

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Bordj Bou Arreridj Mohammed El Bachir El Ibrahimi
Faculté des Mathématiques et d'Informatique
Département d'Informatique

Mémoire

En vue de l'obtention du diplôme de Master

Domaine : Mathématiques et Informatique

Filière : Informatique

Option : Ingénierie de l'Informatique Décisionnelle (IID)

Intitulé

**Classification des IRM cérébrales pathologiques par une approche
hiérarchique**

Présenté par :

- Gouader fatima zohra
- Addala khouloud

Proposé et dirigé par :

- Mme H.CHELLAKH

Devant le jury composé de :

-

Année Universitaire 2019/2020

Remerciment

En premier lieu, nous remercions Dieu, le Tout-Puissant, de nous avoir donné le courage et la patience de mener ce travail durant toute cette année.

Nous adressons notre profond remerciement à Madame H.Chellakh notre encadreur qui nous a aidées.

Nous remercions le professeur Attia Abdelouaheb pour son temps et ses informations, que d'autres n'aurait pas pu aboutir à des résultats, et pour ses conseils.

Nous adressons aussi nos remerciements à tous, nos Enseignante à l'université de Bordj Bou Arreridj, A tous l'équipe pédagogique de la faculté des sciences de l'université de Bordj Bou Arreridj, qui ont veillé sur notre formation.

Enfin, que tous ceux qui nous ont aidés et encouragés de près ou de loin, trouvent ici ma gratitude et mes sincères remerciements.

Dédicace

Je dédie ce modeste travail :

*A mes très chers parents qui m'ont permis de
Réussir dans mes études.*

A toute la famille

A mes frères

A mes sœurs

A tous ceux qui m'ont aidé durant mes études

A mes chers amis

Fatima, Khouloud

Sommaire :

| | |
|--------------------|--|
| Remercîment | |
| Dédicace | |
| Liste des figures | |
| Liste des tableaux | |

Chapitre I : Introduction générale

| | |
|-----------------------------------|---|
| I.1 Contexte..... | 1 |
| I.2 Problématique..... | 1 |
| I.3 Objectif et contribution..... | 1 |
| I.4 Plan de travail..... | 2 |

Chapitre II : Notion d'étude

| | |
|--|----|
| Introduction..... | 3 |
| II.1 IRM..... | 3 |
| II.1.1 Définition..... | 3 |
| II.2 Pathologies Cérébrales..... | 4 |
| II.2.1 Méningiome..... | 4 |
| II.2.2 Gliome..... | 4 |
| II.2.3 Les Tumeurs Hypophysaires..... | 5 |
| II.3 l'apprentissage Automatique..... | 6 |
| II.3.1 Définition..... | 6 |
| II.3.2 Types De Problèmes d'apprentissage Automatique..... | 6 |
| II.3.2.1 Apprentissage Supervisé..... | 6 |
| II.3.2.1.1 Les Algorithmes De Classification..... | 8 |
| II.3.2.1.2 Les Algorithmes De Régression..... | 8 |
| II.3.2.2 Apprentissage Non Supervisé..... | 9 |
| Conclusion..... | 10 |

Chapitre III : Classification

| | |
|--|----|
| Introduction..... | 11 |
| III.1 Définition De Classification..... | 11 |
| III.2 Méthodes De Classification..... | 11 |
| III.2.1 Méthodes Supervisées (Classement)..... | 11 |
| III.2.2 Méthodes Non Supervisées..... | 12 |
| III.2.2.1 Méthodes Clustering..... | 12 |
| III.2.2.1.1 Les Types De Clustering..... | 13 |
| III.3 Quelques Algorithmes De Classification qui nous utilise..... | 14 |
| III.3.1 Support Vector Machines..... | 14 |
| III.3.1.1 Les Avantages De SVM..... | 14 |
| III.3.1.2 Les Inconvénients De SVM..... | 15 |
| III.3.2 L'Arbre De Décision..... | 15 |
| III.3.2.1 Définition..... | 15 |
| III.3.2.2 Les avantages..... | 16 |
| III.3.2.3 Inconvénients..... | 16 |
| III.3.3 Classification Hiérarchique..... | 16 |
| III.3.3.1 Architecture Générale..... | 17 |
| III.3.3.2 Processus D'apprentissage..... | 17 |
| III.3.3.3 Processus De Prise De Décision..... | 21 |
| Conclusion..... | 22 |

Chapitre IV : Implémentation

| | |
|--|----|
| Introduction..... | 23 |
| IV.1 Bases Des Données..... | 23 |
| IV.1.1 Base De Données 1..... | 23 |
| IV.1.2 Base De Données 2..... | 23 |
| IV.2 Langage De Programmation Utilise..... | 24 |

| | |
|---|----|
| IV.3 Les Etapes De L'implémentation..... | 25 |
| IV.3.1 Import La Base De Données..... | 25 |
| IV.3.2 Les Descripteurs..... | 25 |
| IV.3.2.1 Alexnet..... | 25 |
| IV.3.2.2 GIST..... | 26 |
| IV.3.2.3 HOG..... | 27 |
| IV.3.2.4 VGG-16..... | 27 |
| IV.3.3 Les Algorithmes..... | 28 |
| IV.3.3.1 SVM..... | 28 |
| IV.3.3.2 DTree..... | 29 |
| IV.3.3.3 HP..... | 29 |
| IV.4 Les Tableaux De Résultats..... | 30 |
| IV.5 Les mesures de performances..... | 33 |
| IV.5.1 Taux d'erreur : accuracy..... | 33 |
| IV.5.2 Recall vs precision..... | 33 |
| IV.5.3 F-mesure..... | 33 |
| IV.6 La matrice de confusion..... | 34 |
| IV.6.1 Définition..... | 34 |
| IV.6.2 Comment calculer une matrice de confusion..... | 35 |
| IV.6.3 Comprendre les terminologies de la Matrice de Confusion..... | 36 |
| IV.7 Courbe ROC..... | 37 |
| IV.7.1 Définition..... | 37 |
| Conclusion..... | 38 |
| Conclusion général | |
| Bibliographie | |
| Résumé | |

Liste des figures :

| | | |
|------------------|--|-----------|
| Figure 1 | Méningiome en regard du cortex pariétal droit | 4 |
| Figure 2 | Aspects radiologiques typiques de GBM | 5 |
| Figure 3 | Tumeurs Hypophysaires | 5 |
| Figure 4 | fonctionnement l'apprentissage supervisé | 7 |
| Figure 5 | types de classifications | 8 |
| Figure 6 | l'algorithme de régression linéaire | 9 |
| Figure 7 | Exemple de concept d'apprentissage non supervisé | 10 |
| Figure 8 | hierarchial et non hierarchial clustering | 13 |
| Figure 9 | L'architecture Générale Du Classificateur HP | 17 |
| Figure 10 | Pseudo-code de processus d'apprentissage HP. | 20 |
| Figure 11 | Illustration de deux méthodes de recherche de prototype les plus proches pour la prise de décision | 22 |
| Figure 12 | dossiers Training et Testing | 24 |
| Figure 13 | Les quatre classes de tumeurs dans le dossier testing | 24 |
| Figure 14 | Logo MATLAB | 25 |
| Figure 15 | Import La Base De Données | 25 |
| Figure 16 | Code Alexnet. | 26 |
| Figure 17 | Code GIST | 26 |
| Figure 18 | Code HOG | 27 |
| Figure 19 | Code VGG-16 | 28 |
| Figure 20 | Code SVM | 28 |
| Figure 21 | Code DTree | 29 |
| Figure 22 | Code HP | 29 |

| | | |
|------------------|---------------------------------|-----------|
| Figure 23 | predicted class et actual class | 32 |
| Figure 24 | la confusion matrice | 34 |
| Figure 25 | Confusion Matrice HP | 35 |
| Figure 26 | Courbe Roc HP | 36 |

Liste des tableaux :

| | | |
|-------------------|---|-----------|
| Tableau 01 | Nombre de prototypes identifiés par classe et Comparaison sur les bases de données. | 30 |
| Tableau 02 | Comparaison de qualité entre Descripteurs par différents algorithmes sur Base de données 1. | 31 |
| Tableau 03 | Comparaison de qualité entre Descripteurs par différents algorithmes sur Base de données 2. | 31 |
| Tableau 04 | Les mesures de performance pour hp avec hog | 32 |

Chapitre I:

Introduction générale

I.1 Contexte

De nos jours, les images numériques dans le domaine médical sont de plus en plus utilisées pour le diagnostic. De bonne heure L'identification des tumeurs cérébrales est importante pour traiter les tumeurs efficacement. Compte tenu du contraste élevé des tissus mous et de l'exposition nulle aux rayonnements ionisants, l'IRM est la technique la plus populaire pour diagnostiquer les tumeurs cérébrales humaines.

I.2 Problématique

Cependant, la classification des tumeurs cérébrales n'est pas une tâche triviale. La méthode conventionnelle pour la détection et la classification des tumeurs cérébrales par IRM est l'inspection humaine, qui dépend fortement de l'expérience des radiologues qui examinent et analysent les caractéristiques des images. De plus, les méthodes de classification assistées par l'opérateur ne sont pas pratiques pour de grandes quantités de données et sont également non reproductibles. Par conséquent, le diagnostic assisté par ordinateur des outils sont hautement souhaitables pour résoudre ces problèmes.

I.3 Objectif et contribution

La présente étude vise à développer une approche pour classer automatiquement les tumeurs cérébrales appartenant à différents types pathologiques, qui sont généralement difficiles et relativement difficiles et constituent un problème par rapport à la classification binaire (normale et anormale). Des études récentes ont suggéré plusieurs techniques de détection et de classifications des Tumeurs cérébrales.

L'objectif visé dans le cadre de ce mémoire est d'étudier différents algorithmes de classification et tester sur les images cérébrales, nous intéressons en particulier aux tumeurs cérébrales.

I.4 Plan de travail

Le mémoire est organisé en quatre chapitres principaux :

- Le premier chapitre présente l'introduction générale, la problématique, et l'objectif visé.
- Le deuxième chapitre présente le cadre applicatif du travail. Il rappelle quelques notions d'anatomie du cerveau humain, les principes fondamentaux de la technique d'imagerie par résonance magnétique, où la problématique est posée d'une manière détaillée aux maladies du cerveau.
- Dans le troisième chapitre nous allons parler de la technique de classification et présentons les différents algorithmes utilisés pour la classification des tumeurs cérébrales tout en évoquant les avantages et les inconvénients de chaque algorithme, puis nous présenterons notre proposition pour la détection des pathologies cérébrales.
- Le quatrième chapitre est consacré à l'implémentation de l'approche proposée en utilisant deux bases de données différentes et l'interprétation des résultats obtenus.
- La conclusion de ce travail sera présentée à la fin du mémoire.

Chapitre II :

Notion d'étude

Introduction

Ce chapitre s'intéresse à l'aspect médical de ce mémoire en présentant des notions sur :

- Présentation du principe de l'imagerie par résonance magnétique(IRM).
- Les différentes catégories de pathologies cérébrales et la manière dont elles apparaissent. À noter que la dernière partie n'a pas pour but d'être exhaustive. En effet, il existe un nombre impressionnant de pathologies cérébrales dont l'énumération et la description iraient bien au-delà du cadre de ce mémoire. Par conséquent, nous nous contenterons dans ce qui suit de définir des catégories de pathologies.
- Clarifier la définition de l'apprentissage automatique (Machine Learning) et les différents types de problèmes en Machine Learning (supervisé et non supervisé) avec quelques algorithmes qui existent dans tous les types.

