Apprentissage profond interactif et semi-supervisé pour la segmentation volumique en tomographie électronique

Cyril Li

N° d'ordre NNT : 2023STET046

THÈSE de DOCTORAT
DE L'UNIVERSITÉ JEAN MONNET SAINT-ÉTIENNE

Membre de l'Université de LYON

École Doctorale N° 488
Sciences Ingénierie Santé SIS

Spécialité / discipline de doctorat :
Image, vision, signal

Soutenue publiquement le 29/11/2023, par :
Cyril Li

Apprentissage profond interactif et semi-supervisé pour la segmentation volumique en tomographie électronique

---

Devant le jury composé de :

Decencière Etienne	Directeur de recherche, Mines ParisTech	Rapporteur
Duffner Stefan	Maître de conférences HDR, INSA Lyon	Rapporteur
Fromont Elisa	Professeur, Université de Rennes	Examinatrice
Lartizien Carole	Directrice de recherche CNRS	Examinatrice
Ducottet Christophe	Professeur, Université Jean Monnet	Directeur de thèse
Moreaud Maxime	Docteur HDR, IFP Energies Nouvelles	Co-directeur de thèse
Desroziers Sylvain	Docteur, Michelin	Encadrant

Table des matières

1 Introduction **1**
1.1 Contexte 2
1.2 Tomographie électronique 2
1.3 Segmentation sémantique 3
1.4 Contributions 5
1.5 Structure du document 6

2 État de l'art **7**
2.1 Introduction 8
2.2 Segmentation 2D 9
2.2.1 Réseaux de neurones convolutifs 10
2.2.2 Entraînement d'un réseau de neurones 14
2.2.3 Revue des méthodes de segmentation d'image par apprentissage profond 15
2.2.4 Réseaux de type Encodeur-Décodeur 20
2.2.5 Convolution à trous 23
2.3 Segmentation 3D 24
2.3.1 Modèles 3D 25
2.3.2 Modèle 2D par plan et modèle 3D 26
2.4 Segmentation avec peu de données annotées 28
2.4.1 Annotations partielles 28
2.4.2 Few shot Learning 30
2.4.3 Approches auto-supervisées 31
2.4.4 Apprentissage contrastif 34
2.5 Segmentation d'objet vidéo 36
2.5.1 Segmentation interactive 37
2.5.2 Segmentation d'objet semi-supervisée 39
2.6 Récapitulatif 40

3 Tomographie électronique : spécificité des données et appren-

tissage profond **42**
3.1 Introduction . . . 44
3.2 Notions de base sur la reconstruction de données en tomographie électronique . . . 44
3.2.1 Microscope électronique en transmission . . . 44
3.2.2 Principe de la tomographie . . . 45
3.2.3 Principales difficultés de reconstruction des images de tomographie électronique . . . 46
3.3 Présentation des données disponibles . . . 47
3.3.1 Support de catalyseur alumine . . . 48
3.3.2 Zéolithes . . . 48
3.4 Protocole d'évaluation . . . 50
3.4.1 Évaluation du réseau U-Net sur des données 2D de microscopie électronique à balayage . . . 51
3.4.2 Évaluation du réseau U-Net sur des données partiellement annotées 2D de microscopie électronique à balayage 52
3.4.3 Évaluation du réseau U-Net sur les données 3D de tomographie électronique . . . 54
3.5 Résultats . . . 56
3.5.1 Premiers résultats avec le réseau de neurone U-Net sur les données bidimensionnelles de microscopie électronique à balayage . . . 56
3.5.2 Apport de l'annotation partielle . . . 59
3.5.3 Résultat d'un réseau de neurones U-Net sur les données tridimensionnelles de tomographie . . . 60
3.6 Conclusion . . . 62

4 Nouveau modèle basé sur l'apprentissage contrastif **64**
4.1 Introduction . . . 65
4.1.1 Approches semi-supervisées pour la segmentation sémantique . . . 65
4.1.2 Apprentissage contrastif . . . 66
4.2 Méthode proposée pour la segmentation sémantique à partir d'apprentissage contrastif . . . 66
4.2.1 Formalisation de la perte contrastive . . . 67
4.2.2 Architecture . . . 69
4.2.3 Gestion des données non annotées . . . 70
4.3 Expérimentations . . . 72
4.3.1 Paramètres expérimentaux . . . 72
4.3.2 Comparaison avec un entraînement conventionnel . . . 74
4.3.3 Choix du classifieur . . . 75

4.3.4 Apport des fonctions de perte 76
4.4 Conclusion 78

5 Hijacked-STM : un réseau de segmentation de vidéo détourné pour de la segmentation semi-supervisé de matériaux 80
5.1 Introduction 82
5.1.1 Potentiels de la segmentation d'objet vidéo pour les données volumiques 83
5.1.2 Segmentation d'objet vidéo semi-supervisé 84
5.1.3 Segmentation interactive 85
5.2 Détournement d'un réseau à mémoire 86
5.2.1 Procédure de propagation des annotations 86
5.2.2 Encodage Clé-Valeur 87
5.2.3 Lecture partielle de la mémoire 89
5.3 Expérimentation 91
5.3.1 Dispositif expérimental 91
5.3.2 Comparaison avec STM 92
5.3.3 Propagation de la segmentation 93
5.3.4 Comparaison avec des méthodes de segmentation . . . 94
5.4 Conclusion 94

6 Conclusions et perspectives 97
6.1 Contributions 98
6.1.1 Évaluation des données de tomographie électronique . . 98
6.1.2 Apprentissage contrastif pour la segmentation de volumes 99
6.1.3 Segmentation semi-supervisée utilisant des réseaux à mémoire 99
6.2 Perspectives 100
6.2.1 Évaluation sur d'autres types de données 100
6.2.2 Adaptation de contexte pour les vecteurs clés 100
6.2.3 Implantation d'une méthode interactive de segmentation 101

Table des figures

1.1 Image d'un plan d'un volume d'un matériau mésoporeux, ici une zéolithe. La taille de l'image est 592 pixels par 600 pixels avec une résolution 1 nanomètre par voxel. 3
1.2 Photo d'un microscope électronique en transmission JEOL 2100F. 3
1.3 Exemple de segmentation sémantique d'une image de lion. Chaque pixel est annoté en fonction de sa classe. 4
1.4 Exemple de segmentation sémantique d'un voxel d'un plan de matériau mésoporeux (résolution : 1 nm/voxel). Chaque voxel est classifié selon la présence ou non de matière. 5

2.1 Exemple de segmentation sémantique dans plusieurs domaines d'applications. 9
2.2 Résultat de la convolution pour la première valeur de la matrice. Le filtre est appliqué autour de la première valeur et une somme pondérée par les poids du filtre donne le résultat. . . . 11
2.3 Exemple de *max pooling*. 11
2.4 Exemple de neurone d'une couche entièrement connectée. . . . 12
2.5 Exemple de fonction *softmax*. 13
2.6 Exemple d'architecture de CNN (VGG [74]). Ce CNN comportant des blocs de convolution (bleu), *max pooling* (rouge) et entièrement connecté (vert) 13
2.7 Exemple de réseau entièrement convolutif [54]. La dernière couche de classification des pixels est une couche convolutive, contrairement aux CNN utilisés en classification. 15
2.8 Exemple d'encodeur-décodeur [83]. 16
2.9 Schéma de R-CNN [33]. Un module de segmentation sémantique donne le masque de segmentation du masque. . . . 17
2.10 Schéma d'un modèle multiéchelle [32]. Chaque étage correspond à une résolution différente. 17

2.11 Image (gauche) et la carte d'attention générée par un modèle d'attention (droite) [13]. 18
2.12 Schéma de fonctionnement d'un réseau antagoniste génératif. . 19
2.13 Un post-traitement utilisant un modèle de contour actif pour la segmentation d'images médicales [29]. 19
2.14 Schéma d'un transformeur visuel où chaque image est découpée en patchs qui seront encodés avec un transformeur de traitement automatique du langage naturel [26]. 20
2.15 Architecture d'U-Net. La première moitié de l'architecture est l'encodeur, la deuxième partie est le décodeur [78]. 21
2.16 Convolution classique à gauche et convolution à trous avec un taux de dilatation $r = 2$ à droite. Le champ réceptif est représenté en orange. 23
2.17 Champ réceptif à plusieurs taux de dilatation [11] 24
2.18 Architecture de V-Net, analogue à celle de U-Net, mais avec des convolutions 3D [59]. 26
2.19 Différences entre modèle 3D (haut) et modèle 2D plan par plan (bas). 27
2.20 Résultats de la comparaison entre U-Net 2D et 3D. U-Net 2D surclasse la version 3D dans la plupart des cas [39]. 27
2.21 Performance d'U-Net 2D et 3D. U-Net 2D est plus efficace en utilisation de la mémoire du GPU et est également plus rapide qu'U-Net 3D [39]. 28
2.22 Configuration pour la segmentation semi-automatique (haut) et configuration pour segmentation automatique (bas). 29
2.23 Exemple d'image partiellement annotée (2D). 30
2.24 Principe du *few shot learning*. 31
2.25 Principe du *few shot segmentation*. 32
2.26 Exemple d'un réseau guidé où le support est encodé puis utilisé pour guider l'encodeur-décodeur [75]. 32
2.27 La tâche à réaliser dans le cadre de l'utilisation de contexte pour apprendre des représentations visuelles. Le but est de prédire où le patch rouge se situe par rapport au patch bleu. En essayant de répondre aux questions 1 et 2, on remarquera que la tâche est bien plus facile dès que l'on reconnaît l'objet en question [25]. 33
2.28 La tâche à réaliser est de remplir une image partiellement masquée (gauche). Si le résultat est probant (droite), le réseau aura appris une représentation visuelle correcte [68]. 34
2.29 Exemple de paire positive et de paire négative et leur influence sur la fonction de perte contrastive. 34

2.30 Schéma d'un réseau siamois. x correspond à l'image d'entrée modifiée par deux transformations aléatoires $\mathcal{T}$. $\tilde{x_i}$ et $\tilde{x_j}$ forment ainsi une paire positive, qui passe dans deux encodeurs f qui partagent les mêmes poids. $f(\tilde{x_i})$ et $f(\tilde{x_j})$ sont ensuite projetés dans un espace dans lequel il est possible de les comparer grâce à la fonction de similarité [15]. 35
2.31 À gauche, l'approche auto-supervisée et à droite, l'approche supervisée. À gauche, l'image du chien encadré en rouge est incorrectement catégorisée en paire négative avec l'image de chien encadré en noir. Avec l'information de la classe, il n'est plus possible de produire cette erreur. 36
2.32 Différentes étapes d'une étape de segmentation interactive. Tout d'abord, une prédiction de segmentation est produite. L'annotateur indique des corrections, qui seront effectuées par la méthode de segmentation interactive [16]. 38
2.33 Les clics de l'annotateur indiquent les parties à inclure (clic positif en bleu) et les parties à exclure (clics négatifs). Ces informations sont concaténées à l'image d'entrée avec les différents canaux RGB. Le réseau fournit une carte de segmentation prenant en compte l'image et les instructions de l'annotateur [56]. 38

3.1 Schéma d'un MET. À gauche, le mode image, à droite le mode diffraction [4]. 45
3.2 Schéma du principe de tomographie. À gauche, l'acquisition d'images d'un échantillon sous plusieurs angles. À droite, l'utilisation de ces images pour reconstruire l'échantillon [79]. . . 46
3.3 Illustration du manque de projections dû à un intervalle angulaire de balayage limité [84]. 47
3.4 Support catalyseur de zéolithe déplacé (6,64 nm/pixel) entre deux angles d'acquisition. [84]. 47
3.5 Images reconstruites contenant des artéfacts de reconstructions. [84]. 48
3.6 Image de support de catalyseur alumine (résolution : 0.11165 µm/pixel). 49
3.7 Exemple de bruit (a) et d'artéfacts de reconstruction (b) sur un plan de deux volumes de zéolithes différents (résolution : 1 nm/pixel). L'image représentant le bruit a été agrandi d'un facteur 10. 50
3.8 Exemple de plan issu de chacun des volumes (résolution : 1 nm/pixel). La plupart de ces volumes proviennent divers matériaux de type zéolithes et sont visuellement différents. . . 51

3.9 Architecture du réseau U-Net mis en place. 52
3.10 Différence entre les annotations de deux images similaires. . . 53
3.11 Image originale à gauche, image semi-labellisé à droite où les pixels verts correspondent à l'objet, les pixels rouges au fond et les pixels bleus aux pixels non annotés. 54
3.12 Image d'un plan semi-labellisé. En vert, l'objet d'intérêt, en violet le fond et en jaune, les voxels non labellisés. 56
3.13 Illustration de l'IOU. À gauche, le rapport de l'intersection et de l'union du masque et de la segmentation est inférieur à l'exemple de droite. Plus l'aire de l'intersection et de l'union sont proches, plus l'IOU sera proche de 1. 57
3.14 Exemple d'un entraînement par validation croisée par blocs. Ici, avec 24 images par bloc, 4 images sont sélectionnées pour l'entraînement (cases en bleu), 4 pour la validation (cases en rouge) et les 16 restantes pour le test (cases en vert). Pour chaque bloc, les différents groupes d'images sont différents. Un score est calculé pour toutes les images de tests et la moyenne entre les scores des différents blocs constitue le score final. . . 58
3.15 Image originale de taille 1024x768 d'une résolution de 0.11165 µm/pixel à gauche, représentation des résultats à droite avec en vert les pixels correctement segmentés, en rouge les pixels manquants par rapport au masque de segmentation (faux négatifs) et en violet les pixels en trop du masque de segmentation (faux positifs). On peut voir que les principaux objets ont été segmentés correctement, cependant de nombreux petits défauts sont manquants. Il y a relativement peu de faux défauts segmentés. De plus, on observe des effets de bord dû au découpage par patchs. 59
3.16 Image originale à gauche. À droite, le masque de segmentation superposé à l'image originale avec en vert les pixels catégorisés en objet et en rouge les pixels catégorisés en pixel fond. 60
3.17 Influence du nombre de plans d'entraînement. En abscisses, nombre de plans utilisés à l'entraînement, et en ordonnées, l'IOU correspondant. Plus l'IOU est proche de 1, meilleur est le résultat de la segmentation. Pour les valeurs de nombre d'images d'entraînement inférieur à 1, un seul plan d'entraînement est utilisé avec un taux de labellisation r correspondant à cette valeur. 61
3.18 IOU de chaque plan en fonction de la distance au plan central et de plusieurs taux de labellisation. En bleu clair, la courbe montrant la quantité de matière. 62

4.1 Schéma d'un espace latent où chaque point correspond à une image projetée dans cet espace. Les images d'une même classe sont proches, contrairement aux images de classes différentes. . 67
4.2 Stratégie de sélection de paires pour un voxel annoté $z_{s,i}$. . . . 68
4.3 Architecture proposée composée d'un réseau f de segmentation et d'un classifieur h. L'image d'entrée est passée dans un réseau de neurones f entraîné avec une fonction de perte contrastive. En sortie, on obtient une carte des caractéristiques dans lesquelles chaque pixel est projeté dans un espace à grande dimension, construit grâce à l'entraînement contrastif. Cette carte des caractéristiques est ensuite passée dans un classifieur h qui permet d'obtenir la carte de segmentation finale. 69
4.4 Stratégie de sélection de paires pour un voxel non annoté $z_{s,i}$. 70
4.5 Le calcul de la fonction de perte est différent selon la classe des pixels. Pour les pixels labellisés, des paires de pixels sont construites et sont utilisées dans une fonction de perte contrastive supervisée. Pour les pixels labellisés et non labellisé, l'image d'apprentissage et sa version transformée sont utilisées comme paires positives dans une perte contrastive par auto-apprentissage. 72
4.6 Comparaison des IOU pour différents taux de labellisation entre la méthode supervisée par entropie croisée (vert) et notre méthode par fonction de perte contrastive (rouge). Les zones en rouge et vert indiquent l'écart-type mesuré sur l'ensemble des réalisations. 74
4.7 Comparaison entre différents modules de classification avec en rouge un classifieur SVM et en vert un classifieur intégré à l'encodeur-décodeur avec une couche de classification. On remarque que pour un nombre de pixels annoté faible, il est plus facile d'entraîner une couche de convolution qu'un SVM. . 76
4.8 Comparaison des IOU avec différentes fonctions de pertes : l'entropie croisée pondérée, la fonction de perte contrastive supervisée, la fonction de perte contrastive auto-supervisée et la combinaison des fonctions de perte supervisée et non supervisée. 77
4.9 Résultats des segmentations pour un taux de labellisation $r = 0.06$ avec différentes fonctions de pertes. Les pixels verts sont les pixels corrects, les pixels rouges sont les pixels manquants à la segmentation et les pixels violets sont les pixels en trop par rapport à la vérité terrain. 78

4.10 Visualisation des reconstructions 3D de différents volumes. . . 79

5.1 Similarités entre vidéos et volumes. Les deux types de données peuvent être interprétés comme une pile d'images successives avec une continuité d'une image à l'autre. 83
5.2 Principe d'un réseau à mémoire pour la segmentation sémantique de vidéo. Une image et son masque sont encodés dans la mémoire. L'image requête est encodée puis comparée aux données dans la mémoire pour produire une prédiction de masque. Le résultat est stocké dans la mémoire. 85
5.3 Différentes étapes d'une étape de segmentation interactive. Tout d'abord, une prédiction de segmentation est produite. L'annotateur indique des corrections, qui seront effectuées par la méthode de segmentation interactive [16]. 86
5.4 Propagation par patchs : seuls les voxels annotés contribuent à la mémoire. 87
5.5 Propagation par frame : le plan reconstruit est encodé dans la mémoire. 89
5.6 Architecture de la méthode. Un encodeur mémoire et requête produisent des vecteurs clé et valeur qui sont utilisés pour la lecture de la mémoire et permettent de guider le décodeur. . . 90
5.7 Schéma de la lecture partielle de la mémoire. 91
5.8 Un plan partiellement annoté d'une zéolithe (1 nm/pixel). Une fenêtre d'une surface A_w est considérée comme annotée, alors que la classe des voxels à l'extérieur de cette fenêtre est inconnue. Le centre de la fenêtre est choisi au hasard près de la frontière entre l'objet et l'arrière-plan afin d'inclure des pixels des deux classes. 92
5.9 Moyenne des IOU pour plusieurs taux de labellisation r. Toutes les méthodes ne nécessitent pas de procédure d'apprentissage supplémentaire, à l'exception de U-Net et de la version contrastive de U-Net. [45]. 95
5.10 Visualisation 3D de volumes segmentés de zéolithes NaX Siliporite G5. Une fenêtre aléatoire de 6% d'une tranche a été annotée. Les segmentations sont fournies en utilisant notre approche (Algorithme 1). 96

Liste des tableaux

2.1 Table comparant la différence de quantité de pixels entre un jeu de données d'images bidimensionnel et la quantité de voxels de ce même jeu de données avec une hypothétique troisième dimension. 25

