
Rapport final

Projet long

Réalisé par :

Youness AKOURIM

30/03/2021

youness.akourim@ens-rennes.fr

Introduction

Dans la dernière décennie, on constate une nouvelle tendance de l'usage de l'intelligence artificielle dans le secteur médical afin de faciliter l'identification des symptômes d'une maladie neurologique comme l'Alzheimer. Souvent les médecins utilisent des données vocales et musculaires pour diagnostiquer l'état du patient. Or, cette méthode prend beaucoup de temps de la faire et elle est aussi coûteuse. C'est pourquoi, il est préférable d'utiliser un modèle de classification capable de prédire l'état du patient, autrement dit s'il est malade ou bénin.

L'identification des symptômes de l'Alzheimer se fait généralement par l'analyse des mouvements musculaires d'un patient en lui demandant de tracer des formes géométriques bien précises, par exemple une spirale, puis on compare le dessin avec un dessin normal. Du coup, on peut entraîner une architecture CNN prédéfinie pour déterminer les features ensuite les classer. Grâce à la tablette graphique, on peut avoir des caractéristiques statiques et dynamiques du tracé ce qui va nous aider à mieux classer les données.

Après une brève introduction du jeu de données, on verra la structure de l'architecture VGG 16 utilisée pour l'extraction des features, ensuite on traitera la méthode de sélection des features afin de les classer avec 3 modèles statistiques différents. Finalement, on va comparer les performances de chaque estimateur en utilisant plusieurs métriques.

Base de données

La base de données utilisée pour notre étude est une collection des images de spirale dessinées par des patients qui sont de niveaux de gris c'est-à-dire chaque pixel prend des valeurs entre 0 et 255. La taille des images est de (224,224) qui est adaptée à l'architecture qu'on va l'utiliser par la suite pour l'entraînement du modèle.

On a 75 images partagées en deux parties : 45 images pour la classe Control (bénin) et 30 images pour la classe AD (l'Alzheimer Disease). On voit bien dans la figure 1 la différence entre une spirale AD et control dont la première est mal faite avec des vibrations dans le tracé plus des déformations au niveau la forme, et la deuxième est plutôt fluide et bien faite. Cependant, ce critère visuel ne suffit pas pour prédire les symptômes précoces de la maladie d'alzheimer car on peut avoir des images bien faites mais elles sont pour des patients malades et aussi des images mal faites pour des patients bénins.

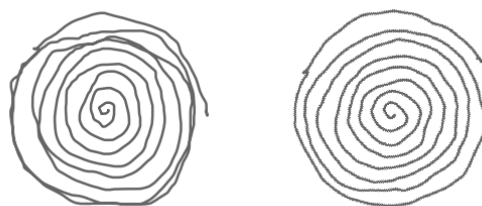


figure 1- spirale AD (à gauche) et spirale control (à droite)

La tablette graphique permet de générer deux catégories d'images : statique et dynamique. Les images statiques présentent des données sur la forme géométrique de la spirale par exemple son épaisseur, ses dimensions (hauteur, largeur) ... etc. C'est souvent les images utilisées chez les médecins pour diagnostiquer la maladie. Les images dynamiques présentent la dynamique du pointeur du stylo utilisé lors du dessin comme l'altitude du pointeur, sa vitesse et sa pression sur la tablette. Le système de coordonnées est celui du plan de la tablette.

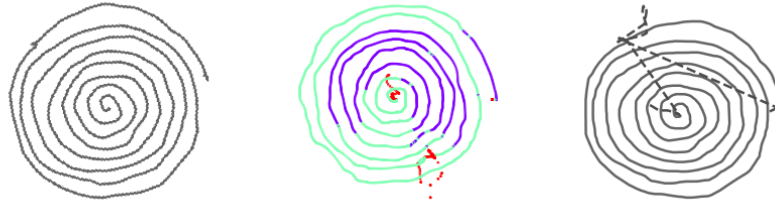


figure 2- spirale control de type RAW (à gauche), Pressure (au milieu) et Penups Raw (à droite)

Pour notre étude, on va utiliser 2 types d'images statiques (Raw : spirale brute et Penups raw : spirale avec le levé du stylo) et un type d'image dynamique (Pressure). Pour les images dynamiques, chaque pixel prend trois valeurs possibles correspondant de 3 niveaux de pressions prédéfinis dans la tablette. Ainsi, les images dynamiques permettent de mieux classer les individus car elles présentent des informations locales du tracé ce qu'on ne peut pas les avoir avec une image statique.