II.1 IRM

II.1.1 Définition

L'IRM est une technique de diagnostic médical puissante qui fournit des images tridimensionnelles et en coupe de grande précision anatomique. L'IRM est une technique radiologique récente, non invasive et sans effets secondaires connus, basée sur le phénomène physique de résonance magnétique nucléaire. Il s'agit simplement d'observer la résonance magnétique nucléaire (RMN) des protons de l'eau contenus dans l'organisme, c'est à dire la réponse des noyaux soumis à un champ magnétique extérieur et à une excitation électromagnétique.

L'intensité recueillie pour un élément de volume (voxel) dépend de la concentration de l'eau à l'endroit considéré et on obtient ainsi une image tridimensionnelle de la répartition de l'eau dans le corps d'un patient. Selon la méthode utilisée, le contraste entre deux voxels peut être augmenté si les temps de relaxation des spins nucléaires (décrivant le retour à l'équilibre des noyaux après l'excitation) diffèrent dans les deux zones. Il est donc possible d'observer des altérations des tissus (telles que des tumeurs) grâce aux différences de densité et de relaxation de l'eau [3].

II.2 Pathologies cérébrales

Les pathologies cérébrales sont divisées en trois catégories

II.2.1 Méningiome

Le méningiome est une tumeur bénigne de l'enveloppe du système nerveux c'est à dire les méninges qui sont l'enveloppe du cerveau et de la moelle épinière. . C'est une maladie assez fréquente. Elle touche plutôt les adultes de plus de 50 ans avec une prédominance féminine [6].

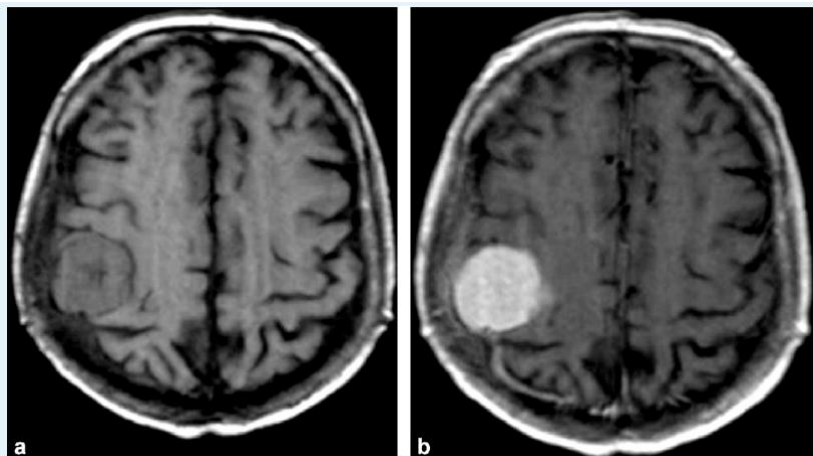


Figure 01 : Méningiome en regard du cortex pariétal droit [7].

II.2.2 Gliome

Les gliomes ou tumeurs gliales sont l'ensemble des tumeurs cérébrales, bénignes ou malignes, issues du tissu de soutien neuronal ou glie. Ce sont des tumeurs rares, dont le pronostic, extrêmement variable, est principalement lié à plusieurs facteurs parmi lesquels la localisation, la taille, le degré d'extension ou certains facteurs immunitaires¹. Elles justifient toujours une prise en charge dans un service spécialisé afin d'adapter au mieux le projet de soins à la situation du patient [8].

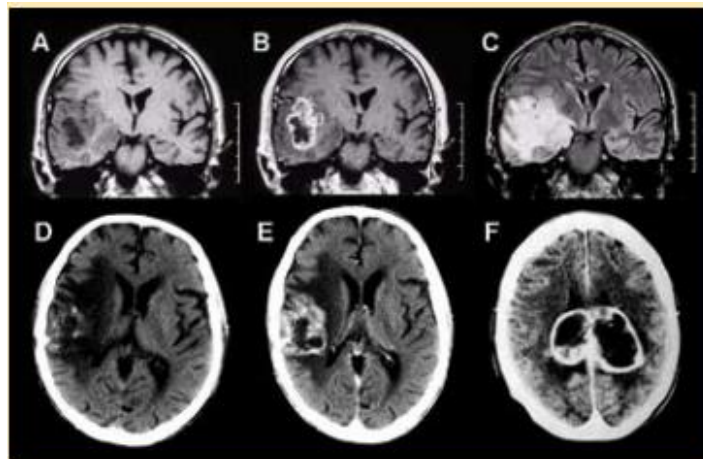


Figure 02 : Aspects radiologiques typiques de GBM [8].

II.2.3 Les tumeurs hypophysaires

Les tumeurs hypophysaires sont des excroissances anormales qui se développent dans votre glande pituitaire. Certaines tumeurs hypophysaires entraînent un trop grand nombre d'hormones qui régulent des fonctions importantes de votre corps. Certaines tumeurs de l'hypophyse peuvent amener votre glande pituitaire à produire des niveaux plus faibles d'hormones. La plupart des tumeurs hypophysaires sont des excroissances non cancéreuses (bénignes) (adénomes). Les adénomes restent dans votre glande pituitaire ou dans les tissus environnants et ne se propagent pas à d'autres parties de votre corps [9].

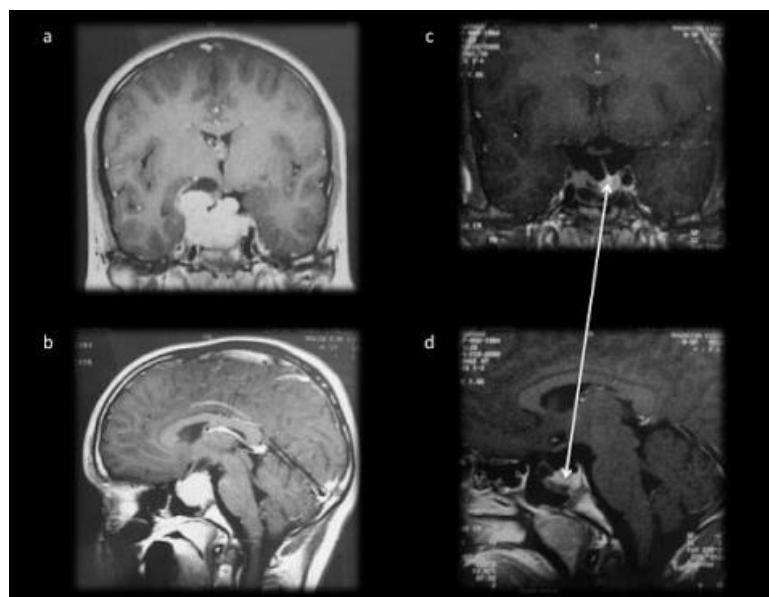


Figure 03 : Tumeurs Hypophysaires [9].

II.3 l'apprentissage automatique

II.3.1 Définition

La Machine Learning est une branche de l'intelligence artificielle qui a pour but de donner la possibilité aux ordinateurs d'apprendre. Un ordinateur n'est pas intelligent, il ne fait qu'exécuter des tâches. On lui décrit sous forme de programmes quoi faire et comment le faire. C'est ce qu'on appelle la programmation.

La Machine Learning traite des sujets complexes où la programmation traditionnelle trouve ses limites. Construire un programme qui conduit une voiture serait très complexe voire impossible. Cela étant dû aux nombres infinis des cas possibles à traiter... ML traite cette problématique différemment. Au lieu de décrire quoi faire, le programme apprendra par lui-même comment conduire en "observant" des expérimentations.

Machine Learning : Donner la possibilité à l'ordinateur d'apprendre sans être programmé. En fonction des données d'expérimentation que prendra l'algorithme d'apprentissage en entrée, il déduira par lui-même une hypothèse de fonctionnement. Il utilisera cette dernière pour de nouveaux cas, et affinera son expérience au fil du temps [5].

II.3.2 Types d'apprentissage automatique

On distingue deux types de problèmes en ML :

- Apprentissage Supervisé (Supervised Learning)
- Apprentissage Non supervisé (unsupervised Learning)

II.3.2.1 Apprentissage supervisé

L'apprentissage supervisé est le concept derrière plusieurs applications sympas de nos jours : reconnaissance faciale de nos photos par les smart phones, filtres anti-spam des emails, etc [5].

Plus formellement, étant donné un ensemble de données D , décrit par un ensemble de caractéristiques X , un algorithme d'apprentissage supervisé va trouver une fonction de mapping entre les variables prédictives en entrée X et la variable à prédire Y . la

fonction de mapping décrivant la relation entre X et Y s'appelle un modèle de prédiction $f(x) \rightarrow Y$ [5] .

Les caractéristiques (features en anglais) X peuvent être des valeurs numériques, alphanumériques, des images... Quant à la variable prédite Y, elle peut être de deux catégories :

- **Variable discrète** : La variable à prédire peut prendre une valeur d'un ensemble fini de valeurs (qu'on appelle des classes). Par exemple, pour prédire si un mail est SPAM ou non, la variable Y peut prendre deux valeurs possible : $Y \in \{SPAM, NONSPAM\}$ [2]
- **Variable continue** : La variable Y peut prendre n'importe quelle valeur. Pour illustrer cette notion, on peut penser à un algorithme qui prend en entrée des caractéristiques d'un véhicule, et tentera de prédire le prix du véhicule (la variable Y) [2].

La catégorie de la variable prédite Y fait décliner l'apprentissage supervisé en deux sous catégories :

- La classification
- La régression

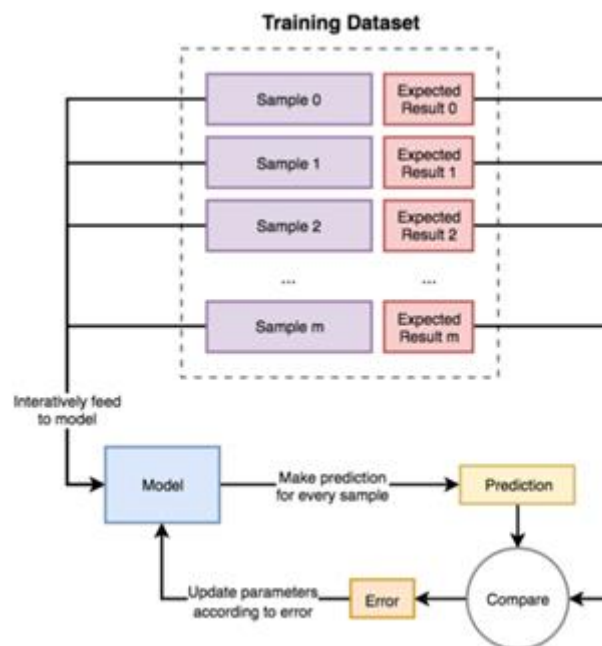


Figure 04 : fonctionnement l'apprentissage supervisé [4].

