

Segmentation automatique des plaques athéromateuses en OCT.

MELLAK Youness,

Université Clermont Auvergne, Master 2 TECHMED.

Encadré par: Pr. Laurent SARRY - Professeur des Universités

Abstract—La tomographie par cohérence optique intravasculaire (IV-OCT) est une modalité d'imagerie in-vivo basée sur l'introduction d'un cathéter intravasculaire permettant de visualiser la paroi interne des vaisseaux sanguins avec une résolution spatiale de 10 μm à 20 μm . L'identification précise de la plaque coronaire est très importante pour les cardiologues lorsqu'ils traitent des patients avec des atteints d'athérosclérose avancée. Nous avons créé une simulation basique des images OCT contenant des plaques, ainsi on a développé une méthode de segmentation sémantique entièrement automatisée de ces plaques à partir des images OCT intravasculaires.

Index Terms—IV-OCT - Simulation des images IV-OCT - Segmentation sémantique - Apprentissage Profond

I. INTRODUCTION

La maladie coronarienne est une cause connue de morbidité et de mortalité pour les deux sexes dans les pays développés, l'imagerie est donc nécessaire pour planifier et évaluer les traitements. L'athérosclérose entraîne différents types de lésions (fibrotiques, lipidiques et calcifiées). La radiographie non invasive; la tomodensitométrie ou l'imagerie par résonance magnétique avec des agents de contraste des artères coronaires montre principalement la lumière, ce qui ne permet qu'une identification limitée des tissus de la paroi artérielle. Les principales méthodes d'imagerie intracoronaire sont l'échographie intravasculaire (IVUS) et la tomographie par cohérence optique intravasculaire (IVOCT). Bien que l'IVUS puisse fournir des informations significatives sur la composition de la plaque, elle présente deux limites importantes. Premièrement, la technique a une résolution axiale limitée de 150 à 250 μm et une résolution latérale de 150 à 300 μm , ce qui est incapable de détecter de minces coiffes fibreuses (65 μm). Deuxièmement, la technique empêche également la détermination de la profondeur de calcification en raison de l'ombre acoustique.

L'un des défis de la planification du traitement en temps réel est que chaque retraite IVOCT comprend +500 images, ce qui est une surcharge de données d'image. L'analyse manuelle de chaque base de retrait demande beaucoup de travail et de temps et souffre d'une forte variabilité inter et intra observateur. L'année précédente un groupe a travaillé sur la segmentation, semi-automatique (par graph-cut) des différentes couches de l'artère, qui a donné des résultats satisfaisants sur des images de scanner et des images d'IRM. Par contre sur des images d'OCT l'algorithme n'a pas arrivé à segmenter les différentes couches des artères coronaires du fait de la présence du

speckle, en outre, l'algorithme nécessite toujours un expert qui doit labéliser des régions différentes avant que l'algorithme segmente la région d'intérêt, ce qui reste un travail à faire et une tâche qui consomme du temps. Cela indique la nécessité d'une méthode de caractérisation de plaque rapide et entièrement automatisée, et comme de nombreuses techniques de vision d'ordinateur inventées au cours des 20 dernières années sont devenues inutiles ces dernières années à cause du deep learning [6], nous allons essayer d'implémenter différents modèles afin de segmenter nos plaques.

II. SIMULATION DES IMAGES OCT DES ATHÉROMATEUSES:

Dans la littérature de l'apprentissage profond on trouve que la performance d'un algorithme DL augmente avec l'augmentation des données d'apprentissage. Alors nous dans le cas d'OCT, l'accès à une base de données géante en matière de nombre d'images n'est pas possible, vu que la technique est nouvelle, et aussi le nombre d'examen d'OCT intravasculaire n'est pas grand en France. Pour cela nous avons pensé à faire une simulation de ces images, essayant de respecter le maximum possible les caractéristiques de chaque tissu.

A. Image de la lumière d'artère:

Nous avons considéré l'artère comme un cercle (ce qui est n'est pas le cas dans la réalité), et la sonde est un cercle et son origine est le centre de l'image, l'origine de l'artère et de la sonde ne sont pas forcément confondus (Figure 1 et 2).

L'acquisition réelle de l'OCT elle n'est pas faite dans le repère cartésien, mais plutôt dans le repère polaire. (Figure 3), et on a :

$$\rho = \sqrt{x^2 + y^2} \quad (1)$$

$$\theta = \arctan\left(\frac{y}{x}\right) \quad (2)$$

L'image de la figure 4 Montre deux images OCT réels dans le domaine polaire et le domaine cartésien.

Nous avons essayé de simuler cette caractéristique, en passant du cartésien vers la polaire utilisant l'algorithme de Bresenham (vue dans le premier TP de l'imagerie médicale). Comme déjà cité auparavant, notre artère est simulé par un cercle, qui n'est pas forcément centré dans l'image. Le centre de notre image (le pixel de coordonnées 128,128) représente la sonde d'acquisition.

