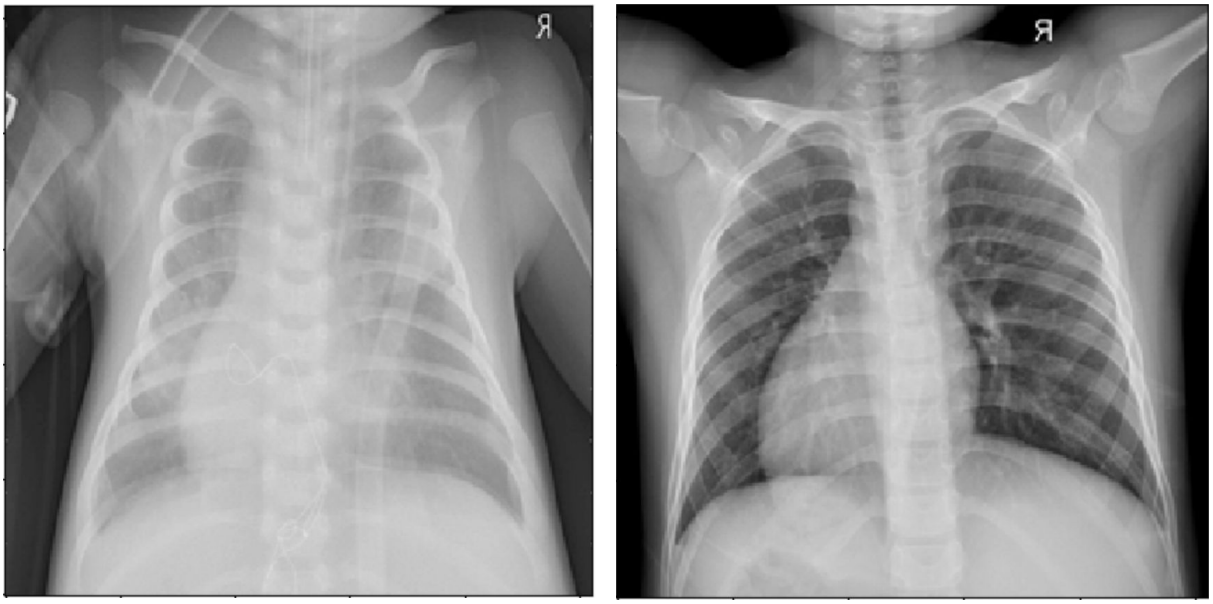
Maîtrise en Informatique

Cours IFT712

Techniques d'Apprentissage



"Classification binaire d'images médicales"

ADOLPHE Maxime

JOUFFROY Emma

MOUGIN Cyril

SOMMAIRE:

1. Introduction et description du projet

Dans le cadre du cours IFT-712, “techniques d’apprentissages”, il a été proposé aux étudiants d’appliquer l’ensemble des stratégies proposées pendant la formation à une problématique concrète. Notre groupe a décidé de travailler sur un problème de classification d’images médicales, à partir d’un jeu de données de radiologies du thorax, il s’agissait de déterminer quels patients été atteint d’une pneumonie. De façon assez évidente, proposer des stratégies automatiques de détection de ce type de pathologies pourraient être d’une grande aide aux praticiens de la santé. Cependant, comme nous avons pu le découvrir, de nombreuses difficultés sont à surmonter : traitement d’objets de grandes dimensions (des images), résolution des clichés (bruit lié à la modalité d’imagerie) ou encore nécessité d’être expert pour définir qu’une image est belle et bien atteinte de la pathologie.

2. Gestion du projet

Avant de rentrer plus en détail sur le travail réalisé, la section suivante décrit l’ensemble de la gestion de projet effectué.

a. Organisation du travail

L’ensemble des tâches effectuées a été regroupé dans le planning figure 1. Le projet s’est découpé autour de trois grandes étapes importantes : choix du jeu de donnée et traitement des images, déploiement des solutions, rédaction du rapport et du notebook pour la présentation des résultats.