II.3.2.1.1 Les Algorithmes de classification

Quand la variable à prédire prend une valeur discrète, on parle d'un problème de classification. Parmi les algorithmes de classification, on retrouve : Support Vector Machine (SVM), Réseaux de neurones, Naïve Bayes, Logistic Regression... Chacun de ses algorithmes a ses propres propriétés mathématiques et statistiques. En fonction des données d'entraînement (Training set), et nos features, on optera pour l'un ou l'autre de ces algorithmes. Toutefois, la finalité est la même : pouvoir prédire à quelle classe appartient une donnée (ex : un nouveau email est-il spam ou non).

Quand l'ensemble des valeurs possibles d'une classification dépasse deux éléments, on parle de classification multi-classes (Multi-class Classification). L'image suivante illustre les deux types de classifications [2].

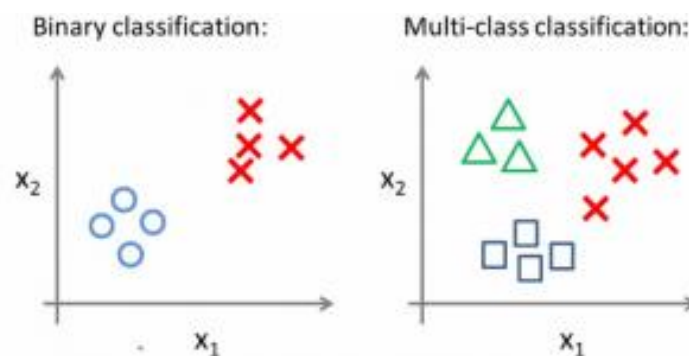


Figure 05 : types de classifications [2].

Dans l'image ci-dessus, les ronds en bleu représentent une classe (mail non spam par exemple), et les croix rouges peuvent représenter des SPAM. L'image à droite est une multi-class classification, car nous avons trois classes possibles (les triangles, les croix, et les carrés) [2].

II.3.2.1.2 Les Algorithmes de régression

Un algorithme de régression permet de trouver un modèle (une fonction mathématique) en fonction des données d'entraînement. Le modèle calculé permettra de donner une estimation sur une nouvelle donnée non encore vue par l'algorithme (qui ne faisait pas partie des données d'entraînement) [2].

Les algorithmes de régression peuvent prendre plusieurs formes en fonction du modèle qu'on souhaite construire. La régression linéaire est le modèle le plus simple : Il consiste à trouver la meilleure droite qui s'approche le plus des données d'apprentissage. La fonction de prédiction sera donc une droite [2].

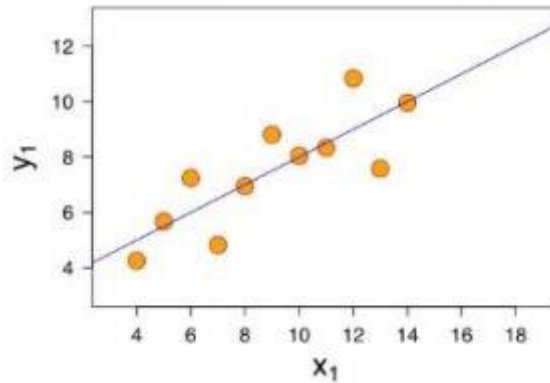


Figure 06 : l'algorithme de régression linéaire [2].

Le modèle prédit par l'algorithme de régression linéaire sera de la

forme $Y = f(X) = \alpha * X_1 + \beta$ (a et b sont les coefficients de la droite).

II.3.2.2 Apprentissage non supervisé

A l'inverse de l'apprentissage supervisé (Supervised Learning) qui tente de trouver un modèle depuis des données labellisées $f(X) \rightarrow Y$, l'apprentissage non supervisé prend uniquement des données sans label (pas de variable à prédire Y). Un algorithme d'apprentissage non supervisé va trouver des patterns ou une structuration dans les données.

Les algorithmes de clustering rentrent dans la catégorie d'apprentissage non supervisé. Ils permettent de regrouper en des ensembles, les données qui sont similaires [1].

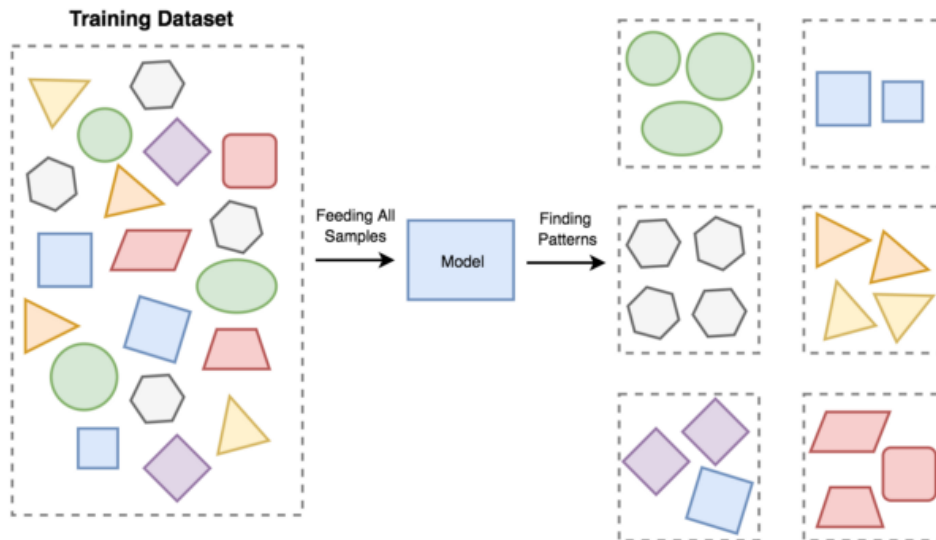


Figure 07 : Exemple de concept d'apprentissage non supervisé [4].

Conclusion

Nous avons présenté brièvement, dans ce chapitre, l'aspect médical de ce mémoire en incluant quelques pathologies cérébrales. Ces lésions peuvent être visualisées in vivo grâce à l'IRM.

Le traitement d'images permet d'aider les médecins dans les tâches de diagnostic et de suivi de la maladie.

Le chapitre suivant présente les principales méthodes existantes pour la classification d'images IRM et permet ainsi de situer notre méthode par rapport à la littérature.

Chapitre III :
Classification

Introduction

Dans ce chapitre, nous allons d'esquisser un état de l'art de la classification d'images. Cette revue des différentes techniques et approches n'a pas pour but d'être exhaustive mais de montrer la grande diversité qui existe au sein de ce vaste domaine. Celles citées en exemples sont celles qui nous ont semblées les plus à même d'illustrer nos propos.

Dans un premier temps, nous réfléchirons sur les tenants et aboutissants de la classification et les critères de véracité d'une classification, puis, dans un second temps, nous nous intéresserons aux quelques méthodes de classification que nous avons classé en deux grands thèmes (supervisé, non supervisé) que sont les approches opérant sur les contours avec certains algorithmes des classifications mentionnés, les avantages et les inconvénients de chaque algorithme

III.1 Définition de classification

Classifier une image est une tâche ou une série de méthodes qu'une théorie unifiée pour pouvoir utiliser les images pour les analyses complémentaires ou pour la cartographie. Elle consiste à déterminer des procédures permettant de rassembler l'ensemble des pixels de l'image en un nombre limité de classes correspondant aux grands éléments structuraux de l'image [10].

“La qualité de la classification peut être jugée sur la base des deux critères suivants :

- Les classes générées doivent être les plus différentes possibles les unes des autres vis-à-vis de certaines caractéristiques
- Chaque classe doit être la plus homogène possible vis-à-vis de ces caractéristiques

III.2 Méthodes de classification

III.2.1 Méthodes supervisées (Classement)

“ Ces sont des méthodes dans lesquelles les classes sont connues a priori avant d'effectuer l'opération d'identification des éléments de l'image. Elles demandent une première phase d'apprentissage sur l'échantillon représentatif [10].

Cette phase d'apprentissage permet de définir des règles de décision à partir d'un ensemble d'objets de référence, dont on connaît l'identité à priori et qui sont représentatifs de chaque classe. Il est décomposé en trois étapes principales :

1- Appeler "acquisition des données", concerne le moyen d'obtention des points de référence représentatifs des différentes classes. On peut, soit utiliser des objets de référence pour chaque classe et les présenter au capteur; soit sélectionner de manière interactive les signaux fournis par le capteur et les assimiler à une classe. L'affectation des groupes est réalisée par un opérateur humain traditionnellement appelé "professeur".

2- Transformer les données issues du capteur pour se placer dans un espace de représentation adapté à la reconnaissance. Chaque objet est caractérisé par un ensemble de paramètres (ou attributs) regroupés sous la forme d'un vecteur. le nombre d'attributs dimensionne l'espace de représentation.

3- Appelée "partitionnement de l'espace des attributs", partitionnement de l'espace en zones associées à chacune des classes prédéfinies.

La phase de classement (reconnaissance) consiste à attribuer les objets que l'on propose de reconnaître à une des classes définies lors de l'apprentissage.

III.2.2 Méthodes non supervisées

Procède de la façon contraire. C'est à dire ne nécessitent aucun apprentissage et aucune tâche préalable d'étiquetage manuel. Elle consiste à représenter un nuage des points d'un espace quelconque en un ensemble de groupes appelé Cluster. Il est lié généralement au domaine de l'analyse des données [10].

III.2.2.1 Méthodes clustering

Le clustering consiste à grouper des points de données en fonction de leurs similitudes, tandis que l'association consiste à découvrir des relations entre les attributs de ces points de données :

Les techniques de clustering cherchent à décomposer un ensemble d'individus en plusieurs sous ensembles les plus homogènes possibles [10]

- On ne connaît pas la classe des exemples (nombre, forme, taille)

- Les méthodes sont très nombreuses, typologies généralement employées pour les distinguer Méthodes de partitionnement / Méthodes hiérarchiques
- Avec recouvrement / sans recouvrement
- Autre : incrémental / non incrémental
- D'éventuelles informations sur les classes ou d'autres informations sur les données n'ont pas d'influence sur la formation des clusters, seulement sur leur interprétation.

III.2.2.1.1 Les Types de clustering

Il existe deux grands types du clustering :

A/ le clustering hiérarchique : d'agglomération («bottom-up»)

Décompose l'ensemble d'individus en une arborescence de groupes.

B/ le clustering non-hiérarchique : de division («top-down»)

Décompose l'ensemble d'individus en K groupes.

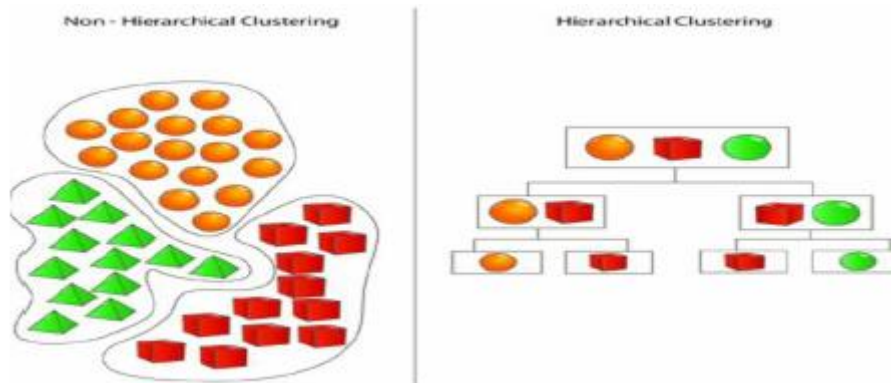


Figure 08 : hierarchical et non hierarchical clustering.

