détail (comme la couleur de la porte d'entrée, le type de plante dans le jardin, etc.), vous risquez un sur ajustement car le modèle apprendra des détails spécifiques à l'ensemble de données d'entraînement qui ne sont pas pertinents pour prédire les prix des logements neufs.

En résumé, l'objectif est de trouver le bon équilibre où le modèle est suffisamment complexe pour capturer les tendances des données sans devenir trop spécifique aux données d'entraînement. C’est ce qu’on appelle la généralisation en apprentissage automatique.

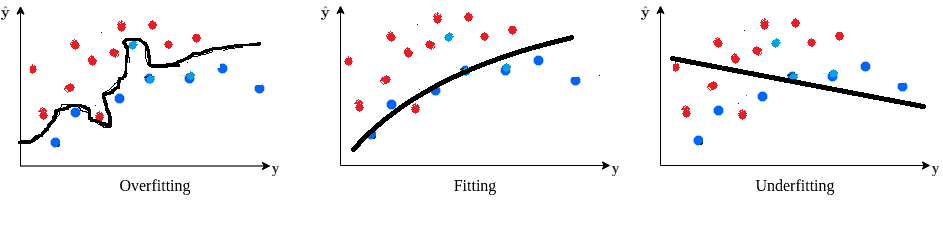


Figure 17: fitting

##### Détection du surapprentissage

Nous pouvons détecter le surapprentissage à différentes étapes du cycle de vie de l’apprentissage automatique à l’aide de divers techniques. Il est crucial d’adopter la méthode d’exclusion et de sauvegarder une partie de l’ensemble de données à des fins de test.

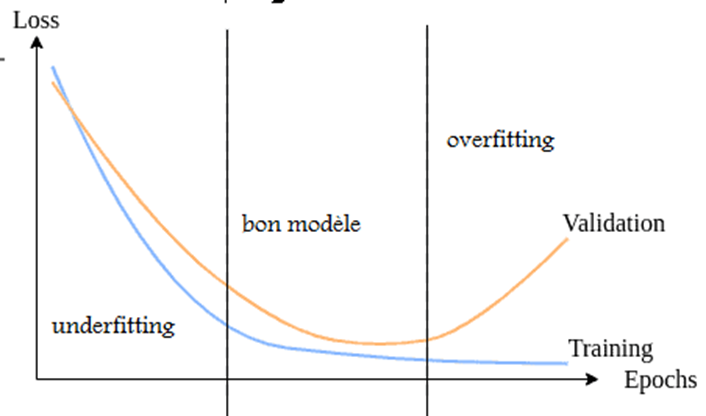


Figure 18: Detection d’overfitting

Pour déterminer quand le surapprentissage commence, nous traçons ensemble l'erreur de formation et l'erreur de validation. Au fur et à mesure que nous entraînons le modèle, nous nous attendons à ce que les deux diminuent au début. Cependant, après un certain point, l’erreur de validation augmenterait, alors que l’erreur de formation continue de diminuer. Un entraînement plus poussé après ce point conduit à un surapprentissage :

### Facteurs de Performance et Comment les Choisir

Pour obtenir de bonnes performances en deep learning, plusieurs facteurs doivent être pris en compte :

#### Qualité et Quantité des Données

Les réseaux de deep learning nécessitent de grandes quantités de données pour être efficaces. La qualité des données est également cruciale; des données bien étiquetées et diversifiées conduisent à de meilleurs modèles [15] .

La quantité de données est un facteur important pour développer un modèle d’apprentissage profond réussi. Une technique couramment utilisée en apprentissage profond pour augmenter la taille d’un ensemble de données est l’augmentation des données, qui vise à créer plusieurs exemples différents du même élément de données. Il existe de nombreuses façons, comme l'ajout de bruit aléatoire, de rotation ou même de miroirs. Ces techniques ne peuvent être appliquées sans tenir compte du manque d’influence sur les caractéristiques qui distinguent chaque catégorie des autres. [4].

#### Choix de l'Architecture

Le choix de l'architecture doit correspondre à la nature du problème à résoudre.

* **CNN pour les images**
* **RNN pour les séquences**
* **Transformer** pour les données de type séquentiel avec des relations complexes à longue distance

#### Hyperparamètres

Les hyperparamètres comme le taux d'apprentissage, la taille des batchs, et le nombre de couches doivent être soigneusement choisis et souvent ajustés à travers des techniques de validation croisée et de recherche hyperparamétrique.

#### Puissance de Calcul

Les modèles de deep learning, particulièrement ceux avec des architectures complexes, nécessitent une grande puissance de calcul. L'utilisation de GPU et de TPU est courante pour accélérer l'entraînement des modèles.

### Conclusion

Le deep learning a transformé de nombreux domaines grâce à sa capacité à apprendre des représentations complexes des données. Comprendre les différents types de réseaux de neurones, leurs composants, et les principes nécessaires à leur fonctionnement est essentiel pour développer des applications efficaces. En tenant compte des facteurs de performance et en choisissant judicieusement les hyperparamètres et l'architecture, les praticiens peuvent maximiser les performances de leurs modèles de deep learning.

# METHODOLOGIE POUR LA CLASSIFICATION

## Collecte et prétraitement des données

Pour évaluer un modèle de classification il faut prendre en considération la taille de notre dataset d’apprentissage et leur nombre de classes et le nombre d’éléments pour chaque classe, pour notre étude actuel on a une dataset contient 4 classes et autre dataset de 2 classes chaque classe contient des images \*.JPG de taille 128x128x1 pixel représenté par les figures suivantes

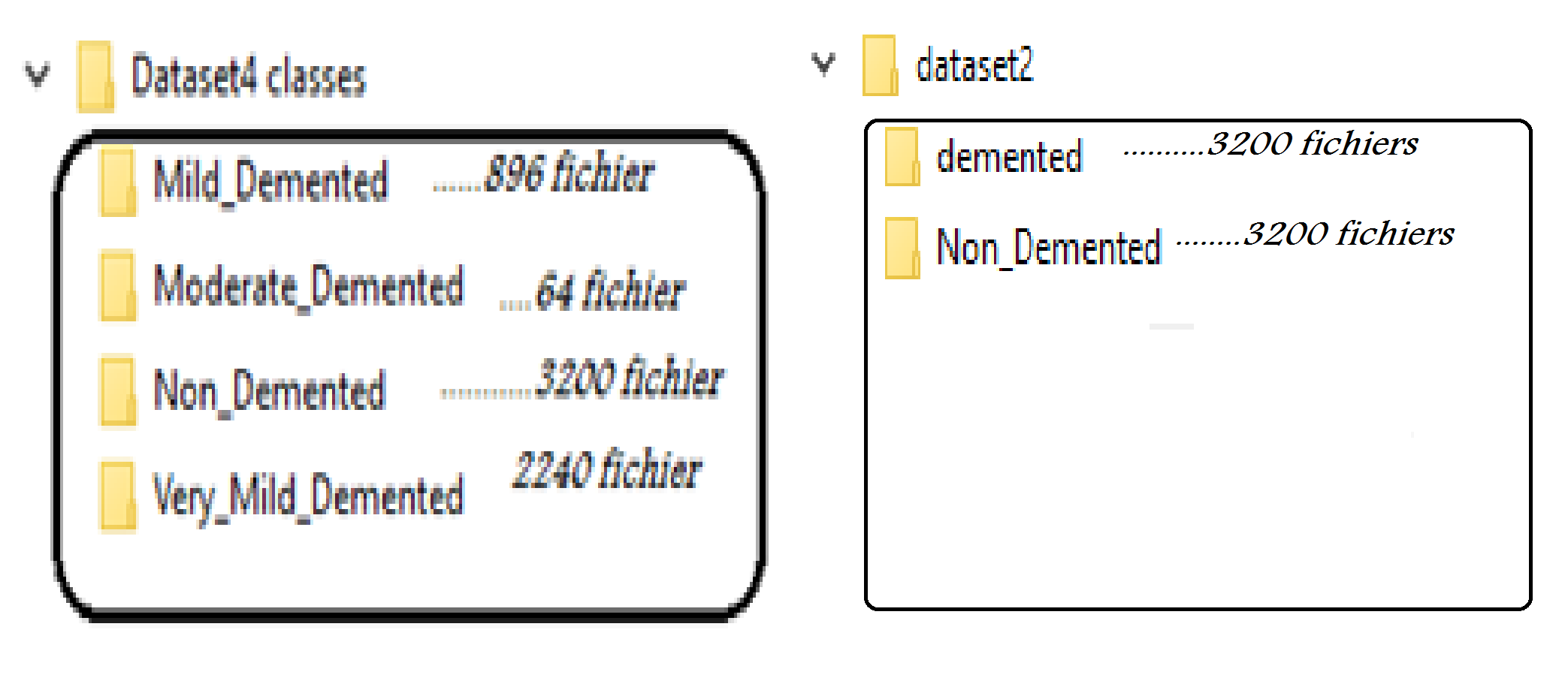


Figure 19: Dataset

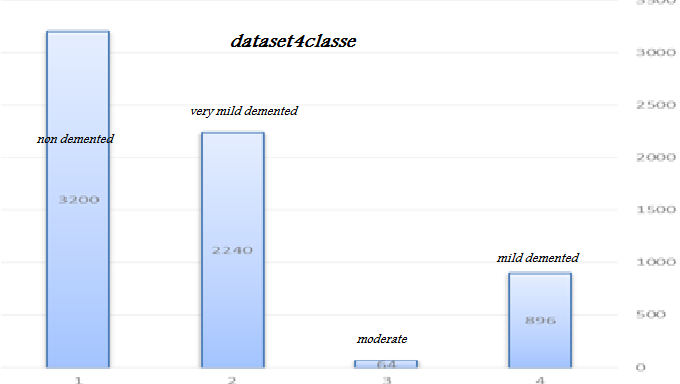


Figure 20:distribution de dataset

La division de données est une étape cruciale dans le processus de formation de modèles de deep Learning. Elle consiste à séparer un ensemble de données en ensembles distincts pour l'entraînement, la validation et l'évaluation finale du modèle. Voici un aperçu général des méthodes et des considérations couramment utilisées dans la division de données pour le deep learning :

### Ensembles d'entraînement, de validation et de test :

La division traditionnelle en trois ensembles est largement utilisée. L'ensemble d'entraînement est utilisé pour entraîner le modèle, l'ensemble de validation est utilisé pour ajuster les hyperparamètres et surveiller la performance du modèle pendant l'entraînement, et l'ensemble de test est utilisé pour évaluer la performance finale du modèle.