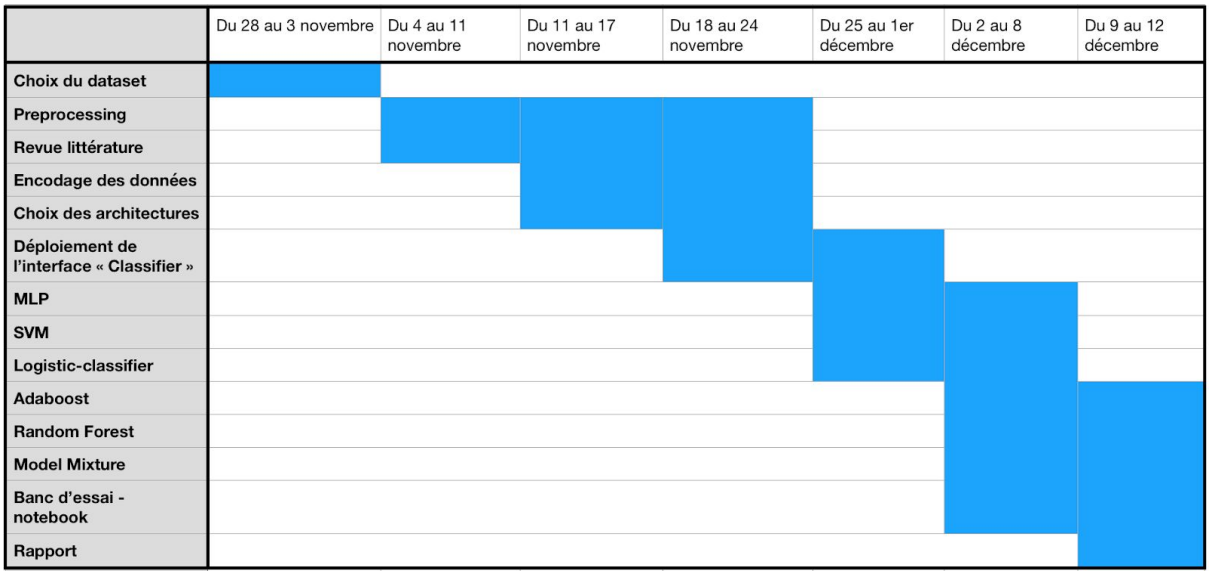


Figure 1 : Planning du projet

b. Outils utilisés

Afin d'assurer une bonne gestion de projet, trois outils principaux ont été utilisés. Google drive a permis un partage des données non liées au code (notes, bibliographies, etc). Un tableau Trello que l'on peut trouver à l'adresse [1] a été utilisé afin que chaque acteur réalise des tâches différentes (et que chacun soit au courant de l'avancé de ses partenaires). Finalement git et github ont été utilisé respectivement pour le versionnage et le partage du code. Comme vous pourrez le voir sur l'outil de visualisation des graphs du repository github [2], le workflow utilisé a été relativement simple mais efficace: le travail réalisé en continu possédant des parties fonctionnelles se trouve sur la branche 'develop', le travail en cours de réalisation se trouvait sur les branches 'features' (portant le nom des modules/solutions développées).

3. Description du jeu de données

a. Informations globales

Le jeu de donnée utilisé pour la mise en place et le test des solutions de classification a été trouvé sur kaggle [3]. Il s'agit de deux types d'images (récoltées à l'aide d'un scanner): des images de poitrine saines et des images de poitrines atteintes d'une pneumonie. Le corpus original contient 3875 images malades et 234 images normales pour l'entraînement, 390 images malades et 234 saines pour le test et finalement 234 saines et 8 malades pour la validation. Comme vu plus tard dans ce rapport, n'ayant pas besoin d'un jeu de validation segmenté du jeu d'entraînement, nous avons regroupé jeu d'entraînement et jeu de validation. Une remarque émise à l'observation de la répartition dans nos deux classes est qu'elles sont fortement débalancées (presque 10 fois plus d'images malades dans

[1] https://trello.com/b/dog6HORT/projet-tech-dapp
[2] https://github.com/madolphe/ProjetTechApp/network
[3] https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

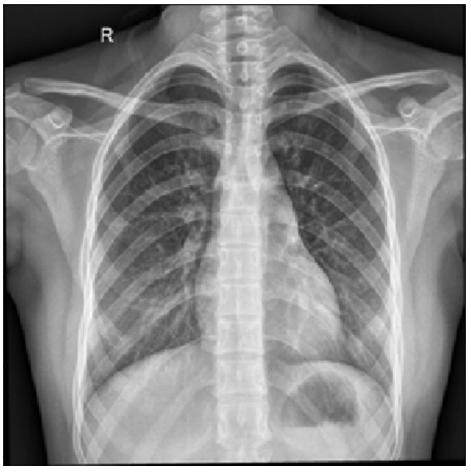
le jeu d'entraînement). Nous prendrons cette remarque en considération au moment d'entraîner nos solutions.

b. Prétraitement

Après déploiement des méthodes d'accès aux corpus, le prétraitement a adressé deux problématiques principales inhérentes au jeu de données.