III.3 Quelques algorithmes de classification qui nous utilise

III.3.1 Support vector machines

Le SVM est une solution à ce problème de classification¹. Le SVM appartient à la catégorie des classificateurs linéaires (qui utilisent une séparation linéaire des données), et qui dispose de sa méthode à lui pour trouver la frontière entre les catégories [11].

Pour que le SVM puisse trouver cette frontière, il est nécessaire de lui donner des données d'entraînement. A partir de ces données, le SVM va estimer l'emplacement le plus plausible de la frontière : c'est la période d'entraînement, nécessaire à tout algorithme d'apprentissage automatique.

Une fois la phase d'entraînement terminée, le SVM a ainsi trouvé, à partir de données d'entraînement, l'emplacement supposé de la frontière. En quelque sorte, il a « appris » l'emplacement de la frontière grâce aux données d'entraînement. Qui plus est, le SVM est maintenant capable de prédire à quelle catégorie appartient une entrée qu'il n'avait jamais vue avant, et sans intervention humaine : c'est là tout l'intérêt de l'apprentissage automatique.

III.3.1.1 Les Avantages de SVM

- L'algorithme SVM a une base théorique solide, basée sur les idées de dimension et de minimisation du risque structurel.
- L'algorithme SVM s'adapte bien à des ensembles de données relativement volumineux.
- L'algorithme SVM est flexible en partie grâce à la robustesse de l'algorithme lui-même, et en partie à la par métrisation du SVM via une large classe de fonctions, appelées fonctions noyau. Le comportement du SVM peut être modifié pour incorporer la connaissance préalable d'une tâche de classification simplement en modifiant la fonction noyau sous-jacente [12].

- Précision: l'explication la plus importante de la popularité de l'algorithme SVM est sa précision. La théorie sous-jacente suggère des explications pour l'excellente performance d'apprentissage des SVM, son application généralisée est due en grande partie au succès empirique de l'algorithme.

III.3.1.2 Les inconvénients de SVM

Les SVM (Support Vector Machines) sont une technique utile pour la classification des données. Bien que SVM soit considéré comme plus facile à utiliser que les réseaux de neurones, les utilisateurs ne connaissent pas il obtient souvent des résultats insatisfaisants au début [12].

III.3.2 L'arbre de décision

III.3.2.1 Définition

L'apprentissage par arbre de décision désigne une méthode basée sur l'utilisation d'un arbre de décision comme modèle prédictif. On l'utilise notamment en fouille de données et en apprentissage automatique.

Dans ces structures d'arbre, les feuilles représentent les valeurs de la variable-cible et les embranchements correspondent à des combinaisons de variables d'entrée qui mènent à ces valeurs. En analyse de décision, un arbre de décision peut être utilisé pour représenter de manière explicite les décisions réalisées et les processus qui les amènent. En apprentissage et en fouille de données, un arbre de décision décrit les données mais pas les décisions elles-mêmes, l'arbre serait utilisé comme point de départ au processus de décision.

C'est une technique d'apprentissage supervisé : on utilise un ensemble de données pour lesquelles on connaît la valeur de la variable-cible afin de construire l'arbre (données dites étiquetées), puis on extrapole les résultats à l'ensemble des données de test. Les arbres de décision font partie des algorithmes les plus populaires en apprentissage automatique.

III.3.2.2 Les avantages

- La simplicité de compréhension et d'interprétation.

- Peu de préparation des données (pas de normalisation, de valeurs vides à supprimer, ou de variable muette).
- Le modèle peut gérer à la fois des valeurs numériques et des catégories.
- Il est possible de valider un modèle à l'aide de tests statistiques, et ainsi de rendre compte de la fiabilité du modèle.
- Performant sur de grands jeux de données.

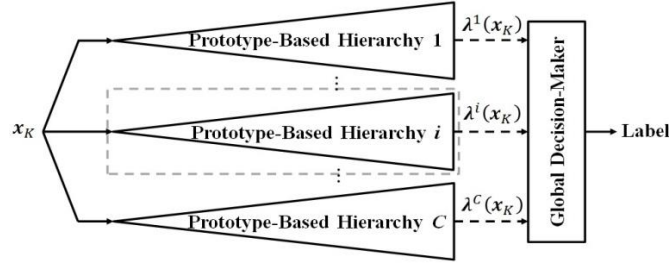
III.3.2.3 Inconvénients

- L'apprentissage de l'arbre de décision optimal est NP-complet concernant plusieurs aspects de l'optimalité. En conséquence, les algorithmes d'apprentissage par arbre de décision sont basés sur des heuristiques telles que les algorithmes gloutons cherchant à optimiser le partage à chaque nœud, et de tels algorithmes ne garantissent pas d'obtenir l'optimum global. Certaines méthodes visent à diminuer l'effet de la recherche gloutonne.
- L'apprentissage par arbre de décision peut amener des arbres de décision très complexes, qui généralisent mal l'ensemble d'apprentissage.
- Lorsque les données incluent des attributs ayant plusieurs niveaux, le gain d'information dans l'arbre est biaisé en faveur de ces attributs.

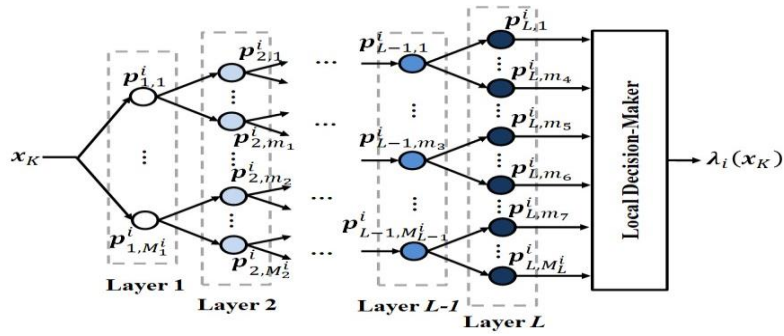
III.3.3 Classification hiérarchique

Une nouvelle approche basée sur un prototype avec de telles caractéristiques est proposée pour classification. Le classificateur HP décompose les problèmes complexes en une série de modèles locaux plus simples avec différents niveaux de granularité et les représente avec des prototypes significatifs agrégés naturellement dans une forme de hiérarchie pyramidale. Ces prototypes sont objectivement identifiés à partir de données en streaming via un site en ligne, processus autonome et hautement efficace en termes de calcul avec différents niveaux de granularité de faible à élevé, ce qui aboutit naturellement à une structure multicouche. Ils représentent les pics locaux des distributions multi-modales dérivées de données à différents niveaux de spécificité sans connaissance préalable des problèmes.

III.3.3.1 Architecture générale



(a) structure System



(b) Structure de zoom avant de la i^{th} hiérarchie basée sur le prototype

Figure 33 : L'architecture Générale Du Classificateur HP [14] .

III.3.3.2 Processus d'apprentissage

Dans cette sous-section, la procédure algorithmique pour identifier de manière autonome une structure de système hiérarchique basée sur un prototype est détaillée comme suit, pour chaque échantillon de données observé du i^{th} classe, x_k^i ($k = 1, 2, \dots, K^i, \dots$), il est d'abord normalisé par sa norme euclidienne:

$$x_k^i \leftarrow \frac{x_k^i}{\|x_k^i\|} \tag{1}$$

Où

$$\|x_k^i\| = \sqrt{\sum_{j=1}^N (x_{k,j}^i)^2}$$

Étape 0. Initialisation du système

$x_{K^i}^i$ ($K^i = 1$) est utilisé pour initialiser la hiérarchie et est traité comme le premier prototype à chaque couche $l = 1, 2, \dots, L$:

$$M_l^i \leftarrow 1; \quad \mathbf{p}_{l,M_l^i}^i \leftarrow \mathbf{x}_{K^i}^i; \quad S_{l,M_l^i}^i \leftarrow 1 \quad (2)$$

Où $S_{l,M_l^i}^i$ est le nombre d'échantillons de données associés à $\mathbf{p}_{l,M_l^i}^i$

Les liens (relations subordonnées) entre ces prototypes de couches successives sont :

$$\mathcal{L}_0^i \leftarrow \{\mathbf{p}_{1,M_1^i}^i\} \quad (3a)$$

et, pour le prototype au l^{th} layer :

$$\mathcal{L}_{l,M_l^i}^i \leftarrow \{\mathbf{p}_{l+1,M_{l+1}^i}^i\} \quad (3b)$$

Par l'équation (3), le i^{th} la hiérarchie est établie dans sa forme initiale ressemblant à une chaîne avec $\mathbf{p}_{1,M_1^i}^i$ comme nœud de départ et $\mathbf{p}_{L,M_L^i}^i$ comme nœud final.

Étape 1. Système en évolution dynamique

Premièrement, le prototype le plus proche au l^{th} layer à $\mathbf{x}_{K^i}^i$ est identifié à l'aide de l'équation suivante:

$$n_l^* = \begin{cases} \operatorname{argmin}_{\mathbf{p} \in \mathcal{L}_0^i} (\|\mathbf{x}_{K^i}^i - \mathbf{p}\|) & \text{if } l = 1 \\ \operatorname{argmin}_{\mathbf{p} \in \mathcal{L}_{l-1, n_{l-1}^*}^i} (\|\mathbf{x}_{K^i}^i - \mathbf{p}\|) & \text{if } l = 2, 3, \dots, L \end{cases} \quad (4)$$

Une fois le prototype le plus proche $\mathbf{p}_{l, n_l^*}^i$ est identifié, la condition suivante est vérifiée pour voir si $\mathbf{x}_{K^i}^i$ a le potentiel de devenir un nouveau prototype du l^{th} layer:

$$\begin{aligned} \textbf{Condition 1:} \quad & \text{IF } \left(\|\mathbf{x}_{K^i}^i - \mathbf{p}_{l, n_l^*}^i\|^2 > r_l \right) \\ & \text{THEN } (\mathbf{x}_{K^i}^i \text{ is a new prototype at the } l^{\text{th}} \text{ layer}) \end{aligned} \quad (5)$$

Dans cet article, par défaut, la valeur de r_l ($l = 1, 2, \dots, L$) est dérivée par l'équation suivante [2]:

$$r_l = 2 \left(1 - \cos \left(\frac{\theta_o}{2^{l-1}} \right) \right) \quad (6)$$

Où

$$\theta_o = \frac{\pi}{2}.$$

Si la **condition 1** n'est pas satisfaite, $\mathbf{x}_{K^i}^i$ est utilisé pour mettre à jour les méta-paramètres du prototype le plus proche au l^{th} couche par les formules suivantes:

$$\mathbf{p}_{l,n_l^*}^i \leftarrow \frac{S_{l,n_l^*}^i}{S_{l,n_l^*}^i + 1} \mathbf{p}_{l,n_l^*}^i + \frac{1}{S_{l,n_l^*}^i + 1} \mathbf{x}_{K^i}^i \quad (7a)$$

$$\mathbf{p}_{l,n_l^*}^i \leftarrow \frac{\mathbf{p}_{l,n_l^*}^i}{\|\mathbf{p}_{l,n_l^*}^i\|} \quad (7b)$$

$$S_{l,n_l^*}^i \leftarrow S_{l,n_l^*}^i + 1 \quad (7c)$$

Par conséquent, une ré-normalisation est nécessaire pour garantir que l'algorithme peut reconnaître correctement les vrais prototypes les plus proches à tout moment par l'équation (4).