Transfert learning

Classifier les images dans son espace n'est pas très optimal en termes de temps de calcul et de la mémoire occupée pour cette tâche. C'est bien qu'on fait recours aux architectures Deep CNN qui nous permettent d'extraire des informations utiles sur des zones locales dans l'image car d'ailleurs on a un fond blanc qui commun pour toutes les images et qui ne va pas servir à classifier les images.

1. Architecture VGG 16

L'architecture abordée pour cette étude est le réseau VGG16 proposé par les chercheurs K. Simonyan et A. Zisserman de l'Université d'Oxford. Il permet d'atteindre 92.7% sur la base de données ImageNet qui contient 14 Millions d'images appartenant à 1000 classes.

L'image d'entrée est de taille (224, 224, 3) car on travaille avec des images de type RGB. La sortie du réseau VGG 16 est un vecteur binaire de taille (1000,1). On peut décomposer l'architecture en deux parties :

- Partie "Extraction des features" où on calcule les features d'une image dans une couche de convolution. Le VGG 16 comme son nom l'indique contient 16 couches de convolutions partagées en 5 blocs. Le nombre de filtres est le même pour les couches qui appartiennent

au même bloc dont le premier bloc a 64 filtres, le deuxième bloc 128 filtres, le troisième bloc 256 filtres et le 4ème et 5ème bloc 512 filtres.

- Partie “classification” où les features calculées dans la dernière couche de convolution (16ème couche) est de taille (25088,1) entraîne un réseau de neurones entièrement connecté de 3 couches avec un sortie softmax.

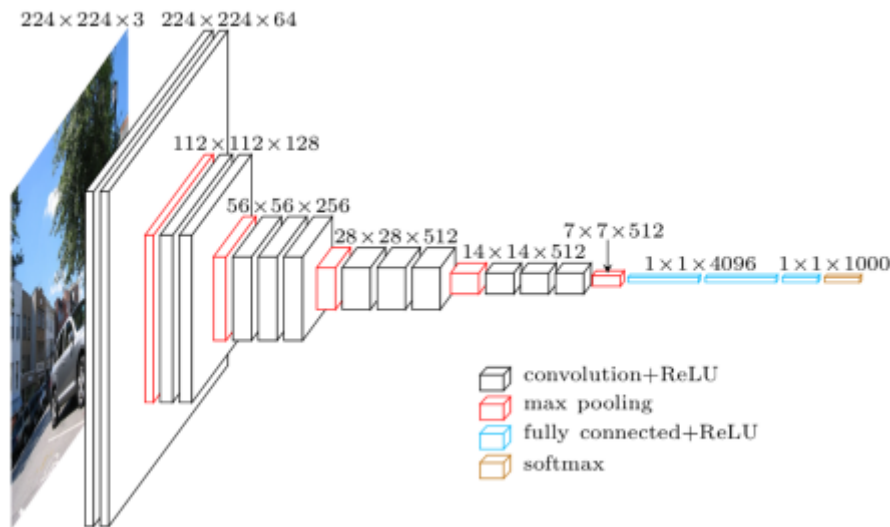


figure 3 - Schéma du réseau VGG 16

Ce qui nous intéresse c’est le partie Extracteur de features pour classifier les images dans un espace réduit.

2. Les Features MAP

Dans cette partie, on va visualiser les features MAP des images de deux classes différentes (AD et control) afin de comprendre les informations extraites dans chaque bloc.

