

Projet Master 2 - I3D.

Profils d'attribut morphologique pour la segmentation d'image.

Mehmet Ozkan

5 janvier 2020

1 Contexte du sujet

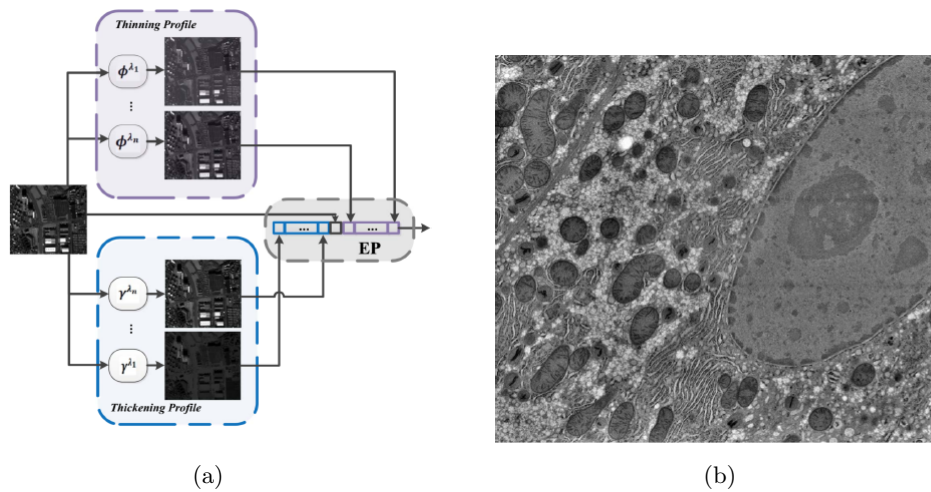


FIGURE 1 – Profil d'attributs morphologiques : les informations données par chaque opérateur sont concaténées dans un vecteur (a) et la coupe d'une image FIB/SEM 3D (b)

L'objectif de mon sujet de projet de Master 2 est d'extraire des profils d'attribut morphologique sur des images médicales (Figure 1) pour permettre de mettre en place des forêts d'arbres de décision pour la classification.

La première partie du projet est d'extraire les profils sur des images quelconques et ainsi obtenir un vecteur de profils, contenant le profil d'épaississement et le profil d'amincissement (Figure 1). La seconde partie est d'appliquer un apprentissage supervisé, en utilisant les forêts d'arbres décisionnels, ou *random forests* en anglais. Les caractéristiques sont extraites à partir des différents profils morphologiques, notamment les profils d'attribut et les profils d'extinction.

Pour répondre à ces deux problématiques, j'ai eu comme support deux articles scientifiques[1, 2]. Ces articles s'appuient sur deux techniques différentes pour extraire des profils, l'un permettant d'extraire des profils d'attribut tandis que l'autre permet d'extraire des profils d'extinction. Ainsi, pour pouvoir développer ce projet j'avais pour seule contrainte d'utiliser le langage de programmation C++ ou le langage de programmation Python. Après discussion avec mon encadrant, nous avons pris la décision de nous orienter plutôt vers le Python, où il y a un nombre important de bibliothèques en rapport avec la problématique du sujet.

2 Explication des différents principes

Dans cette partie, je vais expliquer les différents principes qui m'ont permis de mettre en place les différents algorithmes. Ainsi, je vais parler de la représentation des images sous forme d'arbre, dont il y a plusieurs types différents. De la manière de filtrer pour récupérer les différents profils d'attribut ou d'extinction qui nous intéressent. Ainsi que d'une technique d'apprentissage supervisée utilisée dans le domaine de la reconnaissance d'image, qui sont les forêts d'arbres décisionnels.

2.1 Représentation d'une image en arbre

La représentation d'une image sous forme d'arbre apporte plus de clarté pour permettre d'élaguer, et donc de filtrer des images rapidement et sans prise de tête. Ainsi on peut stocker les différents attributs qui nous permettent de concevoir nos profils. Cette technique est largement utilisée dans le domaine du traitement de l'image, mais aussi dans la segmentation d'image et encore dans l'apprentissage supervisée pour permettre de classer et détecter des similitudes.