Puis, $\mathbf{x}_{K^i}^i$ est passé au suivant layer ($l \leftarrow l + 1$), et la même procédure (à partir de l'équation (4)) est répétée pour mettre à jour la couche suivante de la hiérarchie jusqu'à ce que la couche inférieure soit mise à jour ou interrompue lorsque la **condition 1** est satisfaite.

Si la **condition 1** est satisfaite, un nouveau prototype est ajouté au l^{th} couche ainsi que les couches successives avec méta-paramètres initialisés par équation (9) ($j = l, l + 1, l + 2, \dots, L$):

$$M_j^i \leftarrow M_j^i + 1; \quad \mathbf{p}_{j,M_j^i}^i \leftarrow \mathbf{x}_{K^i}^i; \quad S_{j,M_j^i}^i \leftarrow 1 \quad (9)$$

$\mathbf{x}_{K^i}^i$ lui-même est le nœud de départ et \mathcal{L}^i est mis à jour comme suit:

$$\mathcal{L}_0^i \leftarrow \mathcal{L}_0^i \cup \{\mathbf{p}_{1,M_1^i}^i\} \quad (10a)$$

Autrement, $\mathbf{p}_{l-1,n_{l-1}^*}^i$ est reconnu comme le nœud de départ de la nouvelle branche, et $\mathcal{L}_{l-1,n_{l-1}^*}^i$ est mis à jour par l'équation (10b):

$$\mathcal{L}_{l-1,n_{l-1}^*}^i \leftarrow \mathcal{L}_{l-1,n_{l-1}^*}^i \cup \{\mathbf{p}_{l,M_l^i}^i\} \quad (10b)$$

| |
|--|
| <p><i>Input: the data stream, $\{x\}_K^i$</i></p> <p><i>Algorithm begins</i></p> <p>While (new data sample $x_{K^i}^i$ is available or until interrupted)</p> <p><i>a. Normalize $x_{K^i}^i$ by its Euclidean norm: $x_{K^i}^i \leftarrow x_{K^i}^i / \ x_{K^i}^i\$;</i></p> <p><i>b. If ($K^i = 1$) then</i></p> <p><i>i. For $l = 1$ to L do</i></p> <p><i>1. $M_l^i \leftarrow 1$; $p_{l,M_l^i}^i \leftarrow x_{K^i}^i$; $S_{l,M_l^i}^i \leftarrow 1$;</i></p> <p><i>2. $\left\{ \begin{array}{l} \mathcal{L}_0^i \leftarrow \{p_{l,M_l^i}^i\} \quad \text{if } l = 1 \\ \mathcal{L}_{l-1,M_{l-1}^i}^i \leftarrow \{p_{l,M_l^i}^i\} \quad \text{if } l = 2, 3, \dots, L \end{array} \right.$</i></p> <p><i>ii. End for</i></p> <p><i>c. Else</i></p> <p><i>i. For $l = 1$ to L do</i></p> <p><i>1. Identify the nearest prototype, $p_{i,n_l^i}^i$ to $x_{K^i}^i$ by equation (4);</i></p> <p><i>2. If (Condition 1 is unsatisfied) then</i></p> <p><i>* Update $p_{i,n_l^i}^i$ and $S_{i,n_l^i}^i$ by equation (7);</i></p> <p><i>3. Else</i></p> <p><i>* For $j = l$ to L do</i></p> <p><i>- $M_j^i \leftarrow M_j^i + 1$; $p_{j,M_j^i}^i \leftarrow x_{K^i}^i$; $S_{j,M_j^i}^i \leftarrow 1$;</i></p> <p><i>* End for</i></p> <p><i>* $\left\{ \begin{array}{l} \mathcal{L}_0^i \leftarrow \mathcal{L}_0^i \cup \{p_{l,M_l^i}^i\} \quad \text{if } l = 1 \\ \mathcal{L}_{l-1,n_{l-1}^i}^i \leftarrow \mathcal{L}_{l-1,n_{l-1}^i}^i \cup \{p_{l,M_l^i}^i\} \quad \text{if } l = 2, 3, \dots, L \end{array} \right.$</i></p> <p><i>* For $j = l + 1$ to L do</i></p> <p><i>- $\mathcal{L}_{j-1,M_{j-1}^i}^i \leftarrow \{p_{j,M_j^i}^i\}$;</i></p> <p><i>* End for</i></p> <p><i>* Break the for loop;</i></p> <p><i>4. End if</i></p> <p><i>ii. End for</i></p> <p><i>d. End if</i></p> <p>End while</p> <p><i>Algorithm ends</i></p> <p><i>Output: the i^{th} prototype-based hierarchy</i></p> |
|--|

Figure 10 : Pseudo-code de processus d'apprentissage HP [14].

III.3.3.3 Processus de prise de décision

Dans cette sous-section, deux méthodes optionnelles de recherche de prototypes les plus proches pour calculer $\lambda^i(x_K)$ sont prévus [14].

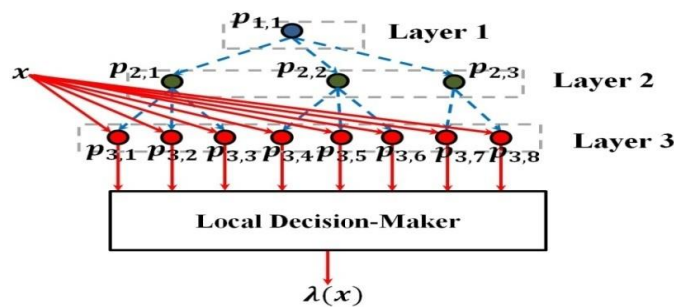
Mode A: la première méthode consiste à rechercher le prototype le plus proche de x_K au l^{th} couche de la hiérarchie directement, et le score de confiance est calculé par l'équation suivante:

$$\lambda^i(x_K) = \max_{p \in \{p\}_l^i} (e^{-\|p-x_K\|^2}) \tag{12}$$

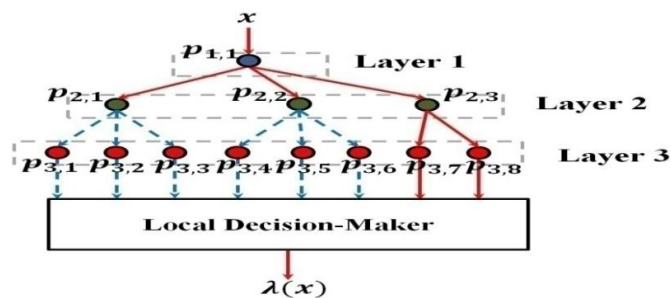
Où $\{p\}_l^i = \{p_{l,1}^i, p_{l,2}^i, \dots, p_{l,M_l}^i\}$ désigne la collection de prototypes au l^{th} layer du i^{th} hiérarchie.

Mode B: alternativement, on peut choisir de rechercher les prototypes les plus proches $p_{1,n_1}^i, p_{2,n_2}^i, \dots, p_{l,n_l}^i$ couche par couche du haut vers le haut l^{th} couche de la hiérarchie utilisant l'équation (4), et le résultat final, à savoir, le score de confiance est donné par:

$$\lambda^i(x_K) = e^{-\|p_{l,n_l}^i - x_K\|^2} \tag{13}$$



(a) *Mode A*



(b) *Mode B*

Figure 11 : Illustration de deux méthodes de recherche de prototype les plus proches pour la prise de décision [14].

Conclusion

La classification est utilisée pour attribuer des éléments à un groupe ou à une classe discrète en fonction d'un ensemble spécifique de fonctionnalités. Les algorithmes de classification sont une composante essentielle l'apprentissage automatique.

Les processus d'apprentissage et de prise de décision du classificateur HP sont entièrement interprétables, tractables, explicables par les humains. L'approche proposée est également capable de visualiser les connaissances acquises, à savoir les prototypes identifiés avec différents niveaux de détails et des liens significatifs entre eux, sous une forme hiérarchique compréhensible par l'homme naturellement. Pour cela nous allons présenter dans le chapitre suivant La structure du classificateur HP proposé est très transparente et ses processus d'apprentissage et de prise de décision sont entièrement explicable et interprétable pour l'homme.

Chapitre IV :
Implémentation

Introduction

Ce dernier chapitre est consacré à l'application des méthodes choisies dans les chapitres précédents, nous allons dérouler notre implémentation sur des images du cerveau humain, tout en vous guidons sur les étapes de son fonctionnement, ainsi qu'une analyse des résultats obtenus.

IV.1 Bases des données

Deux bases d'images sont utilisé dans cette mémoire :

IV.1.1 Base de données 1

Le dossier contient des données IRM. Les images sont déjà divisées en dossiers yes et no.

Source distante : <https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection?>

La taille : 8 MB

Nombre totale d'images IRM : 253

IV.1.2 Base de données 2

Le dossier contient des données IRM. Les images sont déjà divisées en dossiers Training et Testing.

Chaque dossier a plus de quatre sous-dossiers. Ces dossiers ont des IRM des classes de tumeurs respectives.

Source distante: <https://www.kaggle.com/sartajbhuvaji/brain-tumor-classification-mri>

La taille : 88.77 MB

Nombre d'image IRM : 3264

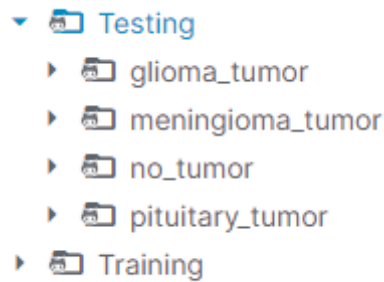


Figure 12 : dossiers Training et Testing.