Le signal dans l'OCT est le résultat de la réflexion de la

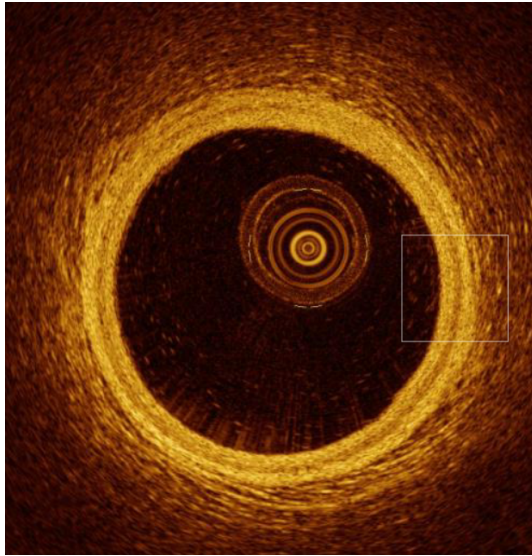


Figure 1. Image réelle d'un artère avec OCT

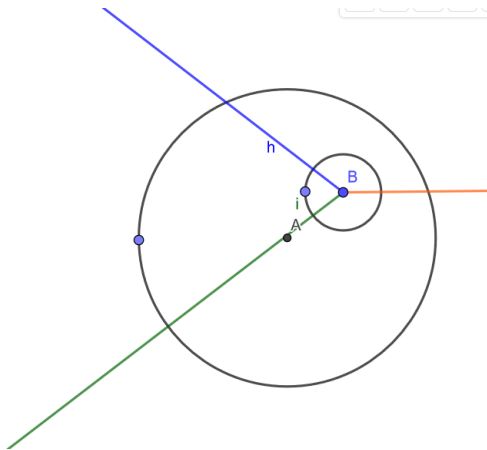


Figure 2. Représentation de la sonde et de l'artère

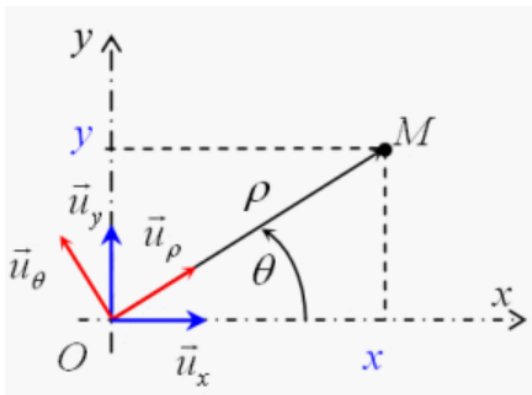


Figure 3. Les coordonnées polaires ρ et θ

lumière, après interaction avec le tissu. L'acquisition est faite sur des lignes de tir de la sonde avec un angle donné. Puisque le centre de la sonde et le vaisseau ne sont pas confondus,

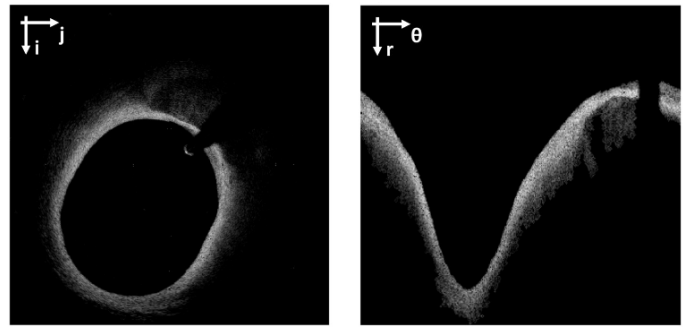


Figure 4. Image OCT dans le domaine cartésien || Image OCT dans le domaine polaire.

donc la distance entre la sonde et la lumière de l'artère va différer d'une position à l'autre.

Dans la figure 2 nous avons essayé de schématiser ce qu'on a cité auparavant. Les lignes (Orange, bleu, vert) représentent les lignes d'acquisition dans des angles différents, le petit cercle notre sonde et le grand cercle la lumière de notre artère.

Nous remarquons que pour la ligne orange la distance entre la sonde et la lumière est faible donc r sera faible. Par contre la ligne verte à un angle θ_3 nous avons une distance plus grande ce qui représente un r plus grand. et comme ça nous aurons les variations dans la distance r donc une caractéristique non stable de la lumière comme l'image droite de la figure 4.

Afin d'avoir les niveaux de gris dans une ligne d'un θ donné nous avons utilisé l'algorithme de Bresenham qui va retourner pour chaque angle un tableau de taille $[r(\theta)]$ contenant les niveaux de gris des pixels de la ligne.

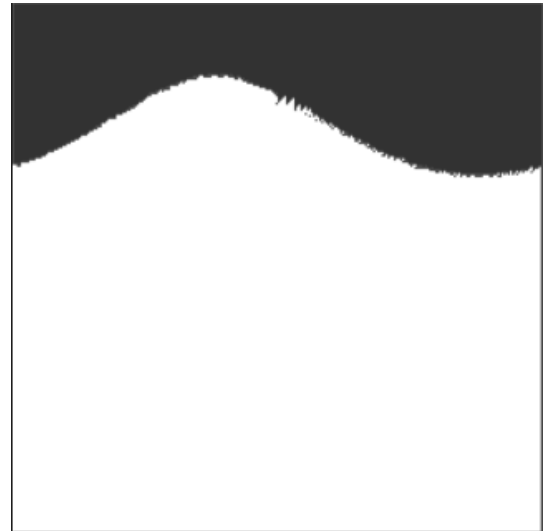


Figure 5. Image OCT simulée dans le domaine domaine polaire.

B. Atténuation de l'intensité dans l'artère:

Les systèmes IVOCT acquièrent plusieurs lignes. En supposant que l'OCT ne détecte que la lumière qui a été diffusée

une fois et que le tissu est homogène, l'intensité d'une telle A-line $I(d)$ diminue avec la profondeur selon la loi de Lambert-Beer :

$$I(d) \propto I_0 \cdot \exp(-\mu_t \cdot d) \quad (3)$$

où d indique la profondeur de pénétration, μ_t le coefficient d'atténuation total (comprenant à la fois l'absorption μ_a et la diffusion μ_s) et I_0 L'intensité sur la surface de la lumière. La figure 6 Montre le résultat après la simulation de l'atténuation de l'intensité dans le vaisseau.