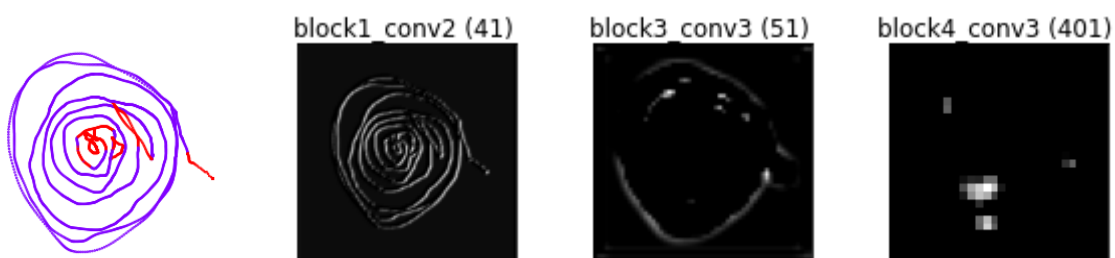


figure 4 - Features MAP de l'image A_0009_0.png

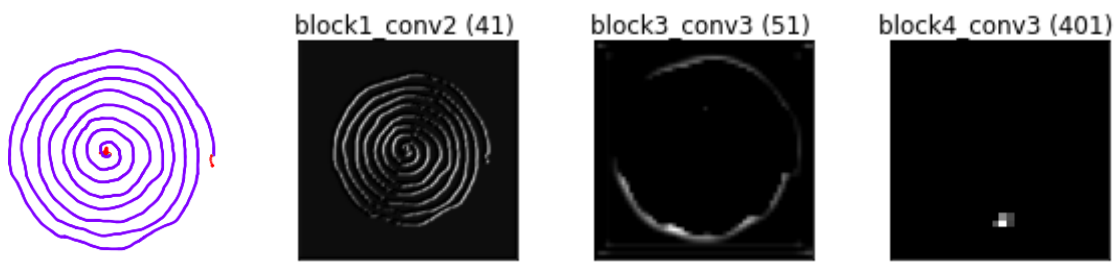


figure 5 Features MAP de l'image C_0009_0.png

Les filtres présentés dans les figures 4 et 5 sont choisis aléatoirement pour les blocs 1, 3 et 4 appliqués sur une image type **Pressure** pour un cas Alzheimer et un cas control. Le numéro de chaque filtre est indiqué entre les parenthèses. Les couches utilisées représentent la dernière couche du bloc.

On voit que dans le bloc 1, les features map donnent une information réelle de l'image où le filtre est activé au niveau du tracé. Alors que pour le bloc 3 le filtre est activé au niveau du bord de la spirale. Pour le bloc 4, on constate que le filtre détecte une forme particulière dans la spirale qu'on ne peut pas voir réellement.

Pour montrer la différence entre une image AD et control dans chaque couche de convolution, il faut visualiser toutes les filtres des 16 couches et rappelant que pour le premier bloc on a 64 filtres pour chaque couche et pour le dernier bloc, chaque couche a 512 filtres ce qui rend cette comparaison entre les images un peu pénible. Cependant, on voit bien que pour la couche 3 du bloc 4 le filtre s'activent dans une grande zone pour le cas AD alors qu'il est quasi faible pour le cas control.

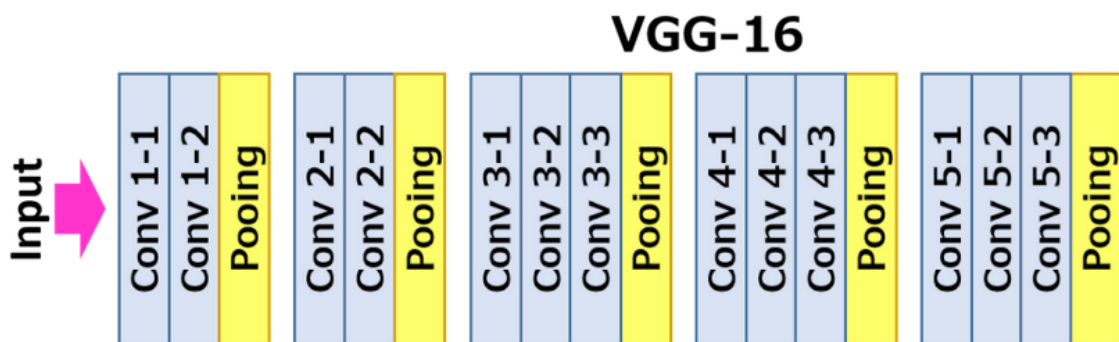


figure 6 - schéma de structure de VGG16

3. Data augmentation

Le manque de données est une première contrainte qu'on trouve pour notre démarche. Notre base de données contient 75 images (30 AD et 45 control). Elle est insuffisante pour l'entraînement d'un modèle de classification. Le deuxième problème se situe au niveau de la distribution des données dans lequel le nombre d'images control est grand par rapport à celui d'Alzheimer.

Pour fixer ces problématiques, on va proposer un protocole d'augmentation des données de telle façon qu'on sera le plus fidèle possible au jeu de données. On peut penser d'utiliser les filtres de convolution comme l'article [1] propose surtout un filtre médian avec un filtre EDGE mais ce protocole peut impacter sur la qualité des images et ainsi la performance du classifieur.

Notre idée est de générer une série de transformations de rotation appliquées pour chaque image, le nombre d'exemple généré pour chaque patient et le pas de rotation est déterminé de telle manière qu'on ait un équilibre des données pour les deux classes. Pour mieux comprendre le protocole, la figure 7 montre pour la classe AD (pour noter AD: Alzheimer Disease) on génère 15 images en appliquant une rotation de 10° pour chaque image, c'est-à-dire, on compte l'image initiale et on tourne l'image d'un pas de 10° jusqu'à 140° pour avoir finalement 15 exemples. Pour la classe control, on génère 10 images pour un pas de 15° .