### Division aléatoire vs stratifiée :

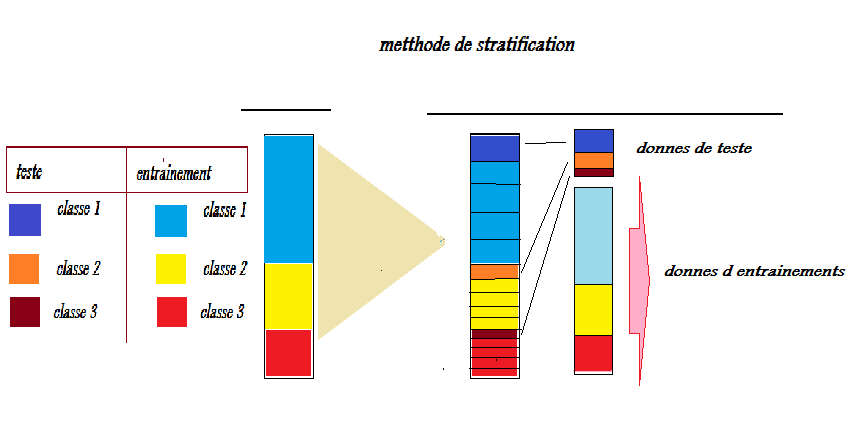
La division aléatoire consiste à répartir les données au hasard entre les ensembles, tandis que la stratification garantit une distribution équilibrée des classes dans chaque ensemble, ce qui est particulièrement important lorsque les classes sont déséquilibrées.

Figure 21: stratification

### Validation croisée :

Cette technique consiste à diviser les données en plusieurs plis et à entraîner et évaluer le modèle sur chacun d'eux de manière itérative. Cela permet une évaluation plus robuste du modèle, en particulier lorsque l'ensemble de données est petit

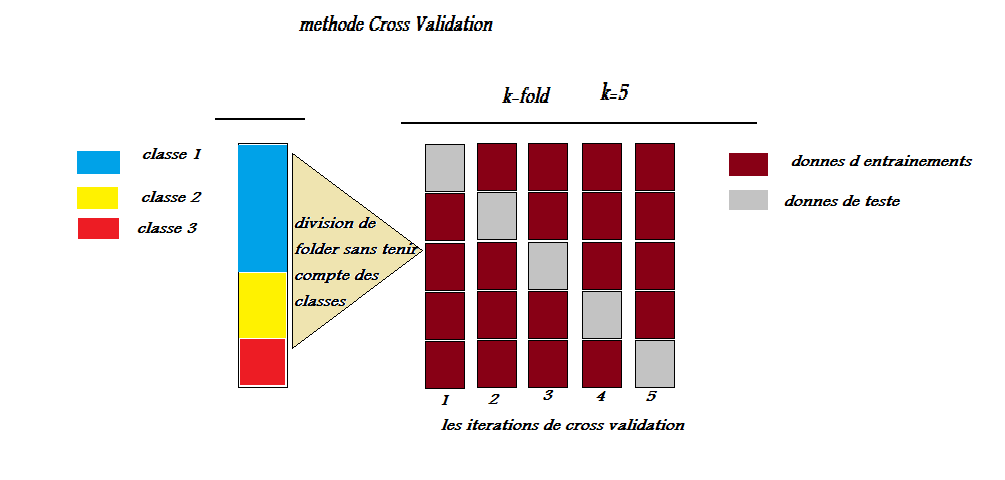
.

Figure 22: Validation Croisée

### Validation croisée stratifiée:

La validation croisée K-Fold stratifiée est une technique couramment utilisée en apprentissage automatique pour évaluer la performance d'un modèle de manière robuste, en particulier lorsque les données sont déséquilibrées en termes de répartition des classes.

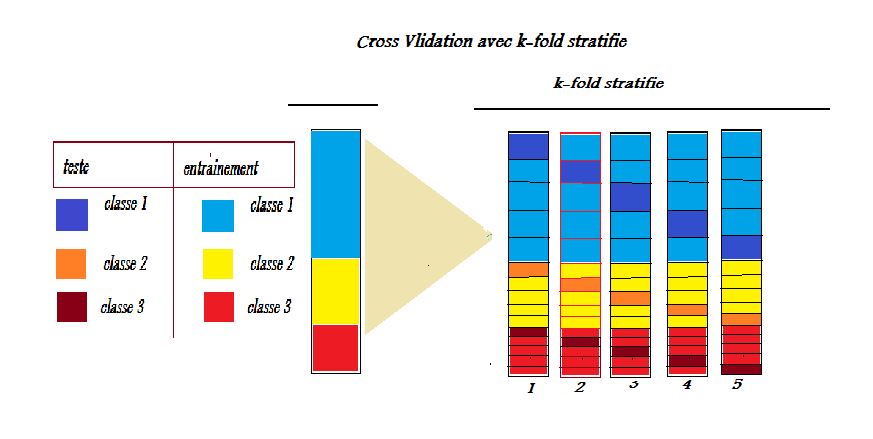


Figure 23:Validation croisée et stratifie

#### Avantage :

* 1. **Utilisation efficace des données :** Toutes les données sont utilisées à la fois pour l'entraînement et l'évaluation, ce qui permet une utilisation efficace des données disponibles.
  2. **Robustesse de l'évaluation** : En évaluant le modèle sur plusieurs partitions des données, la validation croisée K-Fold fournit une estimation plus robuste de la performance du modèle, moins sensible à la répartition spécifique des données.

#### Limitations :

* 1. **Coût computationnel** : La validation croisée K-Fold peut être plus coûteuse en termes de calcul que d'autres méthodes d'évaluation, car le modèle est entraîné K fois.
  2. **Variabilité :** La performance du modèle peut varier entre les différentes partitions des données, ce qui peut rendre l'interprétation des résultats plus complexe.

### Taille des ensembles :

La répartition des données entre les ensembles peut varier en fonction de la taille totale de l'ensemble de données. Pour de grands ensembles de données, une plus petite proportion peut être allouée à l'ensemble de test, tandis que pour de petits ensembles de données, une validation croisée peut être plus appropriée.

### Considérations sur la généralisation :

Il est important de garantir que les données dans chaque ensemble sont représentatives de la distribution réelle des données afin que le modèle puisse généraliser correctement à de nouvelles données.

En résumé, la division de données en deep learning joue un rôle crucial dans l’apprentissage et l'évaluation des modèles. Le choix de la méthode de division dépend de divers facteurs, notamment la taille de l'ensemble de données, la distribution des données, et les objectifs spécifiques du projet. L'objectif principal est de garantir que le modèle est entraîné et évalué de manière robuste et fiable.

## Méthodes d'évaluation de la performance

### Matrice de confusion

La performance d’un modèle deep Learning est directement liée à sa capacité à prédire un résultat. Lorsque l’on cherche à comparer les résultats d’un algorithme à la réalité, on utilise une matrice de confusion. Dans cet article, vous verrez comment lire cette matrice pour interpréter les résultats d’un modèle de classification.

Le deep Learning consiste à nourrir un algorithme à l’aide de données afin qu’il apprenne par lui-même à effectuer une certaine tâche. Dans les problèmes de classification, il prédit des résultats que l’on doit comparer à la réalité pour mesurer son degré de performance. On utilise généralement la matrice de confusion, appelée aussi tableau de contingence. Elle mettra non seulement en valeur les prédictions correctes et incorrectes mais nous donnera surtout un indice sur le type d’erreurs commises. Pour calculer une matrice de confusion, on a besoin d’un ensemble de données de test et un autre de validation qui contient les valeurs des résultats obtenus.

Chaque colonne du tableau contient une classe prédite par l’algorithme et les lignes des classes réelles [11].

On classe les résultats en 4 catégories :

**True Positive (TP)** : la prédiction et la valeur réelle sont positives.

Exemple : Une personne malade et prévu malade.

**True Négative (TN)** : la prédiction et la valeur réelle sont négatives.

Exemple : Une personne saine et prévu saine.

**False Positive (FP)** : la prédiction est positive alors que la valeur réelle est négative.

Exemple : Une personne saine et prévu malade.

**False Négative (FN)** : la prédiction est négative alors que la valeur réelle est négative.

***Exemple*** : Une personne malade et prévu saine.

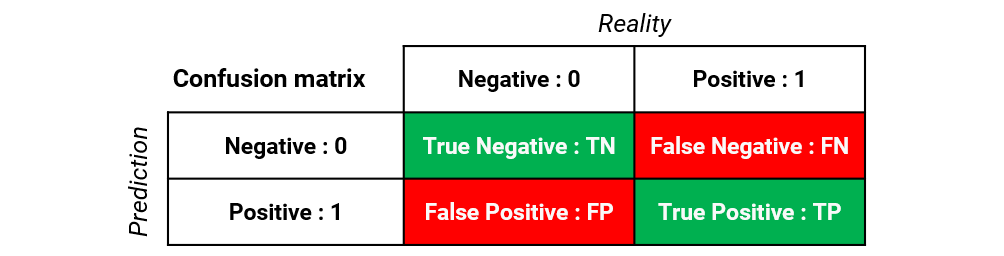


Figure 24: matrice de confusion binaire

***Exemple* :** Nous disposons de données réparties en 3200 images représentant un cas confirmé de maladie d'Alzheimer et 3200 cas confirmés négatifs de maladie d'Alzheimer.

Après avoir entraîné notre modèle, nous avons testé nos données dessus pour déterminer l'efficacité du modèle.