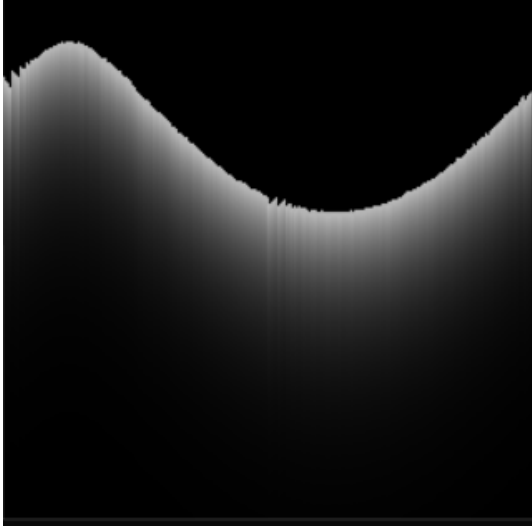


Figure 6. Image OCT simulée dans le domaine polaire avec une atténuation suivant la loi de Beer-Lambert.

C. Simulation du speckle :

Comme il est visible sur la figure 1, les images OCT présentent du speckle, c'est-à-dire que les valeurs de pixels sont distribuées de manière probabiliste. (Le speckle est un artefact déterministe basé sur l'interaction de la lumière avec les caractéristiques du sous-voxel et est mieux modélisé en tant que bruit multiplicatif [Lien]. La littérature statistique contient un grand nombre de modèles probabilistes qui pourraient être appliqués pour modéliser le speckle. Jusqu'à présent, notre étude ne s'est concentrée que sur la modélisation par loi gamma.

1) **La loi gamma:** En théorie des probabilités et en statistiques, une distribution Gamma ou loi Gamma est un type de loi de probabilité de variables aléatoires réelles positives. Une variable aléatoire X suit une loi Gamma de paramètres k et θ (2 nombres strictement positifs), ce que l'on note aussi $X \sim \Gamma(k, \theta)$ si sa fonction de densité de probabilité peut se mettre sous la forme :

$$f(x; k, \theta) = \frac{x^{k-1} e^{-\frac{x}{\theta}}}{\Gamma(k) \theta^k} \quad (4)$$

avec $\Gamma()$ c'est la fonction Gamma d'Euler :

$$\Gamma(k) = \int_0^{+\infty} t^{k-1} e^{-t} dt \quad (5)$$

Et pour être conforme avec la librairie déjà contenant la loi Gamma sur C++ nous allons représenter la loi par le paramètre de forme $\alpha = k$ et le paramètre de densité $\beta = \frac{1}{\theta}$ et on a :

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1} \beta^\alpha e^{-x\beta}}{\Gamma(\alpha)} \quad (6)$$

de moyenne et variance, (Équations 7 et 8) à la suite :

$$\text{Moyenne} = (k - 1)\theta = (\alpha - 1) \frac{1}{\beta} \quad (7)$$

$$\text{Variance} = (k - 1)\theta^2 = (\alpha - 1) \frac{1}{\beta^2} \quad (8)$$

2) **Le speckle sur l'image:** Dans l'image de la figure 7 nous présentons le résultat obtenu après l'ajout d'un bruit Gamma multiplicatif de $\alpha = \frac{\text{pixel_value}}{\beta}$ et $\beta = 5$, (avec pixel_value c'est la valeur du pixel sur lequel en appliquant la distribution Γ), nous avons donc une moyenne = $(\frac{\text{pixel_value}}{5} - 1) \cdot \frac{1}{5}$ et une variance = $(\frac{\text{pixel_value}}{5} - 1) \cdot \frac{1}{5^2}$.

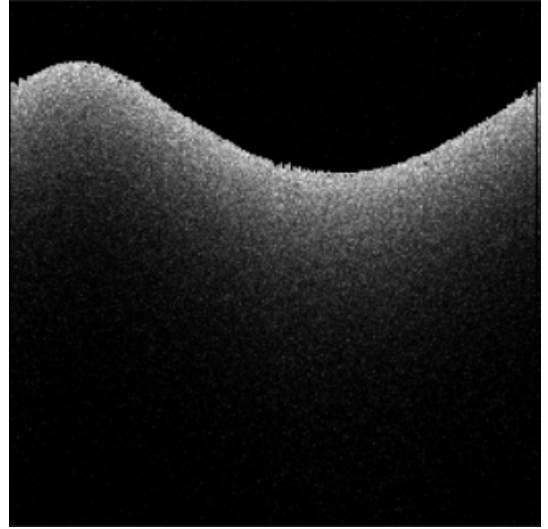


Figure 7. Image OCT simulée avec bruit Gamma (Speckle).

D. Les plaques athéromateuses en OCT :

Puisque le sujet est de segmenter les plaques athéromateuses, nous avons dû simuler 3 types de ces plaques : (Lipidique, Calcifié et Fibrotique voir la figure 8). Les plaques ils ont simulé à l'intérieur de la lumière avec une profondeur très petite, et de taille qui varie (longueur et largeur) aléatoirement respectant des dimensions réelles". Pour garder des caractéristiques réalistes de la plaque, nous avons considérées que la plaque suit la lumière (Nous avons considéré la plaque comme un rectangle qui suit la lumière, comme géométrie il ne reflète pas la réalité).

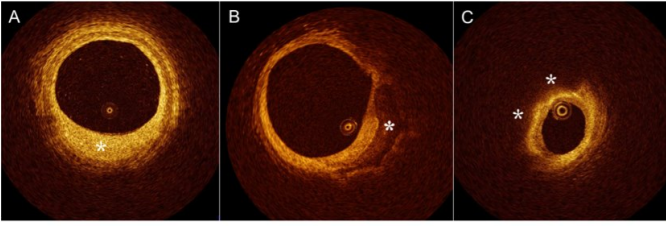


Figure 8. Images d'OCT réelles montrant les 3 plaques : A/Fibrotique // B/Calcifiée // C/Lipidique.

1) **La plaque fibrotique:** Ce type de plaque est caractérisé par le fait qu'il est homogène, elle présente un sur-signal et une faible atténuation du signal.