Le choix du pas de rotation et le nombre d'exemples générés est justifié du fait qu'on cherche à équilibrer les données. Avec ce protocole, on a 450 images = $(14 + 1 \text{ image initiale}) * 30$ images pour la classe AD et 45 images = $(9 + 1 \text{ image initiale}) * 45$ images pour la classe control. Finalement, le nombre d'images du jeu de données est 900 images. Cette méthode fonctionne bien car le fond de l'image est blanc donc il est facile de tourner l'image sans la déformer. En plus, l'idée d'utiliser la rotation pour augmenter les données est plus fidèle que d'utiliser les filtres de convolution. Les images sont aussi bien centrées au milieu du cadre avec une marge vide qui tolère une éventuelle rotation appliquée sur l'image.

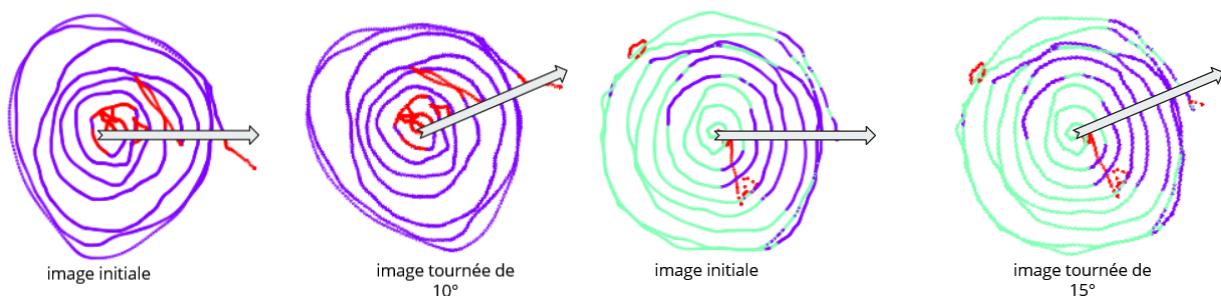


figure 7 - protocole d'augmentation d'images pour AD (à gauche) et pour control (à droite)

On peut penser à comparer deux images augmentées de le même individu en visualisant leurs features MAP. Dans la figure 8, on a deux images augmentées pour une image control de type Pressure, la première est dans la position initiale et la deuxième dans la position 45° . Les filtres utilisés sont choisis aléatoirement pour les dernières couches des blocs 1,3 et 5. Comme il est cité au début, dans le bloc 1 on voit la forme fidèle de la spirale pour les deux images. Le deuxième bloc présente plutôt le bord de la spirale. Jusqu'à ici, les filtres s'activent dans presque les mêmes endroits.

[1] : Assessing visual attributes of handwriting for prediction of neurological disorders— A case study on Parkinson's disease. Momina Moetesum & Al.

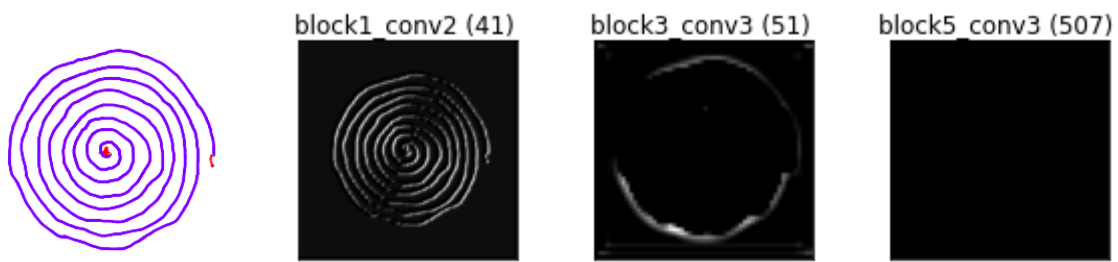


figure 8 - Features MAP de l'image C_0009_0.png

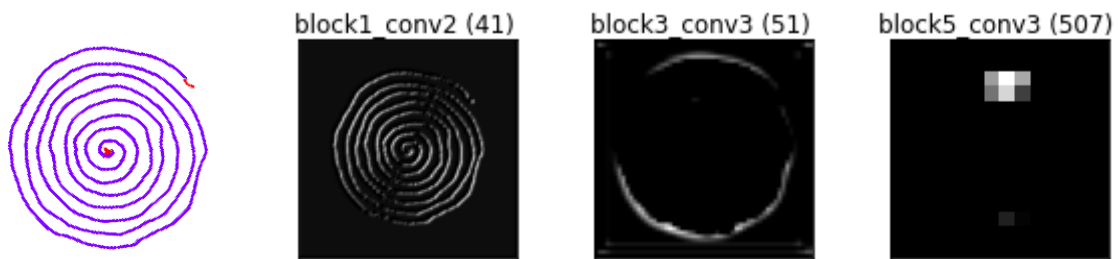


figure 9 - Features MAP de l'image C_0009_3.png

Néanmoins, pour le dernier bloc on constate une différence dans les deux cartes d'activation des deux images augmentées. Alors que le filtre n'est pas activé pour l'image initiale, on voit une nouvelle zone activée pour la deuxième image ce qui montre que l'architecture n'est pas biaisée par la rotation des images.