3.1 Récapitulatif des données à disposition. 50
3.2 Tableau des IOU pour les différents tests de segmentation utilisant le réseau U-NET 2D. Les résutats quantitatifs montrent une bonne stabilité d'un bloc à l'autre (écart type de 0,04) et une moyenne de 62,5% sur l'IOU. Ce résultat n'est pas très élevé à cause des problèmes d'annotation discutés précédemment. 59

4.1 Table des transformations et de leurs paramètres. 71
4.2 Comparaison des IOU pour différents taux de labellisation entre la méthode supervisée par entropie croisée et notre méthode par fonction de perte contrastive. 75
4.3 Résultats sur différents volumes. 75

5.1 Moyenne des IOU sur nos volumes pour notre méthode et un réseau STM non modifié. La modification de notre approche permet à un modèle de type STM de produire une bonne segmentation. 93
5.2 IOU moyen sur nos volumes pour notre méthode avec seulement les parties étiquetées dans la mémoire (propagation par patchs) et notre méthode avec les pseudo-labels du premier plan dans la mémoire (propagation par frame). Notre approche n'utilise que les zones annotées en mémoire. 94

5.3 IOU moyen sur nos volumes pour notre méthode, un U-Net adapté aux zones partiellement segmentées, et un U-Net utilisant une fonction de perte contrastive pour exploiter à la fois les zones annotées et non annotées. La méthode que nous proposons obtient des résultats proches de ces méthodes malgré l'absence de phase d'entraînement. 95

1

Introduction

Outline

1.1 Contexte 2
1.2 Tomographie électronique 2
1.3 Segmentation sémantique 3
1.4 Contributions 5
1.5 Structure du document 6

1.1 Contexte

Les travaux présentés dans ce document sont le fruit d'une collaboration entre le laboratoire Hubert Curien et IFP Energies nouvelles (IFPEN), réalisés au sein du Labex MILYON [1]. Centre de recherche spécialisé dans le secteur de l'énergie, du transport et de l'environnement, IFPEN développe de nouveaux procédés pour la production de biocarburant qui peut être une alternative au carburant classique. Ce type de production fait intervenir des opérations de catalyse et plus particulièrement de catalyse hétérogène, très utilisée dans le domaine de l'énergie. La catalyse correspond à l'accélération ou le ralentissement d'une réaction chimique, à l'aide de catalyseur. La catalyse est hétérogène lorsque le catalyseur et les réactifs sont de différente phase. Ici, les catalyseurs utilisés sont solides. Ces catalyseurs deviennent de plus en plus complexes pour répondre aux contraintes environnementales de plus en plus sévères. De ce fait, les méthodes d'analyses classiques permettant leur évaluation atteignent leurs limites. C'est un challenge à relever puisque de récents travaux ont montré que la microstructure du support du catalyseur a une influence directe sur ses propriétés physico-chimiques. Il faut donc des méthodes innovantes permettant de caractériser ces nouveaux matériaux. Un moyen d'y parvenir est d'utiliser des images multidimensionnelles à une échelle très élevée de ces catalyseurs. Nos travaux se focalisent principalement sur la caractérisation d'images obtenues par tomographie électronique, une technique d'imagerie permettant l'observation des catalyseurs à l'échelle nanométrique [Figure 1.1].

1.2 Tomographie électronique

La tomographie électronique permet l'acquisition de la structure interne d'un objet, tout en effectuant des mesures externes à l'objet. Des projections obtenues par microscope électronique en transmission sont utilisées pour la reconstruction volumique nanométrique de matériaux [Figure 1.2]. Le microscope électronique en transmission acquiert des images en envoyant des rayons d'électrons sur un échantillon. Une image est obtenue avec la projection de ces électrons sur un écran. Un algorithme de reconstruction, nécessitant plusieurs projections, avec plusieurs angles différents, permet l'obtention d'une image tridimensionnelle d'échantillons nanométriques. Dans ce travail, nous nous intéressons à une étape nécessaire à l'analyse de ces images,

1. ANR-10- LABX-0070 de l'Université de Lyon, dans le cadre du programme "Investissements d'Avenir" (ANR-11-IDEX-0007) géré par l'Agence Nationale de la Recherche (ANR).

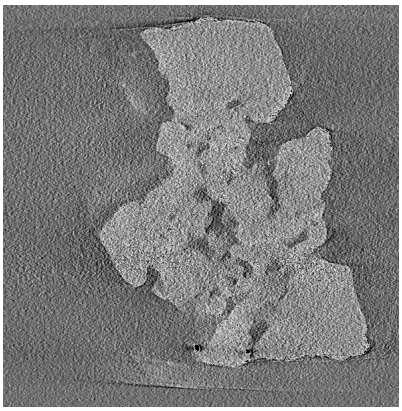


FIGURE 1.1 – Image d'un plan d'un volume d'un matériau mésoporeux, ici une zéolithe. La taille de l'image est 592 pixels par 600 pixels avec une résolution 1 nanomètre par voxel.

la segmentation sémantique.

FIGURE 1.2 – Photo d'un microscope électronique en transmission JEOL 2100F.

1.3 Segmentation sémantique

La segmentation sémantique automatique des images consiste à catégoriser chaque pixel en sa classe respective [Figure 1.3]. Dans notre cas, pour la segmentation sémantique d'image de supports de catalyseurs, on cherche à annoter chaque voxel en fonction de sa composition, matière ou vide [Figure 1.4]. Cette étape de segmentation est nécessaire pour caractériser la texture de ce type d'objets, comme l'analyse la morphologie et topologie de la mi-

crostructure. L'enjeu de la segmentation sémantique est d'obtenir une carte suffisamment précise pour permettre l'utilisation d'algorithmes d'analyse de ces matériaux. Le résultat de la segmentation est le masque de segmentation, un volume binaire où chaque voxel indique la présence ou non de matériaux.

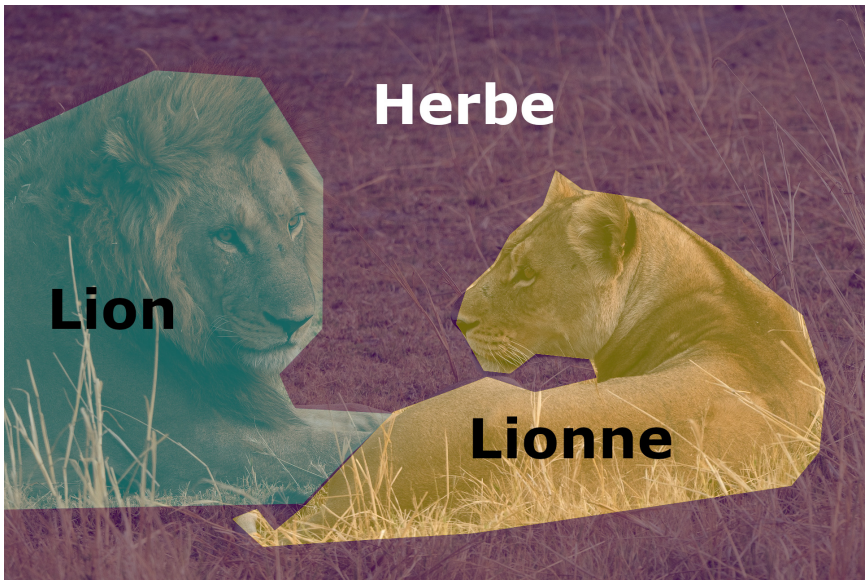


FIGURE 1.3 – Exemple de segmentation sémantique d'une image de lion. Chaque pixel est annoté en fonction de sa classe.

Un grand nombre de méthodes de segmentation automatique par apprentissage profond existent déjà dans la littérature [78, 14, 11]. Cependant, les approches standards nécessitent en général un grand nombre de données préalablement labellisées. Dans notre contexte, il est difficile d'obtenir cette quantité de données. D'une part, les images sont spécifiques, limitant la possibilité d'obtenir une base d'annotations volumineuses en s'appuyant sur la communauté. D'autre part, la segmentation manuelle faite par un expert peut prendre plusieurs jours pour un seul volume. Avoir une stratégie efficace pour gérer le faible volume de données et limiter les sollicitations de l'expert sont des enjeux majeurs que nous souhaitons adresser. Des travaux récents ont proposé des approches permettant d'utiliser peu de données annotées (few shot learning [75]), d'exploiter des données non labellisées (apprentissage auto-supervisé [25]), ou combinant les deux (apprentissage semi-supervisé [69]). Une autre approche tirée du domaine de la segmentation vidéo appelée segmentation interactive permet à l'annotateur de guider à un réseau de neurones en apportant des directions et des corrections afin d'obtenir un masque de segmentation [18]. Dans le contexte de la segmentation volumique

de matériaux, de façon surprenante ces méthodes ont été peu ou pas utilisées à notre connaissance et il n'est pas évident de savoir comment les appliquer et quels bénéfices elles pourraient apporter. Un objectif de cette thèse est d'apporter une réponse.

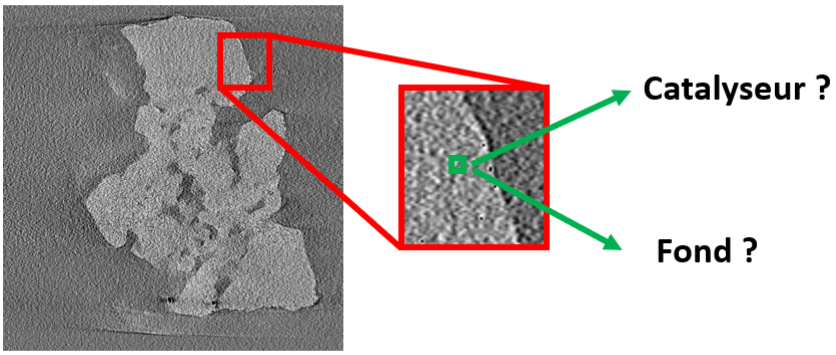


FIGURE 1.4 – Exemple de segmentation sémantique d'un voxel d'un plan de matériau mésoporeux (résolution : 1 nm/voxel). Chaque voxel est classifié selon la présence ou non de matière.

1.4 Contributions

Le but de mon travail de thèse est de proposer de nouvelles méthodes de segmentation sémantique de volume tridimensionnelle par apprentissage profond permettant de réduire le temps passé à effectuer cette tâche fastidieuse. Nous proposons trois principales contributions qui seront présentées dans ce document :

1. Une procédure d'utilisation de volumes partiellement annotés pour l'entrainement d'un réseau neuronal profond.
2. Un nouveau modèle de réseau profond basé sur l'apprentissage contrastif pour exploiter des données partiellement annotées et des données non annotées.
3. Un modèle de segmentation interactive pour des images de tomographie électronique exploitant de travaux effectués dans le domaine de la segmentation vidéo.

1.5 Structure du document

Ce document étudie des méthodes de segmentation sémantique de volume tridimensionnelle par apprentissage profond. Ces méthodes sont destinées à des ingénieurs et chercheurs ayant besoin d'exploiter des images tridimensionnelles de matériaux issues de tomographie électronique sans passer par des étapes fastidieuses d'annotations de volumes entiers. Ce document est organisé de la manière suivante :

- Le chapitre 2 présente un état de l'art des différentes méthodes de segmentation sémantique disponibles dans la littérature. En particulier, ce chapitre introduit les réseaux de neurones convolutifs qui ont récemment démontré de très bonnes performances dans le domaine de la segmentation sémantique. Nous nous intéresserons aux méthodes traitant des images bidimensionnelles, puis des images tridimensionnelles. Enfin, cette revue abordera les méthodes nécessitant peu de données d'entraînement.
- Le chapitre 3 contient des résultats préliminaires obtenus suite à une analyse avancée des données à disposition. En analysant les difficultés posées par ces images vis-à-vis des méthodes de segmentation sémantique classiques, nous proposons un nouveau dispositif d'entraînement de réseau de neurones convolutifs.
- Les travaux présentés dans le chapitre 4 s'appuient sur les expériences présentées dans le chapitre 3. Ce chapitre introduit une nouvelle approche basée sur l'entraînement contrastif. En exploitant complètement toutes les données à notre disposition, nous proposons une approche permettant la segmentation sémantique d'images de catalyseur issues de la tomographie électronique avec très peu de données annotées.
- Le chapitre 5 est dédié aux travaux inspirés des méthodes de segmentation vidéo. Sur la base des résultats du chapitre 4, nous exploitons les résultats prometteurs obtenus dans le domaine de la segmentation sémantique de vidéos, pour proposer une méthode de segmentation interactive dédiée aux images de tomographie électronique.
- En conclusion, nous faisons un récapitulatif des contributions de thèse, présentées dans les chapitres précédents. Nous présentons également plusieurs pistes pour de futurs travaux de recherche.

2

État de l'art

Outline

2.1 Introduction 8
2.2 Segmentation 2D 9
2.2.1 Réseaux de neurones convolutifs 10
2.2.2 Entraînement d'un réseau de neurones 14
2.2.3 Revue des méthodes de segmentation d'image par apprentissage profond 15
2.2.4 Réseaux de type Encodeur-Décodeur 20
2.2.5 Convolution à trous 23
2.3 Segmentation 3D 24
2.3.1 Modèles 3D 25
2.3.2 Modèle 2D par plan et modèle 3D 26
2.4 Segmentation avec peu de données annotées . . 28
2.4.1 Annotations partielles 28
2.4.2 Few shot Learning 30
2.4.3 Approches auto-supervisées 31
2.4.4 Apprentissage contrastif 34
2.5 Segmentation d'objet vidéo 36
2.5.1 Segmentation interactive 37
2.5.2 Segmentation d'objet semi-supervisée 39
2.6 Récapitulatif 40

2.1 Introduction

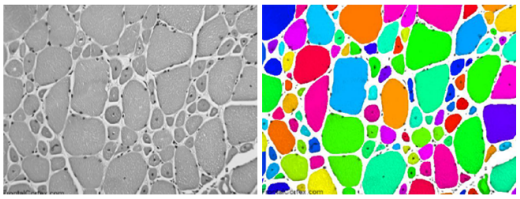
Dans la littérature, de nombreux travaux portent sur la segmentation [11, 14, 59, 92]. Ici, nous nous intéressons à la segmentation dite *sémantique*. La segmentation sémantique consiste à attribuer une catégorie à chaque pixel d'une image. C'est une tâche très importante dans de nombreux domaines, notamment en imagerie médicale [78], pour la conduite de véhicules autonomes [10] ou bien dans notre cas, l'analyse de matériaux [57]. Cette étape est souvent nécessaire préalablement à une analyse de forme par exemple. Les méthodes de segmentation par apprentissage profond se basent typiquement sur l'utilisation d'architectures neuronales entraînées avec de larges bases de données. Ces méthodes présentent des avantages et des inconvénients, dépendant de contraintes liées au type de données, au temps d'exécution ou au temps d'entraînement.

Le cadre applicatif que nous considérons dans ce travail comporte deux contraintes notables vis-à-vis de nombreux travaux de la littérature. Tout d'abord, les données sont volumiques, les approches envisagées doivent donc prendre en compte l'intégration d'une troisième dimension spatiale. La prise en compte de cette dimension augmente le nombre de paramètres des architectures neuronales et en conséquence les ressources matérielles nécessaire à leur traitement telles que la mémoire vive ou la puissance de calcul.

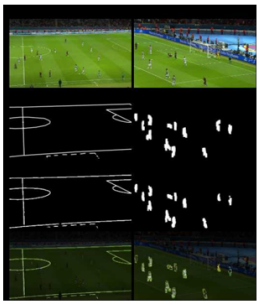
D'autre part, le volume de données annotées à disposition est extrêmement limité en nombre. Là où les méthodes classiques de l'état de l'art nécessitent d'immenses bases de données, il n'est possible d'avoir accès qu'à une quantité très limitée d'images. Pour tenter de palier à ces contraintes, il existe plusieurs approches :

- **Labellisation partielle**, par des méthodes prenant en compte des données partiellement annotées. Avec cette approche, il n'est plus nécessaire de fournir des volumes complètement annotés pour entraîner le réseau.
- **Apprentissage avec peu de données**, optimisant l'utilisation des données pour apprendre. Ce type de méthode exploite au mieux les données disponibles, par exemple en apprenant des espaces de représentation traduisant la structure des données d'apprentissage. Un des moyens pour apprendre des espaces de représentation est d'utiliser l'apprentissage contrastif, en recherchant un espace dans lequel les données qui se ressemblent sont proches selon une certaine fonction de similarité.
- **Apprentissage auto-supervisé**, se passant complètement de

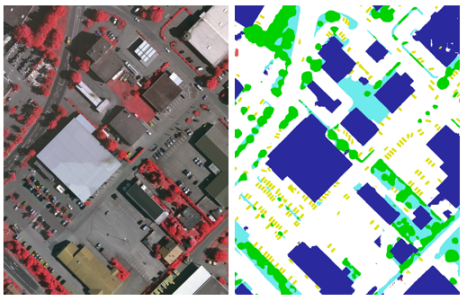
Analyse de paysages urbains [30]



Segmentation de cellules [81]



Compréhension de parties de football [21]



Surveillance d'images satellites [3]

FIGURE 2.1 – Exemple de segmentation sémantique dans plusieurs domaines d'applications.

données annotées, permettant un apprentissage sans supervision. Il faut alors exploiter au maximum les données disponibles, soit en utilisant le contexte des données, soit en transformant les données pour en augmenter leur nombre.

Dans un premier temps, nous présenterons les méthode usuelles de segmentation sémantique d'images bidimensionnelles, et en particulier, aux méthodes d'apprentissage profond. Puis, nous aborderons des approches relatives à la segmentation sémantique volumique. Nous explorerons ensuite les différentes méthodes d'apprentissage avec peu de données.

2.2 Segmentation 2D

De nombreuses méthodes ont ainsi été établies avant l'utilisation de réseaux de neurones : la classification de l'histogramme de l'image par l'estimation automatique de seuil [65] ou de méthodes de classification de l'espace des pixels [66], la fusion de régions [63], les champs aléatoires de Markov [72] pour ne citer que quelques exemples. Nous ne considérons pas ce type de méthodes dans cet état de l'art pour nous focaliser sur les méthodes basées sur l'apprentissage profond.

Nous introduirons tout d'abord le principe de base des réseaux de neurones convolutifs. Puis une revue des différentes méthodes classiques de segmentation sémantique sera établie. Nous détaillerons ensuite l'architecture des réseaux de type encodeur-décodeur. Enfin, nous nous intéressons aux convolutions à trous, qui permettent de réduire la charge en mémoire des méthodes de segmentations.

2.2.1 Réseaux de neurones convolutifs

Depuis les années 2015, les méthodes dominantes en traitement d'image sont basées sur l'apprentissage profond. L'apprentissage profond s'inspire de la façon dont fonctionne le cerveau humain. Les méthodes d'apprentissage profond utilisent des réseaux de neurones pour apprendre à partir d'une grande quantité de données. Récemment, les performances des méthodes d'apprentissage profond ont surpassé les performances humaines dans le cadre de la classification d'images [2]. Un des points forts des méthodes d'apprentissage profond est la diversité des applications dans lesquelles ces méthodes peuvent être utilisées. En effet, ces approches sont performantes dans beaucoup de domaines avec beaucoup de types de données différentes sans modifier la philosophie de base de l'apprentissage profond.