Nous avons essayé les 3 200 images de cas positifs pré-confirmés, et cela nous a donné 3 010 d’entre elles qui correspondaient à la pré-segmentation et 190 cas identifiés comme cas négatifs.

Nous avons essayé 3 200 images de personnes en bonne santé et nous en avons donné 3 110 comme étant en bonne santé et 90 comme malades.

Sur la base de ces données, nous obtenons la matrice de confusion suivante

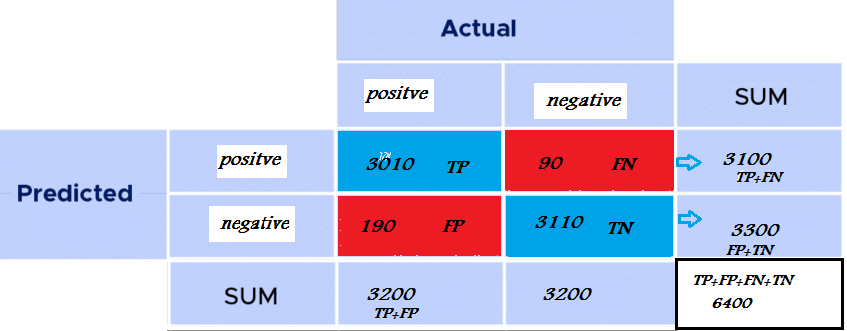


Figure 25: exemple d’une matrice de confusion

**Matrice de confusion multi-classe**

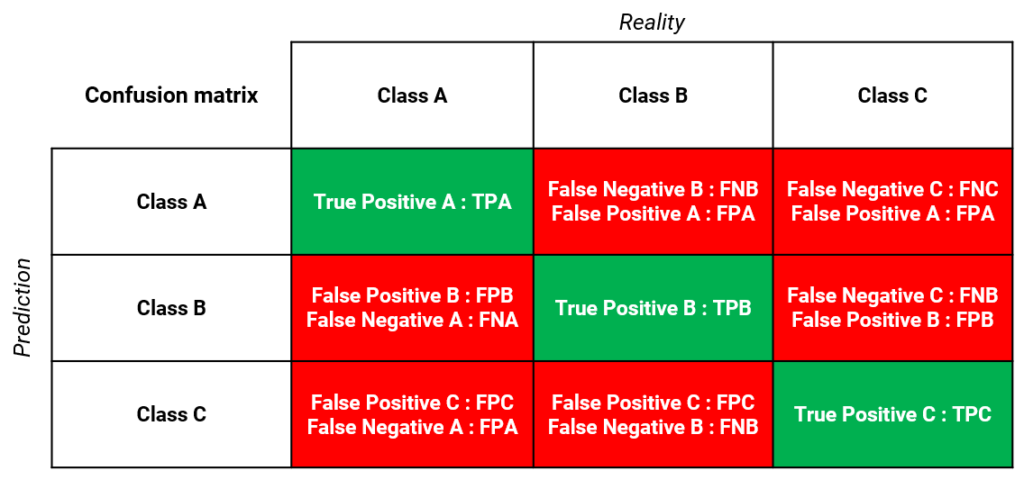


Figure 26: Matrice de confusion multi-classe théorique pour 3 classes

### Métriques

Voici une liste de certaines des métriques couramment utilisées en apprentissage automatique, accompagnées de leurs formules [6]:

1. **Accuracy :** L'exactitude est une mesure de la performance d'un modèle de classification. Elle représente le pourcentage d'exemples correctement classés parmi tous les exemples. Par exemple, si un modèle a une exactitude de 0,85, cela signifie qu'il classe correctement 85% des exemples

Accuracy= …………. 7

1. **Recall ou (Sensitivity ouTPR) :** Le TPR est synonyme de rappel et mesure la proportion d'exemples positifs correctement identifiés parmi tous les exemples réellement positifs. Il est calculé de la même manière que le rappel :

Recall=……………….8

Ces mesures sont toutes importantes pour évaluer différents aspects de la performance d'un modèle de classification binaire.​

1. **Précision (Precision) :** La précision mesure la proportion d'exemples positifs correctement identifiés parmi tous les exemples identifiés comme positifs par le modèle. Elle est calculée comme :

Precision= ​……………….9

1. **F1-score :** Le score F1 est une mesure de la précision d'un modèle de classification. Il combine à la fois la précision et le rappel (recall) en une seule métrique. Le score F1 est calculé comme la moyenne harmonique de la précision et du recall :

F1-score=2\*……………………10

Le score F1 donne une meilleure idée de la performance globale du modèle que la précision ou le rappel seuls, car il prend en compte à la fois les faux positifs et les faux négatifs**.**

1. **Spécificité :** La spécificité mesure la capacité d'un modèle à identifier correctement les exemples négatifs. Elle est calculée comme le ratio des vrais négatifs parmi tous les exemples négatifs réels :

…………………….11

Une spécificité élevée indique que le modèle est bon pour éviter de prédire à tort des exemples négatifs comme positifs**.**

1. **FALS POSITIVE RATE (FPR) :** Le FPR mesure la proportion d'exemples négatifs incorrectement identifiés comme positifs parmi tous les exemples négatifs réels. Il est calculé comme :

ET ………………12

Un faible FPR indique que le modèle a tendance à ne pas classer à tort les exemples négatifs comme positifs.

1. **AUC-ROC (Area Under the ROC Curve) :** L'AUC-ROC est calculée à partir de la courbe ROC, qui est une courbe de performance d'un modèle de classification en fonction de son seuil de discrimination. L'AUC-ROC représente la surface sous cette courbe [6] .

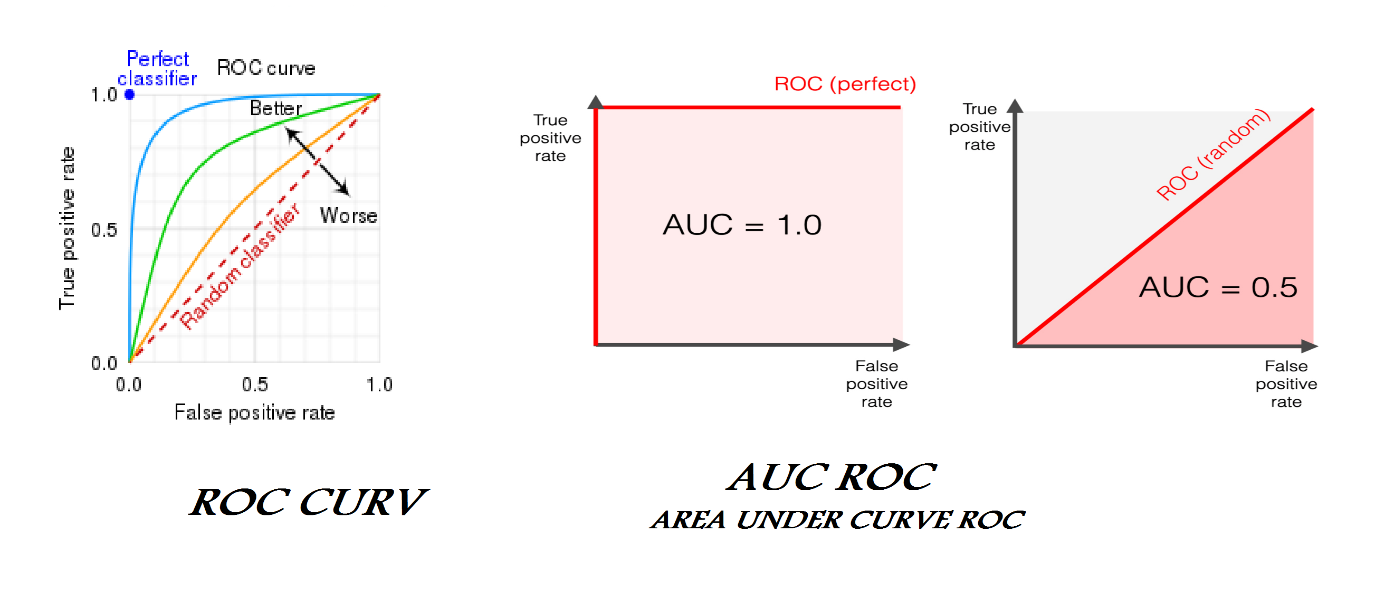


Figure 27: ROC ET AUC

1. **MSE (Mean Squared Error) :**

MSE= ……………13

1. **RMSE (Root Mean Squared Error) :**

RMSE= ​……………….14

1. **MAE (Mean Absolute Error) :**

MAE= ………………15

1. **R-squared (Coefficient de détermination) :**

*=1-………………16*

Où est la moyenne des valeurs réelles et sont les valeurs prédites par le modèle.

### Métrique multi classes

Les métriques de classification usuelles sont définies dans le cas binaire à partir de la [matrice de confusion binaire](https://kobia.fr/classification-metrics-matrice-de-confusion/).

Comment utiliser la matrice de confusion multi-classes pour généraliser les métriques binaires ?

Nous allons voir les 3 méthodes classiques :

* **L’approche “per class”** :

Pour laquelle on s’intéresse à *𝑛 𝑐𝑙𝑎𝑠𝑠* problèmes de classification binaire indépendamment. Cette approche permet de s’intéresser à chaque classe individuellement mais il est difficile d’en tirer une synthèse car on obtient une métrique par classe.

* **L’approche “macro”** :

Qui permet de résumer les métriques obtenues avec l’approche “per class”. Il existe 2 variantes “macro”, une variante classique robuste au [déséquilibre de classe](https://kobia.fr/imbalanced-data-et-machine-learning/) et une variante pondérée, plus représentative des données.

* **L’approche “micro” :**

Qui utilise directement les valeurs de la matrice de confusion multi-classe pour produire une métrique qui résume la performance du modèle. Cette approche présente des limites, que nous détaillerons, liées à la structure de la matrice de confusion.

#### Métriques (per class)

L’approche (per class) consiste à diviser une classification multi-classes en **𝑛**𝑐𝑙𝑎𝑠𝑠 classification binaire. On évalue ensuite chaque classification binaire séparément.