Sélection des features

Le nombre de features générées pour chaque image est très grand (25088 feats), du coup on s'intéresse à choisir le nombre de features minimal qui discrimine mieux les données car on constate l'existence des filtres qui ne s'activent jamais pour l'ensemble d'image vu qu'il s'agit d'une classification binaire pour des images simples (spirales).

L'une des méthodes qu'on l'utilise souvent pour la sélection des features est l'analyse des composantes principales. L'entraînement du réseau VGG 16 exige que la valeur de chaque pixel est comprise entre -1 et 1. Ainsi, on normalise les images (en divisant par 255) pour qu'elle respecte la condition de l'entraînement. Ainsi, la matrice finale Feat est de dimension (900, 25088).

Ensuite, on partage l'ensemble des feature en deux parties : **Ftrain** qui prend 2/3 de la dataset et **Ftest** qui prend 1/3 de la dataset. Cette distribution est bien équilibrée par classe pour les deux ensembles. Pour la décomposition ACP, le modèle est entraîné par **Ftrain** pour transformer toutes les features. Cette procédure va nous garantir l'absence d'un biais lors de l'entraînement du classifieur. Cette méthode est appliquée pour les 3 types d'images : Raw, Penups Raw et Pressure séparément.

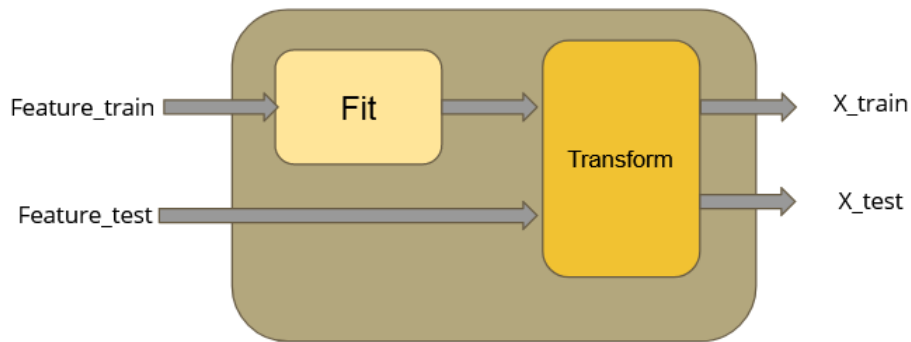


figure 10 - schéma explicatif de la décomposition ACP

Le choix de nombre de features est lié au pourcentage d'inertie portée par des axes (features). Pour notre cas, on exige un ratio de 95% d'information qui doivent être représentés par des features. Par la suite, on applique cette décomposition sur les images RAW, Penups Raw et Pressure séparément. La figure 11 nous montre que les deux classes sont bien superposées dans le plan factoriel. Pour la variable Pressure, on voit une forme typique des nuages d'individus surtout que juste le premier axe porte 70% de l'information alors que pour Penups et RAW il porte respectivement 26,09% et 20.25% de la variance expliquée.