2.1.1 Max-tree et Min-tree (ou component-tree)

$$\mu_h(C_i) = \max_{\forall k \in \text{descendants}(i)} \{h_k - h_i\} \quad (1)$$

$$\mu_a(C_i) = \sum_{\forall x \in C_i} 1 \quad (2)$$

$$\mu_v(C_i) = \mu_a(C_i) + \sum_{k \in \text{descendants}(i)} \mu_a(C_k) \times \text{nlevels}(k) \quad (3)$$

$$\mu_{\text{diag}}(C_i) = \sqrt{(x_{i,\text{max}} - x_{i,\text{min}})^2 + (y_{i,\text{max}} - y_{i,\text{min}})^2} \quad (4)$$

FIGURE 2 – Exemple d'attributs possibles à chaque noeud de l'arbre max-tree.

L'avantage que procure les structures *max-tree* est le fait que l'on peut calculer pour chaque noeud de l'arbre des attributs divers. Tels que, la hauteur de l'arbre qui donne une indication sur le contraste de l'image, la zone (area) qui est un attribut de taille, le volume qui est une combinaison du contraste et de la taille, et bien plus encore. La Figure 2 montre les différentes équations pour calculer ces différents attributs.

Il y a plusieurs algorithmes qui existent pour construire cet arbre, plus précisément il y a trois grandes catégories d'algorithmes. La première étant les algorithmes immersifs, puis les algorithmes de *flooding* et enfin les algorithmes basés sur la fusion d'élément[3].

La structure se représente sous deux grandes formes en mémoire. Premièrement en tant que deux vecteurs, l'un contenant les parents des pixels et l'autre contenant les indices des noeuds pour traverser l'arbre, tout cela, ordonnés. Et l'autre grande représentation consiste à avoir un tableau de la dimension de l'image, contenant pour chaque pixel le noeud correspondant dans l'arbre. Et un autre tableau contenant les parents des noeuds ainsi que les différentes valeurs que l'on peut y mettre, par exemple, la valeur en niveau de gris et les attributs de taille (zone).

Dimensions	Construction	Filtering	Restitution	Total
256 × 256	11.71	9.14	0.12	20.97
512 × 512	47.26	9.35	0.42	57.03
768 × 768	111.10	10.23	0.93	122.26
1024 × 1024	198.98	9.49	1.56	210.03
1280 × 1280	310.42	9.56	2.64	322.62

FIGURE 3 – Temps moyen de traitement nécessaire pour la construction, le filtrage et la restitution d'un max-tree en millisecondes.

Ainsi filtrer et reconstruire un *max-tree* est très rapide contrairement au temps que l'on passe à le construire, voir Figure 3.

Min-tree est le dual du max-tree. En récupérant une image et en appliquant un max-tree sur l'image négatif de cette même image, on obtient un min-tree.

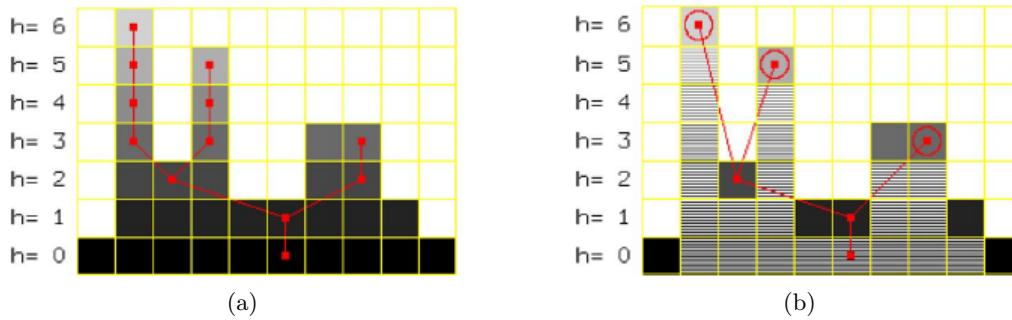


FIGURE 4 – (a) Component-tree et (b) max-tree de l'image à une dimension avec comme valeurs de $f = [0, 6, 2, 5, 1, 1, 3, 3, 1, 0]$.