Pour bien comprendre, reprenons notre exemple et calculons le [recall](https://kobia.fr/classification-metrics-precision-recall/) par classe.

Pour ce faire, nous commençons par établir chaque matrice de confusion en repartant de la matrice de confusion multi-classe standard :

****

Figure 28: passage multi-classes vers bi-classes

*Passage de la matrice de confusion multi-classe à la matrice de confusion binaire****non******demented.*** *Les rectangles de couleurs représentent les mêmes individus à gauche et à droite.*

Pour construire la matrice de confusion **non demented**, on identifie dans la colonne et la ligne **non demented** les vrais positifs, faux négatifs et faux positifs. Les autres cases sont des vrais négatifs.

En appliquant cette transformation pour chaque classe, on obtient les 4 matrices de confusion binaires suivantes :

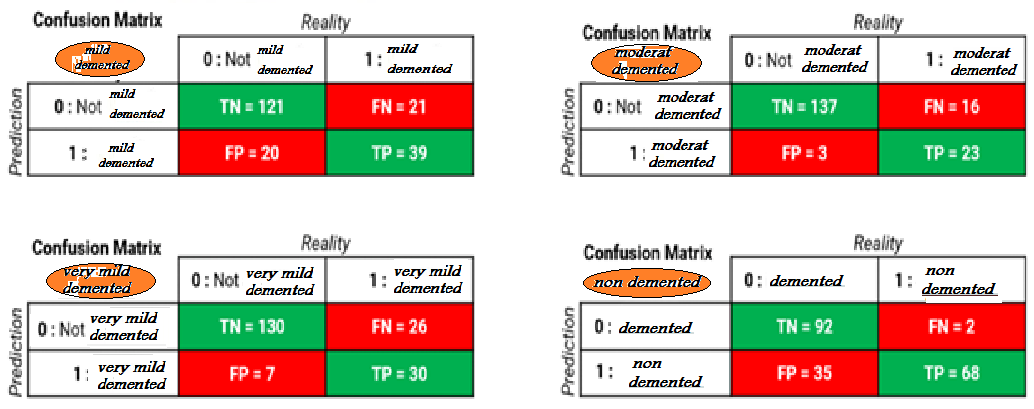
****

Figure 29:Matrices de confusion des 4 classes séparées

***(****Mild demented, very mild demented, moderate demented et non demented.) En considérant la prédiction de chaque classe indépendamment, on construit des matrices de confusion binaires classiques.*

On s’est donc ramené à quatre problèmes de classification binaire, pour lesquels on peut calculer toutes les métriques dérivées de la matrice de confusion. On obtient pour chaque métrique 4 valeurs, correspondant aux 4 classes**.**

En prenant l’exemple du recall, on obtient les 4 recall suivants :

Recall=……………….17

* mild demented: 65%
* moderat demented : 59%
* Very mild demented : 54%
* Non demented : 97%

Les images **Non demented** ont été quasiment toutes détectées par le modèle alors que seulement la moitié des situations **Very mild demented** ont été détectées.

**Les métriques (per class)** permettent de comprendre le détail des performances du modèle sur chaque classe individuellement.  Dans certains cas cependant, on peut préférer une métrique qui résume les performances sur l’ensemble des classes, pour comparer des modèles entre eux par exemple. L’approche (**per class**) est alors insuffisante et il est nécessaire de faire appel aux méthodes (**macro**) et (**micro**).

#### Métrique (macro)

**L’approche (macro)** se base sur les métriques obtenues avec l’approche (per class) pour les résumer en une seule valeur. On peut décomposer cette approche en 2 étapes :

1. Le calcul des métriques **(per class). Par exemple**, on calcule le recall pour les 4 classes d’Alzheimer.
2. Le résumé des métriques **(per class)** en calculant une moyenne**. Par exemple : le** macro-recall sera une moyenne des **recalls** de chaque classe.

Ce résumé peut se faire de 2 façons :

* ***La métrique (macro) classique* :** robuste aux données déséquilibrées
* ***La métrique (macro) pondérée*** : plus représentative de la distribution des classes

##### Métrique (macro) classique

La métrique (macro) classique revient à faire la moyenne des métriques (per class). En termes mathématiques, on obtient la formule suivante :

…………18

Reprenons l’exemple du calcul du recall, on obtient le macro-recall suivant :

Macro-recall**=**

**==** 68%

Le macro-recall est donc la moyenne des recalls de chaque classe.

La métrique macro classique accorde autant d’importance à chacune des classes, peu importe la proportion d’individus qu’elles contiennent. Cela permet de ne pas négliger une classe qui serait sous-représentée dans nos données et rend cette métrique robuste au déséquilibre des classes.

##### Métrique (macro) pondérée

***La variante (macro) pondérée*** revient à faire la moyenne pondérée des métriques (per class). Chaque métrique a une importance proportionnelle à la proportion d’individus dans la classe correspondante. Mathématiquement, cela se traduit par la formule suivante :

…19Où :

……..20

* *𝑛𝑖 désigne le nombre d’individus dans la classe𝑖.*

Le calcul du macro-weighted-recall dans notre exemple est le suivant :

**macro-weighted-recall=(prop***mild***×recall**mild**+prop**Vmild**×recall**Vmild**+prop**dement**×recall**dement**+prop***Moderat***×recall***Moderat***)**

**=27%×65%+17%×59%+25%×54%+31%×97%=71%**

*Le macro-weighted-recall* est la proportion d’individus correctement détectés. Ici, 71% des individus ont été correctement détectés.

La métrique macro-weighted accorde une importance plus grande aux classes les plus nombreuses dans les données, ce qui la rend plus représentative.

C’est un avantage lorsque les classes jouent le même rôle. Cependant si une classe a un rôle particulier et qu’elle est sous-représentée dans les données, une métrique macro-weighted ne permettra pas de détecter des changements de performance sur cette classe particulière.

##### Comparaison des 2 variantes (macro)

Quelle variante macro faut-il privilégier ? Le paramètre qui détermine le choix entre macro classique et macro-weighted est le déséquilibre des données :

* Pour des données équilibrées, la proportion de chaque classe est environ 1/𝑛𝑐𝑙𝑎𝑠𝑠 et les deux variantes sont équivalentes. Pour 4 classes, ces proportions valent toutes 25%.
* Pour des données déséquilibrées, les deux variantes ne seront pas équivalentes. Imaginons que nos images d’Alzheimer soient déséquilibrées :
* 90% d’image **mild demented**
* 4% d’image **moderat demented**
* 3% d’image **Very mild demented**
* 3% d’image **Non demented**

Dans ce cas, la métrique macro-weighted accorde une importance très forte à la classe *couverte,* qui a la proportion la plus grande.

En s’intéressant à un modèle simpliste qui prédit tout le temps la classe majoritaire (**mild demented)**, on voit que le macro-weighted recall lui donne une performance élevée alors même qu’il n’a pas extrait d’informations des données :

Dans la plupart des cas on privilégiera donc la métrique macro classique car :

* on obtient un résultat similaire à la métrique macro-weighted quand les données sont équilibrées
* on obtient un résultat plus robuste quand les données sont déséquilibrées.

#### Métrique micro

**L’approche (micro)** utilise directement la matrice de confusion multi-classe pour faire une **matrice de confusion (synthétique).** Elle reprend les termes de la matrice de confusion binaire : True Positive (TP), True Negative (TN), False Positive (FP) et False Negative (FN). Ces termes sont calculés de la manière suivante **:**



Figure 30: *Matrice de confusion synthétique*

*(Valeurs utilisées pour le calcul des métriques micro).*

En classification multi-classe, il n’y a pas de classe (**négative**) et par conséquent le terme des vrais négatifs est nul :**𝑇𝑁=0.**

Dans un problème multi-classe, toutes les prédictions fausses (cases rouges) sont à la fois un faux positif et un faux négatif. Par conséquent**𝐹𝑃=∑𝑖=1𝑛𝑐𝑙𝑎𝑠𝑠𝐹𝑃𝑖=∑𝑖=1𝑛𝑐𝑙𝑎𝑠𝑠𝐹𝑁𝑖=𝐹𝑁.**

**On fera** donc attention à ne pas sommer**𝐹𝑃**et**𝐹𝑁**lorsque l’on calcule les métriques**.**

Nous retrouvons à nouveau une matrice de confusion binaire à partir de laquelle on peut calculer les différentes métriques de classification.

L’approche “micro” s’intéresse à des proportions d’individus, car chaque somme se fait en nombre d’individus. L’approche macro-weighted, qui pondère par le nombre d’individus dans chaque classe, donne aussi des métriques représentatives de la proportion d’individus même si celles-ci diffèrent, on pourra retrouver des points communs entre ces deux approches.

Prenons l’exemple du calcul du micro-recall pour bien comprendre. On construit la matrice de confusion (**micro**) suivante :



Figure 31: *Matrice confusion micro*

Le calcul du micro-recall est le suivant **:**

…………..21

**=**

Cela signifie que dans notre exemple, **71%** des individus ont été correctement détectés. On obtient le même résultat que pour le macro-weighted-recall car dans le cas du recall, la pondération de l’approche macro-weighted compense le dénominateur. Pour les autres métriques, le dénominateur est différent et les approches micro et macro-weighted diffèrent.

##### Limites de la métrique micro

L’approche micro étant représentative des proportions d’individus comme les métriques macro-weighted, on retrouve en présence de données déséquilibrées la limite qui a déjà été développée lors de la comparaison entre macro classique et macro-weighted. L’approche micro accorde une faible importance aux classes minoritaires.

La deuxième limite de l’approche micro vient du fait que chaque individu dont la prédiction est fausse est à la fois un faux positif de sa classe prédite et un faux négatif de sa classe réelle. Par conséquent, **𝐹𝑃=𝐹𝑁** et l’approche micro ne différencie pas les métriques. La micro-precision, le micro-recall, la micro-accuracy et le micro-F1-score sont tous égaux.