Et pour modéliser ses caractéristiques par une loi Gamma, nous avons diminué β pour enlever la moyenne et diminuer la variance (pour l'homogénéité et le sur signal) ainsi enlevé *pixel_value* vers 200 afin de présenter le sur-signal et on n'a pas modifié la valeur du signal après la plaque, le résultat est représenté dans la figure 9.

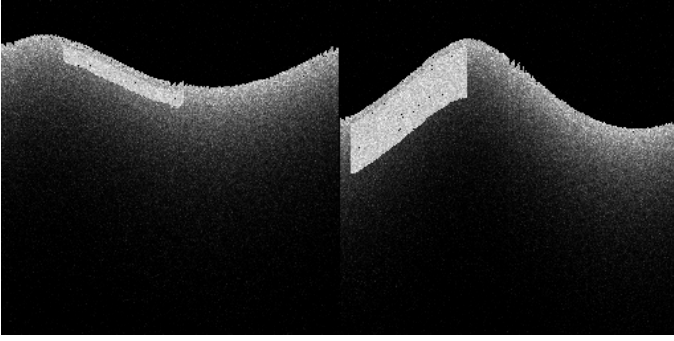


Figure 9. 2 Images simulées présentent des plaques fibrotiques.

2) **La plaque calcifiée:** Ce type de plaque est caractérisé par le fait qu'il a des contours marqués, elle présente un sous-signal.

Et pour modéliser ses caractéristiques par une loi Gamma, nous avons augmenté β et on a affecté au *pixel_value* la valeur 80 pour réduire la moyenne afin de modéliser le sous-signal, nous n'avons pas modifié les contours de cette plaque, ainsi le signal après la plaque, afin de garantir le caractère marqué des contours le résultat est représenté dans la figure 12.

3) **La plaque Lipidique:** Ce type de plaque est caractérisé par le fait qu'il a des contours diffus, elle présente un sous-signal aussi.

Et pour modéliser ses caractéristiques par une loi Gamma, nous avons augmenter β et on a affecté au *pixel_value* une valeur exponentielle décroissante en fonction de la profondeur (r), afin de réduire le signal après la plaque sans qu'il soit nul. Le caractère diffus des images peut être représenté par le critère précédent de la plaque, mais aussi par la transition entre les côtés (droit et gauche) doit être douce. pour cela nous avons ajouté une fonction de lissage des contours qui est

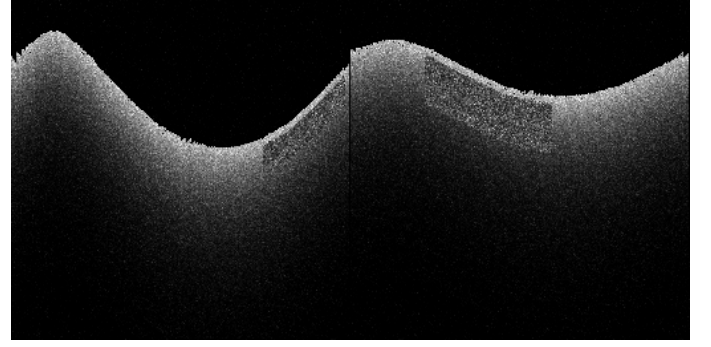


Figure 10. 2 Images simulées présentent des plaques calcifiés.

la fonction d'erreur (Équation 9-figure 9) permettant de mener les valeurs de niveau de gris d'un niveau bas de la plaque à un niveau élevé qui est de la lumière dans le contour droit de la plaque, et une fonction d'erreur complémentaire (Équation 10-figure 10) pour modéliser une diminution lisse des valeurs de la partie gauche qui est la lumière vers l'intérieur de la plaque. Les résultats obtenus sont présentés dans la figure 13.

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (9)$$

$$erfc(x) = 1 - erf(x) \quad (10)$$

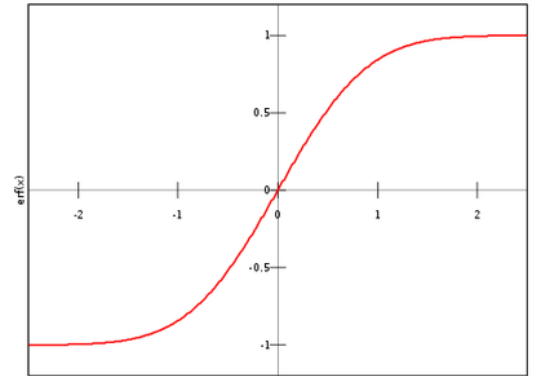


Figure 11. La fonction d'erreur.

III. SEGMENTATION PAR DEEP LEARNING:

A. Préparation des données :

Notre objectif dans ce projet est de faire la segmentation sémantique qui consiste à classifier chaque pixel des images comme des étiquettes des classes qu'on a, nous présentons par la suite 3 types d'algorithmes qui permettent d'exécuter ce type de tâches.

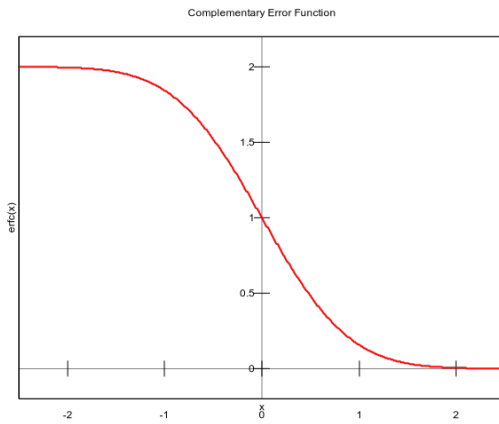


Figure 12. La fonction d'erreur complémentaire.