Les réseaux de neurones convolutifs (ou CNN pour *Convolutional Neural Network*) sont parmi les plus utilisés en apprentissage profond dans le domaine de la vision par ordinateur. Ils sont composés de plusieurs couches successives. Chaque couche est composée de neurones qui possèdent en entrée les neurones de la couche précédente. La valeur du neurone dépend de la fonction de combinaison de la couche. Un réseau de neurones convolutif est constitué de plusieurs couches successives, chacune avec une fonction bien précise : la couche de convolution, la couche d'activation, la couche dite de *pooling*, la couche complètement connectée et enfin la couche de softmax. La couche d'entrée est chargée avec les données de l'image.

- **La couche de convolution** combine les différents neurones d'entrée en effectuant une opération de convolution [Figure 2.2]. La convolution entre la couche $k-1$ d'entrée X^{k-1} et le filtre ou noyau de convolution w est définie comme suit :

$$X_{i,j}^{k} = (x * w)[i,j] = \sum_{m=-K}^{K} \sum_{n=-K}^{K} X_{i-m,j-n}^{k-1}.w_{m,n} \tag{2.1}$$

Avec N_w la taille du filtre et

$$K = \frac{N_w - 1}{2}$$

. Ainsi, chaque neurone ne dépend que d'un voisinage de neurone de la couche précédente. Cette fenêtre est appelée champ réceptif. Le résultat y, appelé carte d'activation (*activation map*) ou plan descripteur (*feature map*), est passé à la couche suivante. Les poids du filtre w sont des paramètres qui seront appris lors de l'entraînement, afin d'extraire les caractéristiques utiles à la classification. En pratique, plusieurs filtres sont utilisés. Si la taille de l'entrée de la couche de convolution est $W \times H \times C$ W et H les dimensions de la couche et C le nombre de canaux de la couche d'entrée, la taille du plan descripteur est $W \times H \times F$ avec F le nombre de filtres.

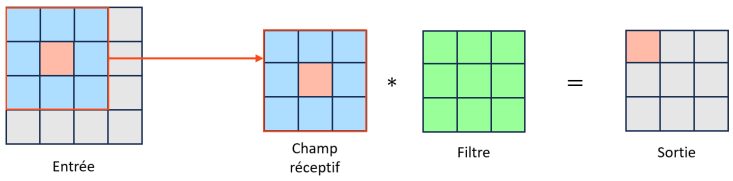


FIGURE 2.2 – Résultat de la convolution pour la première valeur de la matrice. Le filtre est appliqué autour de la première valeur et une somme pondérée par les poids du filtre donne le résultat.

- **La couche de pooling** sert à sous-échantillonner l'entrée. Le but est de réduire des dimensions spatiales de la couche d'entrée. Plusieurs méthodes sont possibles, dont l'*average pooling* ou le *max pooling*. L'*average pooling* calcule la moyenne du voisinage de chaque des vecteurs descripteurs pour réduire la taille d'entrée, tandis que le *max pooling* prend la valeur maximale du voisinage sur chaque dimension du vecteur descripteur [Figure 2.3].

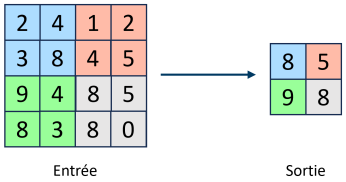


FIGURE 2.3 – Exemple de *max pooling*.

- **La fonction d'activation** sert à introduire une non-linéarité dans le réseau. Plusieurs fonctions d'activation existent, mais la plus commune pour les CNN est la fonction Unité Linéaire Rectifiée (*Rectified Linear Unit* ou ReLU). Elle transforme l'entrée en valeurs positives en fixant à 0 les valeurs négatives :

$$f_{ReLU}(x) = \max(0, x) \tag{2.2}$$

- **La couche entièrement connectée** relie tous les neurones de la couche précédente à chaque neurone de sortie [Figure 2.3]. Chaque neurone $X_{i,k}$ de la couche k est obtenu en effectuant une somme pondérée de tous les neurones de la couche précédente $k-1$, où à chaque neurone $X_{j,k-1}$ de la couche précédente est associé un poids $w_{j,k}$ qui sera appris à l'entraînement. De plus, un biais $w_{0,k}$ qui n'est pas combiné à un neurone est introduit. La valeur du neurone est :

$$X_{i,k} = w_{0,k} + \sum_{j=1}^{N_{k-1}} X_{j,k-1}.w_{j,k} \tag{2.3}$$

Ce type de couche est utilisé soit comme une couche intermédiaire, dîtes neurones cachés, soit comme couche de sortie. Lorsqu'elle est utilisée à la fin du réseau pour les tâches de classifications, le nombre de neurones de sortie correspond au nombre de classes possibles. Chaque neurone de cette couche correspond à la sortie du réseau pour chaque classe.

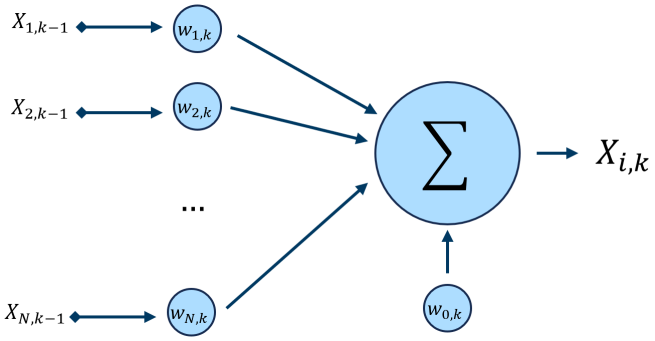


FIGURE 2.4 – Exemple de neurone d'une couche entièrement connectée.

- **La fonction exponentielle normalisée** (*softmax*) est une fonction de normalisation qui transforme la couche précédente en une probabilité $p \in [0, 1]$ [Figure 2.5]. Pour chaque neurone de la couche précédente $\tilde{y}_i$, la fonction *softmax* s'écrit :

$$p_i = \frac{e^{\tilde{y}_i}}{\sum_{k=1}^{N} e^{\tilde{y}_k}} \tag{2.4}$$

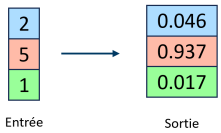


FIGURE 2.5 – Exemple de fonction *softmax*.

Un réseau convolutif est une combinaison de ces différentes couches. Par exemple, le réseau de classification d'image VGG [80] est composé de plusieurs blocs [Figure 2.6]. Chaque bloc est composé de trois couches de convolution suivit d'une couche d'activation ReLU et d'une couche de max pooling. L'avantage d'une approche convolutive est le nombre de paramètres à apprendre bien plus faible par rapport à une approche avec des couches entièrement connectée (appelé *Multi-Layer Perceptron* ou MLP). Le nombre de poids ne dépend plus de la taille de la couche d'entrée. Ce type d'architecture est largement utilisé en vision par ordinateur, les images par nature comportant un nombre de données unitaire important. Par exemple, pour une image de la taille 100 × 100, une couche entièrement connecté nécessite de 10000 poids. Un filtre d'une fonction convolutive de la taille 5 × 5 peut traiter cette image avec seulement 25 poids.

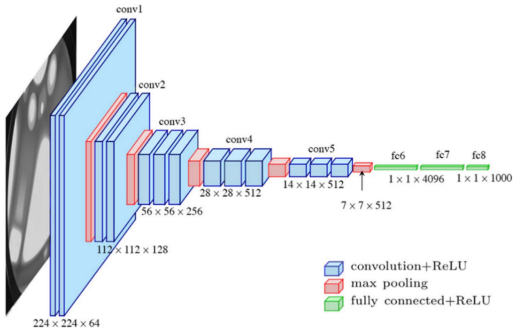


FIGURE 2.6 – Exemple d'architecture de CNN (VGG [74]). Ce CNN comportant des blocs de convolution (bleu), *max pooling* (rouge) et entièrement connecté (vert)

2.2.2 Entraînement d'un réseau de neurones

L'étape d'apprentissage permet d'optimiser tous les poids des couches. Pour ce faire, une fonction de perte (*loss function* en anglais) est utilisée. Elle prend deux paramètres en entrée : la sortie p du réseau et la prédiction attendue y. Une fonction de perte classique pour une tache de classification est l'entropie croisée (*cross-entropy*) :

$$L(p, y) = -\sum_{i=1}^{N} y_i \log p_i \tag{2.5}$$

Le but est alors de minimiser la fonction de coût L en fonction des paramètres θ du réseau :

$$\tilde{\theta} = \text{argmin} L(\theta) \tag{2.6}$$

Une méthode classique d'optimisation est l'algorithme de la descente de gradient stochastique (*Stochastic Gradient Descent* ou SGD). C'est une méthode itérative où les poids sont progressivement mis à jour. Soit θ_j les paramètres de la couche j, les paramètres mis à jour $\tilde{\theta}_j$ sont calculés tel que :

$$\tilde{\theta}_j = \theta_j - \eta \frac{\partial L}{\partial \theta_j} \tag{2.7}$$

Avec η le pas (*learning rate*) et $\frac{\partial L}{\partial \theta_j}$ le gradient de la fonction de coût par rapport aux paramètres θ_j. Ce gradient est calculé grâce à la rétro propagation du gradient de l'erreur de prédiction. L'enchaînement des différentes couches d'un réseau de neurones correspond à une composition de fonctions. Le théorème de dérivation des fonctions composées fournit une décomposition du gradient en produit de gradients locaux associé à chaque couche :

$$\frac{\partial L}{\partial \theta_j} = \frac{\partial L}{\partial \theta_N} \frac{\partial \theta_N}{\partial \theta_{N-1}} \cdots \frac{\partial \theta_{j+2}}{\partial \theta_{j+1}} \frac{\partial \theta_{j+1}}{\partial \theta_j} \tag{2.8}$$

La fonction de perte étant dérivable, le premier facteur peut être calculé. On remonte les différentes couches du réseau pour pouvoir calculer les autres facteurs. Le gradient est propagé jusqu'à l'entrée du réseau, modifiant ainsi tous les poids.

L'entraînement se déroule ainsi :

- L'image d'entraînement est propagée dans le réseau, permettant de fournir les neurones de sorties.
- La sortie du réseau est comparée à la sortie attendue grâce à la fonction de perte.
- Le gradient de la fonction de perte est calculé.
- À partir du gradient de la fonction de perte, le gradient est propagé sur les autres couches, en partant de la dernière couche et en remontant par composition jusqu'à la première couche, mettant à jour les différents paramètres.

Ces opérations sont répétées jusqu'à que les paramètres du réseau convergent. Généralement, au lieu de mettre à jour les poids après chaque passage de chaque image, les images d'entraînement sont regroupées en batchs. La fonction de perte globale à partir de la moyenne des erreurs calculées par les fonctions de perte de chaque image.

2.2.3 Revue des méthodes de segmentation d'image par apprentissage profond

La revue de Minaee *et al.* [60] classifie toutes ces méthodes de segmentation en dix catégories :

- **Réseaux Entièrement Convolutif** (FCN pour *Fully Convolutional Network*) [53, 54, 90] : ces modèles s'appuient sur les CNN utilisés en classification d'images, et sont détournés pour obtenir une classification par pixel [Figure 2.7]. Ces méthodes ont l'avantage d'avoir une base qui a déjà fait ses preuves dans le domaine de la classification. Ce sont des méthodes populaires et efficaces.

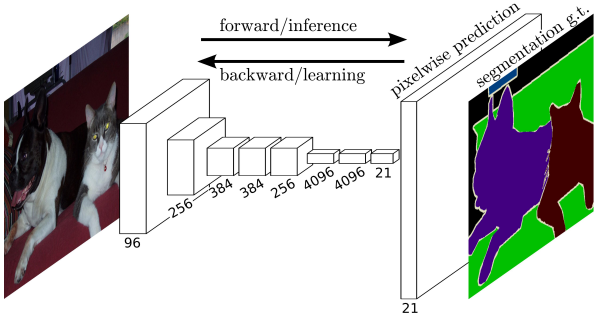


FIGURE 2.7 – Exemple de réseau entièrement convolutif [54]. La dernière couche de classification des pixels est une couche convolutive, contrairement aux CNN utilisés en classification.

- **Réseaux Convolutif avec modèle probabiliste** [11] : pour pallier le manque de contexte des CNN, il est possible de coupler ceux-ci à un modèle probabiliste comme les champs aléatoires conditionnels. On a donc une meilleure précision que les réseaux entièrement convolutifs grâce à l'information apportée par le contexte.
- **Encodeur-Décodeur** : comme son nom l'indique, ce type de réseau est composé de deux parties : l'encodeur et le décodeur. Du côté de l'encodeur, il s'agit d'un réseau convolutif, tiré des architectures existantes et performantes dans le domaine de la classification d'images. Un décodeur est ajouté avec pour but est de reconstruire le masque de segmentation [Figure 2.8]. Comme pour les réseaux entièrement convolutifs, ces méthodes ont l'avantage de s'appuyer sur des modèles performants, mais il est possible d'obtenir une perte de détails lors de l'encodage à cause des réductions successives de résolutions. Un exemple d'encodeur-décodeur est le réseau U-Net [78] et sa version 3D V-Net [59], que nous détaillerons dans les prochaines parties.

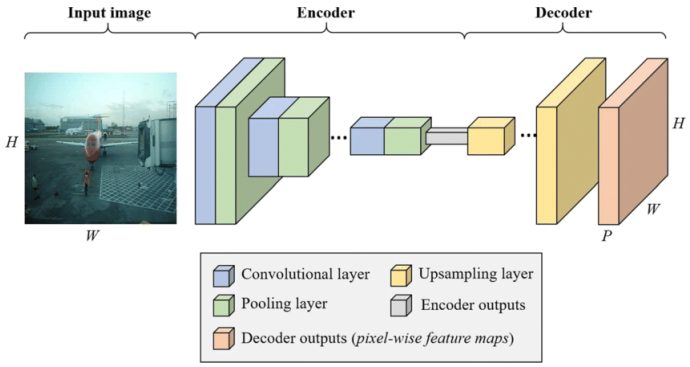


FIGURE 2.8 – Exemple d'encodeur-décodeur [83].

- **Réseaux de détection d'objets R-CNN** [52] : ce sont des architectures fonctionnant en deux étapes. Des réseaux de détections d'objets détectent tout d'abord les objets dans les images avec une boîte englobante, puis chaque boîte est ensuite segmentée pus précisément. Contrairement aux précédents réseaux, il y a trois sorties pour ce type de modèle : une boite englobante de l'objet détecté, sa classe ainsi que sa segmentation dans la boite englobante [Figure 2.9].
- **Modèles convolutif à trous** [14] : un taux de dilatation est introduit aux couches convolutives. Les filtres de convolutions peuvent alors s'appliquer sur des pixels non adjacents, augmentant le champ réceptif, sans augmenter le nombre de paramètres de la couche de convolution. Ce taux de dilatation peut être introduit dans chacune

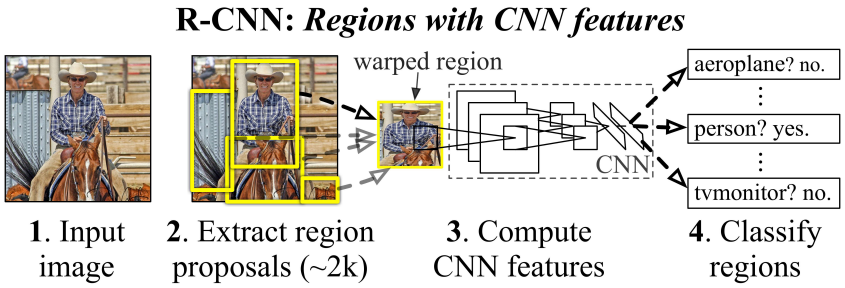


FIGURE 2.9 – Schéma de R-CNN [33]. Un module de segmentation sémantique donne le masque de segmentation du masque.

des méthodes citées ci-dessus, dès lors qu'une couche convolutive a été utilisée. Ce type de convolution sera abordé dans la suite du chapitre.

- **Modèles multiéchelle et pyramidaux** [32] : la sortie de plusieurs couches convolutives est utilisée à plusieurs échelles pour capter l'information localement, mais aussi l'information sur le contexte de l'image [Figure 2.10].

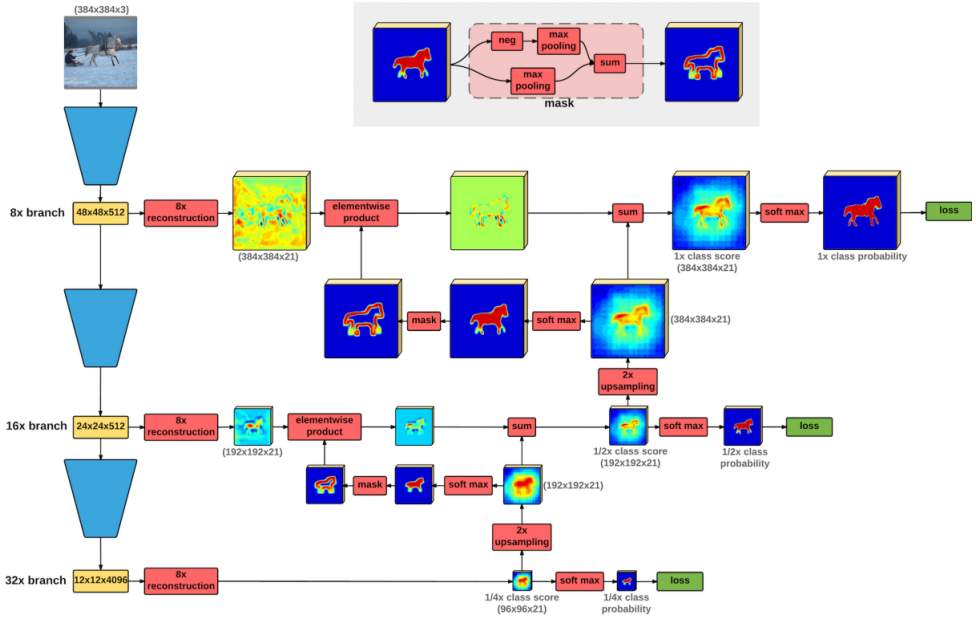


FIGURE 2.10 – Schéma d'un modèle multiéchelle [32]. Chaque étage correspond à une résolution différente.

- **Réseaux de neurones récurrents** [77] : ce type de réseau est utilisé pour modéliser des dépendances des pixels sur le court ou le long

terme. Initialement, ce type de réseau a été développé pour des applications sur des données temporelles comme des signaux ou des vidéos. La notion de court et long terme peut être adaptée à la segmentation d'image en utilisant la notion d'échelle des modèles pyramidaux, ou en 3D, par des notions de plans en fonction d'un axe. Les réseaux récurrents s'appuient sur des réseaux convolutifs pour parcourir les images et en extraire les caractéristiques.

- **Modèles basés sur l'attention** [13] : ces modèles utilisent le principe d'attention des réseaux de classification. Le mécanisme d'attention permet de guider le réseau sur les zones de l'image jugée importantes [Figure 2.11]. Ce principe est utilisé initialement dans des réseaux de traduction de phrases pour mettre en évidence sur les liens entre les mots. Cette technique peut également être adaptée pour la segmentation d'images.