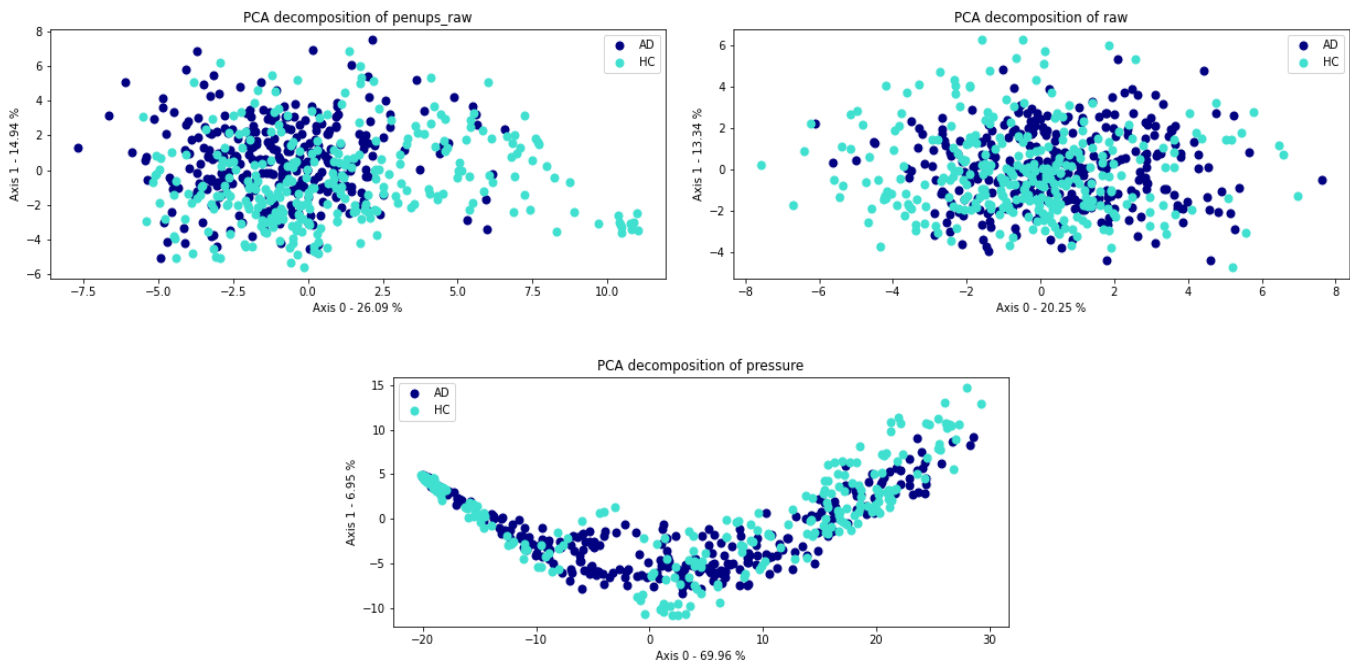


figure 11 - visualisation des individus dans le plan factoriel pour chaque Type

Le tableau 1 montre que pour les types Raw et Penups Raw le plan factoriel porte environ 33.6% et 41% de l'inertie expliquée respectivement. Les inerties portées pour les deux premiers sont proches (26% et 15% pour Penups et 20% et 13% pour Raw. Cependant, pour le type Pressure le plan factoriel porte environ 77% d'inertie avec 70% pour le premier axe et 7% pour le deuxième axe. Ce coude d'inertie va assurer d'avoir un nombre de features très réduit par rapport aux Raw et Penups.

Type d'image	% d'inertie portée par le plan factoriel
Raw	33,59
Penups Raw	41,03
Pressure	76,91

Tableau 1 - Pourcentage d'inertie expliquée par le plan factoriel

La figure 12 nous montre l'évolution de l'inertie intra-classe en fonction de nombre d'axes pour les types d'images. On voit bien que pour tous les types on arrive à bien sélectionner les features les plus discriminantes à partir de 25088 feats. Avec le type Pressure, on peut utiliser juste 23 features pour représenter 95% de l'inertie expliquée et cela peut être justifié par le fait qu'il y a plus de données locales dans les images avec trois niveaux de valeur. C'est le grand avantage des images dynamiques.