On dit également que le *component-tree* et le *max-tree* sont des synonymes, même si certains considèrent que le *max-tree* est une forme compacte du *component-tree*, voir la Figure 4.

2.1.2 Tree of shapes

L'arbre des formes est une structure d'arbre représentant une image. Elle est définie en tant que la fusion des min-tree et max-tree. Ainsi on peut éliminer les redondances dans ces deux derniers arbres. Cet arbre encode l'inclusion spatiale des composantes connexes en niveau de gris. L'accès est facile et peut-être utilisé dans des domaines variés, liés au traitement de l'image.

2.2 Profil d'attribut

$$AP = \{\mathcal{P}_\phi^- \setminus \phi^{T_0}, \mathcal{P}_\gamma\}.$$

FIGURE 5 – Equation mathématique des profils d'attribut. Le profil est composé de $2n+1$ images.

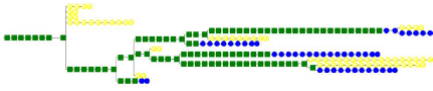
Le profil d'attribut est un vecteur comportant n -ouvertures morphologiques et n -fermetures morphologiques ainsi que l'image originelle. Ce profil est calculé en utilisant les max-tree et min-tree. Ainsi on filtre en utilisant des filtres d'attribut, dans notre cas par la zone (area). Le profil est la concaténation de tous ces éléments pour un vecteur de seuil donné. L'équation mathématique correspondant au profil d'attribut est donnée dans la Figure 5.

Donc si l'on a un vecteur de seuil de taille N , on aura un vecteur de profil de taille $2*N + 1$, comportant N -fermetures, l'image originelle et N -ouvertures.

2.3 Profil d'extinction

$$EP(f) = \left\{ \underbrace{\phi^{\lambda_1}(f), \dots, \phi^{\lambda_{n-1}}(f), \phi^{\lambda_n}(f)}_{\text{Thickening Profile } (\Pi_\phi)}, \underbrace{f, \gamma^{\lambda_n}(f), \gamma^{\lambda_{n-1}}(f), \dots, f, \gamma^{\lambda_1}(f)}_{\text{Thinning Profile } (\Pi_\gamma)} \right\}$$

(a)



(b)

FIGURE 6 – (a) Equation mathématique du profil d'extinction. Le profil est composé de $2n+1$ images. (b) Max-tree correspondant à une image où les noeuds en vert et bleu représente le filtre d'extinction avec $n = 5$.

Tout comme le profil d'attribut, le profil d'extinction se compose de la même manière. Sauf qu'au lieu d'utiliser des filtres d'attribut on utilise des filtres d'extinction, qui sont les valeurs extremes des arbres max-tree et min-tree. L'équation mathématique correspondant au profil d'extinction est donnée dans la Figure 6.

2.4 Automatisation de la recherche des seuils

Le principale problème des profils d'attributs morphologiques est de trouver le bon seuil pour les différents critères pour avoir les bonnes valeurs de profils. Ce problème est exclue des profils d'extinction qui n'utilise pas des seuils à proprement parlé mais des extrêmes. Comme on s'intéresse uniquement aux valeurs des extrêmes dans l'arbre *max-tree*. Pour remédier à ce problème l'article scientifique[2] parle de la manière d'automatiser cette recherche. Pour cela on s'intéresse aux *Granulometric Characteristics Functions*, ou GCF. L'algorithme est donné dans l'article[2].

2.4.1 Granulometric characteristic function (GCF)

$$\text{GCF}(\mathcal{P}_\psi(f)) = \{\mathcal{M}(\psi_i)\}_{i=1}^L.$$

FIGURE 7 – Equation mathématique de la fonction GCF où M est une mesure possible.

La GCF est une fonction permettant de décrire/quantifier l'effet d'un filtre sur une image en nuance de gris. L'équation mathématique est donnée par la Figure 7. Dans l'article[2], on nous décrit trois grandes fonctions GCF. La GCF qui quantifie la différence des valeurs de pixel entre l'image d'origine et le profil, celle qui nous décrit le nombre de pixels différents entre l'image d'origine et le profil, et enfin, celle qui quantifie la différence entre la cardinalité des composantes connexes de l'image de base et celle du profil.