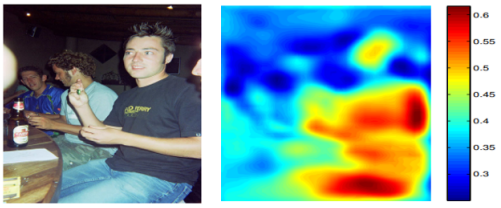


FIGURE 2.11 – Image (gauche) et la carte d'attention générée par un modèle d'attention (droite) [13].

- **Réseaux antagonistes génératifs** (GAN) [55] : ces réseaux possèdent deux sous-réseaux rentrant en compétition. Le réseau dit générateur crée un masque de segmentation et un réseau dit discriminateur essaie de déterminer si le masque provient du générateur ou est un masque réel. La structure du générateur peut être celle d'une méthode citée au précédemment, le but du discriminateur étant d'améliorer la précision du générateur [Figure 2.12].
- **CNN avec un modèle de contour actif** [43] [29] : Un modèle de contour actif modélise une courbe déformable pour épouser le contour des objets à segmenter [Figure 2.13]. Cette technique s'applique soit en post traitement, soit comme guide pour formuler une fonction de perte.
- **Transformeur visuel** (*Vision Transformer* ou ViT) [26] : récemment, les architectures de type transformeur ont permis d'énormes progrès dans le domaine du traitement automatique du langage naturel (*Natural Language Processing* ou NLP) [86]. Ces

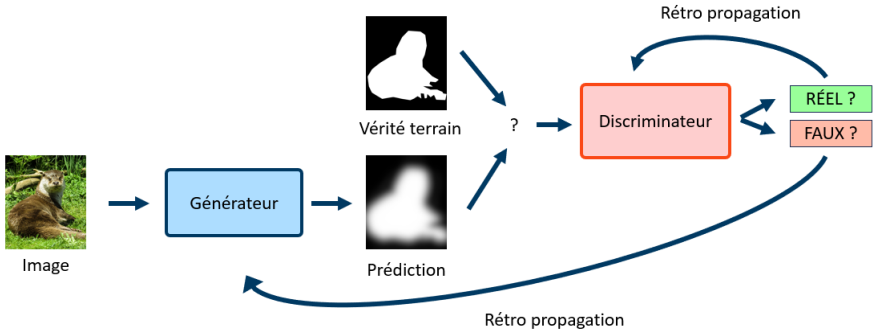


FIGURE 2.12 – Schéma de fonctionnement d'un réseau antagoniste génératif.

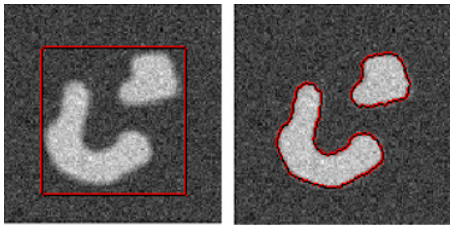


FIGURE 2.13 – Un post-traitement utilisant un modèle de contour actif pour la segmentation d'images médicales [29].

méthodes fonctionnent à l'aide d'un mécanisme d'attention. Cette couche d'attention fait office de mémoire et permet de traiter un nombre important de données séquentielles tout en gardant un contexte général. Un des avantages des transformeurs est leur capacité à être facilement parallélisable, permettant un entraînement avec un grand nombre de données. Les transformeurs visuels traitent une image découpée en de nombreux patchs sous forme de vecteur 1D encodés avec un transformeur classique [Figure 2.14]. Il est à noter que les transformeurs visuels sont une des rares méthodes de segmentation sémantique avec apprentissage profond qui ne possède pas de couche de convolution.

Pour conclure cette revue, beaucoup de ces méthodes s'appuient sur des encodeurs-décodeurs. En effet, ce sont ces types de réseau qui ont réussi les premiers à obtenir de bons résultats lors des challenges majeurs de segmentation sémantique tel que PASCAL VOC [28].

Dans la suite de ce chapitre, nous étudions le réseau U-Net qui est une ar-

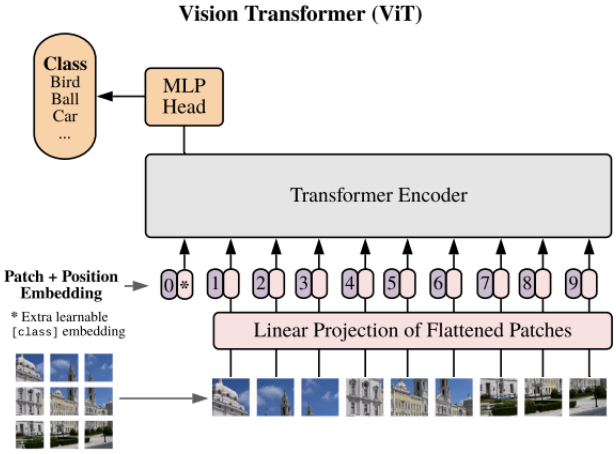


FIGURE 2.14 – Schéma d'un transformeur visuel où chaque image est découpée en patchs qui seront encodés avec un transformeur de traitement automatique du langage naturel [26].

chitecture type encodeur-décodeur. Ce réseau a été initialement mis au point pour la segmentation d'images médicales, où peu de données d'entraînement sont disponibles [78]. Également, dans notre application où l'aspect volumique est primordial, les méthodes de convolutions à trous seront abordées dans la suite de ce chapitre [11]. Ces méthodes permettent d'augmenter le champ réceptif tout en réduisant la charge en mémoire du réseau. Cette réduction peut alors permettre d'implémenter des approches de segmentation volumique, plus gourmandes en ressources que leurs contreparties ne segmentant que des images bidimensionnelles.

2.2.4 Réseaux de type Encodeur-Décodeur

Les réseaux de ce type sont les premiers à avoir été introduits en segmentation sémantique [31]. À titre d'exemple, nous allons ici nous focaliser sur le réseau U-Net qui est l'un des plus utilisés. L'article original d'U-Net est en passe de devenir l'article scientifique le plus cité, tous domaines confondus. U-Net [78] est un réseau classique de l'état de l'art en segmentation sémantique par approches d'apprentissage profond.

Comme indiqué précédemment, il fait partie des réseaux de type encodeur-décodeur, qui ont déjà fait leurs preuves sur des bases de données telles que PASCAL VOC [28], contenant 20 classes d'objets visuels dans une scène

réaliste [54]. La spécificité d'U-Net est d'avoir été conçu pour la segmentation d'images médicales, dont le volume de données disponibles n'est pas comparable aux jeux de données d'images plus classiques. Les exemples d'utilisation de U-Net montrent qu'il est possible apprendre une tache de segmentation avec très peu de données, grâce à des connexions entre l'encodeur et le décodeur qui permettent d'améliorer la précision du réseau en apportant de l'information haute définition lors du décodage [Figure 2.15].

Une innovation de U-Net est l'introduction de courts-circuit entre l'encodeur et le décodeur. Cette caractéristique permet de limiter le problème de diminution du gradient. Lors de la rétro propagation du gradient, plus le gradient remonte de couches, plus les erreurs d'approximations numériques dues au calcul discrétisé augmentent. Cette diminution de la valeur du gradient rend difficile d'entraîner les premières couches d'un réseau très profond. Ces courts-circuits entre encodeur et décodeur réduisent fortement ce phénomène.

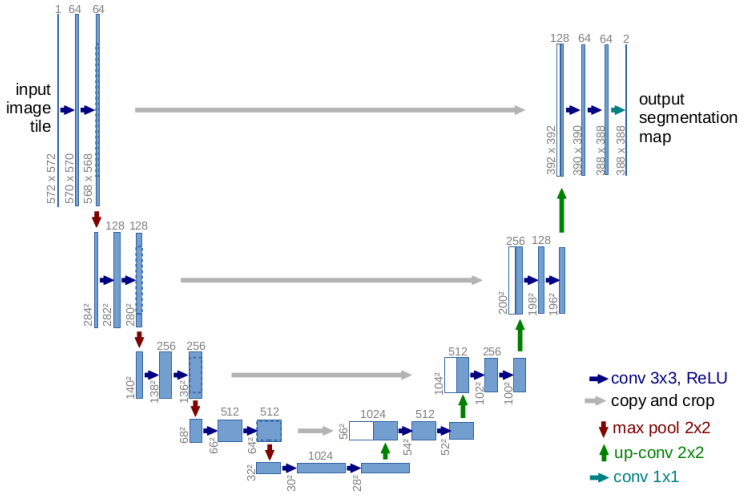


FIGURE 2.15 – Architecture d'U-Net. La première moitié de l'architecture est l'encodeur, la deuxième partie est le décodeur [78].

Encodeur. L'encodeur est construit avec plusieurs blocs composés de 2 couches de convolution avec un fitre 3x3, suivie d'une fonction d'activation de type ReLU (ReLU pour *Rectified Linear Unit*) puis d'une couche de max pooling 2*times*2. À chaque étape, la dimension des vecteurs descripteurs est doublée dû aux couches convolutives et la résolution est réduite de moitié avec la couche de max pooling. À la sortie de l'encodeur, une représentation de l'image d'entrée dans l'espace des descripteurs est obtenue.

Décodeur. Le décodeur est composé tout d'abord d'une augmentation de résolution avec une couche convolutive transposée (déconvolution ou *up-convolution*). En même temps, comme précédemment, le nombre de dimensions des vecteurs descripteurs est réduit d'un facteur deux à chaque augmentation d'échelle. Ici, à chaque couche du décodeur, le plan descripteur de la couche correspondante de l'encodeur est concaténée au résultat. Puis une convolution 3x3 est réappliquée. C'est une des spécificités d'U-Net qui améliore la reconstruction de la carte de segmentation en incluant des caractéristiques haute définition lors du décodage. À la sortie du décodeur, l'image obtenue est de même résolution que l'image d'entrée. Une couche convolutive 1x1 permet de réduire le nombre de dimensions des vecteurs descripteurs au nombre de classes souhaitées. Une fonction *soft-max* est ensuite calculée, permettant d'obtenir la probabilité d'appartenance de chaque pixel à chaque classe. Ce sont ces sorties qui sont utilisées pour le calcul de la fonction de perte, pour chaque pixel.

Entraînement. Les images et leur masque de segmentation sont utilisés pour l'entraînement. La fonction de perte utilisée, dans le cas d'une segmentation sémantique binaire, est une entropie croisée appliquée à chaque pixel x, puis la moyenne est calculée pour l'ensemble des pixels. Le réseau est représenté par une fonction F prédisant la probabilité p d'appartenir à une classe objet. Les données estimées par le réseau sont notées $p = F(x)$. La vérité terrain du pixel x est notée $\tilde{x}$ où $\tilde{x} = 1$ si le pixel est un pixel objet et $\tilde{x} = 0$ si le pixel appartient au fond. Ω correspond à l'ensemble des N pixels de l'image.

$$E(x) = -\frac{1}{N} \sum_{x \in \Omega} \tilde{x} \log p \tag{2.9}$$

Augmentation de données. L'augmentation de données est une étape importante lorsque l'on dispose de peu d'images. Elle consiste à augmenter le jeu d'entraînement en ajoutant numériquement des images et des masques obtenus, en transformant une image d'entraînement à partir de transformations. Comme exemple de transformations classiques, il est possible de citer des opérations géométriques de rotation, translation, changement d'échelle ou des opérations modifiant le niveau de gris de l'image. Elle permet d'augmenter artificiellement le set d'entraînement pour le rendre plus résilient aux transformations.

2.2.5 Convolution à trous

Les convolutions sont des opérateurs utilisés dans tous les CNN présentés précédemment. modifier cette brique permet d'avoir un impact considérable sur les résultats, quelle que soit la méthode choisie.

La convolution à trous est un outil permettant de contrôler explicitement la résolution et le champ réceptif des descripteurs grâce à l'introduction d'un taux de dilatation à la convolution classique. Le taux de dilatation r permet de contrôler le champ réceptif de la convolution. La convolution à trous est définie comme suit :

$$b[i] = \sum_k a[i + r.k].w[k] \tag{2.10}$$

Avec i la position dans le plan descripteur de sortie b, w le filtre de convolution et a le plan descripteur d'entrée.

Avec cet opérateur, il est possible d'obtenir un champ réceptif plus grand sans augmenter le nombre de poids des filtres. Le cas particulier de $r = 1$ permet d'obtenir une convolution classique [Figure 2.16].

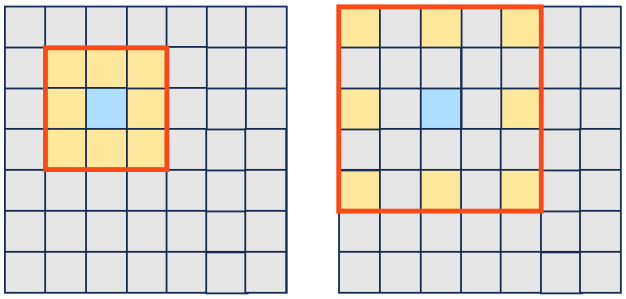


FIGURE 2.16 – Convolution classique à gauche et convolution à trous avec un taux de dilatation $r = 2$ à droite. Le champ réceptif est représenté en orange.

La convolution à trous est l'opérateur du réseau DeepLab [11]. Les auteurs de DeepLab utilisent une architecture multiéchelle : l'image d'entrée est utilisée dans plusieurs réseaux convolutifs à différents champs réceptifs pour à la fois capturer l'information locale et l'information sur le contexte. Un décodeur simple combine ensuite les résultats à plusieurs échelles pour obtenir la carte de segmentation. En utilisant ainsi les convolutions à trous, des convolutions

avec des grands champs réceptifs sont calculées sans augmenter le nombre de paramètres [Figure 2.17]. Les différentes versions de DeepLab (DeeplLabv3 [12], DeeplLabv3+ [14]) proposent des améliorations de l'intégration de la convolution à trous, notamment par l'utilisation de la convolution à trous dans les couches très profondes ou encore l'utilisation des convolutions à trous séparables.

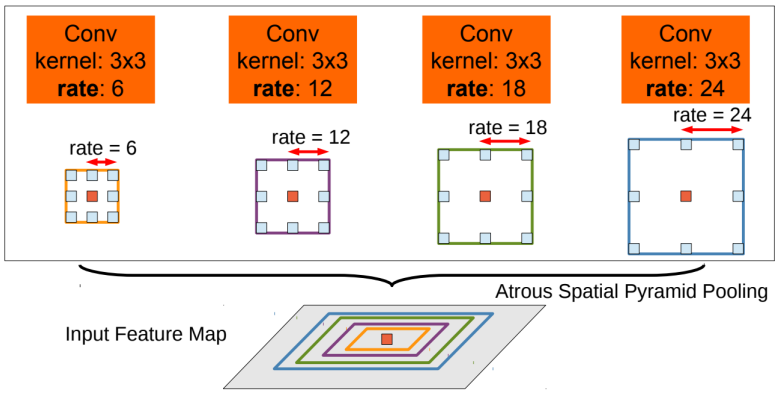


FIGURE 2.17 – Champ réceptif à plusieurs taux de dilatation [11]

Un autre avantage de cet opérateur est la réduction de la mémoire utilisée. Dans le cadre applicatif avec des données volumiques, les types de réseaux 3D possèdent plus de paramètres à optimiser, multipliant toutes les opérations par une nouvelle dimension. Cette convolution "creuse" est une solution pour gagner en performance et en utilisation de ressources. La partie suivante aborde les architectures intégrant la troisième dimension.

2.3 Segmentation 3D

Dans le cas de la segmentation 3D, un des plus grands défis à relever est la contention sur les ressources, calcul et mémoire, associées aux réseaux convolutifs. En effet, en ajoutant une dimension supplémentaire, les couches de convolutions comportent plus de paramètres. La table 2.1 compare le nombre de pixels d'une image d'un jeu de données populaire et d'un équivalent volumique hypothétique en ajoutant une dimension de taille égale à la plus petite dimension spatiale de l'image 2D. Ces réseaux s'appuient sur un grand nombre de filtres de convolution pour encoder l'information. Le stockage nécessaire pour chaque pixel en 2D, doit être effectué pour chaque voxel d'un volume en 3D. Il en est de même avec les filtres de convolutions. Pour une convolution 2D avec un filtre 3×3, il y a 9 poids à entraîner. Comparative-

Jeu de données	Taille des images	Nombre de pixels	Voxels si volume
MNIST [23]	28 × 28	784	21 952
CIFAR-10 [42]	32 × 32	1 024	32 768
PASCAL VOC [28]	480 × 364	174720	63 598 080
COCO [48]	640 × 480	307 200	147 456 000
DIV2K [1]	2560 × 1440	3 686 400	5 308 416 000

TABLE 2.1 – Table comparant la différence de quantité de pixels entre un jeu de données d'images bidimensionnel et la quantité de voxels de ce même jeu de données avec une hypothétique troisième dimension.

ment, une convolution 3D avec un filtre $3 \times 3 \times 3$, nécessite d'entraîner 27 poids. Il faut donc des méthodes pour traiter cette quantité de données de manière efficace, sous peine de manquer de ressources.

2.3.1 Modèles 3D

Il est intéressant d'étudier une architecture comme U-Net se comporte avec des convolutions 3D, et si l'ajout de la 3^{e}dimension a un impact significatif.

V-Net [59] est l'extension en 3D du réseau U-Net vu précédemment. La structure de V-Net est similaire à celle de U-Net [Figure 2.18] :

- **Couches convolutives** : les convolutions sont maintenant des convolutions 3D avec un noyau de convolution $5 \times 5 \times 5$.
- **Entropie** : dans le cas de l'entropie croisée [Formule 3.1], le nombre de voxels dans chaque classe induit un biais sur la fonction de perte, ce qui peut provoquer des erreurs de segmentation dans le cas où l'objet est faiblement représenté. C'est généralement un problème pour les volumes 3D d'imagerie médicale où les ratios des nombres de voxels entre classes peuvent être très importants. Un poids pour contrôler l'influence de chaque classe en fonction de leur nombre peut être introduit. Cette nouvelle fonction de perte est appelée entropie croisée pondérée. Cependant, le ratio des populations entre les voxels de la classe objet et les voxels de la classe fond doit être préalablement connu.
- **Fonction de perte** : les auteurs de V-Net introduisent une fonction de perte basée sur le coefficient de Sørensen-Dice, où A est le volume prédit et B la vérité terrain :

$$D(A, B) = 1 - \frac{2|A \cap B|}{|A| + |B|} \tag{2.11}$$

Avec |.| l'opérateur de cardinalité.