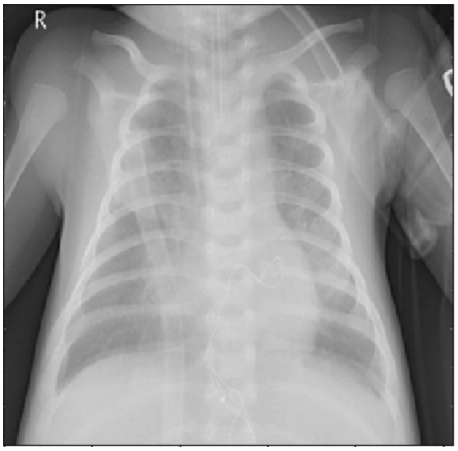
i. Normalisation des images

D'abord, il a fallu effectué une normalisation des images du corpus. En effet, toutes les images n'étaient pas à la même dimension ou dans les mêmes plages d'intensité. Il a donc été choisi de les transformer dans une taille arbitrairement sélectionnée à 256 par 256 pixels (bon compromis entre résolution / perte d'information) à l'aide d'une interpolation de Lanczos. Ensuite, l'observation du dataset nous a permis de nous rendre compte de la basse qualité des images. Il a alors été décidé d'égaliser les histogrammes afin d'obtenir un meilleur contraste. Cette démarche nous semblait cohérente, cependant nous pensons qu'il aurait été judicieux de faire valider les images transformées auprès de professionnels (les modifications apportées pourraient entraîner une perte d'informations que nos yeux non initiés ne pourraient détecter).

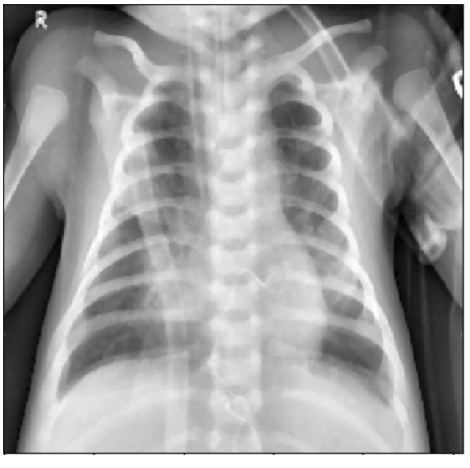


(a)

(b)



(c)



(d)

Figure 2: (a) image saine sans égalisation, (b) image saine avec égalisation, (c) image malade sans égalisation, (d) image malade avec égalisation

ii. Réduction de dimensions

Après que toutes les images "subissent" le prétraitement précédent, nous nous sommes intéressés à la manière dont nous allions traiter ces objets de hautes dimensions. En effet, si nous choisissons de considérer les images entières, les individus de nos corpus auraient été des objets de dimensions 65 536. Nous avons donc mis en place une extraction de caractéristiques afin de diminuer la dimensionnalité tout en conservant le maximum d'information. Pour cette partie du travail, nous avons étudié la littérature et notamment l'article de Yang et al, de 2018 [1].

1. Caractéristiques de texture

Pour commencer nous avons calculé la matrice de co-occurence des niveaux de gris ('GLCM' en anglais). Il s'agit d'évaluer le nombre de pixels adjacents à une distance et une direction donnée pour les toutes les intensités présentes dans l'image. Pour exemple, on peut considérer l'image suivante (où chaque case est un pixel et le numéro de cellule son intensité):

1	1	2
2	1	3
1	2	1

Table 1: Image d'exemple

La matrice de co-occurence des niveaux de gris pour la distance 1 et la direction 0 associée est:

	1	2	3
1	1	2	1
2	2	0	0
3	0	0	0

Table 2: Matrice de co-occurence des niveaux de gris pour la distance 1 et la direction 0 degré.