**Vérifions-le sur notre exemple :**

* **Micro­­\_precision=**
* **Micro\_recall=**
* **Micro\_f1-scor=**
* **Micro\_accuracy=**

Où **𝐻** est l’opérateur moyenne harmonique et (+**𝐹𝑁**) signifie que l’on ne prend pas en compte **𝐹𝑁** dans la somme, car c’est un doublon de **𝐹𝑃**.

Pour la prédiction de la météo, la **micro-precision**, le **micro-recall**, la **micro-accuracy** et le **micro-F1-score** sont bien tous égaux

### La généralisation peut se faire de 3 façons :

#### Per class :

permet d’avoir le détail des performances sur chaque classe mais multiplie le nombre de métriques à surveiller.

#### Macro :

* + - 1. **(Macro classique)** : résume les performances obtenues par classe en restant *robuste au déséquilibre des classes.*
      2. (**macro-weighted)** : résume les performances par classe *en préservant les proportions des classes.*

#### *Micro* : résume les performances obtenues par individu et prend en compte les proportions d’individus dans chaque classe.

Pour avoir le détail des performances sur chaque classe, on utilise l’approche **(per class).** Et si à l’inverse on a besoin de résumer les performances avec une seule valeur, l’approche à privilégier est la **macro\_classiqu**

**CHAPITRE III**

**RESULTATS DU MODELE PROPOSE**

# Architecture du modèle de deep learning

### Model pré-entrainé sqeeznet

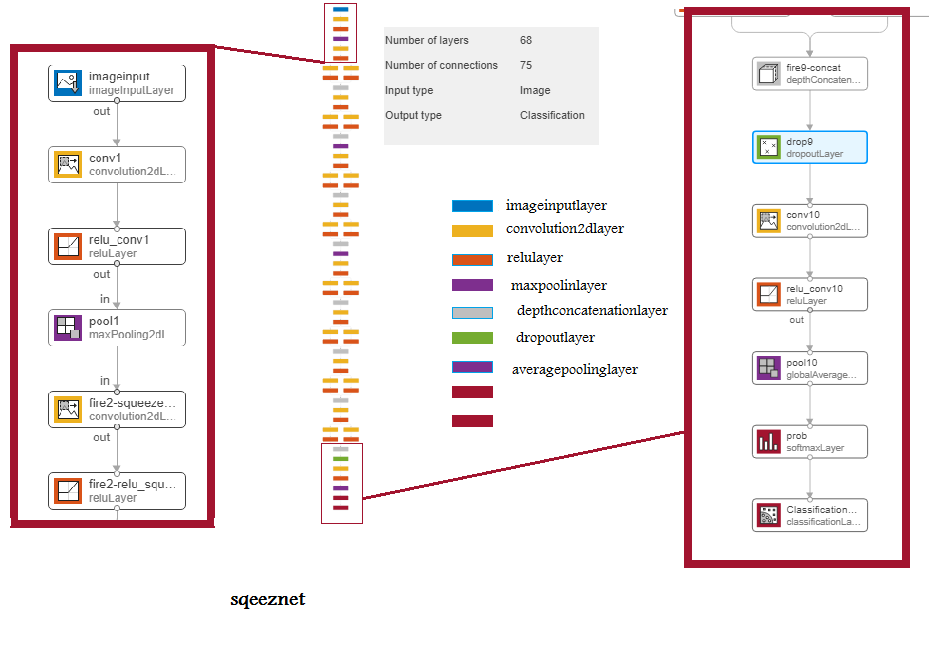


Figure 32: architecture de modèle sqeeznet

### Model proposé :

#### Schémas de modèle

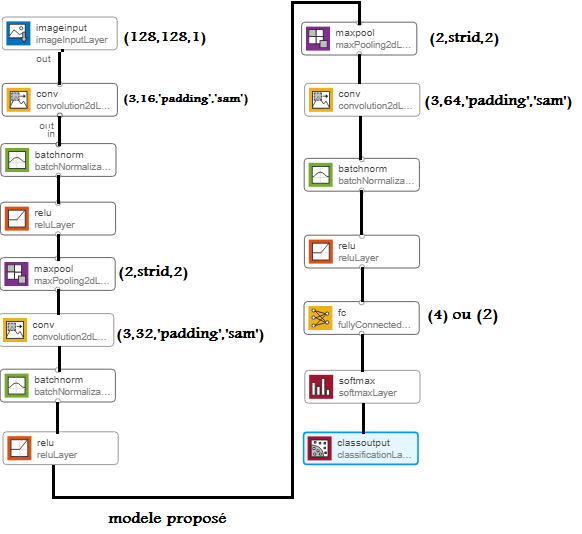


Figure 33: architecture de modèle proposé

#### Code Matlab de notre modèle :

digitDatasetPath = fullfile(matlabroot,'toolbox','nnet','nndemos','nndatasets','Dataset');

imds = imageDatastore(digitDatasetPath,'IncludeSubfolders',true,'LabelSource','foldernames');

%Répartition des données en ensembles de formation, de validation et de test

[imdsTrain, imdsValidation] = splitEachLabel(imds, 0.7, 'randomized');

% Définir l'architecture du réseau de neurones

layers = [

imageInputLayer([128 128 1])

convolution2dLayer(3,16,'Padding','same')

batchNormalizationLayer

reluLayer

maxPooling2dLayer(2,'Stride',2)

convolution2dLayer(3,32,'Padding','same')

batchNormalizationLayer

reluLayer

maxPooling2dLayer(2,'Stride',2)

convolution2dLayer(3,64,'Padding','same')

batchNormalizationLayer

reluLayer

fullyConnectedLayer(4)

softmaxLayer

classificationLayer];

% Définir les options d'entraînement

options = trainingOptions('adam', ...

'MaxEpochs',10, ...

'MiniBatchSize',32, ...

'Shuffle','every-epoch', ...

'InitialLearnRate',1e-3, ...

'ValidationData',imdsValidation, ...

'ValidationFrequency',30, ...

'Verbose',false, ...

'Plots','training-progress');

% Entraîner le réseau de neurones

net = trainNetwork(imdsTrain, layers, options);

% Évaluer le modèle sur les données de test

YPred = classify(net, imdsValidation);

YTest= categorical(imdsValidation.Labels);

# RESULTATS

## Description des données utilisées

Nous disposons d'une **dataset** originale composée de quatre paquets de fichiers d'images sensiblement inégaux en termes de taille de données. Chaque paquet représente une catégorie d'images, et ces données comportent quatre catégories réparties comme suit : Une catégorie représente des images IRM de personnes en bonne santé, et les trois catégories restantes représentent des cas malades, chaque cas avec un degré différent, que nous avons utilisés pour **entrainer** et **tester** deux modèles avec quatre sorties **Sqeeznet** et notre modèle que nous avons développé, afin de les comparer.

Dans la même **dataset**, nous l'avons réorganisé en deux classes de données comme suit : images IRM de personnes saines et autres de personnes malades, nous avons donc obtenu deux catégories équilibrées en termes de volume de données, 3200 images dans chaque catégorie. Entrainés et testés après les avoir convertis en un modèle binaire.