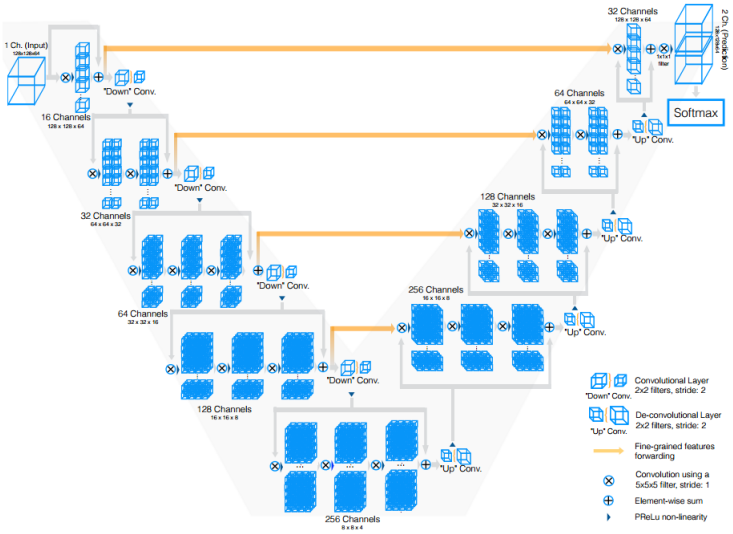


FIGURE 2.18 – Architecture de V-Net, analogue à celle de U-Net, mais avec des convolutions 3D [59].

V-Net parvient ainsi à obtenir de très bons résultats lors du challenge PROMISE2012 [50] en utilisant 50 IRM de prostates pour l'entraînement.

2.3.2 Modèle 2D par plan et modèle 3D

Dans les parties précédentes, des exemples d'architecture réseau 2D et 3D ont été présentés. Dans le cas de la segmentation volumique, il est possible d'avoir recours à un modèle 2D même si les données sont volumiques. En effet, un volume de données 3D peut être vu comme une pile d'images en 2D. Dans ce cas, il est possible d'appliquer un modèle 2D sur chaque plan de manière indépendante. Cette procédure a l'avantage d'éviter le problème de mémoire des approches 3D. Cependant, la prise en compte de la continuité des plans est perdue [Figue 2.19]. On peut alors se demander quelle approche est la plus efficace.

Une telle comparaison a déjà été réalisée [39]. Un réseau de type U-Net en 2D et 3D ont été mis à l'épreuve sur des volumes d'imagerie médicale (foie, rein, rate, pancréas). Les deux types de réseaux ont ainsi été entraînés sur 100 époques sur la même base d'apprentissage. La fonction entropie croisée pondérée est utilisée pour les deux architectures. Les deux modèles sont évalués sur les mêmes 16 volumes, avec une étape de cross-validation, c'est-à-dire que les images sont différentes entre chaque expérimentation, mais iden-

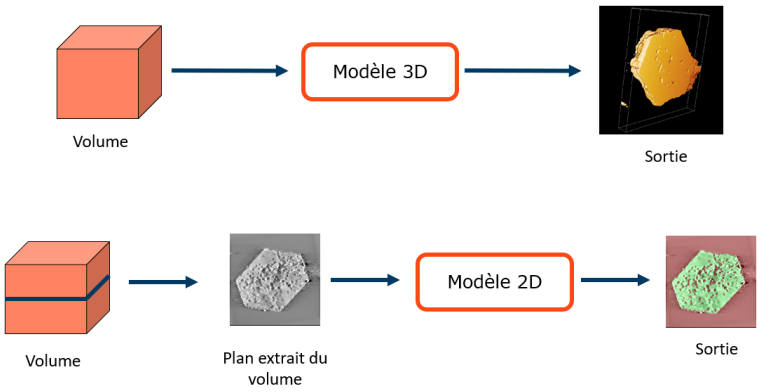


FIGURE 2.19 – Différences entre modèle 3D (haut) et modèle 2D plan par plan (bas).

tiques entre la version 2D et 3D. L'expérimentation est répétée 80 fois pour chaque organe. Les résultats [Figue 2.20] montrent que l'approche U-Net 2D est bien plus performante que la variante 3D sur la plupart des organes testés (foie, rate et reins). Pour le pancréas, où les images ont un faible contraste, U-Net 3D obtient cependant des résultats légèrement meilleurs.

DSCs:	**Mean±Std.**		**Median~IQR**	
Organ	2D U-Net	3D U-Net	2D U-Net	3D U-Net
Liver	**0.94±0.03***	0.93±0.04	**0.95~0.02****	0.94~0.03
R. kidney	**0.91±0.05*****	0.89±0.05	**0.92~0.03*****	0.90~0.05
L. kidney	**0.92±0.05*****	0.86±0.14	**0.93~0.03*****	0.89~0.08
Spleen	**0.93±0.04**	0.92±0.04	**0.94~0.03**	0.93~0.03
Pancreas	0.57±0.19	**0.59±0.15**	0.60~0.21	**0.61~0.21**

FIGURE 2.20 – Résultats de la comparaison entre U-Net 2D et 3D. U-Net 2D surclasse la version 3D dans la plupart des cas [39].

Ces travaux s'intéressent également à l'utilisation de ressources de calcul et de mémoire et montrent que l'approche 2D consomme bien moins de mémoire et est également plus rapide que sa contrepartie 3D [Figure 2.21].

Pour conclure, les approches 3D, en plus de demander plus de ressources à cause de leur structure, sont dans le meilleur des cas présentés aussi performant que les approches "plan par plan". Pour la suite de ces travaux, nous nous intéresserons exclusivement à des méthodes 2D, plan par plan.

GPU-Performance:	**Memory**				**Time**			
	Training [MiB]		**Application [MiB]**		**Training [min:sec]**		**Application [sec]**	
Organ	2D U-Net	3D U-Net	2D U-Net	3D U-Net	2D U-Net	3D U-Net	2D U-Net	3D U-Net
Liver	1693	10957	1693	9117	9:26	10:07	1.47	3.18
R. kidney	1693	10957	1693	9117	9:28	10:07	0.40	0.55
L. kidney	1693	10957	1693	9117	9:26	10:07	0.40	0.55
Spleen	1693	10957	1693	9117	9:27	10:08	0.40	0.55
Pancreas	1693	10957	1693	9117	9:28	10:08	0.40	0.55
Mean±Std.	**1693±0*****	10957±0	**1693±0*****	9117±0	**9:27±0:01*****	10:07±0:01	**0.61±0.43**	1.07±1.05
Median~IQR	**1693~0*****	10957~0	**1693~0*****	9117~0	**9:27~0:01**$^{+}$	10:07~0:01	**0.40~0.003**$^{+}$	0.55~0.001
Mean Improvement	**647.19%**	N/A	**538.51%**	N/A	**107.13%**	N/A	**175.22%**	N/A
Median Improvement	**647.19%**	N/A	**538.51%**	N/A	**107.07%**	N/A	**137.10%**	N/A

FIGURE 2.21 – Performance d'U-Net 2D et 3D. U-Net 2D est plus efficace en utilisation de la mémoire du GPU et est également plus rapide qu'U-Net 3D [39].

2.4 Segmentation avec peu de données annotées

Dans le cas de méthodes classiques par apprentissage profond, il est important d'avoir un nombre conséquent de données annotées qui seront utilisées pendant la phase d'entrainement. Les données de matériaux issues de tomographie électronique ne sont pas courantes dans la littérature. Comparativement aux 250 000 pixels d'une image 500×500 à annoter, il y a 125 000 000 de voxels à annoter pour un volume 500×500×500. De plus, la nature complexe des images nécessite de faire appel à un expert. Il est ainsi fastidieux et coûteux en temps d'annoter une telle quantité de données. Cette partie met en avant les méthodes utilisées dans la littérature pour entrainer un réseau de neurones profond avec peu de données annotées.

2.4.1 Annotations partielles

Cette première approche consiste à exploiter des données qui n'ont pas été complètement annotées. En demandant de ne labelliser qu'une infime partie des données disponibles, la charge de travail pour l'annotateur est grandement réduite. Dans cette configuration, deux cas de figure peuvent nous intéresser [Figure 2.22] :

- **Segmentation semi-automatique** : l'utilisateur segmente une partie du volume (quelques plans) qui sera utilisée pour l'entraînement, puis la totalité du volume est passée dans le réseau afin d'obtenir une carte de segmentation du volume entier.
- **Segmentation automatique** : le réseau de neurones est entrainé avec plusieurs plans provenant de plusieurs autres volumes. La méthode est ainsi capable de segmenter de nouveaux volumes sans

nouvel apport de connaissance.

Pour ces deux approches, il existe déjà dans la littérature des exemples concluants.

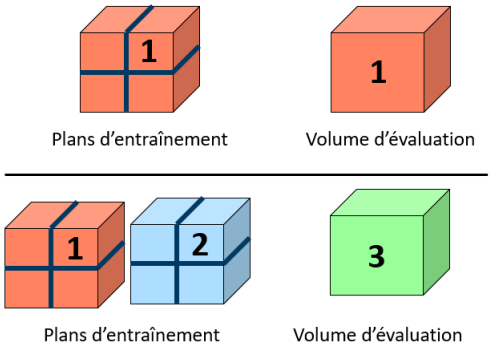


FIGURE 2.22 – Configuration pour la segmentation semi-automatique (haut) et configuration pour segmentation automatique (bas).

Dans [20], l'architecture est V-Net et trois volumes de cellules embryonnaires de rein d'amphibien (Xenopus) sont utilisés. Pour chaque volume, une image est labellisée pour chaque plan xy, xz et yz. Il y a donc trois images pour l'entraînement pour chaque volume. Les données non annotées sont prises en compte par l'introduction d'une nouvelle classe "non labellisé". Trois classes différentes sont utilisées : "non labellisé", "objet" et "fond" [Figure 2.23]. La fonction de perte employée est l'entropie croisée pondérée où le poids dépend de chaque classe. Le poids pour les classes "objet" et "fond" est fixé à 1 et le poids de la classe non labellisé à 0. Les voxels de la classe "non labellisée" ne contribuent ainsi plus à la fonction de perte, le réseau se focalisant uniquement sur les voxels des classes "objet" et "fond", comme lors de l'entraînement classique d'un réseau U-Net. Le réseau peut ainsi prendre en charge des volumes non annotée en entrée.

Cette approche présente une solution potentielle à notre problématique générale. En effet, la reconstruction de volume à partir de quelques plans annotés est un problème intéressant à étudier dans le cas où l'on veut minimiser le temps d'annotation. Nous verrons dans la partie suivante que certains travaux vont encore plus loin en se passant complétement de l'étape d'annotation.

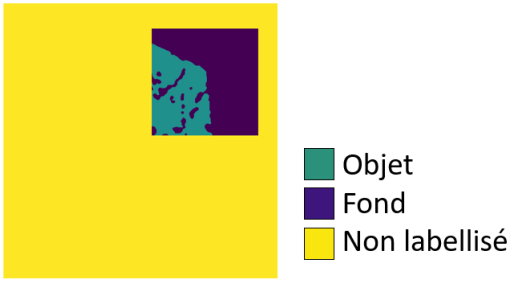


FIGURE 2.23 – Exemple d'image partiellement annotée (2D).

2.4.2 Few shot Learning

Le *few shot learning* est une autre approche de la segmentation avec peu de données labellisées. Le but est d'entraîner un encodeur avec peu d'exemples de plusieurs objets différents, avec pour objectif d'avoir appris l'acte d'apprendre. Avec peu de données, cette méthode est inspirée de la manière dont les enfants apprennent à reconnaître des objets. Par exemple, lorsqu'un enfant n'a jamais vu un animal, il saura reconnaître ses principales caractéristiques, telles que le nombre de pattes, la forme des oreilles, le type de pelage, du fait de son expérience avec d'autres espèces animales. Un adulte va alors lui donner le nom de l'animal. L'enfant pourra ainsi reconnaître d'autres individus de la même espèce, en ayant comme base d'apprentissage un seul exemple [22, 76]. L'objectif recherché est d'apprendre au réseau à répliquer ce comportement, en apprenant tout d'abord sur une petite base de données, appelée base support, constitué de N classes comportant un petit nombre K d'exemples chacune (typiquement $K < 10$). Pendant la phase de test, le réseau devra prédire la classe d'images requêtes qu'il n'a jamais vues [Figure 2.24].

Le *Few Shot Learning* peut être appliqué au cas de la segmentation. On parle de *Few Shot Segmentation*. Pour une tâche de segmentation, il y a ainsi deux entrées : un support et une requête. Le support est composé de paires image et de sa carte de segmentation correspondante. La requête correspond à l'image à segmenter. La sortie est le masque de segmentation de l'image requête. Le réseau va donc utiliser l'information dans le support pour segmenter la requête. Par exemple, considérons une image avec un chat et un chien. C'est l'image requête. Si le support est composé d'une image de chien et de son masque de segmentation, la sortie attendu est le masque de segmentation du chien dans l'image requête. Si le support est composé d'une image de chat, la sortie attendu est le masque de segmentation du chat. Le support

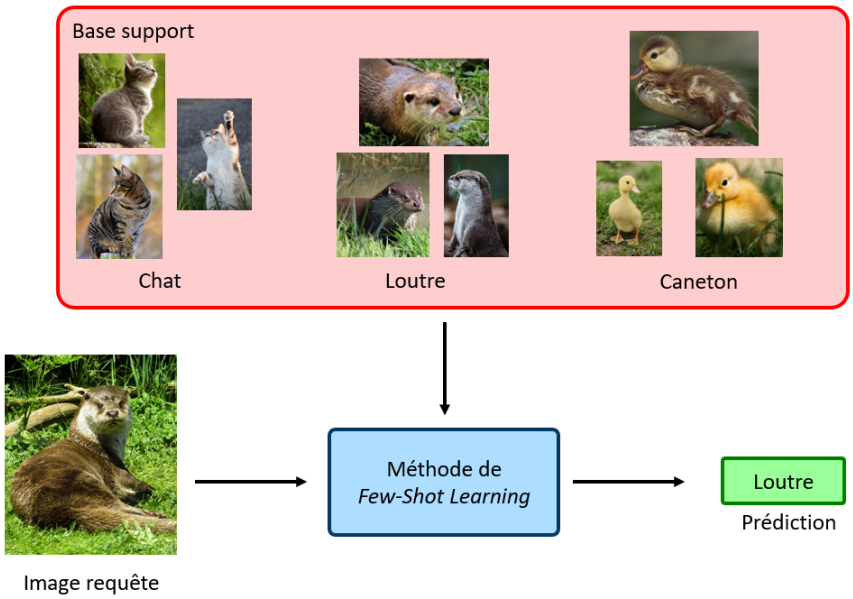


FIGURE 2.24 – Principe du *few shot learning*.

donne ainsi la tâche de segmentation à effectuer. L'objectif principal est de construire un réseau suffisamment généraliste pour extraire les informations générales de l'image, puis de fournir une requête au réseau pour le guider dans la segmentation des images de classes inconnues du réseau [Figure 2.25].

Pour effectuer une tâche de *Few Shot Segmentation*, il est possible d'utiliser un réseau guidé [75]. Le réseau guidé est composé d'un encodeur-décodeur classique, avec en entrée la requête. L'image du support et ses annotations sont encodées par un réseau convolutif. Le plan descripteur résultant est utilisé pour le guidage du décodeur en le fusionnant au plan descripteur de l'encodeur [Figure 2.26]. Il est possible de paramétrer de plusieurs manières ce type de réseau, en modifiant par exemple la façon d'intégrer l'image du support aux annotations.

2.4.3 Approches auto-supervisées

Il existe des méthodes n'utilisant aucune de labellisation. Sans données annotées, l'apprentissage ne peut s'effectuer qu'avec l'image d'entrée. Plusieurs pistes sont envisageables. Par exemple, le contexte est utilisé pour apprendre des représentations visuelles [25].

Dans ces travaux, une tâche prétexte est définie. Elle consiste par exemple

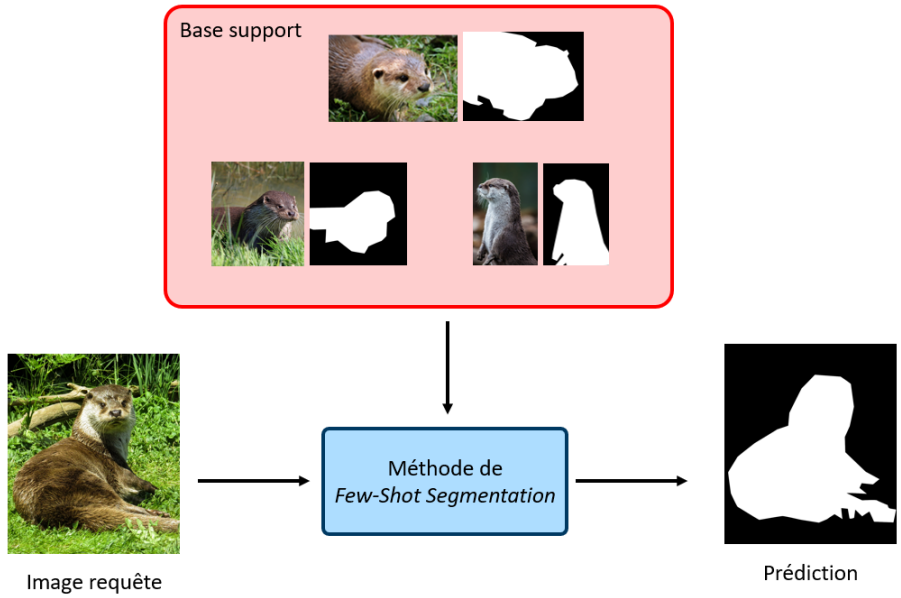


FIGURE 2.25 – Principe du *few shot segmentation*.

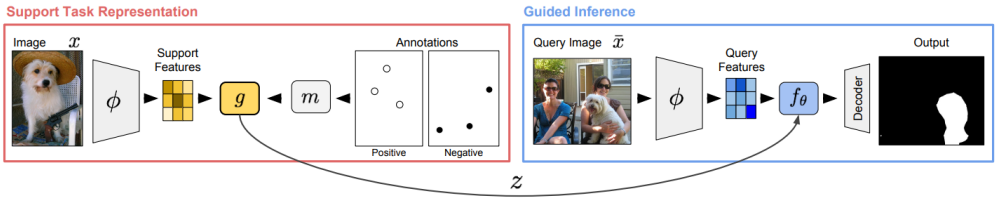


FIGURE 2.26 – Exemple d'un réseau guidé où le support est encodé puis utilisé pour guider l'encodeur-décodeur [75].

à prédire le positionnement relatif de plusieurs patchs d'images. Avec une image d'entraînement en entrée, deux patchs sont extraits à des endroits aléatoires de l'image. La tâche à accomplir est de prédire le positionnement du premier patch par rapport au second patch, c'est-à-dire si le premier patch est à droite, à gauche, plus haut ou plus bas dans l'image que le second patch [Figure 2.28]. L'intuition derrière cette méthodologie est qu'il est plus facile d'effectuer cette tâche lorsque l'on a reconnu le contenu de l'image. En entraînant un réseau qui a de bons résultats sur cet exercice, celui-ci sera capable d'apprendre de bonnes représentations visuelles.