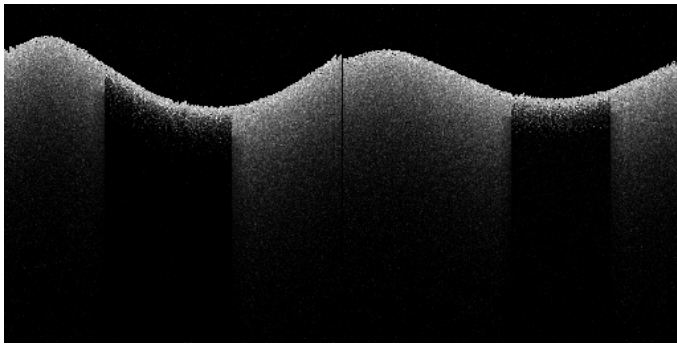


Figure 13. 2 Images simulées présentent des plaques Lipidiques.

B. Segnet :

SegNet dispose d'un réseau de codage et d'un réseau de décodage correspondant, suivis d'une couche finale de classification par pixel. Cette architecture est illustrée sur la figure 14. Le réseau de codeurs se compose de 13 couches de conversion (pour la classification d'objets). Le SegNet évite l'utilisation des couches entièrement connectées pour la conservation des cartes de caractéristiques de résolution plus élevée à la sortie d'encodeur le plus profond. Cela a réduit le nombre de paramètres dans le réseau de codage SegNet de manière significative (de 134 M à 14,7 M) par rapport aux autres architectures. Chaque couche de codeur a une couche de décodage correspondante et donc le réseau de décodage a 13 couches. La sortie finale du décodeur est transmise à un classificateur multi-class 'Softmax' pour produire des probabilités de classe pour chaque pixel indépendamment.

Chaque encodeurs du réseau des encodeurs effectue une convolution avec des filtres pour produire un ensemble de cartes de caractéristiques. Ensuite, une $(ReLU)=\max(0,x)$ est appliquée. Ensuite, le max-pooling avec une fenêtre de 2×2 et un stride de 2 (fenêtre sans chevauchement) sont effectués et la sortie résultante est sous-échantillonnée par un facteur de 2. Le sous-échantillonnage donne un grand contexte d'image d'entrée (fenêtre spatiale) pour chaque pixel de la carte des caractéristiques. Alors que plusieurs couches de max-pooling

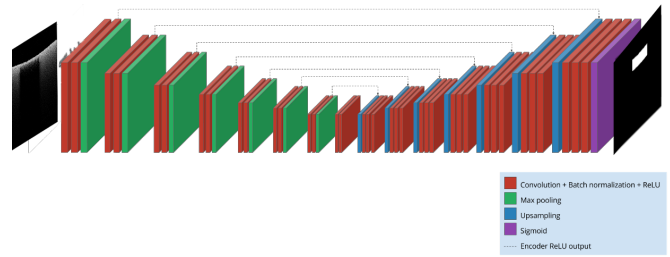


Figure 14. L'architecture de SegNet

et de sous-échantillonnage donnent une perte de résolution spatiale des cartes d'entités.

La représentation de l'image perte de plus en plus de détail si on s'en va plus en plus profond dans l'encodeur (surtout le détail des contours), ce qui n'est pas bénéfique pour la segmentation où la délimitation des contours importants. Par conséquent, il est nécessaire de capturer et de stocker des informations de contours dans les cartes de caractéristiques du codeur avant d'effectuer le sous-échantillonnage (sub-sampling). Si la mémoire pendant l'entraînement n'est pas limitée, alors toutes les cartes de caractéristiques de l'encodeur (après le sous-échantillonnage) peuvent être stockées. Ce n'est généralement pas le cas dans les applications pratiques. L'architecture SegNet propose donc un moyen plus efficace de stocker ces informations. Cela implique de stocker uniquement les indices de max-pooling, c'est-à-dire que les emplacements des caractéristiques qui ont une valeur maximale dans chaque fenêtre de regroupement sont mémorisés pour chaque carte de caractéristiques de l'encodeur. En principe, cela peut être fait en utilisant 2 bits pour chaque fenêtre de pooling 2×2 et est donc beaucoup plus efficace à stocker par rapport à la mémorisation de la carte en précision flottante.

Le décodeur approprié dans le réseau de décodeurs échantillonne la carte de caractéristiques d'entrée en utilisant les indices de regroupement maximum mémorisés à partir de la carte de caractéristiques de l'encodeur correspondant. Cette étape produit des cartes de caractéristiques clairsemées. Cette technique de décodage SegNet est illustrée sur la figure 15. Ces cartes de caractéristiques sont ensuite convoluées avec un filtre de décodeur entraînable pour produire des cartes de caractéristiques denses. Une étape de normalisation des batches est ensuite appliquée à chacune de ces cartes. Notez que le décodeur correspondant au premier encodeur (le plus proche de l'image d'entrée) produit une carte de caractéristiques multicanaux, bien que son entrée d'encodeur ait 3 canaux (RGB). Ceci est différent des autres décodeurs du réseau qui produisent des cartes de caractéristiques avec le même nombre de tailles et de canaux que leurs entrées de codeur. La représentation des caractéristiques de haute dimension à la sortie du décodeur final est transmise à un classificateur softmax pouvant être entraîné. La softmax classe chaque pixel indépendamment. La sortie du soft-maxclassifieur est une image de canal K de probabilités où K est le nombre de classes. La segmentation prédite correspond à la classe avec une

probabilité maximale à chaque pixel.

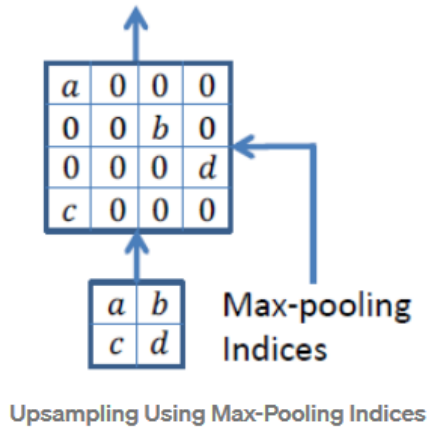


Figure 15. Le décodeur de SegNet

C. UNet :

Nous ajoutons ici une autre architecture, U-Net qui partage une architecture similaire à SegNet mais avec quelques différences. Par rapport à cette dernière, le U-Net (proposé pour la communauté de l'imagerie médicale) ne réutilise pas les indices de mise en commun, mais transfère à la place l'intégralité de la carte des caractéristiques (ici nous aurons besoin de plus de mémoire) aux décodeurs correspondants et les concatène à une carte des caractéristiques du décodeur sur-échantillonné ou (upsampled) (via déconvolution).