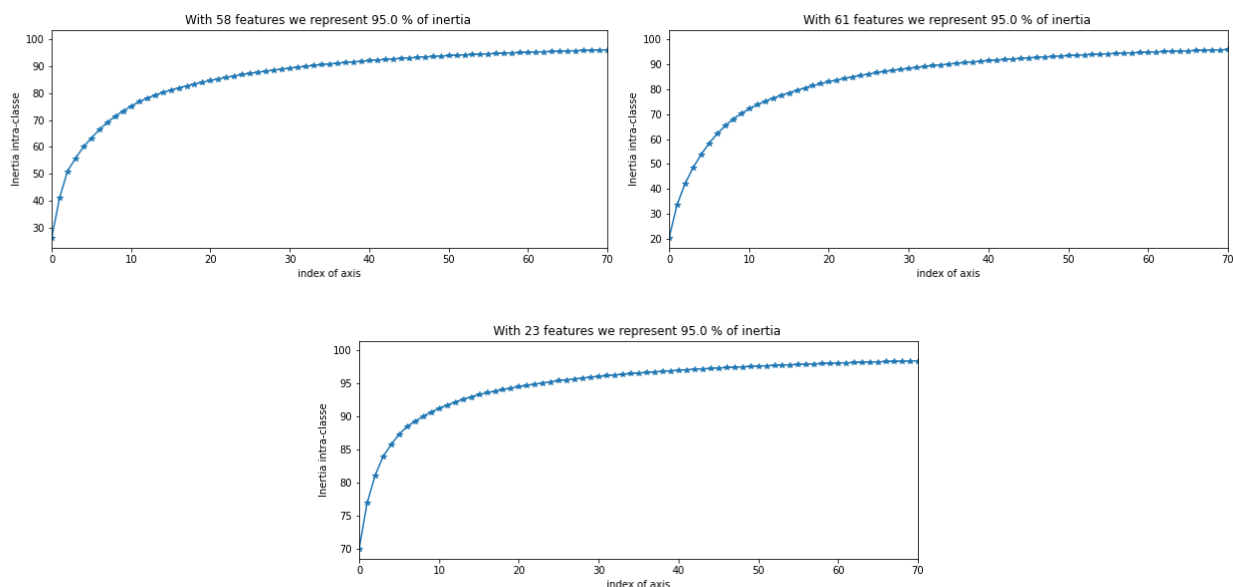


figure 12 - évolution de l'inertie intra-classe en fonction de nombre d'axe pour penups (a gauche), Raw (à droite) et pressure (au milieu)

Type d'image	nombre d'axes pour 95% d'info
Raw	61
Penups Raw	58
Pressure	23

Tableau 2 - Nombre d'axes nécessaire pour représenter 95% de l'information

On voit aussi que le fait d'ajouter le trajet du pointeur du stylo sur l'image brute (Raw) permet de gagner 3 features de plus. Ainsi, la matrice des données X_train et X_test sera dans la forme suivante :

Type d'image	Xtrain	Xtest
Raw	(600,61)	(300,61)
Penups Raw	(600,58)	(300,58)
Pressure	(600,23)	(300,23)

Tableau 3 - Dimensions de l'ensemble de données X_train et X_test

Classification des features

Dans cette partie, on va classer les features calculées après la décomposition ACP. Le but est de comparer les performances de trois modèles de classification ensuite voir la qualité des features du réseau VGG 16. Vu qu'il s'agit d'une classification binaire, on peut utiliser les métriques suivantes pour la comparaison :

- **Accuracy** : le score qui mesure la capacité du réseau à classer correctement les individus (AD ou control)

$$accuracy = \frac{tp+fn}{tp+fn+tn+fp}$$

Avec tp (True positive), fn (False negative), tn (True negative), fp (False positive)

- **Sensitivité** : le score qui indique la probabilité d'un individu d'appartenir à la classe sachant qu'il devrait y appartenir.

$$Sensibilité = \frac{tp}{tp+fn}$$

- **Spécificité** : le score qui indique la probabilité qu'un individu n'appartienne pas à la classe à juste titre.

$$Spécificité = \frac{tn}{tn+fp}$$

- **AUC** : il signifie "aire sous la courbe ROC". Ce score indique la probabilité pour que le modèle classe un exemple positif aléatoire au-dessus d'un exemple négatif aléatoire. Les valeurs d'AUC sont comprises dans une plage de 0 à 1. Un modèle dont 100 % des prédictions sont erronées a un AUC de 0,0. Si toutes ses prédictions sont correctes, son

AUC est de 1,0.

Pour notre étude, on va utiliser les classifieurs SVM (Support Vector Machine), MLP (Multiple Layer Perceptron) et KNN (K Nearest Neighbors). Pour l'ensemble des expériences, on détermine les paramètres de l'estimateur optimal qui maximise l'AUC à 10-fold validation croisée. La valeur des métriques est celle de l'ensemble de Test pour mieux les performances des modèles.

Comparaison des classifieurs

Pour chaque type d'image, on détermine les paramètres du meilleur estimateur pour chaque modèle (SVM, MLP et KNN) puis on calcule les métriques sur l'ensemble de tests.

- Type : Raw

On constate que le SVM est le plus performant par rapport au MLP et KNN même si l'écart est quasi faible mais il reste le plus spécifié avec un score de 0.78. Les résultats sont plutôt similaires pour les trois classifieurs.

	SVM	MLP	KNN
Accuracy	0.77 \pm 0.13	0.75 \pm 0.11	0.74 \pm 0.14
sensibilité	0.77 \pm 0.15	0.76 \pm 0.12	0.75 \pm 0.22
spécificité	0.78 \pm 0.15	0.74 \pm 0.16	0.74 \pm 0.15
AUC	0.85 \pm 0.09	0.84 \pm 0.11	0.85 \pm 0.13
paramètres	C: 100.0, gamma: 0.01, kernel: 'rbf'	'alpha': 0.1, 'hidden_layer': 14, 'max_iter': 15000, 'solver': 'lbfgs'	'algorithm': 'ball_tree', 'n_neighbors': 3, 'weights': 'distance'

- Type : Penups Raw

Pour le type Penups Raw, on trouve que le SVM est l'estimateur le plus performant avec un accuracy de 84% et un AUC 92%. Le KNN est le modèle le plus spécifié avec un score de 88% alors le SVM est le modèle le plus sensible avec un score de 86%. On remarque qu'il n'y a pas d'une grande différence entre les meilleurs estimateurs de chaque modèle au niveau de l'accuracy et l'AUC.