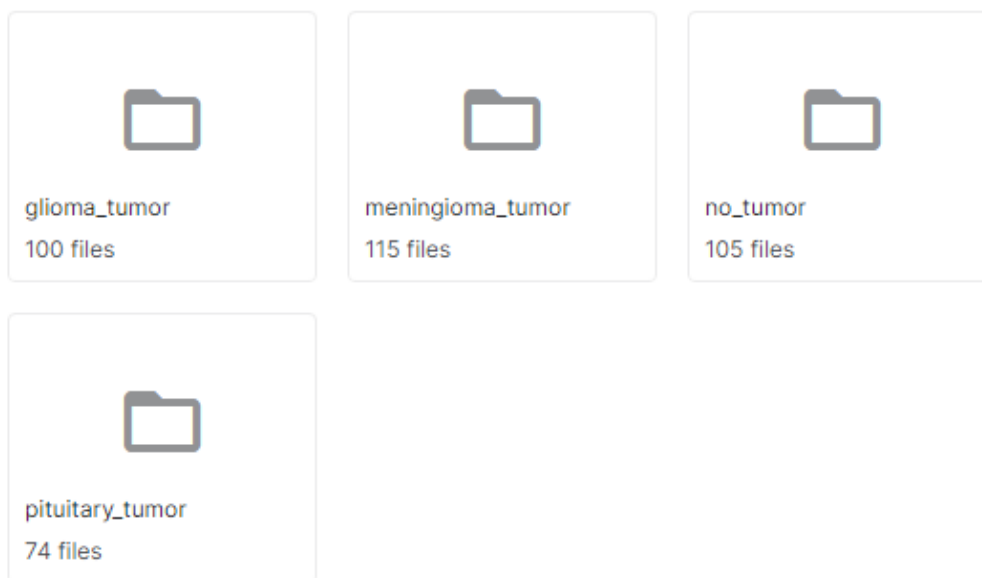


Figure 13 : Les quatre classes de tumeurs dans le dossier testing.

IV.2 Langage de programmation utilise

MATLAB (matrix laboratory) est un langage de programmation de quatrième génération émulé par un environnement de développement du même nom, il est utilisé à des fins de calcul numérique. Développé par la société The Math Works, MATLAB permet de manipuler des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes, de créer des interfaces utilisateurs, et peut s'interfacer avec d'autres langages comme le C, C++, Java, et Fortran. Les utilisateurs de MATLAB (environ un million en 2004) sont de milieux très différents comme l'ingénierie, les sciences et l'économie dans un contexte aussi bien industriel que pour la recherche. Matlab peut s'utiliser seul ou bien avec des *toolbox*(« boîte à outils »). [15]



Figure 14 : Logo MATLAB [15] .

IV.3 Les étapes de l'implémentation

IV.3.1 Import la base de données

A screenshot of a MATLAB script editor window titled 'hp_alexnet.m'. The script contains two lines of code: line 3: 'imdsTest = imageDatastore('D:\brain_tumor_dataset\Testing', 'IncludeSubfolders', true, 'LabelSource', 'foldernames');' and line 4: 'imdsTrain = imageDatastore('D:\brain_tumor_dataset\Training', 'IncludeSubfolders', true, 'LabelSource', 'foldernames');'. The editor shows line numbers 1 through 7 on the left margin. The Command Window is visible at the bottom.

```
1  
2 %dataset  
3 imdsTest = imageDatastore('D:\brain_tumor_dataset\Testing', 'IncludeSubfolders', true, 'LabelSource', 'foldernames');  
4 imdsTrain = imageDatastore('D:\brain_tumor_dataset\Training', 'IncludeSubfolders', true, 'LabelSource', 'foldernames');  
5  
6  
7
```

Figure 15 : Import La Base De Données.

IV.3.2 Les descripteurs

IV.3.2.1 Alexnet

Alexnet est formé à une image de plus d'un million et peut classer les images en 1000 catégories d'objets

En conséquence, le modèle a appris des représentations de fonctionnalités riches pour une large gamme d'images

```
hp_alexnet.m x +
1
2 %dataset
3 imdsTest = imageDatastore('D:\brain_tumor_dataset\Testing','IncludeSubfolders',true,'LabelSource','foldernames');
4 imdsTrain = imageDatastore('D:\brain_tumor_dataset\Training','IncludeSubfolders',true,'LabelSource','foldernames');
5
6
7 net = alexnet;
8 layer='fc8';
9 inputSize = net.Layers(1).InputSize;
10
11 TrainingImages =augmentedImageDatastore(inputSize,imdsTrain,'ColorPreprocessing','gray2rgb');
12 UnlabelledImages =augmentedImageDatastore(inputSize,imdsTest,'ColorPreprocessing','gray2rgb');
13
14 DTrai = activations(net,TrainingImages,layer,'OutputAs','rows');
15 DTesi = activations(net,UnlabelledImages,layer,'OutputAs','rows');
16
17 LTrai = double(imdsTrain.Labels);
18 LTesi = double(imdsTest.Labels);
19
20
Command Window
```

Figure 16 : Code Alexnet.

IV.3.2.2 GIST

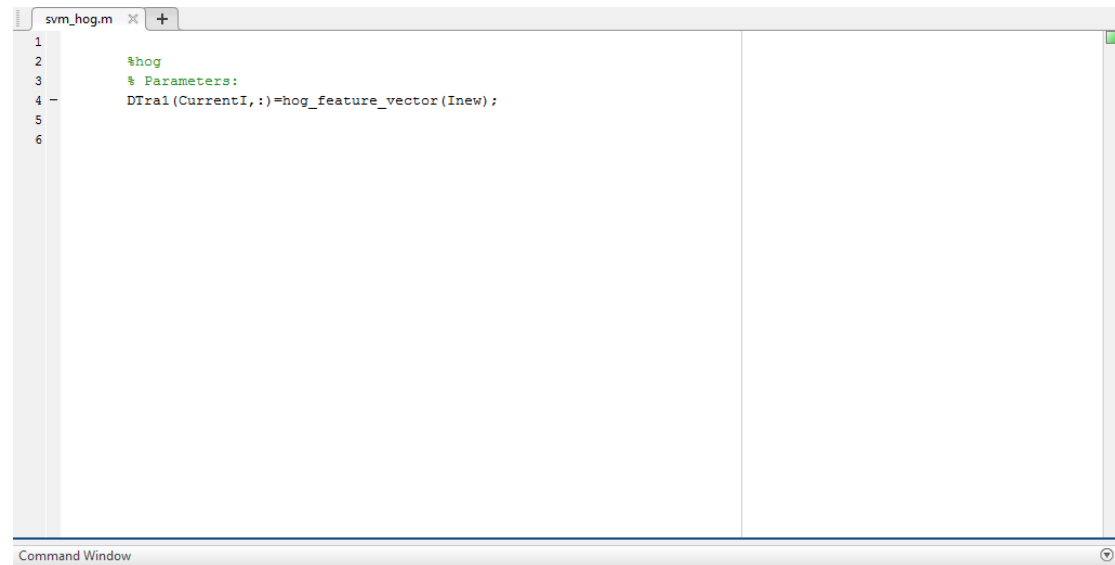
Le descripteur GIST (Oliva et Torralba, 2001) calcule l'énergie d'une banque de filtres de type Gabor évaluée à 8 orientations et 4 échelles différentes. La sortie carrée de chaque filtre est ensuite moyennée sur une grille 4×4 .

```
hp_gist.m x +
1
2 %Gist
3 % Parameters:
4 clear param
5 param.imageSize = [256 256]; % it works also with non-square images
6 param.orientationsPerScale = [8 8 8 8];
7 param.numberBlocks = 4;
8 param.fc_prefilt = 4;
9 % Computing gist requires 1) prefilter image, 2) filter image and collect
10 % output energies
11 [gist, param] = LMgist(I,'', param);
12 DTrai(CurrentI,:)=gist;
13
Command Window
```

Figure 17 : Code GIST.

IV.3.2.3 HOG

Une façon de voir HOG (Dalal et Triggs, 2005) est une version simplifiée de SIFT. HOG calcule les gradients d'intensité d'un pixel à l'autre et sélectionne une case d'histogramme correspondante pour chaque pixel en fonction de la direction du gradient.



```
svm_hog.m  x +
1
2      %hog
3      % Parameters:
4      DTral(CurrentI, :) = hog_feature_vector(Inew);
5
6
```

Figure 18 : Code HOG.

IV.3.2.4 VGG-16

Le VGG-16 est actuellement l'un des descripteurs de fonctionnalités pré-entraînés les plus performants au monde

Utilisé dans différents travaux. Il a une structure plus simple, mais est capable de fournir de meilleures performances sur divers problèmes.

```
DTree_Vgg16.m  x +
1  %dataset
2  imdsTest = imageDatastore('D:\archive (1)\Testing','IncludeSubfolders',true,'LabelSource','foldernames');
3  imdsTrain = imageDatastore('D:\archive (1)\Training','IncludeSubfolders',true,'LabelSource','foldernames');
4
5
6  net = vgg16;
7  %analyzeNetwork(net)
8  layer='fc8';
9  inputSize = net.Layers(1).InputSize;
10
11  TrainingImages =augmentedImageDatastore(inputSize,imdsTrain,'ColorPreprocessing','gray2rgb');
12  UnlabelledImages =augmentedImageDatastore(inputSize,imdsTest,'ColorPreprocessing','gray2rgb');
13
14  DTral = activations(net,TrainingImages,layer,'OutputAs','rows');
15  DTes1 = activations(net,UnlabelledImages,layer,'OutputAs','rows');
16
17  LTral = double(imdsTrain.Labels);
18  LTes1 = double(imdsTest.Labels);
19
Command Window
```

Figure 19 : Code VGG-16.

IV.3.3 Les algorithmes

IV.3.3.1 SVM

```
svm_hog.m  x +
1
2  %svm
3
4  classifier = fitcecoc(DTral,LTral);
5  %classify the test image
6  YPred = predict(classifier,DTes1);
7
8  confMat = confusionmat(LTes1,YPred);
9
10 % Convert confusion matrix into percentage form
11 confMat = bsxfun(@rdivide,confMat,sum(confMat,2))
12 % Display the mean accuracy
13 mean(diag(confMat))
Command Window
```

Figure 20 : Code SVM.

IV.3.3.2 DTree

```
DTree_Vgg16.m x +
1
2 % DTree
3 Mdl = fitctree(DTral,LTral)
4 label = predict(Mdl,DTest1);
5 accuracy_DTREE=sum(label==LTest1)/size(label,1)
6
```

Command Window

Figure 21 : Code DTree.

IV.3.3.3HP

```
hp_alexnet.m x +
1 % Train the HP classifier
2 Input.Data_Train=DTral; % Training data
3 Input.Label_Train=LTral; % Ground truth
4 Input.LayerNum=5; % Layer number
5 [Output0]=HP(Input,'learning');
6 % Validate the trained HP classifier with Mode A
7 Input1.Data_Test=DTest1; % Validation data
8 Input1.Syst=Output0.Syst; % Trained HP classifier
9 [Output1]=HP(Input1,'testinga');
10 ConfusionMatrix=confusionmat(LTest1,Output1.Labels); % Confusion matrix of the prediction result
11 Acc1=sum(sum(ConfusionMatrix.*eye(size(ConfusionMatrix,1))))/sum(sum(ConfusionMatrix)) % Accuracy of the prediction re
12 % Validate the trained HP classifier with Mode B
13 Input1.Data_Test=DTest1; % Validation data
14 Input1.Syst=Output0.Syst; % Trained HP classifier
15 [Output1]=HP(Input1,'testingb');
16 ConfusionMatrix=confusionmat(LTest1,Output1.Labels); % Confusion matrix of the prediction result
17 Acc2=sum(sum(ConfusionMatrix.*eye(size(ConfusionMatrix,1))))/sum(sum(ConfusionMatrix)) % Accuracy of the prediction re
18
```

Command Window

Figure 22 : Code HP.