2.5 Apprentissage

Dans cette section, je vais expliquer le principe des forêts d'arbres décisionnels.

2.5.1 Arbre de décision

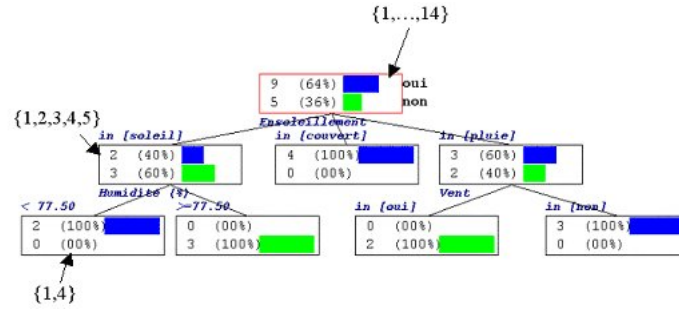


FIGURE 8 – Arbre de décision permettant de prendre une décision binaire (oui ou non).

Un arbre de décision est un arbre où chaque noeud est une condition menant vers un autre noeud pour au final arriver sur une décision finale se situant sur une feuille de l'arbre. Ainsi à chaque noeud de l'arbre, on effectue une décision intermédiaire, qui en fonction de la décision nous conduit vers un autre noeud, voir vers la décision finale. L'arbre est construit en fonction des critères (caractéristiques) passés. Un exemple d'arbre de décision est décrit par la Figure 1.

2.5.2 Forêts d'arbres décisionnels

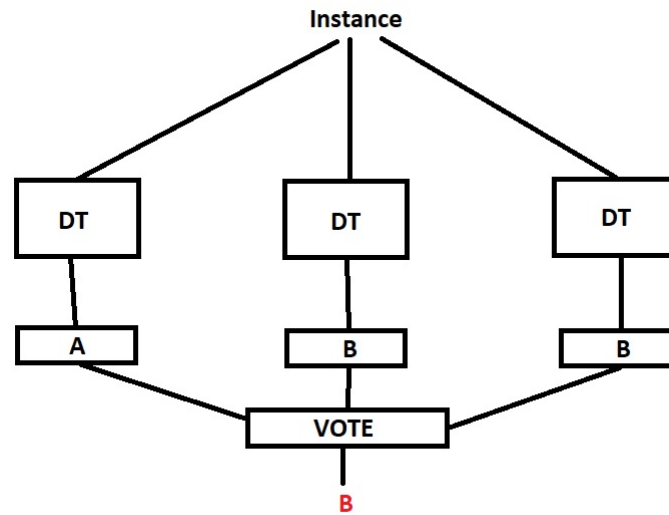


FIGURE 9 – Explication des random forests. DT = Arbre de décision. A et B sont deux classes.

L'algorithme des forêts d'arbres décisionnels effectue un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents[5], voir Figure 9. A la fin, la classe est déterminée en effectuant un vote à la majorité. Par exemple, si sur trois arbres de décision, l'un donne comme décision que la classe est A et que les deux autres me donnent comme décision B alors la classe sera prédite comme étant B. Ce modèle est utilisé pour la classification de données.

3 Application

Dans cette partie, je vais parler des outils utilisés, des moyens mis en place pour palier à certains problèmes. Ainsi que des étapes du développement appliquées pour répondre à la problématique du sujet.

3.1 Outils

Comme évoqué dans les sections précédentes, j'ai utilisé pour ce projet le langage de programmation Python, qui contient énormément de bibliothèques concernant le traitement de l'image, la construction des différentes représentation d'arbre, ainsi que des modèles pour l'apprentissage supervisé et non-supervisé.

Pour la construction des *max-tree* et pour récupérer les profils d'extinction et d'attribut, j'ai utilisé la librairie *siamxt*[4], conçu par les auteurs ayant écrit l'article scientifique[1], qui est mit à disposition.