D. HRNet :

Ici, nous continuons à présenter le troisième modèle de segmentation d'image célèbre, HRNet[v2]. Le premier article de HRNet a été publié en 2019. Il a été utilisé pour la détection des poses, et HRNetv2 était basé sur le HRNet original et l'a légèrement modifié pour devenir un réseau de segmentation. Cependant, en raison des riches fonctionnalités extraites par HRNet, avec différentes résolutions, et les fonctionnalités haute résolution sont maintenues tout au long du réseau, le réseau est utilisé dans différentes tâches (telles que la reconnaissance d'image, la détection de cible, la détection de points de caractéristiques faciales, la segmentation sémantique, etc.). on se limite juste sur l'utilisation de HRNetv2 afin d'effectuer uniquement la tâche de la segmentation des plaques.

Comme le réseau Unet, dans le processus de propagation du réseau, il réduit progressivement la résolution spatiale de la carte des caractéristiques, peut être considéré comme un processus d'encodeur, puis on ajoute un décodeur de queue pour décoder les caractéristiques. Les informations sont décodées et la résolution de la carte des caractéristiques est agrandie en une seule fois ou (étape par étape) jusqu'à ce que la résolution d'entrée d'origine soit atteinte pour obtenir la carte de segmentation. Cela peut également être avec la fusion d'informations de résolution différentes.

Les premiers réseaux (SegNet et U-Net), entraînent une perte d'informations lors du processus de réduction de la résolution de la carte des caractéristiques. Même si une connexion par saut et une fusion multi-résolution sont ajoutées, cela peut ne pas être bien compensé. Il existe donc une autre façon de penser, en conservant une résolution plus grande tout au long du processus d'extraction de caractéristiques, en prédisant la carte de segmentation sur la carte de caractéristiques à grande résolution finale.

L'idée de conception de HRNet issue de la méthode de maintien d'une carte de caractéristiques de résolution plus grande. Au fur et à mesure que le réseau avance, il maintient une carte des caractéristiques plus grande, mais dans le processus de progression du réseau, il sera également parallèle (Fais un sous-échantillonnage pour réduire la carte des caractéristiques). Enfin, plusieurs groupes de cartes d'entités avec des résolutions différentes sont générées, et ces cartes d'entités sont fusionnées pour faire des prédictions de carte de segmentation.

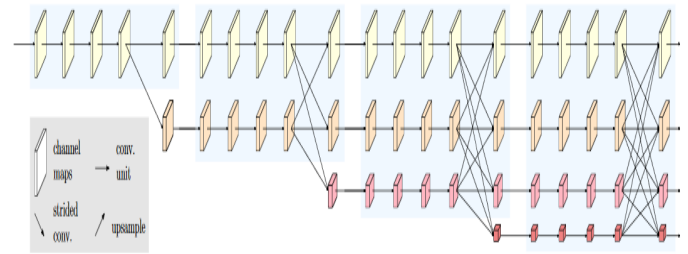


Figure 16. Un exemple de réseau haute résolution. Seul le corps principal est illustré et la potence (deux conversions stride-2 3×3) n'est pas incluse.

La figure 16 est un diagramme schématique simple de HRNet, qui génère une variété de fonctionnalités avec différentes résolutions. Le détail auquel il convient de prêter attention ici est qu'il présente une fusion dans le premier, moyen et dernier trois segments du réseau, et n'est pas seulement sur la carte des caractéristiques finale.

Fusion multi-résolution :

Les caractéristiques extraites par HRnet ont des résolutions différentes et la méthode de fusion doit être sélectionnée en fonction de différentes tâches.

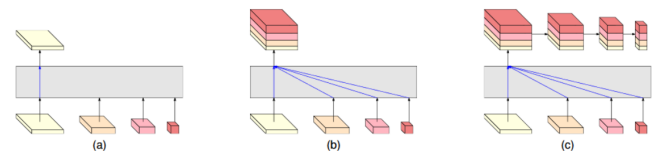


Figure 17. Les représentations à quatre résolutions en bas de chaque sous-figure sont sorties du réseau de la figure 16, et la boîte grise indique comment la représentation de sortie est obtenue à partir des représentations d'entrée à quatre résolutions.

Dans la version originale de HRNet pour la détection de la posture humaine, la méthode de fusion de la figure 17-a- est utilisée, c'est-à-dire que les fonctionnalités de basse résolution

sont ignorées et que seules les fonctionnalités de résolution maximale sont utilisées.

Si vous effectuez une segmentation sémantique ou un positionnement de point de caractéristique faciale, comme le montre la figure 17-b-, les caractéristiques des différentes résolutions sont sur-échantillonnées pour obtenir une carte de caractéristiques à grande résolution cohérente, puis concaténées pour la fusion.

Si vous effectuez une détection de cible, comme le montre la figure 17-c-, construisez une pyramide de caractéristiques multi-résolution sur la base de b.

E. Entraînement :

1) *Base de donnée*: Nous avons généré une base de donnée de 3000 images avec 3000 masques des trois classes des plaques (1000 pour chacune). La figure 18 représente une image avec son masque.

