

Document explicatif du contenu du code

Principe des algorithmes

1. Pré-traitement des images (*image_preprocessing.ipynb*)

Pour que les étapes suivantes fonctionnent telles que nous le souhaitons, une étape de pré-traitement était nécessaire. Les images qui nous ont été fournies avaient une gamme dynamique allant de -2000 à 3000. Après une observation permettant de déterminer quelles zones de l'histogramme correspondaient à quelles régions de l'image, il a été décidé de se concentrer sur les valeurs allant de 0 à 1000 pour mettre en valeur les vaisseaux sanguins. Nous avons alors réalisé une transformation linéaire avec saturation permettant de sélectionner les valeurs de cet intervalle et de les ramener à une gamme dynamique allant de 0 à 255.

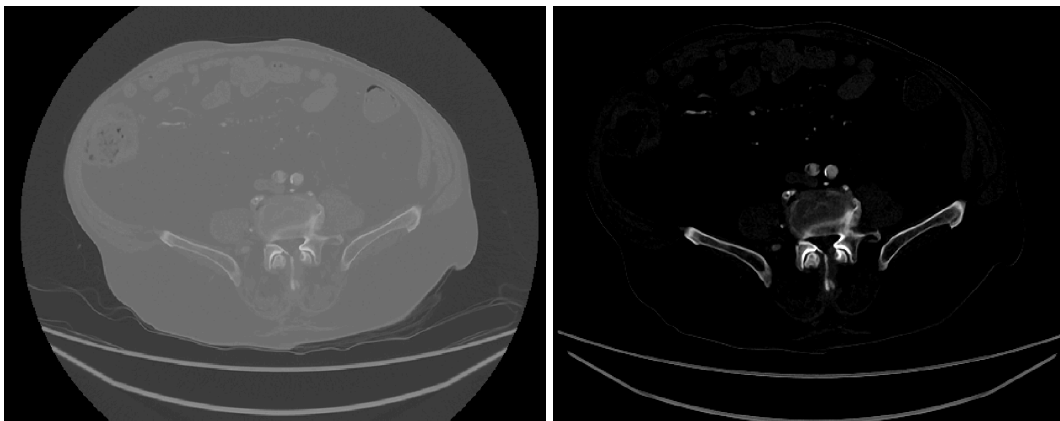


Figure 1 : Coupe 1240 du patient 1 avant (à gauche) et après (à droite) pré-traitement.

2. Construction du réseau d'artères (*network_construction.ipynb*)

La première étape de la construction du réseau artériel consiste à déterminer l'emplacement de l'aorte abdominale sur la première coupe de l'angioscanner du patient. Pour cela, un seuil d'Otsu est calculé sur cette coupe afin de pouvoir créer une image binaire. Une segmentation est ensuite réalisée en utilisant les composantes connectées, et les objets considérés comme trop petits sont enlevés. Finalement, parmi les régions segmentées restantes, celle avec la plus faible excentricité, c'est-à-dire la forme se rapprochant le plus d'un cercle, est sélectionnée car elle correspond à l'aorte abdominale.

Le code est articulé autour de la classe `NetworkEngine` créée pour ce projet qui permet de manipuler l'image et créer son réseau. On utilise une méthode de région growing pour récupérer la morphologie complète de l'artère à un étage précis puis on descend sur l'étage suivant en cherchant un bloc géographiquement proche avec une intensité lumineuse cohérente et on répète ce schéma.

Afin de gérer les séparations, on traque les rétrécissements soudain de diamètre de l'artère étant très généralement lié à une bifurcation. On teste l'hypothèse de la bifurcation, si on

aboutit à quelque chose de cohérent, alors on garde la nouvelle branche et on continue l'exploration.

Afin d'obtenir de meilleurs résultats (dû à la densité des artefacts et du réseau au niveau des hanches, on perd parfois des éléments important) on effectue deux autres explorations. En partant de l'hypothèse qu'il est facile de segmenter le genou, on segmente celui ci afin d'obtenir l'emplacement de l'artère poplitée, facilement identifiable derrière le genou. À partir de ce nouveau point, on descend jusqu'au pied de nouveau, mais on tente également de remonter vers les hanches.

Enfin on nettoie le graphe en fusionnant les branches similaires (notamment du à la montée et à la descente) et on segmente le réseau de tel façon que une branche parente donne deux nouvelles branches enfants : chaque portion de vaisseau est une branche différente.

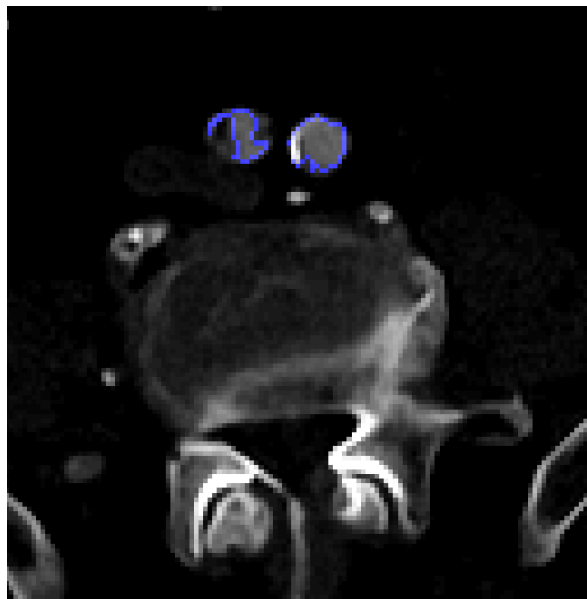


Figure 2 : Coupe 1240 du patient 1 avec les contours issus du réseau d'artères.