Cette matrice est ensuite utilisée pour calculer facilement des métriques de texture dans nos images. Nous avons donc pu définir :

- Le contraste :

$$\sum_{i,j=0}^{niveau-1} P_{i,j}(i-j)^2$$

- La corrélation:

$$\sum_{i,j=0}^{niveau-1} P_{i,j}\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{\sigma_i^2\sigma_j^2}}$$

- L'homogénéité:

$$\sum_{i,j=0}^{niveau-1} \frac{P_{i,j}}{1+(i-j)^2}$$

- L'énergie:

$$\sqrt{\sum_{i,j=0}^{niveau-1} P_{i,j}^2}$$

Ainsi pour une direction est une distance, une GLCM nous donne quatre caractéristiques. Dans notre étude, nous avons utilisé la distance 1 et les orientations 0, 45, 90 et 180 degrés. Finalement, nos images sont donc toutes représentées par un vecteur de 16 caractéristiques de texture.

2. Caractéristiques fréquentielles

Pour récupérer plus d'informations dans nos images, nous nous sommes ensuite intéressés aux caractéristiques fréquentielles. Notre première idée fut d'utiliser la transformée de Fourier. Cependant nous n'étions pas convaincu par l'efficacité de cette méthode en raison de l'absence de localisation en espace de cette transformée. Nous nous sommes donc plutôt intéressés à la transformée en ondelette [2][3]: parfait compromis entre localisation en espace et en fréquence (bien que merveilleuse, nous ne reviendrons pas sur la théorie de cette décomposition dans ce rapport). Notre démarche a été de décomposer en plusieurs niveaux notre image à l'aide d'ondelettes de haute et de basse fréquence. La figure suivante présente ces différentes transformations (notez que l'image voit sa taille diminuée par 2 à chaque transformation et que pour des raisons de présentation seulement 3 niveaux sont affichées).

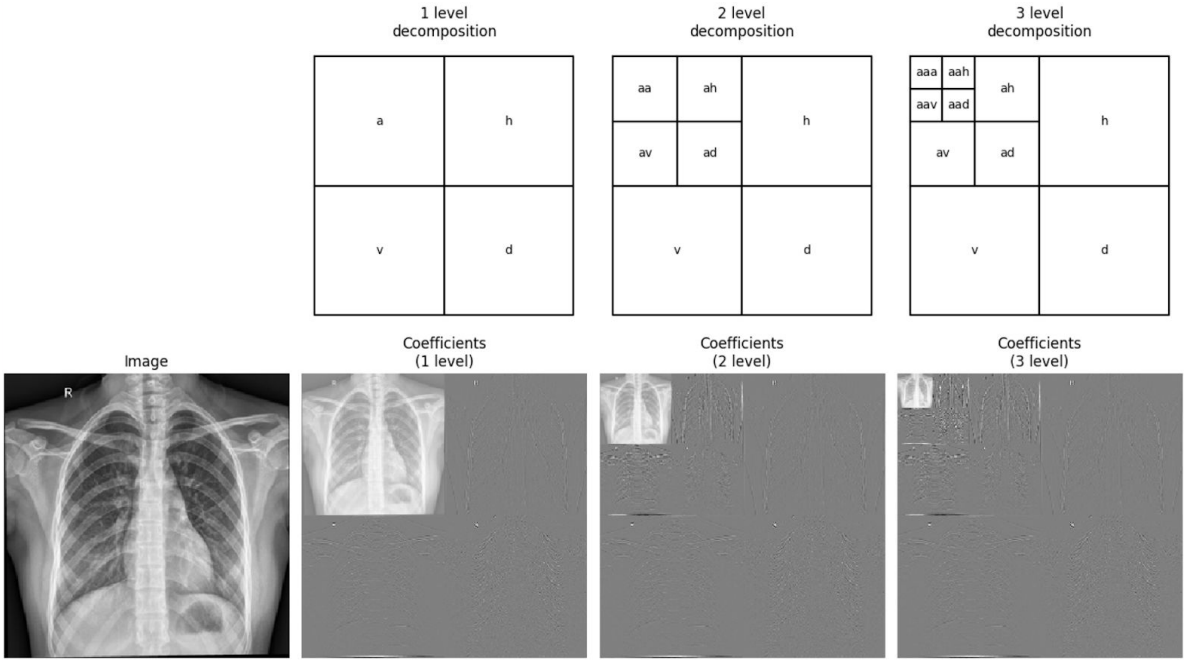


Figure 3: Décomposition en ondelettes db2 pour 3 niveaux

Pour chacune des décompositions, on applique: une ondelette de basse fréquence (ex: a dans le niveau 1) et 3 ondelettes de haute fréquence (ex: horizontal h, vertical v et diagonal d dans le niveau 1). Se référant à la littérature, nous avons ensuite pris la moyenne et la variance des images obtenues. Nous n'avons utilisé que les décompositions hautes-fréquences des 3 premiers niveaux et la décomposition basse fréquence du dernier niveau. Ainsi, nous avons obtenu un vecteur de caractéristiques fréquentielles de 20 dimensions.