## Evaluation de modèle (*Squeezent*) qui entrainer sur dataset a 2 classes avec nbr des individués équilibrés :

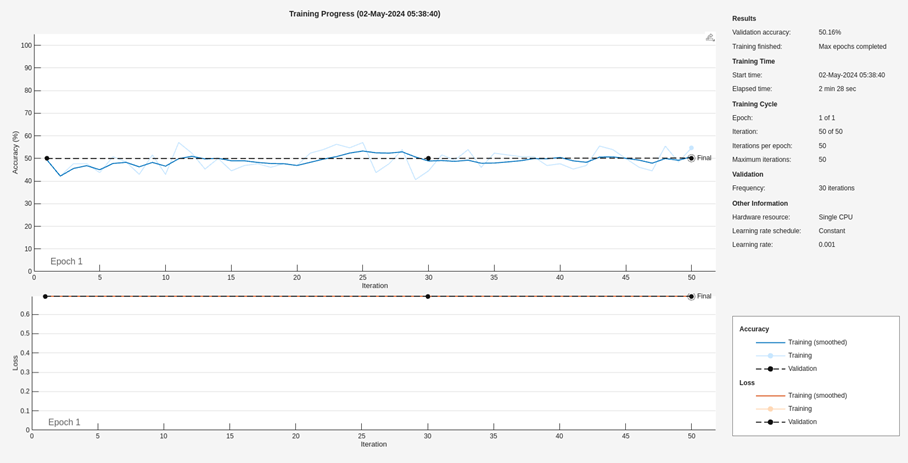
****

Figure 34: progression d’apprentissage pour sqeeznet a 2 classes

### Matrice de confusion

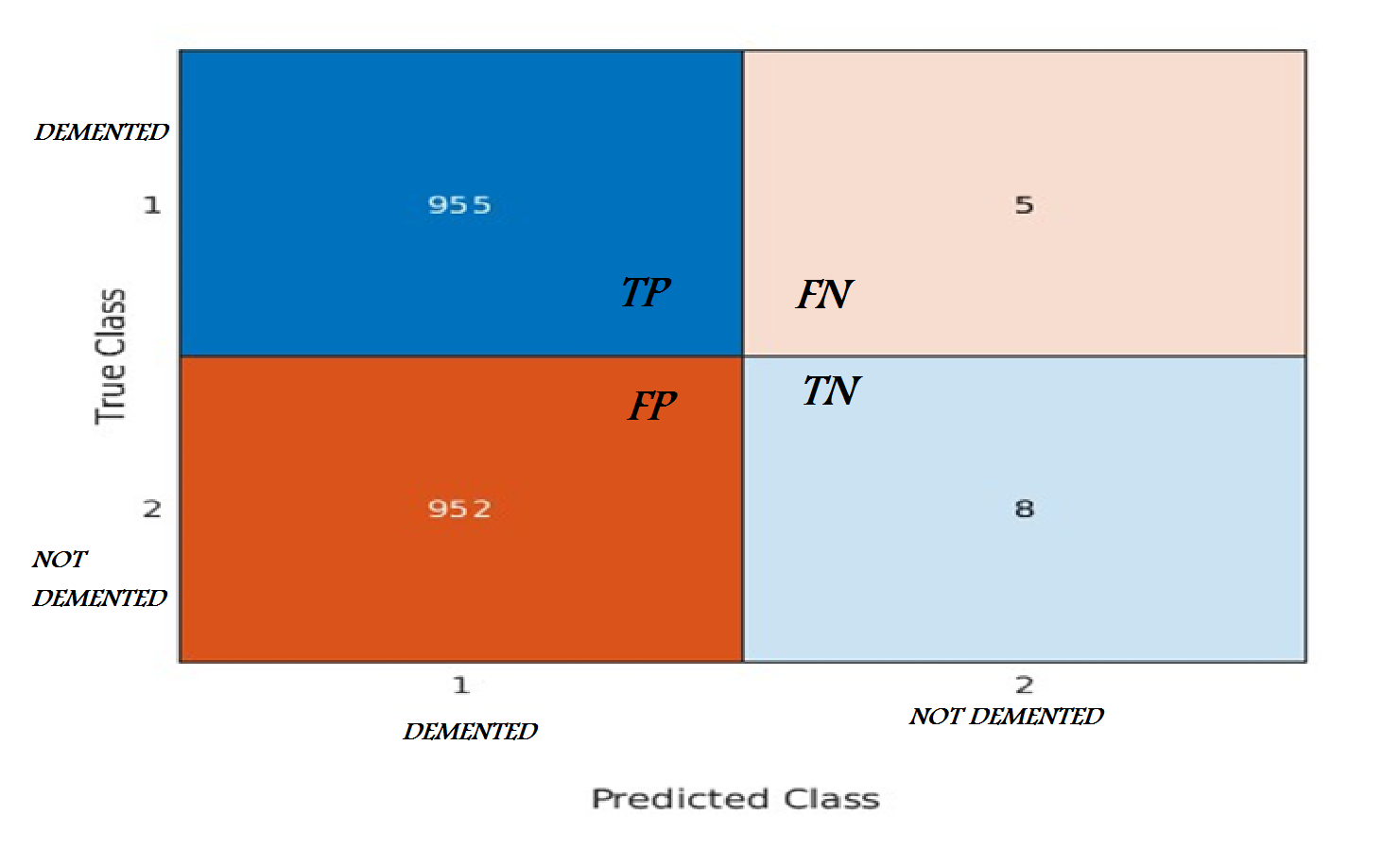


Figure 35: matrice de confusion pour sqeeznet à 2 classes

### Métriques

* **Accuracy = = 0.5016**
* **Precision=0.50079**
* **Recall= 0.99479**
* **F1 Score=2\*( Recall \* Precision)/( Recall + Precision)= 0.6662**

Commentaire1 :

D'après les valeurs obtenues, le modèle montre un échec complet selon toutes les métriques.

## Evaluation de notre modèle qui entrainer sur dataset a 2 classes avec nbr des individués équilibrés :

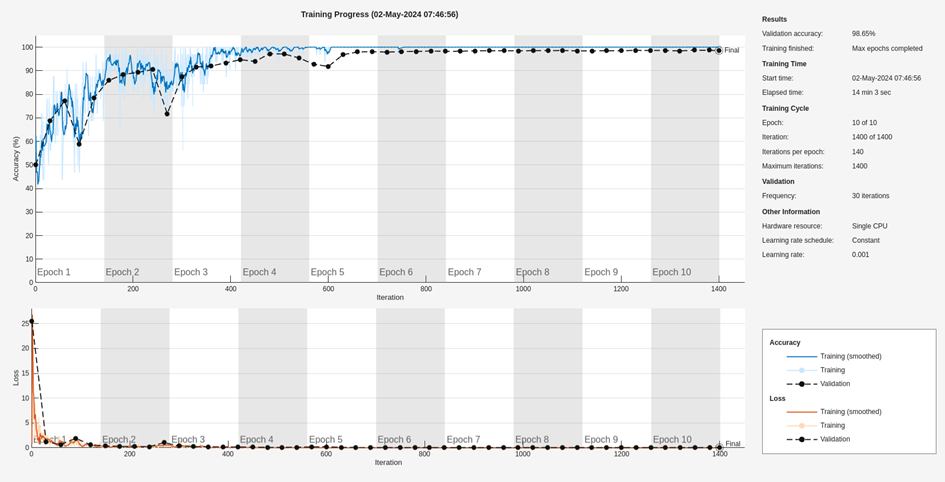
****

Figure 36: progression d’apprentissage pour notre modèle a 2 classes

### Matrice De Confusion

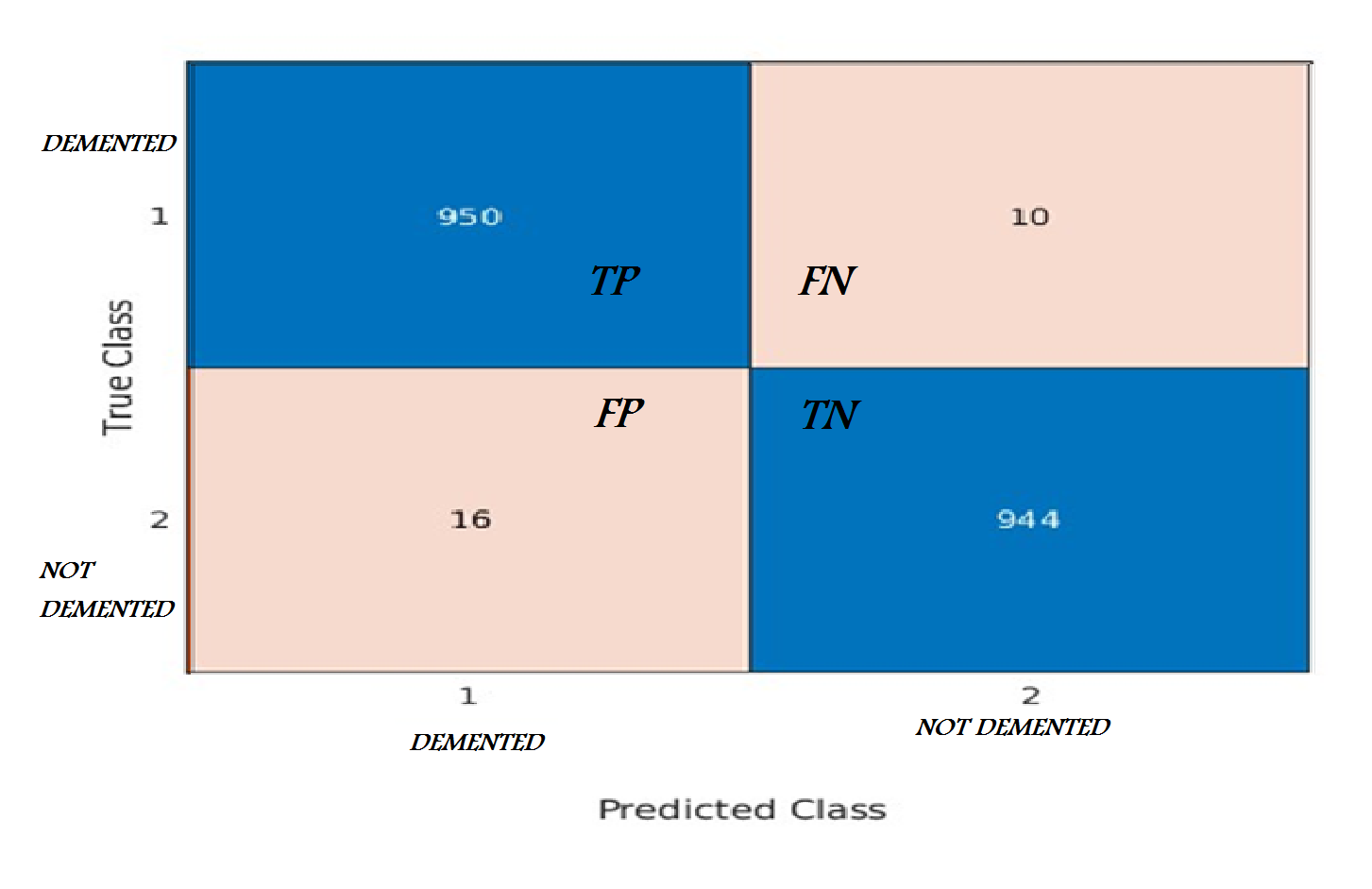
****

Figure 37: matrice de confusion pour notre modèle à 2 classes

### Métriques

|  |
| --- |
| * **Accuracy = = 0.9864** |
| * **Precision=0.9834** |
| * **Recall= 0.9895** |
| * **F1 Score=2\*( Recall \* Precision)/( Recall + Precision)= 0.9864** |

Commentaire 2 :

D'après les valeurs obtenues, le modèle présente un grand succès et des performances élevées selon toutes les métriques.