3. Construction du réseau d'ellipses et détection de calcifications et d'occlusions (*anomaly_detection.ipynb*)

Pour cette partie, tout est traité de manière à travailler indépendamment sur chaque masque extrait, que ce soit la construction et l'adaptation des paramètres des ellipses ou l'analyse réalisée ensuite.

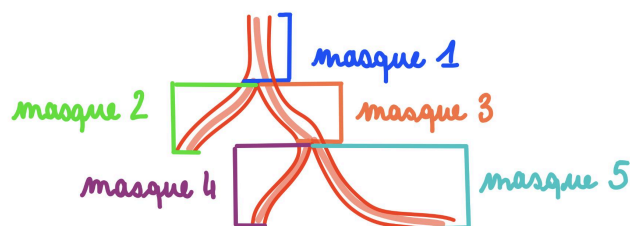


Figure 3 : Schéma illustrant la création des masques et leur numérotation

Construction des ellipses

En partant des masques du réseau segmenté et des contours présents sur chaque tranche, l'idée est de construire un réseau d'ellipses qui s'apparenterait au réseau d'artères réelles. En utilisant les géométries de chaque contour, une première approximation des paramètres de chaque ellipse est faite de manière individuelle.

Puis, de manière à éviter les ellipses qui dévient trop de l'axe principal et à lisser le réseau, une régression linéaire du produit $a \times b = \text{grand axe}_{\text{ellipse}} \times \text{petit axe}_{\text{ellipse}}$ est faite.

Toute ellipse s'éloignant de celle-ci est alors modifiée en reprenant des paramètres proches de celle précédente, en respectant le sens de parcours du masque. Ensuite, pour être sûr que tous les pixels d'intérêts sont pris en compte, une petite dilatation de chaque ellipse est réalisée. La sortie est alors un réseau d'ellipses.

Analyses statistiques

Pour réaliser l'analyse statistique, on récupère la liste des pixels de l'image à l'intérieur de chaque ellipse, qui correspondent donc aux pixels des artères. La proportion de pixels blancs (supérieurs à 250) est calculée pour chacune des coupes et correspond au pourcentage de calcification. Pour les occlusions, un Z score est calculé pour chaque coupe selon la formule suivante :

$$Z = \frac{\mu_{\text{coupe}} - \mu_{\text{masque}}}{\sqrt{\sigma_{\text{coupe}} + \sigma_{\text{masque}}}}$$

Ce score permet de quantifier la différence entre deux distributions. Dans notre cas, on compare la distribution des valeurs de pixels de la coupe à celle de la moyenne des distributions de tout le masque. Les occlusions sont alors détectées par une valeur de Z score faible. Les pixels considérés comme des calcifications ne sont pas pris en compte lors de ce calcul.

Finalement, une heatmap peut être réalisée pour représenter la valeur de ces statistiques (pourcentage de calcification ou Z score) sur les images, avec l'utilisation d'un code couleur facilitant la visualisation. Le vert correspond à une coupe sans occlusion, tandis que le rouge correspond à une coupe avec une occlusion importante.

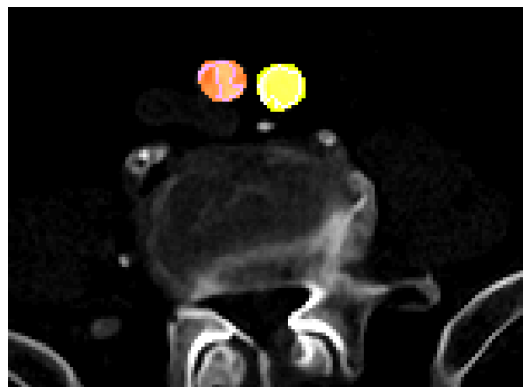


Figure 4 : Coupe 1240 du patient 1 avec les heatmap pour visualiser les occlusions.

Quelques tips pour réutiliser le code

A titre d'information, tout ce travail a été réalisé en utilisant Python 3.11.5.

Pour utiliser les notebooks sur un patient précis, il est nécessaire que les fichiers .mhd et .zraw correspondant à l'image soient placés dans le dossier /data.

Par la suite, chaque notebook utilise des entrées et des sorties spécifiques qui sont sauvegardées dans le dossier /temp, cela permet de pouvoir exécuter et garder en mémoire les résultats de chaque étape. Le diagramme de la figure 5 illustre quel fichier est utilisé à quel moment.

Figure 5 : Diagramme illustrant les entrées et sorties de chaque notebook.