Le traitement de l'image est rendu possible en utilisant la librairie matricielle *numpy* et de la bibliothèque *scikit-image* permettant de travailler sur ces matrices, pour le traitement de l'information matricielle, mais aussi pour l'import et export des images et des profils.

La partie apprentissage a été rendu possible grâce à l'utilisation de la librairie *scikit-learn* qui se compose de plusieurs modèle, dont les forêts d'arbres décisionnels.

Pour le visuel, j'ai utilisé *matplotlib*. Pour afficher les images et les graphiques. Ainsi que de *tqdm* pour mettre en place des barres de progression au sein du terminal pour voir l'avancée des étapes. La partie *extraction* a été entièrement parallélisée pour permettre d'extraire les profils de plusieurs images à la fois, si le processeur le permet.

3.2 Implémentation

La première partie du projet consiste à récupérer les profils d'attribut morphologique des différentes images mise à disposition. Dans notre cas des images FIB/SEM.

Pour récupérer les profils en utilisant les images d'entrées, je construis des min-tree et max-tree de ces différentes images. Par la suite, j'applique un filtrage d'attribut pour les profils d'attribut et un filtrage d'extinction pour les profils d'extinction (dans notre cas on s'intéresse uniquement aux zones, mais on peut également regarder le contraste). Le profil est ainsi la concaténation de profil d'épaississement et d'amincissement qui sont récupérés à chaque étape de filtrage. Dans le cas, des profils d'attribut on parle également de fermeture morphologique et d'ouverture morphologique. Pour construire les max-tree et min-tree, j'ai recouru à la bibliothèque *siamxt*[4] permettant de concevoir ces arbres et de faire des traitement dessus. Cette librairie a été entièrement développée par les auteurs de l'article scientifique[1]. Lorsque j'ai les différents profils, je les sauvegarde sur le disque au format *npz* en utilisant la librairie *numpy*.

L'étape permettant de récupérer les profils est complètement parallélisée pour permettre d'utiliser tous les coeurs disponibles sur le processeur pour accélérer la tâche. Pour cela, j'utilise la librairie *multithreading* fournit par Python 3.

Par la suite, je récupère ces profils et je les utilise sur des forêts d'arbres de décision pour faire de l'apprentissage supervisé. Pour cela je récupère les fichiers se trouvant dans le répertoire de sortie où il y a en plus des profils, les étiquettes correspondant aux images d'entraînement. Puis je construis le modèle en utilisant la bibliothèque *scikit-learn*, et notamment le modèle *RandomForestClassifier*, dont les paramètres sont définis dans le fichier *option.py* dans le code source. Par exemple, le nombre d'arbres de décision que va comporter le modèle.

J'effectue ensuite une étape de validation croisée pour que l'on puisse alterner entre le jeu d'entraînement et le jeu de test à la volée. Ainsi je sépare les données en dix, pour avoir un jeu de test étant dix pourcent des données initiales. Ainsi, on sélectionne le meilleur modèle correspondant à la reconnaissance de nos informations, après n -séparations. La librairie *scikit-learn* intègre la fonction *KFold* permettant de faire la validation croisée.

3.3 Problèmes rencontrés

L'un des problèmes majeurs que j'ai rencontré était la gestion des profils d'extinction et de la construction du max-tree. Il y avait très peu d'information sur les profils d'extinction, tandis que pour la max-tree il y a des algorithmes, comme cité plus haut, mais qui ne sont pas si simple à mettre en place. Heureusement j'ai pu avoir accès au GitHub de l'un des auteurs de l'article scientifique[1]. Ainsi, j'ai utilisé la librairie *siamxt* pour palier à ce problème.

Le deuxième problème était de savoir quoi donner aux forêts d'arbres décisionnels. Sachant que dans les deux articles, ils utilisent ce modèle de classification sans pour autant donner des détails quant aux entrées fournies.

Le troisième problème était d'utiliser le *piecewise linear regression* pour estimer la valeur des GCF, qui est utilisée dans l'algorithme fournit dans l'article[2] pour calculer les seuils automatiquement, pour les profils d'attribut.