## Evaluation de notre modèle qui entrainer sur dataset a 4 classes avec nbr des individués déséquilibrés :

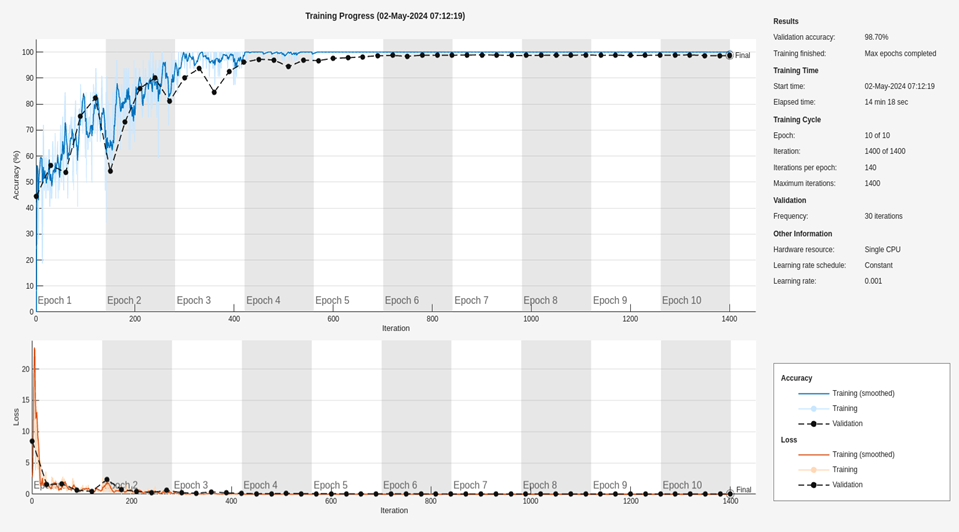


Figure 38: progression d’apprentissage pour notre modèle a 4 classes

### Matrice de confusion

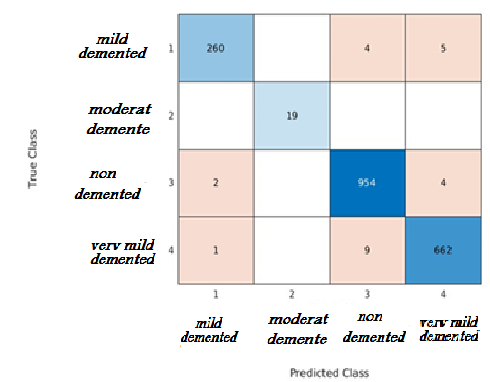
****

Figure 39: matrice de confusion pour notre modèle a 4 classes

**ACCURACY===0.987=98.7%**

### Métriques

#### Per class :

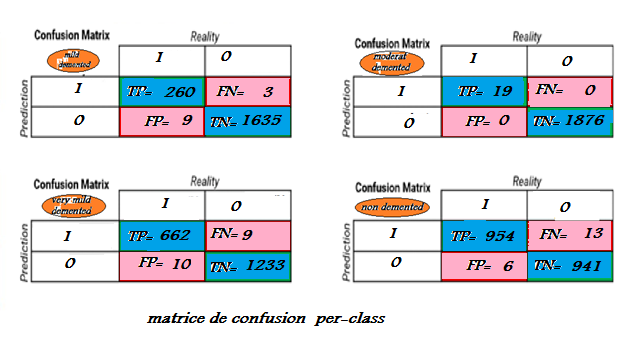
****

Figure 40: matrices binaires pour chaque classe

|  |  |  |
| --- | --- | --- |
| Accuracy(mild demented) | = | 0.99369 |
| Accuracy(moderat dement) | = | 1 |
| Accuracy(very-mild demented) | = | 0.990 |
| Accuracy(non demented) | = | 0.9900 |

|  |  |  |
| --- | --- | --- |
| Recall (mild demented) | = | 0.9885 |
| Recall (moderat dement) | = | 1 |
| Recall (very-mild demented) | = | 0.9865 |
| Recall (non demented) | = | 0.9865 |

|  |  |  |
| --- | --- | --- |
| Precision(mild demented) | = | 0.966 |
| Precision (moderat dement) | = | 1 |
| Precision (very-mild demented) | = | 0.985 |
| Precision (non demented) | = | 0.993 |

|  |  |  |
| --- | --- | --- |
| F1-score (mild demented) | = | 0.977 |
| F1-score (moderate dement) | = | 1 |
| F1-score (very-mild demented) | = | 0.975 |
| F1-score (non demented) | = | 0.9897 |

#### Macro :

###### **La métrique (macro) classique :**

Macro-accuracy**=**

**= = 0.9934**

Macro-recall**=**

**==0.9903**

Macro-precision**=**

**==0.979**

Macro-**F1-score=**

**==0.983**

###### **La métrique (macro)pondérée :**

Où

= **n**(non demented) **= 3200, n**(very mild demented)**= 2240, n**(moderate demented)**= 64, n**(mild demented)**= 896}**

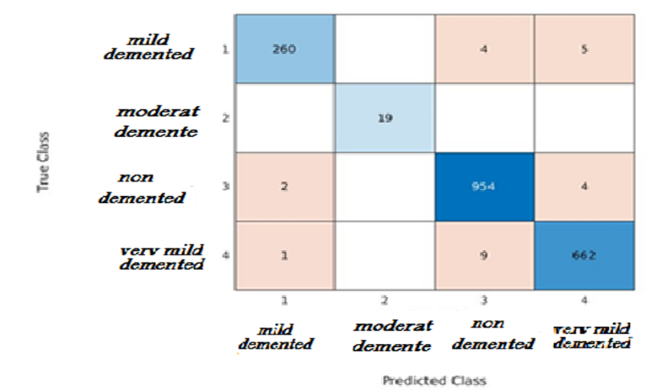
=**3200+2240+64+896=6400**

|  |  |  |
| --- | --- | --- |
|  | = | % |
|  | === 0.14 | =14% |
|  | == 0.01 | =1% |
|  | === 0.35 | =35% |
|  | === 0.5 | =50% |

|  |
| --- |
|  |

|  |
| --- |
| Macro-weighted-accuracy=  (**prop**(mild demented)\***accuracy**(mild demented)+ **prop**(moderate demented)\***accuracy** (moderat dement)+  **prop**(very mild demented)\***accuracy** (very-mild demented)+ **prop**(non demented)\***accuracy** (non demented))  =14%\*0.9936+1%\*1+35%\*099+50%\*0.99= 0.9906 |
| Macro-weighted-recall=  (**prop**(mild demented)\***recall**(mild demented)+ **prop**(moderate demented)\***recall** (moderat dement)+  **prop**(very mild demented)\***recall** (very-mild demented)+ **prop**(non demented)\***recall** (non demented))  =14%\***0.9885**+1%\*1+35%\***0.9865**+50%\***0.9865**=0.9869 |
| Macro-weighted-precision=  (**prop**(mild demented)\* **precision** (mild demented)+ **prop**(moderate demented)\***precision** (moderat dement)+  **prop**(very mild demented)\***precision** (very-mild demented)+ **prop**(non demented)\***precision** (non demented))  =14%\***0.966**+1%\*1+35%\***0.963**+50%\***0.993**=0.9787 |
| Macro-weighted-F1-score =  (**prop**(mild demented)\* **F1-score** (mild demented)+ **prop**(moderate demented)\* **F1-score**  (moderat dement)+  **prop**(very mild demented)\* **F1-score**  (very-mild demented)+ **prop**(non demented)\* **F1-score**  (non demented))  =14%\***0.977**+1%\*1+35%\***0.975**+50%\***0.9897**=0.9787 |

#### Micro-metrique



* **𝑇𝑁=0**

Figure 41: matrices a 4 classes pour notre modèle

**Passage vers matrice synthétique :**

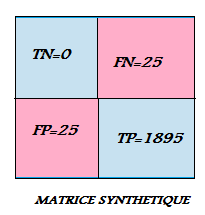
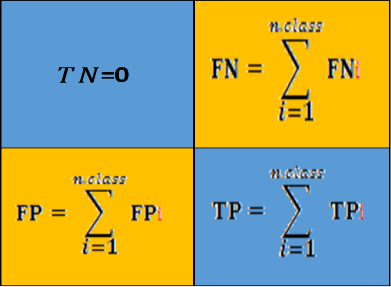


Figure 42: matrice synthétique

|  |  |  |
| --- | --- | --- |
| Micro-Accuracy= | = | = 0.9742 |
| Micro-Recall= | = | = 0.9869 |
| Micro-Precision= | ​= | = 0.9869 |
| Micro-F1-score= | 2\*= | = 0.9869 |

### Synthèse :

|  |  |  |  |
| --- | --- | --- | --- |
|  | Macro classique | Macro pondérée | micro |
| Accuracy | 0.9934 | 0.9906 | 0.9742 |
| Recall | 0.9903 | 0.9869 | 0.9869 |
| Précision | 0.9790 | 0.9787 | 0.9869 |
| F1-SCORE | 0.9830 | 0.9787 | 0.9869 |

Commentaire 3 :

D'après les valeurs obtenues, le modèle présente un grand succès et des performances élevées selon toutes les métriques.