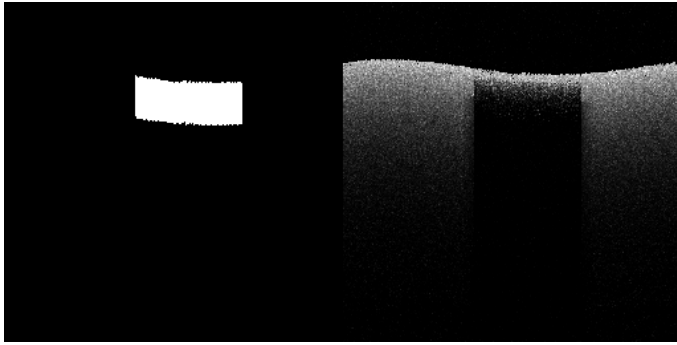


Figure 18. Masque de la plaque || Image de la plaque.

2) *Préparation des images*: Notre but est de la segmentation sémantique multi classe, c'est-à-dire que le sort de chaque modèle doit être une segmentation de la plaque avec une classification des plaques. Pour cela dans la liésons des plaques avec les masques, nous avons différencié les différents masques par des valeurs de labels différents 1-2-3, afin de passer au modèle en question les images réelles comme données d'entraînement, et les masques labelliser comme labels au données.

3) *Optimisation*: Un réseau neuronal de Deep Learning apprend à mapper un ensemble d'entrées à un ensemble de sorties à partir de données d'apprentissage. Nous ne pouvons pas calculer les poids parfaits pour un réseau neuronal; il y a trop d'inconnues. Au lieu de cela, le problème de l'apprentissage est présenté comme un problème de recherche ou d'optimisation et un algorithme sont utilisés pour naviguer dans l'espace des ensembles possibles de poids que le modèle peut utiliser afin de faire des prédictions bonnes ou suffisamment bonnes. Les optimiseurs sont des algorithmes ou des méthodes utilisés pour modifier les attributs du réseau neuronal tel que les poids et le taux d'apprentissage afin de réduire les pertes.

Dans notre cas nous avons utilisé Adam Optimisation qui a

donné les meilleurs résultats en matière de temps de convergence.

4) *Évaluation de la performance - Coefficient de Dice*: L'idée est simple, nous comptons les pixels similaires dans les deux images (Y_{true} C'est l'image de masque et Y_{pred} C'est l'image de masque prédite) et le multiplions par 2 puis on le divise par le nombre total de pixels dans les deux images. (comme ça on prend l'intersection, présents dans les deux masques) (voir 19).

et on a :

$$Dice_coef = 2 \frac{|Y_{true} \cap Y_{pred}|}{|Y_{true}| + |Y_{pred}|} \quad (11)$$

Avec :

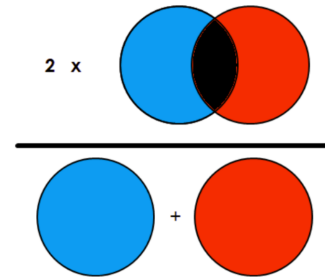


Figure 19. Représentation de coefficient de Dice.

F. La fonction a optimisé:

La fonction a optimisé va être prise du coefficient de dice c'est la fonction de perte de dice.

$$Dice_loss = 1 - dice_coef \quad (12)$$

1) *Optimisation d'entraînement par GPU*: Nous avons utilisé le GPU pour l'entraînement de notre algorithme, et elle a donné des résultats plus rapides que le processeur, cela peut réduire le temps d'entraînement de plusieurs heures. L'utilisation de la carte graphique n'est pas l'option par défaut dans le tensorflow, pour cela nous avons dû à installer le Tensorflow-GPU, Cuda et cudnn. Et afin de donner un ordre de grandeur nous présentons le tableau de la figure 20, de comparaison entre 8 core Ryzen 2700x CPU (16 threads, 20MB cache, 4.3GHz max boost) et Nvidia RTX 2080 (8192MB GDDR6 memory) sur une autre base de données, prit dû ce site.

Package:	tensorflow 2.0	tensorflow-gpu 2.0
Total Time [sec]:	4787	745
Seconds / Epoch:	480	75
Seconds / Step:	3	0.5
CPU Utilization:	80%	60%
GPU Utilization:	1%	11%
GPU Memory Used:	0.5GB	8GB (full)

Figure 20. Carte graphique Vs CPU.

IV. RÉSULTATS :

La sortie de chaque architecture décrite précédemment sera une image de 4 dimension (n,img_height,img_width,labels) (voir24), avec :

- n : l'image d'indice n, dans la base à prédire.
- img_height : sera la hauteur de l'image, ici 256.
- img_width : sera la largeur de l'image, ici 256.
- labels c'est le nombre de classe à prédire, ici 3 classe Plaque lipidique, calcifiée et fibrotique plus l'arrière-plans ce qui fait en total 4 classes.

Suivant le code qu'on a écrit, dans l'image de sortie sera :
image de label 0 : l'arrière plan.

image de label 1,2,3 à la suite c'est : plaque lipidique , plaque fibrotique et plaque calcifiée.

A. SegNet :

Dû à la pénurie de mémoire nous avons compilé le SegNet puisque c'est le plus léger parmi les différentes architectures que nous avons proposées.

Pour les résultats obtenus pour les données de test est un coefficient de dice = 0.8769537210464478 et une perte de dice = 0.12299085408449173. La figure 21 montre les résultats obtenus dans le domaine polaire.



Figure 21. Image de la plaque Classe 1 || Masque réel || Masque prédit || Masque prédit avec classification.

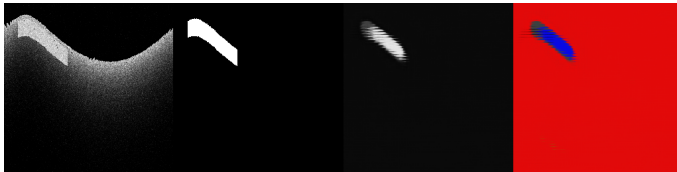


Figure 22. Image de la plaque Classe 2 || Masque réel || Masque prédit || Masque prédit avec classification.



Figure 23. Image de la plaque Classe 3 || Masque réel || Masque prédit || Masque prédit avec classification.