Enfin le dernier problème étant que ce travail de développement s'est vite retrouvé à être un travail de recherche. Je devais palier au manque d'informations sur les articles[1, 2] pour comprendre au mieux les différents principes. En m'intéressant aux références des articles.

3.4 Résultats

Les résultats sont plutôt concluant même sur de petits jeu d'essai. Comme expliqué dans l'article scientifique[1], les profils d'extinction apportent une meilleure précision que les profils d'attribut.

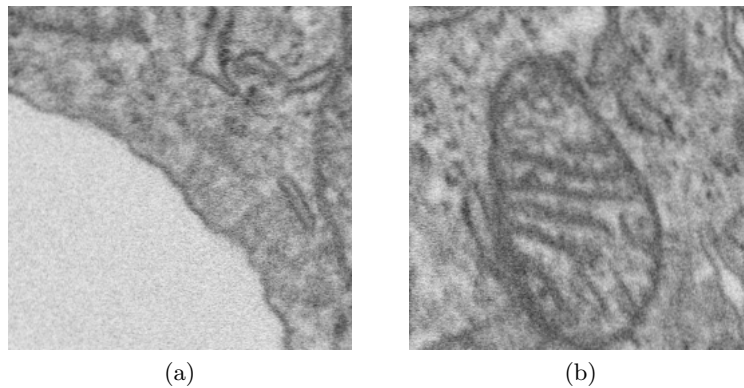


FIGURE 10 – (a) Patch 58 et (b) patch 78. Images FIB.

La classification me donne le résultat suivant pour ces deux images. Le patch (a) de la Figure 10 est classifié en tant que membrane nucléaire en utilisant les profils d'extinction, tandis que le (b) est classifié en tant que mitochondrie.

La matrice de confusion pour une taille de données de 500 images est indiqué par la Figure 11. Ainsi on peut voir que les résultats ne sont pas mauvais sachant que plus les résultats sont dans la diagonale de la matrice mieux c'est.

[223	7	2	0	0]
[7	141	1	0	1]
[0	4	60	0	0]
[1	0	0	21	0]
[0	3	0	0	29]]

FIGURE 11 – Matrice de confusion pour 500 images.

4 Conclusion et ouverture

En conclusion, ce projet m’a permis de mieux comprendre les principes des attributs morphologiques ainsi que de leurs utilités dans les différentes techniques de classification et détection d’images. J’ai également pu revoir mes bases concernant les différentes méthodes d’apprentissage, dans mon cas, surtout l’apprentissage supervisé automatique. Je n’ai malheureusement, pas eu les moyens d’implémenter tout ce qui a été demandé.

On peut par la suite, essayer d’utiliser la technique des profils d’attribut et d’extinction avec d’autres modèles d’apprentissage supervisé. Faire une comparaison entre l’apprentissage profond, qui récupère les caractéristiques importantes automatiquement en fonction d’un très grand nombre de données passées en paramètres. On peut également créer des données en utilisant les profils dans les domaines où le nombre de données est un critère important.

Références

- [1] Pedram Ghamisi, Roberto Souza, Jon Atli Benediktsson, Xiao Xiang Zhu, Letícia Rittner, Roberto A. Lotufo, Extinction profile for the classification of remote sensing data IEEE Transactions on Geoscience and Remote Sensing, 54(10), pp. 5631-5645, 2016.
- [2] Gabriele Cavallaro, Nicola Falco, Mauro Dalla Mura, Jon Benediktsson, Automatic attribute profiles IEEE Transactions on Image Processing, 26(4), pp.1859-1872, 2017.
- [3] Roberto Souza, Luís Tavares, Letícia Rittner, Roberto Lotufo, An overview of max-tree principles, algorithms and applications, 2016.
- [4] iamxt : Max-tree toolbox for image processing and analysis, Souza, Roberto and Rittner, Letícia and Machado, Rubens and Lotufo, Roberto, SoftwareX, Elsevier, v. 6, pp. 81-84, 2017.
- [5] Wikipédia, Forêt d’arbres décisionnels, 30 juin 2019.