L'architecture utilisée est un réseau convolutif pour la représentation visuelle, puis une couche complètement connectée pour la prédiction de la position

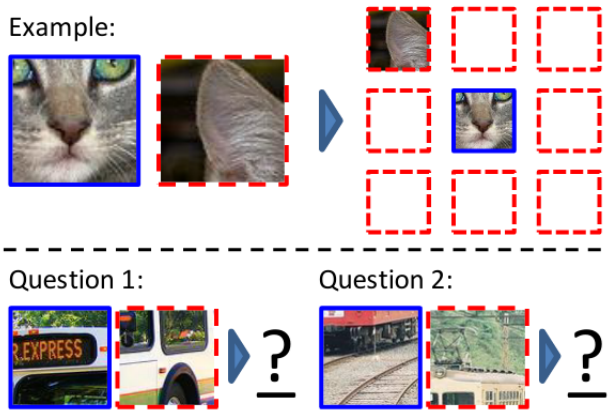


FIGURE 2.27 – La tâche à réaliser dans le cadre de l'utilisation de contexte pour apprendre des représentations visuelles. Le but est de prédire où le patch rouge se situe par rapport au patch bleu. En essayant de répondre aux questions 1 et 2, on remarquera que la tâche est bien plus facile dès que l'on reconnaît l'objet en question [25].

relative. Les deux patchs sont passés dans le réseau convolutif puis, les deux sorties sont mises en entrée de la couche complètement connectée. Le résultat est comparé à la vérité terrain qui est connue même sans annotations. Lorsque ce réseau est utilisé pour reconnaître des objets, seul le réseau convolutif est gardé.

Une autre approche consiste à cacher une partie d'une image. La tâche prétexte est de reconstruire la partie manquante de l'image [68, 24, 6]. L'image prédite est ensuite comparée à l'image originale. De même que la prédiction du contexte, il est plus simple de remplir la section manquante si le contenu de l'image est reconnu. En apprenant à reconstruire les pixels manquants, le réseau apprend ainsi une bonne représentation visuelle.

Dans [68], un réseau antagoniste génératif (*Generative Adversarial Network* ou GAN) est utilisé. Pour générer le patch manquant, un encodeur-décodeur est entraîné avec une fonction de perte antagoniste. Cette fonction de perte est calculée grâce à un deuxième réseau, le discriminateur. Le but de ce discriminateur est de reconnaître si le patch en entrée est la vérité terrain ou un patch généré par l'encodeur décodeur. Une fois appris, pour effectuer la tâche de classification, la partie décodeur est remplacée par une couche de classification.

FIGURE 2.28 – La tâche à réaliser est de remplir une image partiellement masquée (gauche). Si le résultat est probant (droite), le réseau aura appris une représentation visuelle correcte [68].

2.4.4 Apprentissage contrastif

Le *Contrastive Learning* est une méthode d'apprentissage d'identification d'objets similaires. Le principe est de se servir d'un réseau profond pour projeter les données d'entrées dans un espace latent de plus petite dimension en imposant que les données similaires seront proches dans l'espace latent et les données différentes seront éloignées. Ce module est appelé projecteur. On utilise ainsi une fonction de perte dite "contrastive" opérant sur des paires d'exemples soit positives si les exemples sont similaires, soit négatives s'ils sont différents. La fonction de perte sera minimale soit lorsque les exemples d'une paire positive sont proches, soit lorsque les exemples d'une paire négative sont éloignés [Figure 2.29]. Comme les deux exemples d'une paire sont transformés par le même réseau, qui possède donc deux entrées, celui-ci est qualifié de réseau siamois. L'avantage de cette méthode est que l'on peut construire des paires positives sans connaître le moindre label. De plus, lorsque cette information de classe est disponible, les éléments de la paire peuvent être choisis avec soin pour réduire le nombre de données d'entraînement.

FIGURE 2.29 – Exemple de paire positive et de paire négative et leur influence sur la fonction de perte contrastive.

- **Approche auto-supervisée** : pour une approche auto-supervisée du *Contrastive Learning*, l'information sur les classes des images n'est pas disponible. Des paires positives et négatives peuvent tout de même être construites. Une paire positive est constituée de deux images ayant subi chacune une transformation différente [Figure 2.30]. Les deux images étant issues de la même image originale, elles possèdent la même classe sémantique. Pour une paire négative, deux images différentes de la base d'entraînement forment la paire. La fonction de perte contrastive permet d'entrainer un projecteur sans posséder les annotations de la base de données. Pour effectuer une tâche de segmentation, un module de classification est concaténé à la sortie du projecteur. Classiquement, une machine à vecteurs de support (*Support-Vector Machine* ou SVM) est entraînée de manière supervisée. Cependant, les SVM nécessite bien moins de données annotées. Il est alors possible d'entraîner une approche auto-supervisée sur une très large banque d'images, où l'annotation n'est pas nécessaire. Le classifieur est ensuite entraîné avec un nombre plus réduit de données labellisées. Plusieurs méthodes de la littérature ont mis en place cette architecture, étudiant notamment différentes fonctions de similarité, la présence de projecteur, ou bien des modifications sur la propagation du gradient [15, 34, 17].

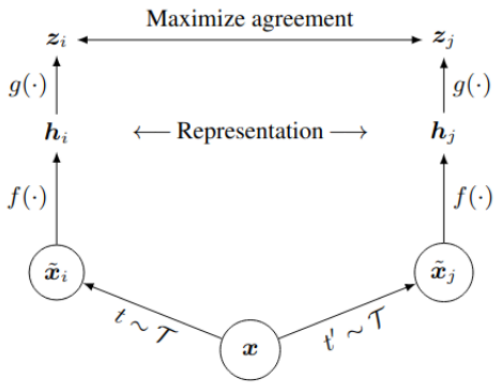


FIGURE 2.30 – Schéma d'un réseau siamois. x correspond à l'image d'entrée modifiée par deux transformations aléatoires $\mathcal{T}$. $\tilde{x_i}$ et $\tilde{x_j}$ forment ainsi une paire positive, qui passe dans deux encodeurs f qui partagent les mêmes poids. $f(\tilde{x_i})$ et $f(\tilde{x_j})$ sont ensuite projetés dans un espace dans lequel il est possible de les comparer grâce à la fonction de similarité [15].

- **Approche supervisée** : dans une approche auto-supervisée, lors de la création de paires négatives, deux images de la même classe peuvent

être sélectionnées puisque l'information de classes n'est pas connue. Les méthodes auto-supervisées comptent sur le nombre d'exemples présent dans la base de données d'entrainement pour minimiser ce problème, contraignant cette approche à un nombre important de données. Cependant, cette erreur de catégorisation contribue tout même à une perte de performance. L'apport de l'information de la classe peut corriger cette erreur [Figure 2.31]. De travaux récents [40] introduisent de nouvelles fonctions de perte et prennent mieux en compte ces erreurs potentielles dans la création des paires négatives.

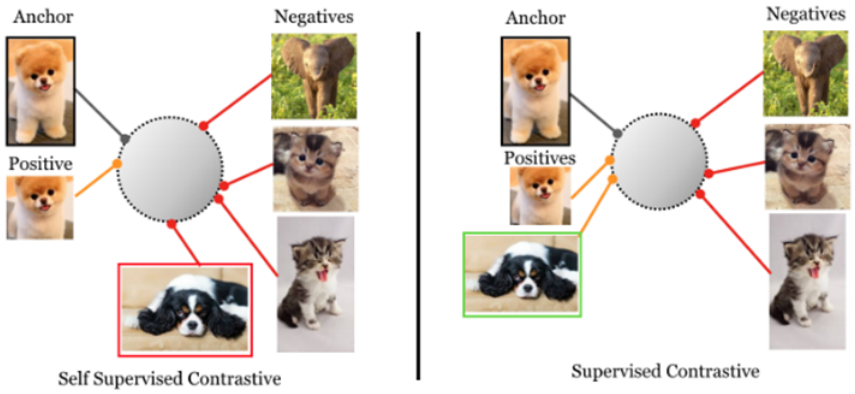


FIGURE 2.31 – À gauche, l'approche auto-supervisée et à droite, l'approche supervisée. À gauche, l'image du chien encadré en rouge est incorrectement catégorisée en paire négative avec l'image de chien encadré en noir. Avec l'information de la classe, il n'est plus possible de produire cette erreur.

Les approches contrastives sont intéressantes parce qu'elles exploitent au mieux les données disponibles, qu'elles soient annotées ou non. Pour notre problématique, comme la quantité de données est limitée, il est intéressant de pouvoir exploiter toutes les données disponibles.

2.5 Segmentation d'objet vidéo

Récemment, des approches de segmentation d'objet vidéo mettent en avant de nouvelles méthodes permettant de combler le manque d'annotations. En effet, dans le domaine de la segmentation vidéo, il est également coûteux d'annoter manuellement une vidéo entière. La tâche est fastidieuse et l'énorme quantité de données disponible rend difficile l'apprentissage de méthode de segmentation de vidéo. Une approche intéressante pour notre domaine d'application est la segmentation interactive. Par ce type d'approche, un an-

notateur guide un réseau de neurone profond afin de segmenter une vidéo à l'aide de quelques clics. De telles performances sont possibles grâce aux réseaux à mémoire, conçu pour propager un masque de segmentation à travers toute une vidéo. À l'aide d'un module de mémoire, une référence de l'objet à segmenter est sauvegardée puis mise à jour au fil de la progression de la segmentation de la vidéo.

Ces méthodes issues du domaine de la segmentation vidéo peuvent faire l'objet d'une adaptation pour la segmentation de volume. En effet, vidéos et volumes sont analogues, puisqu'elles sont composées d'une succession d'image 2D : une image pour la vidéo et un plan pour un volume.

2.5.1 Segmentation interactive

Les approches de segmentation interactive permettent de segmenter une image de manière itérative grâce à une communication avec un utilisateur. La segmentation interactive a pour objectif de segmenter en quelques clics une image. La procédure standard des méthodes interactives est la suivante :

1. L'annotateur indique par quelques clics l'objet à segmenter. L'annotateur fournit des clics positifs pour des zones à inclure ainsi que des clics négatifs pour des zones à exclure de la segmentation.
2. Le réseau fournit une proposition de segmentation.
3. L'annotateur propose des corrections, toujours à l'aide de clics positifs et négatifs. L'annotateur peut ainsi, soit corriger à nouveau la proposition ou bien terminer la procédure si la segmentation est satisfaisante.

Les réseaux de segmentation interactive doivent apprendre, en plus des représentations visuelles, les différentes interactions de l'utilisateur [Figure 2.33].

L'architecture prévalente dans la littérature [56, 51] est celle d'un encodeur-décodeur classique.

- **Architecture** : les interactions sont ajoutées dans l'image d'entrée comme un quatrième canal avec les canaux de couleurs rouge, bleu et vert. L'ajout de ce quatrième canal ne modifie pas l'architecture et le réseau se comporte comme un encodeur-décodeur traditionnel. Il est donc possible de commencer l'entrainement à partir de poids d'un réseau pré-entrainé.

FIGURE 2.32 – Différentes étapes d'une étape de segmentation interactive. Tout d'abord, une prédiction de segmentation est produite. L'annotateur indique des corrections, qui seront effectuées par la méthode de segmentation interactive [16].

- **Apprentissage des interactions** : pendant la phase d'entrainement, les interactions sont simulées, afin de donner aux réseaux des exemples de combinaison de segmentations et d'interactions. Il existe de nombreuses manières de simuler ces interactions. [56] recommande un entrainement itératif. À l'itération 0, des points positifs sont aléatoirement répartis sur l'objet à segmenter et des points négatifs dans les zones en dehors de l'objet. Puis grâce au masque obtenu, de nouveaux clics sont ajoutés automatiquement dans la zone la plus incorrectement annotée. Vingt itérations sont effectuées.

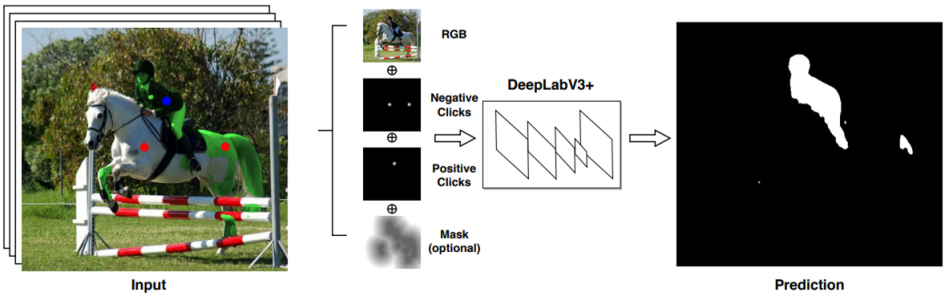


FIGURE 2.33 – Les clics de l'annotateur indiquent les parties à inclure (clic positif en bleu) et les parties à exclure (clics négatifs). Ces informations sont concaténées à l'image d'entrée avec les différents canaux RGB. Le réseau fournit une carte de segmentation prenant en compte l'image et les instructions de l'annotateur [56].

D'autres approches basées sur cette méthode ont été développées, comme un raffinement local à l'aide de fenêtre [49]. Une application de la segmentation interactive a été effectuée sur des images de cellules [41]. Dans [7], les comportements des annotateurs ont été étudiés. Les auteurs de cette étude ont demandé à un grand nombre d'annotateurs de segmenter des images à

l'aide d'un réseau de segmentation interactif. L'analyse des comportements des annotateurs ainsi que les performances de la segmentation ont fourni des recommandations pour de nombreux hyper-paramètres comme le nombre recommandé de clics par itérations ainsi que la manière d'encoder les clics.

2.5.2 Segmentation d'objet semi-supervisée

Pour segmenter une série d'images comme une vidéo, il est possible de segmenter une première image puis de propager la segmentation sur la totalité des images [5, 62, 85]. Cette tâche qui a été introduite dans le domaine de la vidéo correspond à de la segmentation d'objet semi-supervisée. En effet, dans une vidéo, le contenu des images adjacentes sont proches et cette approche permet d'exploiter les connaissances déjà acquises. Un réseau à mémoire (*memory network*) [64] s'appuie sur le fait qu'un objet est similaire sur des images proches, mais devient de plus en plus différent au fur et à mesure que la vidéo avance dans le temps. Les réseaux à mémoire propagent la segmentation initiale sur les images voisines et le résultat est stocké en mémoire pour aider à segmenter les images plus éloignées de l'image de référence. Ces méthodes fonctionnent par clé et valeur pour stocker les informations utiles à la segmentation dans la mémoire.

Les réseaux à mémoires sont composés de deux encodeurs et d'un décodeur.

- **Encodeurs** : Un encodeur est dédié aux éléments de la mémoire tandis que l'autre encodeur, à l'image à segmenter. Les encodeurs fournissent un vecteur clé et un vecteur valeur.
- **Mémoire** : Pour garder une image dans la mémoire, le vecteur clé et le vecteur valeur de l'image sont stockés en mémoire. Lorsqu'une nouvelle image doit être segmentée, la clé de l'image est comparée aux clés de la mémoire. Un vecteur valeur est obtenue en fonction de la similarité avec les clés de la mémoire.
- **Décodeur** Le vecteur valeur obtenu avec la mémoire est passé au décodeur qui fourni une carte de segmentation dépendant de la nouvelle image et des informations stockées en mémoire

De récents travaux [18] proposent une approche modulaire : un module interactif pour la segmentation de la première image, puis un module de propagation. L'annotateur peut segmenter rapidement une première image grâce au module de segmentation interactive. Le module de propagation permet de segmenter la vidéo entière. De plus, l'annotateur peut effectuer des modifications supplémentaires si une partie de la vidéo a été mal segmentée. Ce principe sera détaillé dans le chapitre 5.

Ce type d'approche est intéressant pour aider les annotateurs dans leur travail en exploitant tout le travail effectué pour la segmentation interactive. Cependant, le peu de données disponibles dans le cas de la segmentation volumique de matériaux peut être limitant, comme c'est le cas dans la segmentation d'objets non spécialisés.

2.6 Récapitulatif

Cet aperçu rapide du domaine de la segmentation illustre quelques pistes d'intérêt :

- Une revue des méthodes de segmentation classiques a été effectuée. Les réseaux de neurones convolutifs sont largement utilisés dans la littérature. En particulier, les approches de type encodeur-décodeur ont été appliquées dans de nombreux domaines. Un exemple de ce type d'architecture est U-Net. Un défaut des approches convolutives est le champ réceptif restreint, ce qui a pour conséquence de limiter la perception en détail du contexte de l'image. Pour augmenter le champ réceptif sans augmenter le nombre de paramètres à apprendre, les couches de convolutions peuvent être remplacées par des couches de convolutions à trous.
- Un des obstacles à la conversion des méthodes de segmentation d'images à des méthodes de segmentation de volumes est l'explosion de la taille du réseau pour des images volumiques. Des méthodes ont néanmoins été développées, comme V-Net qui est une extension de U-Net avec des couches de convolution 3D. Cependant, des travaux ont montré que des approches 2D plan par plan sont moins consommatrices de ressources en plus d'être plus performantes que des méthodes entièrement 3D.
- Un des problèmes en lien avec la segmentation de données spécifiques comme des images issues de tomographie électronique est le manque de données annotées. Des méthodes de la littérature permettent néanmoins d'optimiser un modèle profond, même avec des données annotées limitées en nombre. Avec l'annotation partielle, il est possible de tout simplement fournir à un réseau des données qui n'ont pas été entièrement annotées. D'autres méthodes changent le paradigme classique d'apprentissage d'une approche par apprentissage profond, comme le *few shot learning*, ou l'apprentissage contrastif. Enfin, il existe également des méthodes d'apprentissage auto-supervisé qui ne nécessite aucune données annotées.

- Dans le domaine de la segmentation d'objet vidéo, il est également nécessaire d'annoter une quantité importante de données. Une des façons de procéder est l'utilisation de méthodes semi-supervisées et de méthodes interactives. Un annotateur peut guider un réseau de neurones, en apportant des corrections à des prédictions, pour annoter rapidement une image de la vidéo. Puis un module de propagation segmente le reste de la vidéo. Récemment, les réseaux à mémoire se sont montrés efficaces, grâce à leur module de mémoire, qui enregistre une référence de l'objet à segmenter tout au long de la vidéo.

3

Tomographie électronique : spécificité des données et apprentissage profond

Outline

3.1 Introduction 44
3.2 Notions de base sur la reconstruction de données en tomographie électronique 44
3.2.1 Microscope électronique en transmission 44
3.2.2 Principe de la tomographie 45
3.2.3 Principales difficultés de reconstruction des images de tomographie électronique 46
3.3 Présentation des données disponibles 47
3.3.1 Support de catalyseur alumine 48
3.3.2 Zéolithes 48
3.4 Protocole d'évaluation 50
3.4.1 Évaluation du réseau U-Net sur des données 2D de microscopie électronique à balayage 51
3.4.2 Évaluation du réseau U-Net sur des données partiellement annotées 2D de microscopie électronique à balayage 52
3.4.3 Évaluation du réseau U-Net sur les données 3D de tomographie électronique 54
3.5 Résultats 56

3.5.1 Premiers résultats avec le réseau de neurone U-Net sur les données bidimensionnelles de microscopie électronique à balayage 56
3.5.2 Apport de l'annotation partielle 59
3.5.3 Résultat d'un réseau de neurones U-Net sur les données tridimensionnelles de tomographie 60
3.6 Conclusion 62

Les données issues de tomographie électronique sont spécifiques et il est important de bien comprendre leurs particularités pour des images de matériaux mésoporeux. Une première évaluation de ces données avec des réseaux de neurones classique de segmentation permet d'identifier les situations sur lesquelles la segmentation est satisfaisante, mais aussi les faiblesses de ces réseaux.