	SVM	MLP	KNN
Accuracy	0.84 \pm 0.08	0.82 \pm 0.07	0.81 \pm 0.13
sensibilité	0.86 \pm 0.12	0.83 \pm 0.09	0.74 \pm 0.25
spécificité	0.82 \pm 0.15	0.82 \pm 0.11	0.88 \pm 0.10
AUC	0.92 \pm 0.07	0.90 \pm 0.07	0.92 \pm 0.09
paramètres	C: 100.0, gamma: 0.01, kernel: 'rbf'	'alpha': 0.1, 'hidden_layer': 12, 'max_iter': 5000, 'solver': 'lbfgs'	'algorithm': 'ball_tree', 'n_neighbors': 3, 'weights': 'distance'

- Type : Pressure

Pour le type Pressure, on constate que le KNN est le classifieur le plus performant avec un accuracy de 91% et un AUC de 97%. Le modèle le plus sensible est le SVM avec un score de 94% tandis que le modèle le plus spécifié est KNN avec un score de 93%.

	SVM	MLP	KNN
Accuracy	0.89 \pm 0.06	0.89 \pm 0.11	0.91 \pm 0.05
sensibilité	0.94 \pm 0.13	0.89 \pm 0.07	0.89 \pm 0.07
spécificité	0.85 \pm 0.09	0.88 \pm 0.17	0.93 \pm 0.09
AUC	0.97 \pm 0.04	0.94 \pm 0.08	0.97 \pm 0.05
paramètres	C: 10.0, gamma: 0.01, kernel: 'rbf'	'alpha': 0.1, 'hidden_layer': 10, 'max_iter': 5000, 'solver': 'lbfgs'	'algorithm': 'ball_tree', 'n_neighbors': 3, 'weights': 'distance'

Globalement, les meilleurs estimateurs des modèles de classification atteignent quasiment les mêmes performances pour tous les types d'images grâce à l'extracteur des features du réseau VGG 16 et la décomposition ACP. On remarque aussi que le SVM est performant pour les features statiques (Raw et Penups Raw) alors que le KNN est plutôt bon pour les features dynamiques (Pressure).

Comparaison des types d'image

Dans cette partie, on veut comparer les performances des meilleurs estimateurs en fonction du type d'image. On remarque que le fait d'ajouter le trajet du levé de pointeur améliore quasiment les performances du modèle. En d'autre part, on trouve que les performances du classifieur sur les features pressure sont très optimales ce qui explique la bonne qualité de ces features. D'ailleurs, l'utilisation des images dynamiques est très utile pour la prédiction des symptômes d'Alzheimer car elles fournissent des informations locales sur les caractéristiques dynamiques du pointeur du stylo ce qu'on ne peut pas y avoir avec une image statique.

	Raw	Penups Raw	Pressure
Accuracy	0.77 ± 0.13	0.84 ± 0.09	0.91 ± 0.05
sensibilité	0.77 ± 0.15	0.86 ± 0.12	0.89 ± 0.07
spécificité	0.78 ± 0.15	0.82 ± 0.15	0.93 ± 0.09
AUC	0.85 ± 0.09	0.92 ± 0.07	0.97 ± 0.05
classifieur paramètres	SVM C: 100.0, gamma: 0.01, kernel: 'rbf'	SVM C: 100.0, gamma: 0.01, kernel: 'rbf'	KNN 'algorithm': 'ball_tree', 'n_neighbors': 3, 'weights': 'distance'

Conclusion

La prédiction des symptômes d'une maladie comme l'Alzheimer peut être efficace lors de l'utilisation d'un modèle statistique en passant par l'extraction des features par un transfert learning d'une architecture CNN profond comme VGG 16 puis la sélection des features discriminants en utilisant une décomposition ACP. Vu le manque des données sur les patients d'alzheimer, on peut appliquer des méthodes d'augmentation des données de telle façon qu'on aurait des images non biaisées.

La comparaison des classifieurs permet de montrer la qualité des features sélectionnées, on a vu que pour les images statiques, les performances restent acceptables surtout pour le type Penups ou on constate une bonne amélioration des scores du meilleur estimateur. Avec les features dynamiques comme Pressure, les résultats de la classification sont très optimales.