IV.4 Les tableau de résultat

| Data | layer | mode | accuracy | Temps |
|---------------------|-------|------|----------|---------------|
| Base de données (1) | 1 | A | 0.7816 | 00 :00 :13.04 |
| | | B | 0.7816 | |
| | 2 | A | 0.7931 | 00 :00 :14.48 |
| | | B | 0.7931 | |
| | 3 | A | 0.7701 | 00 :00 :13.61 |
| | | B | 0.7701 | |
| | 4 | A | 0.8046 | 00 :00 :17.34 |
| | | B | 0.7931 | |
| | 5 | A | 0.8046 | 00 :00 :15.74 |
| | | B | 0.7931 | |
| Base de données (2) | 1 | A | 0.4239 | 07 :31 :12 |
| | | B | 0.4239 | |
| | 2 | A | 0.4619 | 07 :30 :10 |
| | | B | 0.4619 | |
| | 3 | A | 0.7665 | 09 :00 :10 |
| | | B | 0.7360 | |
| | 4 | A | 0.7817 | 06 :45 :00 |
| | | B | 0.7259 | |
| | 5 | A | 0.8122 | 05.30.05 |
| | | B | 0.7284 | |

Tableau 01: Nombre de prototypes identifiés par classe et Comparaison sur les bases de données.

Comme on peut le voir dans le tableau 1, le classificateur HP est capable de percevoir l'espace de données et d'identifier les prototypes à partir de données à différents niveaux de granularité. En général, une couche supérieure de la hiérarchie contient généralement moins de prototypes mais plus abstraits, et il est capable d'effectuer une classification plus efficace mais plus grossière. En revanche, une couche inférieure contient plus prototypes et partitionne l'espace de données de manière plus fine. En conséquence, il peut être utilisé pour effectuer la classification avec une plus grande précision. Plus le classificateur HP peut obtenir des résultats de partitionnement de données plus détaillés. les échantillons sont reconnus comme des prototypes dans les couches inférieures des hiérarchies basées sur les prototypes. Dans de tels cas, le classificateur démontre de meilleures performances de classification lorsque le mode A est utilisé pendant l'étape de validation. Pour évaluer l'influence du numéro de couche sur l'efficacité du calcul, un classificateur HP est formé sur le

précédemment utilisé 2 ensembles de données. La relation entre le temps de formation la consommation et le nombre de couches sont représentés sur tableau 1. On peut voir que, le calcul l'efficacité du classificateur HP est (presque) linéairement corrélé au numéro de couche. Plus le HP à de couches classificateur a, plus le temps est consommé pendant le processus d'apprentissage. Cependant, il convient de noter que si le plus haut niveau de granularité pour le partitionnement des données a été atteint et tous les échantillons de données ont été reconnus comme des prototypes dans les couches inférieures, l'ajout de couches supplémentaires au classificateur HP ne sera pas de manière significative augmenter la consommation de temps d'entraînement.

| Algorithme | Descripteurs | Mode | Accuracy |
|------------|--------------|------|----------|
| HP | HOG | A | 0.8161 |
| | | B | 0.7816 |
| | GIST | A | 0.7356 |
| | | B | 0.6552 |
| | ALEXNET | A | 0.8046 |
| | | B | 0.7931 |
| | VGG-16 | A | 0.7586 |
| | | B | 0.7586 |
| SVM | HOG | | 0.8099 |
| | GIST | | 0.7645 |
| | ALEXNET | | 0.8818 |
| | VGG-16 | | 0.8818 |
| DTree | HOG | | 0.5057 |
| | GIST | | 0.5632 |
| | ALEXNET | | 0.6322 |
| | VGG-16 | | 0.7241 |

Tableau 02 : Comparaison de qualité entre Descripteurs par différents algorithmes sur Base de données 1.

Comme on peut le voir dans le tableau 2, pendant la phase d'apprentissage, le classificateur HP est efficace en résultat avec le descripteur: Hog et Alexnet. Son efficacité de résultat est également supérieure à DTree et SVM avec descripteur : hog. On peut remarquer que le classificateur hp produit les résultats de classification les plus précis sur ce problème avec 4 descripteurs. Cela indique que la précision de classification HP peut être encore améliorée en ajoutant plus de descripteur. Il est également à noter que pendant la phase de prise de décision, le classificateur HP utilisant le mode A se traduit par de meilleures performances de classification. en termes de précision.

| Algorithme | Descripteurs | Mode | Accuracy |
|------------|--------------|------|----------|
| HP | HOG | A | 0.8061 |
| | | B | 0.7216 |
| | GIST | A | 0.7690 |
| | | B | 0.7183 |
| | ALEXNET | A | 0.7817 |
| | | B | 0.7259 |
| | VGG-16 | A | 0.8122 |
| | | B | 0.7284 |
| SVM | HOG | | 0.6301 |
| | GIST | | 0.4452 |
| | ALEXNET | | 0.7619 |
| | VGG-16 | | 0.7619 |
| DTree | HOG | | 0.1472 |
| | GIST | | 0.0685 |
| | ALEXNET | | 0.7132 |
| | VGG-16 | | 0.7284 |

Tableau 03 : Comparaison de qualité entre Descripteurs par différents algorithmes sur Base de données 2.

Comme on peut le voir dans le tableau 3, pendant la phase d'apprentissage, le classificateur HP est efficace en résultat avec le 4 descripteur. Son efficacité de résultat est également supérieure à DTree et SVM avec descripteur : hog et gist. On peut remarquer que hp peut gérer des bases des données volumineuses contrairement à DTree, qui nécessite des descripteur puissants tels que ALEXNET, et contrairement à SVM, qui a perdu en efficacité lors du changement des bases des données en bases des données plus volumineuses.

IV.5 Les mesure de performances

IV.5 .1 Taux d'erreur : accuracy

| | | PREDICTED CLASS | |
|--------------|-----------|-----------------|-----------|
| | | Class=Yes | Class=No |
| ACTUAL CLASS | Class=Yes | a (TP) | b (FN) |
| | Class=No | c (FP) | d (TN) |

Figure 23: predicted class et actual class.

$$\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

Quelques limitations on considère un problème a 2 classes avec : 9990 instances de classe 0 et 10 instances de classe 1. Si le modèle prédit que tout instance est de classe 0, on a Accuracy = 9990/10000 = 99,9

IV.5 .2 Recall vs precision

Recall (True positive rate sensitivity) de ceux qui existent, combien l'algorithme a pu trouver

$$TPR = \frac{TP}{TP + FN}$$

Precision de ceux que l'algorithme a pu classer, combien sont corrects.

$$PPV = \frac{TP}{TP + FP}$$

IV.5 .3 F-mesure

Moyenne harmonique entre la precision et le rappel

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2TP}{2TP + FP + FN}$$

| accuracy | sensitivity | specificity | precision | recall | f_measure | gmean |
|-------------|-------------|-------------|-----------|---------|-----------|--------|
| 0.793103448 | 0.84375000 | 0.76363636 | 0.675000 | 0.84375 | 0.750000 | 0.8026 |

Tableau 04 : les mesures de performance pour hp avec hog

IV.6 La matrice de confusion

IV.6.1 Définition

Une Confusion Matrice est un résumé des résultats de prédictions sur un problème de classification. Les prédictions correctes et incorrectes sont mises en lumière et réparties par classe. Les résultats sont ainsi comparés avec les valeurs réelles.

Cette matrice permet de comprendre de quelle façon le modèle de classification est confus lorsqu'il effectue des prédictions. Ceci permet non seulement de savoir quelles sont les erreurs commises, mais surtout le type d'erreurs commises. Les utilisateurs peuvent les analyser pour déterminer quels résultats indiquent comment les erreurs sont commises.

Outre le Machine Learning, les matrices de confusion sont aussi utilisées dans le domaine des statistiques, du Data Mining et de l'intelligence artificielle. De manière générale, elles permettent d'analyser des données statistiques plus rapidement et de rendre les résultats plus simples à déchiffrer via la Data Visualisation. Elles offrent l'opportunité d'analyser les erreurs dans les statistiques, le forage de données, ou même les examens médicaux.

IV.6.2 Comment calculer une confusion matrice

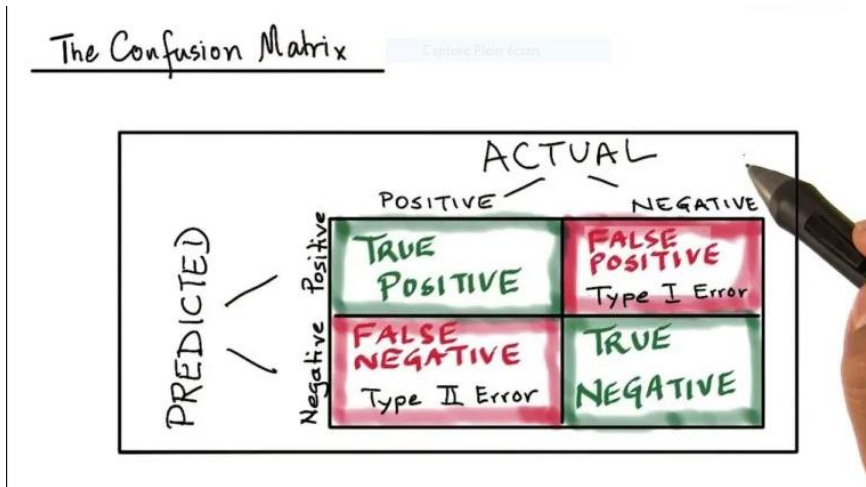


Figure 24 : la confusion matrice.

Pour calculer une matrice de confusion, il est nécessaire de disposer d'un ensemble de données de test (test dataset) ou d'un ensemble de données de validation (validation dataset) avec les valeurs de résultat attendues. On fait ensuite une prédiction pour chaque ligne du "test dataset".

A partir des résultats escomptés et des prédictions, la matrice indique le nombre de prédictions correctes pour chaque classe et le nombre de prédictions incorrectes pour chaque classe organisées en fonction de la classe prédite. Chaque ligne du tableau correspond à une classe prédite, et chaque colonne correspond à une classe réelle. Dans les lignes sous les classes réelles, les prédictions ou les résultats sont inscrits. Ces résultats peuvent être l'indication correcte d'une prédiction positive comme "vraie positive" (true positive) et d'une prédiction négative comme "vraie négative" (true négative), ou une prédiction positive incorrecte comme "fausse positive" (false positive) et une prédiction négative incorrecte comme "fausse négative" (false négative).

L'avantage de ces matrices est qu'elles sont très simples à lire et à comprendre. Elles permettent de visualiser très rapidement les données et les statistiques afin d'analyser les performances d'un modèle et d'identifier les tendances qui peuvent aider à modifier les paramètres. Pour les problèmes de classification avec trois classes ou plus, il est aussi possible d'utiliser une Confusion Matrix en ajoutant des lignes et des colonnes.