3.1 Introduction

U-Net est un réseau de neurones convolutif de type encodeur-décodeur, permettant la segmentation sémantique d'images bidimensionnelles [78]. U-Net est utilisé dans de nombreuses applications. C'est un réseau satisfaisant dans de différents domaines [58, 8]. C'est également un réseau très étudié, cité dans plusieurs dizaines de milliers de travaux scientifiques, dont de nombreuses améliorations ont été proposées [39, 37, 47]. Dans ce chapitre, nous effectuons une première évaluation de U-Net sur nos données. Cette première évaluation est axée sur l'étude des particularités des données de tomographie électronique ainsi que les spécificités de notre application, afin de proposer dans les chapitres suivants des contributions pertinentes dans notre cadre d'application. Nous commençons par rappeler quelques notions de base sur la reconstruction tomographique, puis nous présenterons les jeux de données disponibles.

3.2 Notions de base sur la reconstruction de données en tomographie électronique

Avant de s'intéresser aux images, nous nous intéressons tout d'abord aux notions de base de la reconstruction de données en tomographie électronique. Il est important de bien comprendre l'acquisition des images pour comprendre leurs défauts. Dans un premier temps, nous nous intéresserons à la première composante de la tomographie électronique, l'acquisition des images à l'aide d'un microscope électronique en transmission. La deuxième composante de la tomographie électronique est l'étape de reconstruction. Enfin, nous décrirons les principaux défauts des images de tomographie électronique.

3.2.1 Microscope électronique en transmission

Le microscope électronique en transmission (MET) est un instrument permettant d'acquérir des images bidimensionnelles à l'échelle du nanomètre.

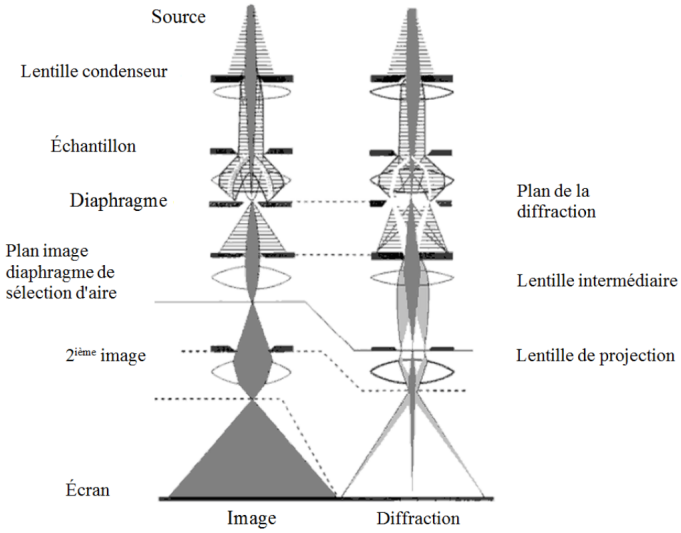


FIGURE 3.1 – Schéma d'un MET. À gauche, le mode image, à droite le mode diffraction [4].

En mode image, un faisceau d'électrons est envoyé à travers un échantillon à imager, puis ce faisceau traverse un réseau de lentilles magnétiques afin de former une image sur un écran fluorescent ou un capteur électronique. Afin de calibrer les différentes lentilles, le MET peut se configurer en mode diffraction [Figure 3.1]. Le MET permet d'obtenir des images 2D à de très petites échelles. Pour acquérir une information volumique, la tomographie est utilisée. La géométrie d'acquisition est en faisceau parallèle.

3.2.2 Principe de la tomographie

La tomographie est une technique d'imagerie permettant la reconstruction de l'intérieur d'un objet à partir de mesures extérieures à l'objet. Dans le cas de la tomographie électronique, une série d'images de l'échantillon sont acquises avec différents angles à l'aide d'un MET. Un algorithme de reconstruction compile l'information des différentes projections pour fournir une image volumique de l'échantillon [Figure 3.2]. Une méthode de reconstruction classiquement utilisée est la rétroprojection filtrée [38], basée sur la transformée de Radon [73]. Ces méthodes permettent d'obtenir un nombre important d'informations en fonction des mesures captées. Ici, la tomographie permet de reconstruire l'intérieur de matériaux jouant le rôle de catalyseurs et ainsi avoir accès à la porosité des matériaux. Cette technique produit cependant

des défauts dans ces images dus au fait de certaines limitations que nous abordons ci-après.

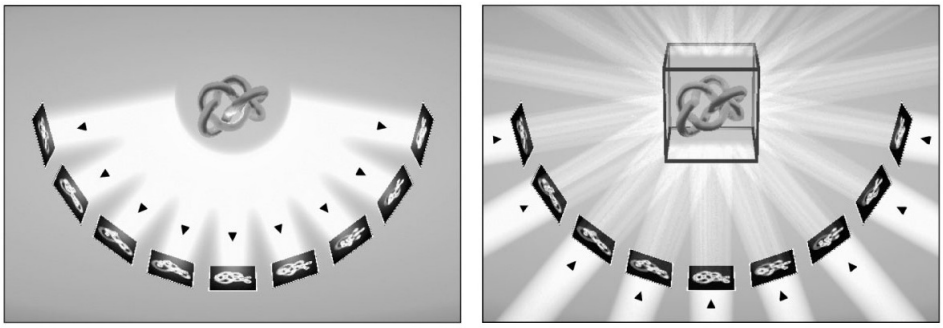


FIGURE 3.2 – Schéma du principe de tomographie. À gauche, l'acquisition d'images d'un échantillon sous plusieurs angles. À droite, l'utilisation de ces images pour reconstruire l'échantillon [79].

3.2.3 Principales difficultés de reconstruction des images de tomographie électronique

Certaines limitations techniques rendent la reconstruction de tomogrammes difficile. Il y a deux principales difficultés :

- Le nombre de projections joue un rôle majeur dans la qualité de la reconstruction. Plus le nombre de projections est élevé, plus précise sera la reconstruction. Dans le cas de projections acquises avec un MET, l'intervalle angulaire balayé, c'est-à-dire l'amplitude autour de laquelle l'échantillon est pivoté, n'est pas complet. Dans le cadre d'images acquises avec un MET, avec un porte-objet adapté à la tomographie, la variation angulaire est de -70° et +70° [Figure 3.3].
- L'alignement des projections est également un problème majeur. En effet, lorsque l'angle d'acquisition est modifié pour préparer la prochaine image, l'objet est déplacé dans le champ d'observation [Figure 3.4]. Même si ce déplacement est partiellement corrigé par les MET les plus récents, l'erreur d'alignement est responsable de certains défauts de reconstruction.

Le bruit en halo observé dû à ces deux problèmes est représenté sur la figure 3.5.

La tomographie électronique nécessite deux étapes : l'acquisition de l'information et la reconstruction du volume. Ces étapes permettent d'avoir une information sur la structure interne d'un objet en n'effectuant que des mesures

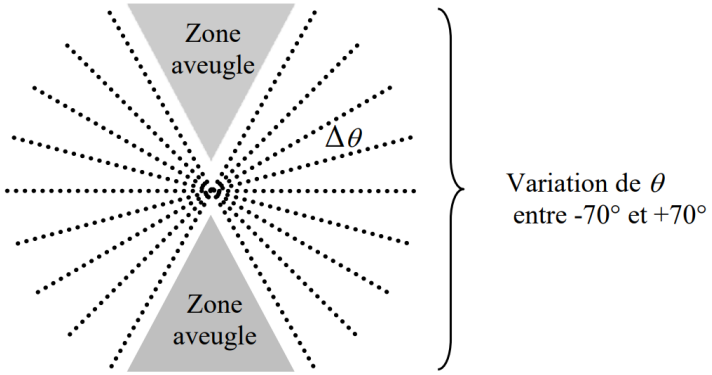


FIGURE 3.3 – Illustration du manque de projections dû à un intervalle angulaire de balayage limité [84].

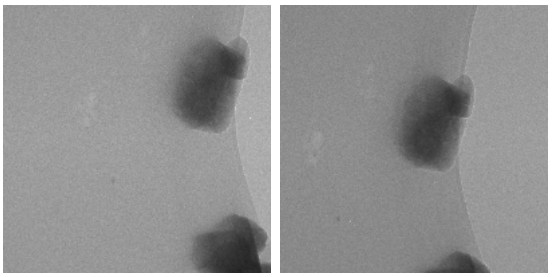


FIGURE 3.4 – Support catalyseur de zéolithe déplacé (6,64 nm/pixel) entre deux angles d'acquisition. [84].

extérieures. Les limitations techniques au niveau de l'acquisition provoquent des imprécisions lors de la reconstruction, diminuant la qualité des images. Dans la partie suivante, nous analyserons des images issues de tomographie électronique pour en comprendre toute leur complexité.

3.3 Présentation des données disponibles

Avant de détailler les expérimentations, il est important d'analyser les données disponibles. Avoir un *a priori* sur les images est intéressant pour déterminer s'il y a besoin de prétraitement ou d'augmentations de données à effectuer. Nous avons à notre disposition deux types d'images : des images de test de dépôt d'alumine bidimensionnelles obtenues par microscopie à balayage. Ces images ne sont pas entachées par des artéfacts de reconstruction de la tomographie électronique. Puis nos images d'intérêt de zéolithe tridi-

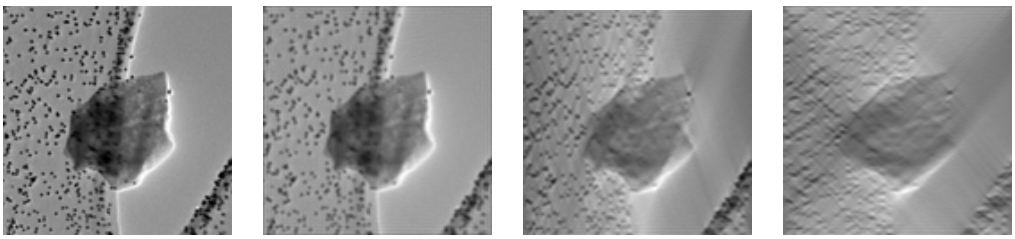


Image originale 180 projections 140 projections 100 projections

FIGURE 3.5 – Images reconstruites contenant des artéfacts de reconstructions. [84].

mensionnelles.

3.3.1 Support de catalyseur alumine

Nous possédons un jeu d'images de support de catalyseur alumine, qui sont des images 2D issues de microscopie électronique à balayage. L'ensemble est composé de 24 images d'une taille de 1024×768 et d'une résolution de 0.11165 µm/pixel [Figure 3.6]. Ces images ont été utilisées principalement pour des tests d'architectures de segmentation 2D. Par la suite, nous avons utilisé principalement les images de zéolithe. Ces données sont imparfaitement annotées. En effet, plusieurs annotateurs différents ont travaillé sur ce jeu de données, causant des erreurs de labellisation : certains objets ne sont pas classifiés de la même manière selon l'annotateur. Cette particularité est à prendre en compte lors de la segmentation sémantique de ces images.

3.3.2 Zéolithes

Les matériaux d'intérêt étudiés sont des zéolithes [Figure 1.1], un cristal poreux d'aluminosilicate. Un matériau poreux est composé d'une multitude de pores ou cavités qui peuvent être interconnectées entre elles. Bien rendre en compte cette complexité au niveau de la structure interne lors de la segmentation est important si on garde à l'esprit que la carte de segmentation sera utilisée pour des calculs sur les caractéristiques du catalyseur.

Nous avons à disposition quatre volumes de zéolithe de taille (592,600,623), (512,512,108), (592,840,296) et (512,510,72) voxels ainsi que leurs annotations. Ces volumes présentent des difficultés majeures pour une tache de segmentation avec apprentissage profond :

1. Les images sont particulièrement bruitées [Figure 3.7.a]. Cette particu-

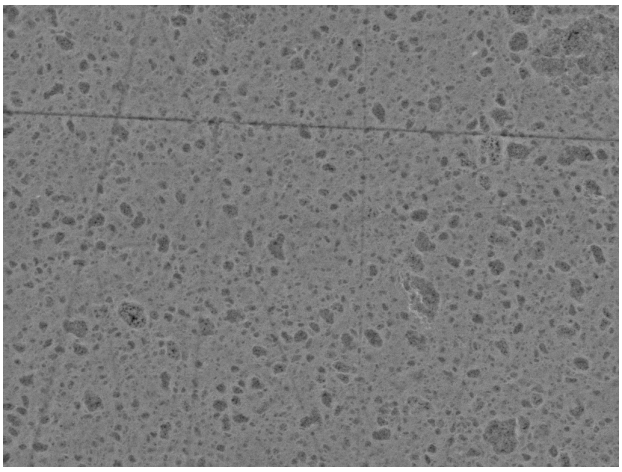
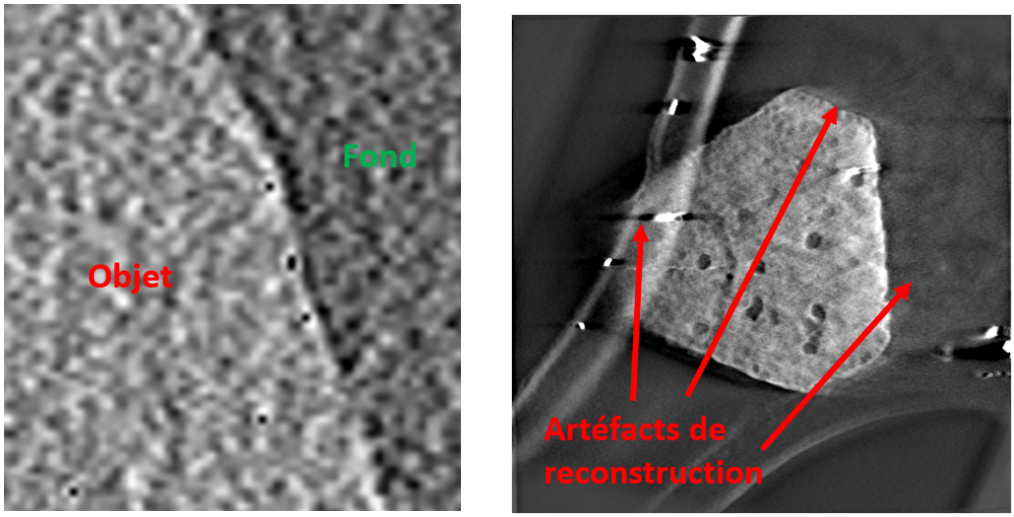


FIGURE 3.6 – Image de support de catalyseur alumine (résolution : 0.11165 µm/pixel).

larité rend les méthodes de segmentation classiques inefficaces. C'est la texture seule et non l'intensité qui permet d'identifier l'objet du fond, ce qui rend les approches convolutives prometteuses.

2. Les images possèdent des artéfacts de reconstructions [Figure 3.7.b]. La cause de ces artéfacts a été explicitée dans la partie précédente. En effet, pour obtenir ces volumes, plusieurs images sont utilisées avec plusieurs angles d'inclinaison. Les données récoltées sont utilisées pour reconstruire ces volumes. C'est le manque d'angles différents ainsi que la difficulté à aligner ces différentes projections qui produisent ces artéfacts. Ces artéfacts compliquent la segmentation.
3. Les volumes sont différents les uns des autres. Pour chaque volume, la texture de l'objet et du fond, la forme, la densité du réseau poreux sont tous uniques [Figure 3.8]. On note également que l'objet à segmenter peut être soit plus clair que le fond, soit plus foncé. Avec une telle diversité de données, il faut une grande quantité d'exemples pour qu'un réseau de neurones réussisse à apprendre à partir d'objets aussi divers et produire une segmentation satisfaisante.
4. Il y a extrêmement peu de volumes annotés. Nous avons au total quatre volumes différents. Même si les quatre volumes représentent un nombre important d'images, ces images sont très similaires à l'intérieur d'un même volume et n'apportent que peu d'informations supplémentaires. Nous avons un nombre d'exemples très faible par rapport aux jeux d'entraînement utilisés dans la littérature.



(a) (b)

FIGURE 3.7 – Exemple de bruit (a) et d'artéfacts de reconstruction (b) sur un plan de deux volumes de zéolithes différents (résolution : 1 nm/pixel). L'image représentant le bruit a été agrandi d'un facteur 10.

	Support de catalyseur alumine	Zéolithes
Méthode d'acquisition	Microscope électronique à balayage	Tomographie électronique
Type de données	2D	3D
Résolution	0.11165 µm/pixel	1 nm/pixel
Nombre d'images	24	4

TABLE 3.1 – Récapitulatif des données à disposition.

Nous avons donc quatre volumes aux caractéristiques très différentes et difficiles à segmenter. L'enjeu est de développer une méthode qui gère à la fois le bruit et les artéfacts de reconstructions, avec une base de donnée d'entraînement limitée. Nous proposons d'évaluer des méthodes de segmentation sémantique classique sur nos données 2D issues de la microscopie électronique à balayage ainsi que sur des données 3D issues de la tomographie électronique.

La table 3.1 récapitule les données à disposition.

3.4 Protocole d'évaluation

Nous proposons une série d'expériences afin d'évaluer le jeu de données avec le réseau U-Net. Tout d'abord, le réseau sera testé sur le jeu de données

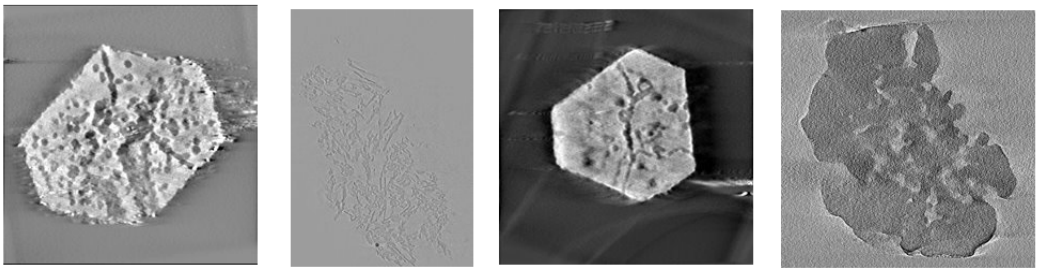


FIGURE 3.8 – Exemple de plan issu de chacun des volumes (résolution : 1 nm/pixel). La plupart de ces volumes proviennent divers matériaux de type zéolithes et sont visuellement différents.

de support de catalyseur alumine obtenu par microscopie électronique à balayage. Sur ces images sont annotées les zones où localement l'alumine est moins dense. Ensuite, pour explorer l'effet de l'incohérence de l'annotation de ces images, nous étudions le comportement de U-Net avec des données dont l'annotation est incertaine. Enfin, nous évaluons le réseau sur des images 3D de matériaux mésoporeux issus de tomographie électronique.

3.4.1 Évaluation du réseau U-Net sur des données 2D de microscopie électronique à balayage

Le but de cette partie est d'établir une référence pour la segmentation sémantique avec une méthode classique de segmentation sémantique par apprentissage profond. Les résultats de cette expérience serviront à analyser la particularité et le comportement de cette méthode vis-à-vis de ces images particulières. Nous reprenons l'architecture classique U-Net [78]. Ce réseau, couramment utilisé dans la littérature pour de nombreuses applications, servira de point de référence.