Les figures 21, 22 et 23 montrent les résultats obtenus par la segmentation sémantique sur les trois différentes plaques, en comparant avec les masques réels.

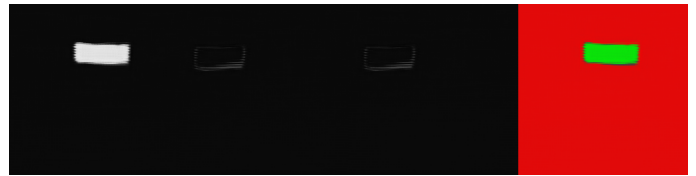


Figure 24. Prédiction du masque de la classe 1 || masque de la classe 2 || masque de la classe 3 || image d'indice n de prédiction.

V. DISCUSSION :

Certes le SegNet est limité et ne présente pas l'état de l'art concernant les réseaux convolutives. La figure 25 montre une fausse classification d'une plaque comme quoi elle est un mélange de classes 1 et 3, ainsi la réalité c'est juste classe 3. La figure 26 montre une mauvaise localisation de l'extrémité de la plaque qui est mélangée avec la lumière.

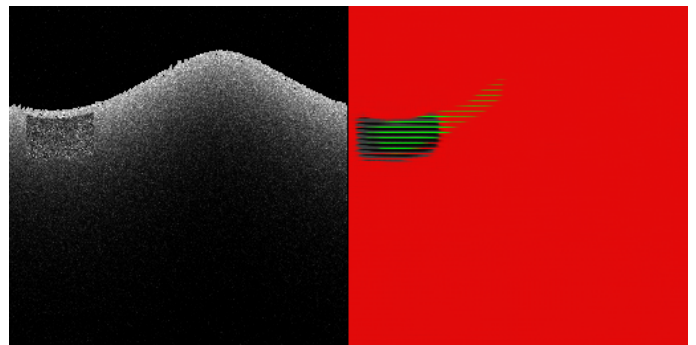


Figure 25. Image de la plaque Classe 3 || Masque prédit avec classification.

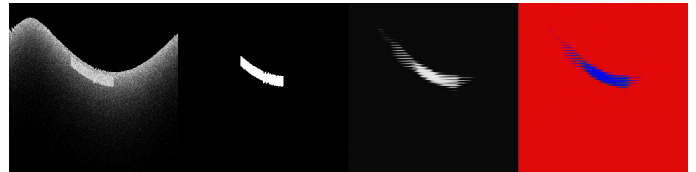


Figure 26. Image de la plaque Classe 2 || Masque réel || Masque prédit || Masque prédit avec classification.

Afin d'améliorer la segmentation nous pouvons penser à implémenter un algorithme plus puissant qui est le H. rnet. Notre hardware est limité et nous avons fait le training juste sur 10 epochs et un seul batch, et nous avons obtenu un dice coefficient de 0.9593227505683899 ce qui est très amélioré par rapport au SegNet sur 100 epochs et avec 8 batches. Le HRNet a donné meilleur résultat concernant la localisation des plaques, mais la classification n'était pas bonne comme il est affiché dans la figure 27.

VI. CONCLUSION

Pour les ressources locales qu'on a le meilleur algorithme à implémenter c'est le SegNet qui a donné des résultat visuelle satisfaisante, malgré qu'il trouve des problèmes pour les plaques qui on une courbure plus importante, le Unet sera

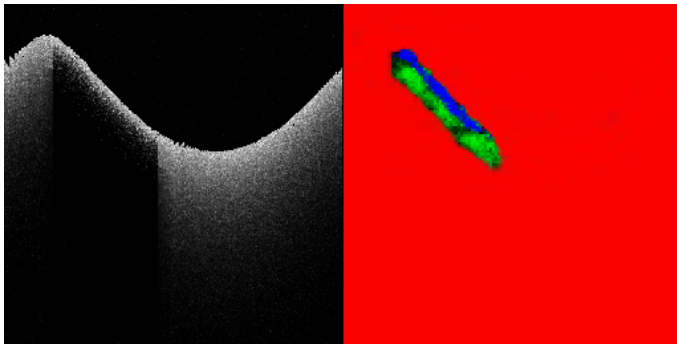


Figure 27. Image de la plaque Classe 1 || Masque prédit avec classification.

une alternatif qui est proche du Segnet mais par contre le HRNet avec une machine assez puissante et un bon réglage des hyperparametres nous prédisons des bons résultats surtout que cet algorithme est l'état de l'art pour toutes les taches liées à la vision d'ordinateur. Il ne faut pas oublier que ces résultats sont obtenus sur une base de données simulée, ce qui est très différent à la réalité ou en peut avoir des formes non régulières, du speckle sur toutes l'image est non pas juste sur la paroi ainsi la présence d'autres structures comme le thrombus....

REFERENCES

- [1] Deep High-Resolution Representation Learning for Visual Recognition
- [2] SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image-Segmentation
- [3] UNet++: A Nested U-Net Architecture for Medical Image Segmentation
- [4] Automated plaque characterization using deep learning on coronary intravascular optical coherence tomographic images
- [5] Use HRNet for segmentation
- [6] Deep Learning vs. Traditional Computer Vision Niall O' Mahony et al.
- [7] https://en.wikipedia.org/wiki/Gamma_distribution
- [8] Coronary Artery Calcification and its Progression
- [9] Principles of Coronary Imaging Techniques
- [10] Bridging Finite Element and Machine Learning Modeling: Stress Prediction of Arterial Walls in Atherosclerosis
- [11] An Automatic Segmentation Method Applied To Images Of IV-OCT In Human Arteries With Atherosclerotic Plaques
- [12] Expert review document on methodology, terminology, and clinical applications of optical coherence tomography: Physical principles, methodology of image acquisition, and clinical application for assessment of coronary arteries and atherosclerosis
- [13] Image Processing in Intravascular OCT