## Evaluation de model (*Squeezent*) qui entrainer sur dataset a 4 classes avec nbr des individués déséquilibrés :

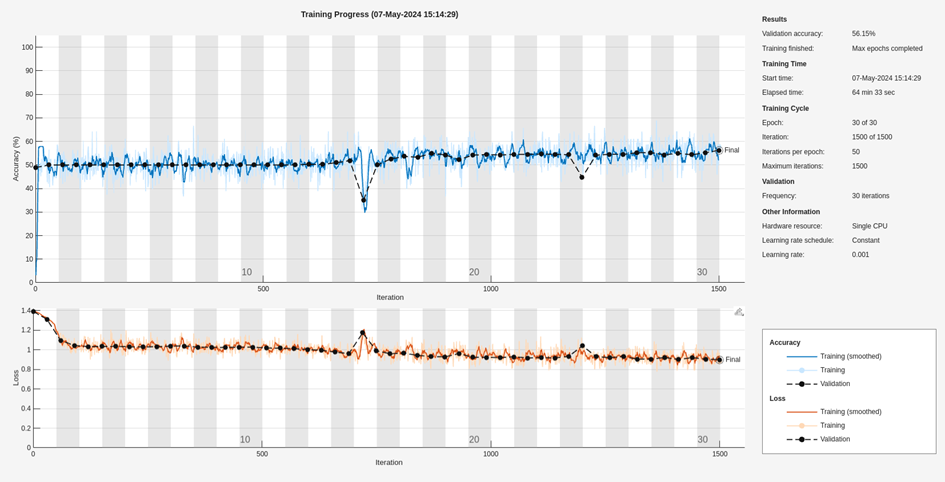
****

Figure 43: progression d’apprentissage de sqeeznet a 4 classes

### Matrice de confusion

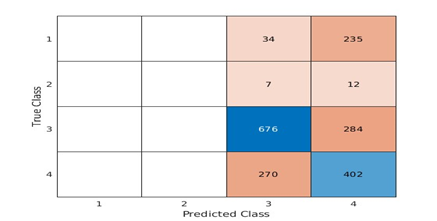
****

Figure 44: matrice de confusion de sqeeznet a 4 classes

**ACCURACY===0.5614=56.14%**

### Métriques

#### Per class:

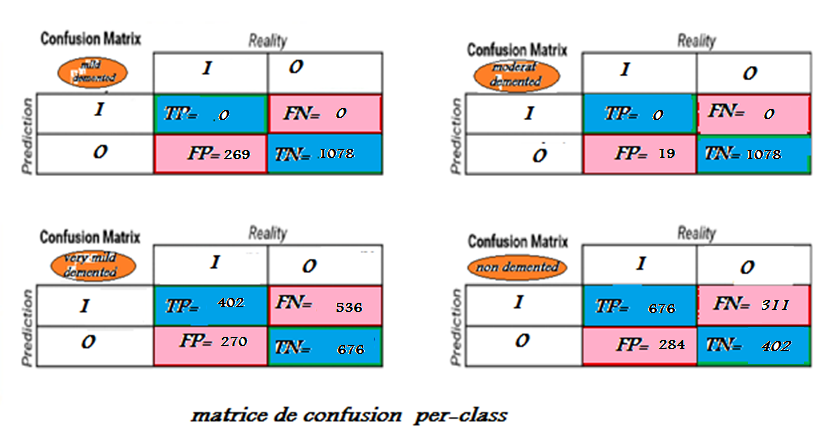
****

Figure 45: matrice binaire de sqeeznet a 4 classes

|  |
| --- |
| Accuracy(mild demented)= = 0.8002 |
| Accuracy(moderat dement)= = 0.9826 |
| Accuracy(very-mild demented)= = 0.5721 |
| Accuracy(non demented)= = 0.6443 |

|  |
| --- |
| Recall (mild demented)= = X |
| Recall (moderat dement)= = X |
| Recall (very-mild demented)= = 0.4285 |
| Recall (non demented)= = 0.6849 |

|  |
| --- |
| Precision(mild demented)= = 0 |
| Precision (moderat dement)= = 0 |
| Precision (very-mild demented)= = 0.5982 |
| Precision (non demented)= = 0.7041 |
|  |

|  |
| --- |
| F1-score (mild demented)= = x |
| F1-score (moderat dement)= = x |
| F1-score (very-mild demented)= = 0.4993 |
| F1-score (non demented)= = 0.6944 |
|  |

#### Macro :

* + - * 1. ***La métrique (macro) classique :***

|  |  |
| --- | --- |
| Macro-accuracy | **==0.7498** |
| Macro-recall | **= == X** |
| Macro-precision | **=**  **== 0.3256** |
| Macro-F1-score | **= == X** |

* + - * 1. **La métrique (macro) pondérée :**

Où

:

= **n**(non demented) **= 3200, n**(very mild demented)**= 2240, n**(moderate demented)**= 64, n**(mild demented)**= 896}**

=**3200+2240+64+896=6400**

|  |  |
| --- | --- |
|  | =% |
|  | ==0.14= 14% |
|  | ==0.35= 35% |
|  | ==0.5= 50% |
|  | ==0.01=1% |

|  |
| --- |
|  |
|  |

Macro-weighted-accuracy

|  |
| --- |
| = (**prop**(mild demented)\***accuracy**(mild demented)+ **prop**(moderate demented)\***accuracy** (moderat dement)+  **prop** (very mild demented)\***accuracy** (very-mild demented)+ **prop**(non demented)\***accuracy** (non demented)) |
| =14%\*0.8002+1%\*0.9826+35%\*0.5721+50%\*0.6443=0.6442 |

Macro-weighted-recall =

|  |
| --- |
| =(**prop**(mild demented)\***recall**(mild demented)+ **prop**(moderate demented)\***recall** (moderat dement)+  **prop**(very mild demented)\***recall** (very-mild demented)+ **prop**(non demented)\***recall** (non demented)) |
| =14%\* x +1%\* x +35%\***0.4285**+50%\***0.6849**= x |

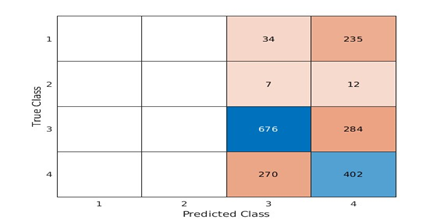
Macro-weighted-precision=

|  |
| --- |
| (**prop**(mild demented)\* **precision** (mild demented)+ **prop**(moderate demented)\***precision** (moderat dement)+ **prop**(very mild demented)\***precision** (very-mild demented)+ **prop**(non demented)\***precision** (non demented)) |
| =14%\***0**+1%\*0+35%\***0.5982**+50%\***0.7041**=0.5614 |

Macro-weighted-F1-score =

|  |
| --- |
| (**prop**(mild demented)\* **F1-score** (mild demented)+ **prop**(moderate demented)\* **F1-score**  (moderat dement)+  **prop** (very mild demented)\* **F1-score**  (very-mild demented) + **prop**(non demented)\* **F1-score**  (non demented)) |
| =14%\* **x** +1%\* x +35%\***0.4993**+50%\***0.6944**= x |

#### Micro-métriques



* **𝑇𝑁=0**

Figure 46: matrices à 4 classes pour sqeeznet

**Passage vers matrice synthétique**

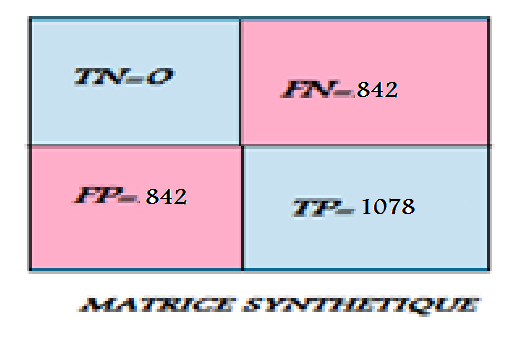
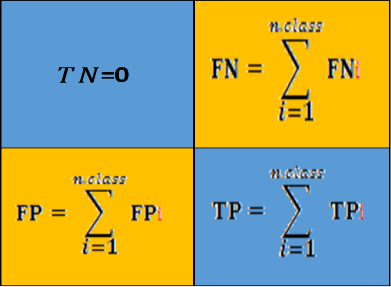


Figure 47: passage vers matrice synthétique

|  |  |  |
| --- | --- | --- |
| Micro-Accuracy | == | = 0.3902 |
| Micro-Recall | == | = 0.5614 |
| Micr-Precision | = ​= | = 0.5614 |
| Micro-F1-score | =2\*= | = 0.5614 |

### Synthèse :

|  |  |  |  |
| --- | --- | --- | --- |
|  | Macro classique | Macro ponderie | micro |
| accuracy | **0.7498** | 0.6442 | **0.3902** |
| recall | **x** | x | **0.5614** |
| précision | **0.3256** | 0.5614 | **0.5614** |
| F1-SCORE | x | x | **0.5614** |

Commentaire 4 :

D'après les valeurs obtenues, le modèle montre un échec complet selon toutes les métriques

## Discussion :

Suite aux résultats obtenues pour les 4 évaluations on conclue que :

* Le modèle pré-entrainé sqeeznet a montré un grand échec lors du processus d'évaluation avec les deux configurations à 4 classes et à 2 classes et selon toutes les métriques
* Notre modèle proposé montre un grand succès pour leurs deux configurations à 2 et a 4 classes selon toutes les métriques et avec **dataset** à 2 classes équilibré et autre **dataset** à 4 classes gravement déséquilibré.

Le dataset que nous avons utilisée pour l’apprentissage se caractérise par une petite taille en termes de nombre d'échantillons, ainsi que par le fait qu'elles sont principalement constituées d'images pré-classées d'une taille de 128\*128 pixels (relativement petite) et chaque image de type monochrome.

Un outil portable et simple à utiliser peut-être développer pour être mis au service des médecins afin de les aider à poser un diagnostic précoce plus précis et plus rapide.

**CONCLUSION GENERALE**

# CONCLUSION GENERALE

La performance de tout modèle d'apprentissage profond (deep learning) n'est pas liée à la complexité du modèle, au nombre de ses couches ou au nombre de ses paramètres, mais dépend plutôt de trouver le modèle approprié pour chaque **dataset** en termes de taille et de caractéristiques discriminante, et le nombre de classes que nous voulons prédire, afin d’éviter le sur-apprentissage (overfitting) ou le sous-apprentissage (underfitting).

**Bibliographie**

**[1]** [*https://www.revmed.ch/revue-medicale-suisse/*](https://www.revmed.ch/revue-medicale-suisse/)Dernier accès au site *14/04/2024*

**[2]** <https://alzheimer-recherche.org/> Dernier accès au site 14/04/2024

[3] [*https://www.mayoclinic.org/*](https://www.mayoclinic.org/)Dernier accès au site *14/04/2024*

[4] Utilisation de l’intelligence artificielle pour l’aide au diagnostic des patients atteints de pathologies neuro dégénératives par Edouard VILLAIN

[5] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Springer Science & Business Media.

[6]Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc.

[7]Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[8]James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer Science & Business Media.

[9]Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. Springer Science & Business Media.

[10]<https://datascientest.com/matrice-de-confusion/> Dernier accès au site 28/04/2024

[11]<https://kobia.fr/classification-metrics/> Dernier accès au site 01/05/2024

[12]LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.(cnn rnn lstm)

[13]Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. Regularisation math

[14]Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85-117.

[15]Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

cnn Image net

.

[16]MathWorks. (2023). Deep Learning Toolbox. Retrieved from MathWorks website.

[17]<https://machinelearningmastery.com/a-gentle-introduction-to-sigmoid-function/> Dernier accès au site 02/06/2024

[18] <https://www.baeldung.com/java-cnn-deeplearning4j> Dernier accès au site 02/06/2024

[19] <https://imtech.imt.fr/2018/03/13/analyse-irm-enfants/> Dernier accès au site 05/06/2024