c. Choix des caractéristiques

A l'issue de la phase d'extraction des caractéristiques, nous nous sommes retrouvés avec des vecteurs à 36 dimensions représentant nos images. Avant de passer à l'étape finale de classification, nous voulions nous assurer de la pertinence des choix d'extraction. Nous avons donc mis en place une double stratégie de sélection de caractéristiques. Tout d'abord, nous avons calculé la variance des données d'entraînement sur chacune des caractéristiques. Ensuite nous avons calculé le score du Chi carré des données d'entraînement sur chacune des caractéristiques. Comme le montre le tableau suivant, en recoupant les deux indices, nous avons éliminé les caractéristiques avec les scores les plus bas. Finalement, nous avons conservé 30 caractéristiques pour la phase de classification.

Variance inférieure à 0.001:	Score Chi-2 inférieure à 1:
mean2HF_H 0.003197	mean3HF_H 0.000055
mean1HF_V 0.003621	mean3HF_D 0.000711
mean1HF_H 0.003965	mean1HF_D 0.001371
mean3HF_H 0.004098	mean2HF_V 0.001837
mean2HF_V 0.005265	correlation_0 0.017736
mean3HF_D 0.009617	mean2HF_D 0.057077
correlation_45 0.009621	mean2HF_H 0.099090
std1HF_V 0.009724	mean1HF_V 0.228127
correlation_135 0.009780	mean1HF_H 0.758234

Table 3: Caractéristiques avec les scores les plus bas en variance et au test du Chi-2 (en gris les caractéristiques supprimées)

4. Structure du projet

a. Environnement virtuel

Afin de faciliter l'utilisation des différentes bibliothèques utilisées, il a été décidé de travailler avec le gestionnaire d'environnement virtuel : pipenv. Simple d'utilisation, cet utilitaire a permis un partage serein des pré-requis pour l'utilisation du projet.

b. Architecture

La structure du projet (figure 4) est relativement simple: l'ensemble des données sont regroupées dans le dossier data. Ce dossier est lui même subdivisé avec nos images stockées dans "chest_xray" et nos fichiers csv dans "chest_xray_feature_vect". Les images

ne sont plus utilisées pour la classification. Tout le traitement du corpus et le chargement du jeu de données dans un dataframe s'effectue dans le dossier "get_data".
L'ensemble des solutions sont regroupées dans le dossier models. La classe Classifier est un modèle parent dont toutes nos solutions vont hériter. À la manière d'une interface, elle implémente certaines méthodes que nos solutions devront impérativement surcharger. Elle permet aussi de proposer des méthodes communes à toutes nos classes (que nous n'auront pas à dupliquer!) telles que la cross-validation, le calcul d'une matrice de confusion, l'affichage de courbes de performance ou encore l'affichage des paramètres du modèle lors de l'appel à la fonction print().
Pour finir, un notebook "banc d'essai" fait un lien entre nos différentes solutions et les teste en utilisant les mêmes métriques / conditions pour toutes nos méthodes.