IV.6.3 Comprendre les terminologies de la Confusion Matrice

Pour bien comprendre le fonctionnement d'une matrice de confusion, il convient de bien comprendre les quatre terminologies principales : TP, TN, FP et FN. Voici la définition précise de chacun de ces termes :

- **TP (True Positives)** : les cas où la prédiction est positive, et où la valeur réelle est effectivement positive. Exemple : le médecin vous annonce que vous êtes enceinte, et vous êtes bel et bien enceinte.
- **TN (True Negatives)** : les cas où la prédiction est négative, et où la valeur réelle est effectivement négative. Exemple : le médecin vous annonce que vous n'êtes pas enceinte, et vous n'êtes effectivement pas enceinte.
- **FP (False Positive)** : les cas où la prédiction est positive, mais où la valeur réelle est négative. Exemple : le médecin vous annonce que vous êtes enceinte, mais vous n'êtes pas enceinte.
- **FN (False Negative)** : les cas où la prédiction est négative, mais où la valeur réelle est positive. Exemple : le médecin vous annonce que vous n'êtes pas enceinte, mais vous êtes enceinte.

| | | | |
|---|--------------|--------------|--------------|
| 0 | 0 0.0% | 0 0.0% | NaN% NaN% |
| 1 | 0 0.0% | 87 100% | 100% 0.0% |
| | NaN% NaN% | 100% 0.0% | 100% 0.0% |
| | 0 | 1 | |
| | Target Class | | |

Figure 25 : Confusion Matrix HP.

IV.7 Courbe roc

IV.7.1 Définition

La courbe ROC (Receiver Operating Characteristic) dessine l'évolution du taux du vrai positif (TPR) en fonction du taux du faux positif (FPR) en faisant varier un seuillage sur la confiance (probabilité) qu'un exemple soit dans la classe positif

Rappel :

$$TPR = \frac{TP}{TP+FN}, FPR = \frac{FP}{TN+FP}$$

Différent méthode peut mieux fonctionner dans différentes parties de l'espace ROC. Cela dépend du cout de false + vs. false -

Soit x_i positif (+) si $p(y = 1 | x_i) > \theta$, sinon il est négatif (-) ($y = 0$) $\hat{y}_i = 1 \Leftrightarrow p(y=1 | x_i) > \theta$

Le nombre des TPs et FPs dépend du seuillage θ , varier θ donne des points (TPR, FPR) différents.

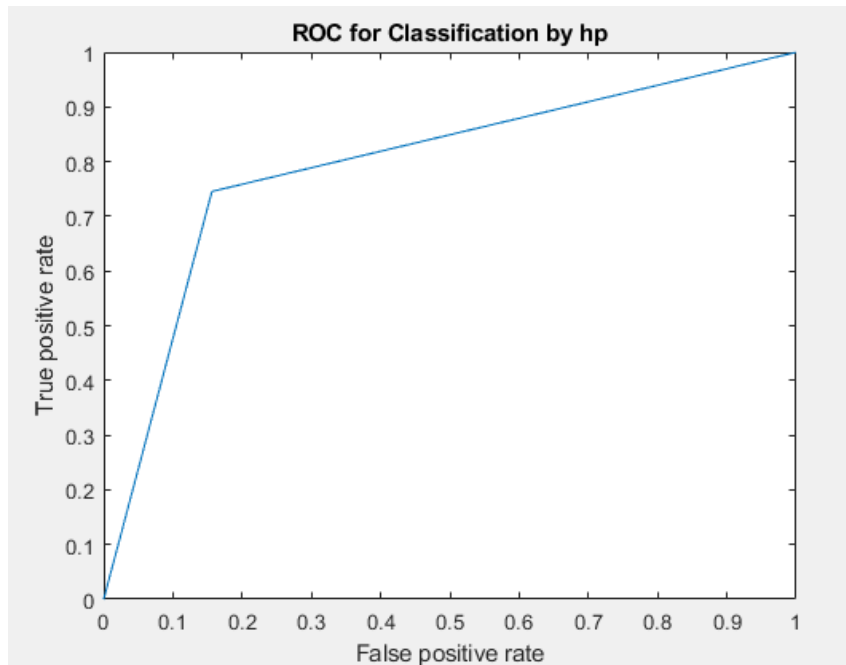


Figure 26 : Courbe Roc HP.

Conclusion

Dans ce chapitre nous avons présenté notre résultat ainsi que les démarches à suivre pour son fonctionnement.

Nous avons décrit les étapes de la classification de l'approche proposée pour répondre à la problématique de la détection des pathologies cérébrales. Donc nos objectifs principaux sont atteints par l'obtention de bons résultats de la classification des différentes des images médicales ainsi que la fixation de meilleurs paramètres.

Conclusion générale

Conclusion générale

Ce mémoire présente une nouvelle approche de classification basée sur un prototype hiérarchique (HP). La proposition de l'approche a une structure de système hiérarchique hautement transparente composée de prototypes significatifs. Celles-ci les prototypes sont identifiés par un processus d'apprentissage autonome et non itératif, et ils représentent naturellement le pics locaux de distributions multimodales dérivées de données à différents niveaux de granularité. Le classificateur HP peut commencer à travailler à partir du tout premier exemple de formation par classe et apprendre continuellement de nouvelles observations dans un calcul simplifié, «en un seul passage» et auto-évolutif de manière dynamique pour suivre l'évolution rapide des données motifs. Grâce à sa nature prototype, les processus d'apprentissage et de prise de décision du classificateur HP sont entièrement interprétables, traçables, explicables à / par les humains. L'approche proposée est également capable de visualiser les connaissances acquises, à savoir les prototypes identifiés avec différents niveaux de détails et les liens significatifs entre eux, sous une forme hiérarchique compréhensible par l'homme naturellement. Les utilisateurs peuvent utiliser les couches inférieures plus sophistiquées du classificateur HP pour une classification plus fine ou / et la partie supérieure plus généralisée couches pour comprendre rapidement des problèmes complexes et effectuer une classification grossière. Exemples numériques sur divers problèmes de référence ont démontré que le classificateur HP peut effectuer une classification très précise surpassant ou, du moins, à égalité avec les approches de pointe. De plus, le très haut niveau de calcul l'efficacité du classificateur HP sur des problèmes à grande échelle est également justifiée.

Il y a plusieurs considérations pour les travaux futurs. Premièrement, l'optimalité du classificateur HP doit être enquêté. Ceci est d'une importance capitale pour l'approche proposée car il détermine la validité et l'efficacité des résultats d'apprentissage. Il y a quelques modifications potentielles qui peuvent améliorer l'objectivité et l'efficacité du classificateur HP et renforcer sa capacité d'adaptation et son applicabilité au monde réel problèmes. Par exemple, le classificateur HP actuel nécessite une reconversion complète si des couches supplémentaires doivent être ajoutés à la structure du système. Une approche pour ajouter dynamiquement de nouvelles couches en fonction des besoins lors de la le processus d'apprentissage en ligne sera précieux. Ce sera une forte nouveauté si certains mécanismes de fonctionnement sont introduits dans le classificateur HP qui permet à l'approche de déterminer

de manière autonome le numéro de couche optimal pour chaque hiérarchie en fonction des propriétés d'ensemble des données. Alternativement, développer des critères qui peuvent aider les utilisateurs à déterminer le numéro de couche le plus souhaitable pour un problème donné sans connaissance préalable sera également soit utile. Une autre direction importante sera l'introduction d'une approche basée sur les données pour déterminer les paramètres réglage (rayons) pour remplacer la méthode de codage en dur utilisée par la version actuelle.

Bibliographie :

- [1] <https://mrmint.fr/lapprentissage-non-supervise-machine-learning>
- [2] <https://mrmint.fr/apprentissage-supervise-machine-learning>
- [3] <http://chimie.sup.pagesperso-orange.fr/IRM.html>
- [4] <https://towardsdatascience.com/coding-deep-learning-for-beginners-types-of-machine-learning>
- [5] <https://mrmint.fr/introduction-machine-learning>
- [6] <http://association-amas.e-monsite.com/pages/informations-pathologies/le-meningiome.html>
- [7] <https://www.info-radiologie.ch/meningiome-irm.php>
- [8] <https://fr.wikipedia.org/wiki/Gliome>
- [9] <https://www.mayoclinic.org/diseases-conditions/pituitary-tumors/symptoms-causes/syc-20350548>
- [10] Thème/ La logique floue pour Classification Des Feuilles de vigne / gharbifatimazahra - ben namiahalima
- [11] <https://le-datascientist.fr/les-svm-support-vector-machine>
- [12] <https://medium.com/@dhiraj8899/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107>
- [13] MÉMOIRE / APPLICATION DE LA CLASSIFICATION TEXTUELLE POUR L'EXTRACTION DES RÈGLES D'ASSOCIATION MAXIMALES
- [14] Journal Pre-proof / A Hierarchical Prototype-Based Approach for Classification
- [15] <https://fr.wikipedia.org/wiki/MATLAB>

Résumé

Dans ce travail, nous extraire les caractéristiques locales dans le ROI, en utilisant différents types d'entités descripteurs utilisés dans la classification d'apprentissage automatique. Dans ce mémoire, nous avons utilisé deux descripteurs de caractéristiques de bas niveau (GIST et HOG) et un descripteur de caractéristiques de haut niveau (un VGG-VD et alexnet), enfin une décision est prise en utilisant un nouvelle approche basée sur un prototype hiérarchique (HP) avec de telles caractéristiques est proposée pour classification.

A fin de traiter la problématique centrale de cette thèse, nous décrivons en dernier lieu une nouvelle méthode pour la détection des pathologies cérébrales. Le formalisme de cette méthode est assez puissant pour, d'une part caractériser plusieurs types de pathologies et d'autre part, être appliqué afin d'optimiser d'autres méthodes (déjà existantes). La phase expérimentale montre que ces quatre systèmes obtiennent des résultats prometteurs à différents niveaux.

Abstract

In this work, we are extract the local characteristics in the ROI, using different types of descriptor entities used in classification of the machine learning. In this thesis we have used two low level feature descriptors (GIST and HOG) and one high level feature descriptor (a VGG-VD and alexnet), finally a decision is made using a new prototype based approach. Hierarchical (HP) with such characteristics is proposed for classification..

To address the central issue of this thesis, we describe lastly a new method for brain pathologies detection. The formalism of this method is powerful enough to characterize several types of diseases on the one hand and to be applied for optimizing other existing methods on the other hand.

The experimental phase shows that the proposed systems achieve promising results at different levels.

ملخص

في هذا العمل، استخرجنا الخصائص المحلية، باستخدام أنواع مختلفة من الوصفات المستخدمة في مجال تصنيف خاصة بتعلم الآلي. في هذه الأطروحة استخدمنا اثنين من واصفات الميزات منخفضة المستوى (GIST و HOG) وواصف ميزة عالي المستوى (VGG-VD و alexnet)، وأخيراً تم اتخاذ القرار باستخدام نهج جديد قائم على النموذج الأولي (HP) مع هذه الخصائص المقترحة للتصنيف.

لمعالجة القضية الرئيسية لهذه الأطروحة، نصف أخيراً طريقة جديدة للكشف عن أمراض الدماغ. هذه الطريقة قادرة على وصف عدة أنواع من الأمراض من ناحية، ومن ناحية أخرى، يمكن تطبيقها من أجل تحسين الأساليب الأخرى (الموجودة بالفعل).

تبين المرحلة التجريبية أن هذه الأنظمة الأربعة تحصل على نتائج واعدة على مستويات مختلفة.