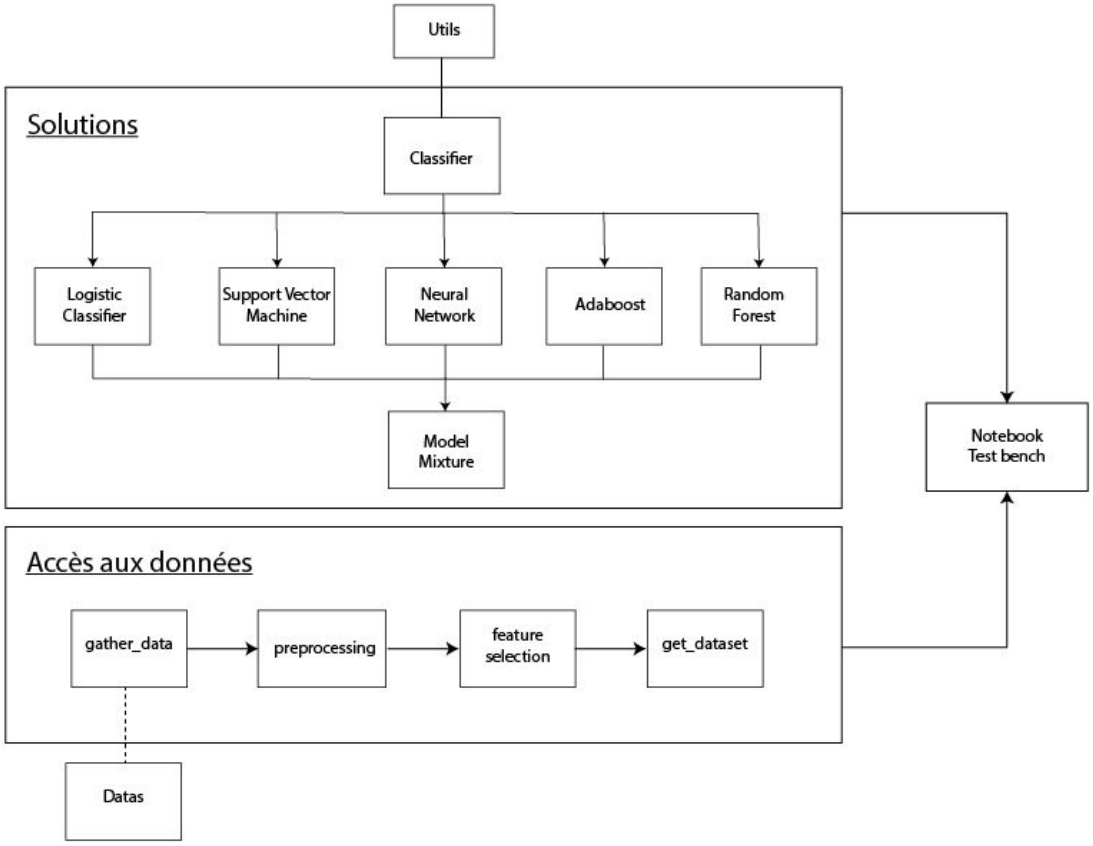


Figure 4: Architecture du projet

5. Résultats

Pour l'ensemble des résultats, le lecteur pourra se référer au notebook intitulé "test-bench.ipynb".

6. Conclusion

a. Analyse des résultats

Parmi toutes les techniques implémentées, le meilleur résultat obtenu est celui de la random Forest (près de 76% de justesse sur le jeu de test et courbes ROC et précision-rappel de bonne “qualité”). De très près, le multi-layer perceptron obtient aussi de bons résultats (presque 75%). Comme attendu le classifieur linéaire obtient les moins bons résultats (70%). Pour finir, nous pensions que la combinaison de modèle aurait été la plus performante. Cependant même avec l’utilisation de nos modèles avec les meilleurs paramètres, les résultats se positionnent dans la moyenne du reste des solutions.

b. Bilan

Les résultats sont peu satisfaisants, ils présentent une trop grosse marge d’erreur pour une éventuelle utilisation dans un cadre réel de santé. Cela peut s’expliquer par plusieurs facteurs: capacité de nos modèles, nombre de données et choix dans l’extraction des caractéristiques. En effet, toutes nos stratégies d’apprentissage ne sont pas du “end-to-end training” et reposent sur une extraction de caractéristiques “à la main”. Nous pensons qu’il serait intéressant d’ouvrir ce travail à des méthodes impliquant de l’apprentissage dans le choix des features intéressantes (tels que les réseaux de neurones convolutifs par exemple).

7. Bibliographie

[1] - Fang Yang, Murat Hamit, Chuan B. Yan, Juan Yao, Abdugheni Kutluk, Xi M. Kong, and Sui X. Zhang - Feature Extraction and Classification on Esophageal X-Ray Images of Xinjiang Kazak Nationality

[2] - Gilles Chardon (2018) - Bases d’ondelettes

[3] - Guan-Chen Pan - A Tutorial of Wavelet for Pattern Recognition - r99942126@ntu.edu.tw Graduate Institute of Communication Engineering National Taiwan University, Taipei, Taiwan, ROC

[4] - T. J. Penfold, I. Travernelli, C. J. Milne et al., “A wavelet analysis for the X-ray absorption spectra of molecules,” Journal of Chemical Physics, vol. 138, no. 1, p. 014104, 2013.