Le réseau [Figure 3.9] se compose d'un encodeur de trois couches de deux convolutions successives doublant la dimension des vecteurs descripteurs, puis d'une couche de *max pooling* réduisant de moitié la résolution. Le nombre de caractéristiques à la première couche est fixé à 32. Toutes les convolutions sont de taille 3×3. Pour le décodeur, les plans descripteurs sont concaténés au résultat de la couche précédente, suivi d'une convolution. L'augmentation de la résolution est réalisée par une convolution transposée, accompagnée d'une réduction de moitié de la dimension des vecteurs descripteurs. Sur la dernière couche, une combinaison linéaire est appliquée pour réduire le nombre de caractéristiques au nombre de classes, ici deux : une pour le fond et une pour

l'objet. On obtient une matrice dans laquelle chaque coefficient correspond à la probabilité de chaque pixel d'appartenir soit à la classe fond, soit la classe objet. À chaque pixel aux coordonnées (i, j), une paire de probabilités $(p_{i,j,fond}, p_{i,j,objet})$ est calculé. La classe du pixel est obtenue par la classe du maximum de ces deux valeurs $\bar{y}(i, j) = \text{argmax}(p_{i,j,fond}, p_{i,j,objet})$. Ce résultat fournit ainsi le masque de segmentation.

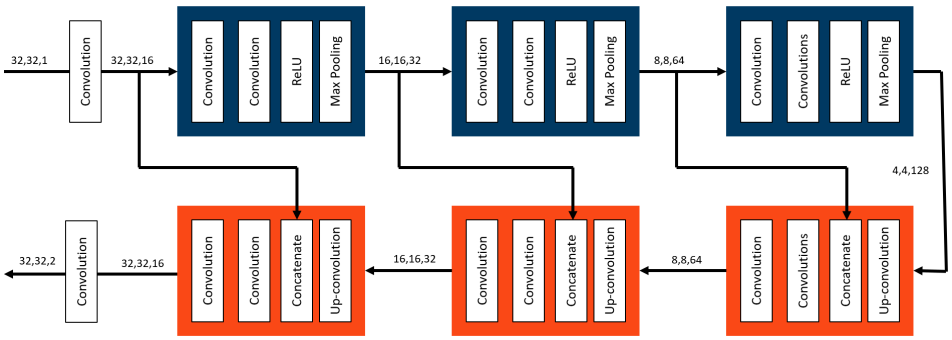


FIGURE 3.9 – Architecture du réseau U-Net mis en place.

Vingt patchs de taille 64 par 64 pour chaque image d'entrainement sont extraits de manière aléatoire pour nourrir notre réseau. Lors de l'entraînement, les patchs sont augmentés à l'aide de petites rotations, et de légères déformations. À la reconstruction, l'image à segmenter est découpée en patch de 64 pixels par 64 pixels qui sont fournis au réseau pour l'évaluation. Les différents patchs sont assemblés pour reformer l'image complète. Cette expérience donne une première évaluation d'une méthode avec un réseau de segmentation sémantique classique sur les données. Cependant, l'annotation de ces images n'est pas parfaite, une nouvelle expérience peut être menée.

3.4.2 Évaluation du réseau U-Net sur des données partiellement annotées 2D de microscopie électronique à balayage

Plusieurs experts ont annoté une partie des images des supports de catalyseurs alumine [Figure 3.10]. Certains experts ayant travaillé ces images n'ont pas annoté complètement certains supports. Cette incohérence dans les vérités terrain pose un problème lors de l'entraînement d'une méthode par apprentissage profond. En effet, il est important que les données extraites des images d'entraînement soient cohérentes afin de ne pas induire le réseau en erreur. Nous avons remarqué que sur un nombre important d'images, les plus

petits objets n'ont pas été annotés. Il y a donc des pixels objets considérés comme des pixels fond sur ces images. Il s'agit de la classe fond qui est imprécise. Il semble intéressant d'introduire une méthode pour apprendre un réseau de segmentation sémantique de type U-Net en ignorant certaines régions du fond [20].

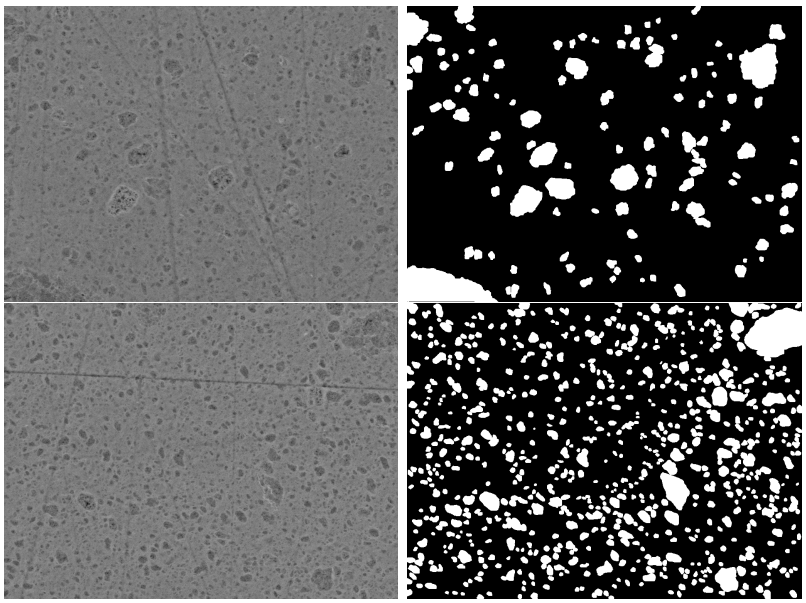


FIGURE 3.10 – Différence entre les annotations de deux images similaires.

La fonction de perte est adaptée pour prendre en compte des vérités terrain partiellement annotées. Il y a ainsi trois classes dans un masque de segmentation partiellement annoté : objet, fond et une classe qui correspond aux pixels dont l'information de classe n'est pas disponible, ou ici potentiellement incomplète. L'entropie croisée pondérée est définie de la manière suivante, pour chaque pixel x et sa classe associée $y(x)$:

$$E(x) = \sum_{x \in \Omega} w(x) y(x) \log(p_l(x)) \tag{3.1}$$

$$\text{avec } w(x) = \begin{cases} 1 & \text{si } y(x) = \text{fond} \\ 1 & \text{si } y(x) = \text{objet} \\ 0 & \text{si } y(x) = \text{non labellisé} \end{cases}$$

Le masque de segmentation des images est modifié pour les transformer en données partiellement annotées. Les pixels objets n'ont pas été modifiés. Les

pixels annotés comme fond peuvent contenir des pixels objets, c'est pourquoi n'est conservée qu'une partie du fond autour des objets déjà annotés. Le reste du fond est considéré comme non annoté. Pour ce faire, des opérateurs de morphologie mathématique sont utilisés. Sur la vérité terrain, une fermeture morphologique [61] avec un disque de rayon 25 pixels est effectué pour agglomérer les objets proches en un seul objet, puis une dilatation morphologique avec un disque de rayon 10 pixels pour prendre en compte le fond autour des objets. Cela produit ainsi une séparation entre pixels annotés et pixels non annotés pour séparer les images en trois classes [Figure 3.11]. Un autre point d'intérêt reste à explorer, à savoir si une approche de segmentation sémantique d'images bidimensionnelles peut répondre de façon satisfaisante à la segmentation de données tridimensionnelles.

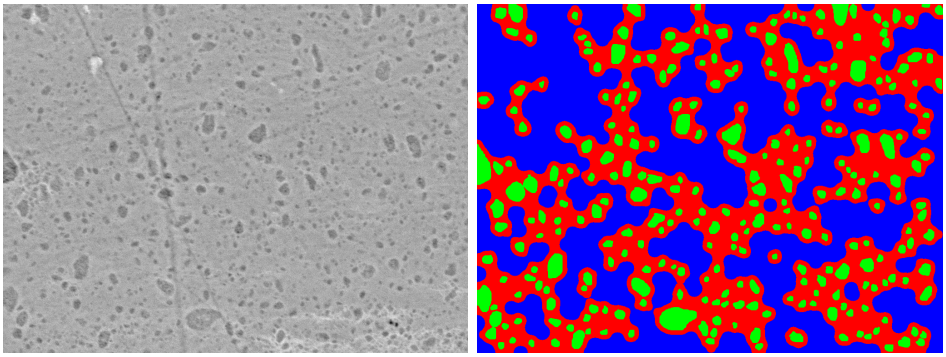


FIGURE 3.11 – Image originale à gauche, image semi-labellisé à droite où les pixels verts correspondent à l'objet, les pixels rouges au fond et les pixels bleus aux pixels non annotés.

3.4.3 Évaluation du réseau U-Net sur les données 3D de tomographie électronique

Pour traiter les données 3D à notre disposition, nous implémentons une architecture U-Net 2D, en traitant chaque plan des données 3D de manière indépendante. Il est ainsi possible d'obtenir des résultats similaires à une approche 3D [39]. Cette expérimentation montrera si une approche plan par plan est satisfaisante pour les données de tomographie électronique.

Soit le volume d'entrée V, composé d'un ensemble de S plans $\{X_s\}_{s=1,...S}$ tel que chaque plan $X_s \in \mathbb{R}^{W \times H \times 1}$ où W est la longueur du plan, H la largeur du plan. Comme chaque plan est une image en niveaux de gris, la dernière dimension du plan est 1. $x_{s,i}$ est alors le niveau de gris du voxel $i \in \mathcal{I}$ du plan X_s, $\mathcal{I}$ étant le support spatial du plan. Ainsi, $X_s = (x_{s,i})_{i \in \mathcal{I}}$.

Une annotation manuelle sur quelques plans du volume est effectuée, avec E, l'ensemble des indices des plans annotés. Le réseau est entrainé sur les plans de E, puis le reste des plans $\{X_s\}_{s \notin E}$ est passé en entrée au réseau entraîné. Cette méthode permet de reconstruire un volume à partir de quelques plans annotés. Ainsi, le temps d'annotation est drastiquement réduit.

Il est intéressant d'enquêter sur le nombre de plans nécessaires à l'entraînement permettant d'obtenir une reconstruction correcte. Pour cela, le réseau est entraîné sur $N_E = \{1, 2, 3, 4, 5, 10, 20, 50, 100\}$ plans pris aléatoirement dans le volume complétement annoté, avec $N_E = \text{Card}(E)$, puis testé sur le reste des plans. Cependant, l'annotation d'un plan entier requiert tout de même un temps non négligeable. Si le réseau parvient à apprendre avec des images partiellement annotées, l'expert métier pourrait ne fournir qu'une portion d'un seul plan, réduisant encore plus la quantité d'image à annoter. On introduit un taux de labellisation r qui indique la proportion de l'image qui est labellisée. On définit r tel que :

$$r = \frac{A}{W \times H} \tag{3.2}$$

Avec A l'aire de la surface annotée, W la longueur du plan et H sa largeur. Il est ainsi possible d'entrainer le réseau avec un seul plan, mais un taux de labellisation inférieur à 1. Nous nous plaçons dans le cas où seul le plan central partiellement annoté est fourni à l'entrainement. Plusieurs valeurs de r sont testées (0.1, 0.25, 0.5, 0.75 et 0.9).

Nous proposons de générer des images partiellement annotées en cachant volontairement une partie de l'image. Le plan est découpé en 100 patchs de tailles égales. En fonction de ce taux r, les patchs définis comme non labellisé sont sélectionnés de manière aléatoire, en ne considérant d'abord que les patchs avec lesquels il y a suffisamment de pixels des deux classes. Les patchs contenant les deux classes sont d'abord choisis de manière aléatoire. Si plus de données ont besoin d'être choisies, les patchs restants sont sélectionnés. Cette méthode produit une semi-labellisation proche de ce qu'un expérimentateur fournit au réseau, avec les deux classes bien représentées [Figure 3.12].

Lors d'un entraînement avec un seul plan, le choix de ce plan est d'autant plus important. Pour étudier l'influence de ce plan dans le résultat des plans voisins, nous étudions le score des plans en fonction de leur position par rapport au plan d'entraînement. L'ensemble de ces expérimentations renseigneront sur le comportement d'un réseau de neurones de type U-Net pour la segmentation sémantique de matériaux mésoporeux issus de la tomographie électronique.

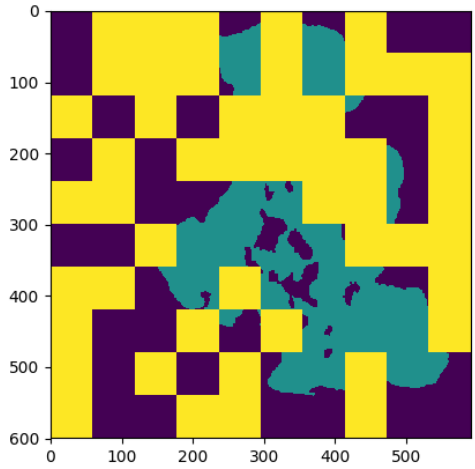


FIGURE 3.12 – Image d'un plan semi-labellisé. En vert, l'objet d'intérêt, en violet le fond et en jaune, les voxels non labellisés.

3.5 Résultats

Dans cette section, les résultats des différentes expérimentations décrites dans la partie précédente sont présentés. Tout d'abord, nous discuterons des premiers tests d'une application d'un réseau de neurone de type U-Net sur les données bidimensionnelles de microscopie électronique à balayage. Nous étudierons ensuite l'apport de la labellisation partielle de la vérité terrain sur ces mêmes images. Enfin, le même réseau U-Net sera déployé sur les données tridimensionnelles de tomographie électronique.

3.5.1 Premiers résultats avec le réseau de neurone U-Net sur les données bidimensionnelles de microscopie électronique à balayage

Une première évaluation est effectuée sur les images bidimensionnelles de support de catalyseur alumine. Pour évaluer nos résultats, nous utilisons comme métrique l'indice de Jaccard (*Intersection Over Union* ou IOU). Elle est définie par le rapport entre l'intersection entre la vérité terrain et le masque de segmentation et leur union :

$$IOU(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{3.3}$$

Avec |.| l'opérateur de cardinalité. Le numérateur correspond au chevauchement entre la prédiction et le masque tandis que le dénominateur correspond à l'association de la prédiction et du masque. Plus on est proche de 1, plus la segmentation est considérée comme satisfaisante [Figure 3.13]. Cette valeur est égale à 1 si le masque de segmentation et la vérité terrain sont identiques.

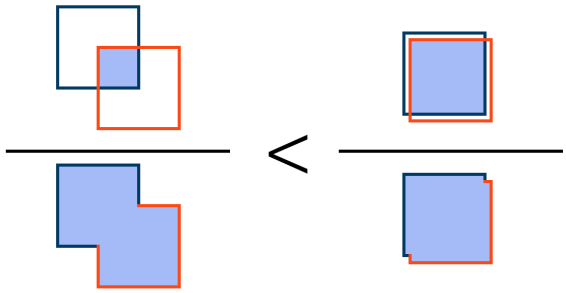


FIGURE 3.13 – Illustration de l'IOU. À gauche, le rapport de l'intersection et de l'union du masque et de la segmentation est inférieur à l'exemple de droite. Plus l'aire de l'intersection et de l'union sont proches, plus l'IOU sera proche de 1.

Nous utilisons une étape de validation croisée par bloc (k-fold cross validation). Cette méthode consiste à répéter un entraînement sur toutes les données disponibles afin de s'assurer que les résultats ne soient pas dépendants des images d'entraînement choisies. On obtient ainsi une estimation plus robuste des résultats. Pour chaque entraînement, trois groupes d'images sont définis : un groupe d'entraînement, un groupe de test ainsi qu'un groupe de validation. Les images d'entraînement sont utilisées pour calculer la fonction de perte et ainsi rétro propager le gradient dans le réseau. Les images de validations permettent d'avoir des images témoins pendant l'entraînement. Elles servent à surveiller l'entraînement sans utiliser des images biaisées du groupe d'entraînement. Enfin, les images de test servent à calculer les scores de performances finaux. Un entrainement est appelé un bloc (fold) et on effectue un nombre k d'entraînements tout en modifiant les groupes d'entraînement, de test et de validation. Un score moyen calculé sur tous les blocs donne un résultat qui n'est pas biaisé par les images d'entraînement choisies [Figure 3.14].

Sur les 24 images, 6 blocs sont effectués avec 4 images d'entrainement, 4 images de validation et les 16 images restantes en images de test sont sélectionnées pour chaque bloc. Les images d'entrainement, de validation et de test, sont choisies tel que chaque image soit prise une fois pour l'entraînement.

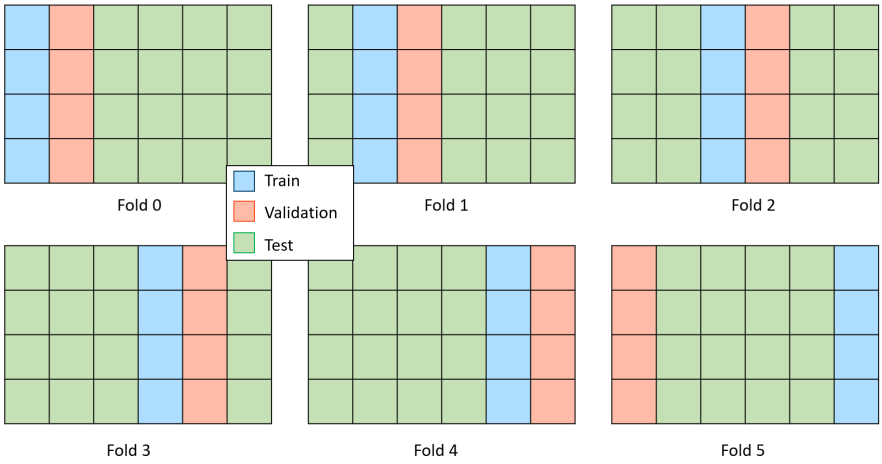


FIGURE 3.14 – Exemple d'un entraînement par validation croisée par blocs. Ici, avec 24 images par bloc, 4 images sont sélectionnées pour l'entraînement (cases en bleu), 4 pour la validation (cases en rouge) et les 16 restantes pour le test (cases en vert). Pour chaque bloc, les différents groupes d'images sont différents. Un score est calculé pour toutes les images de tests et la moyenne entre les scores des différents blocs constitue le score final.

Les résultats peuvent être observés sur la figure 3.15 et la table 3.2. Sur la figure 3.15, les plus gros objets ont été segmentés même si leurs contours ne sont pas bien définis. De plus, des artefacts dus aux effets de bord sont observés lors de la reconstitution des patchs. En effet, des coupures nettes sont aperçues sur les bordures des patchs. Ce manque de continuité peut être réduit en adaptant le réseau pour qu'il prenne en entrée l'image entière ou en utilisant une approche de calcul d'inférence stochastique [35].

Le réseau détecte également mal les petits défauts de densité d'alumine. La principale cause de ces erreurs vient de la mauvaise qualité des annotations sur certaines images. En effet, les images n'ont pas toutes été annotées par le même opérateur. Certains opérateurs ont annoté tous les objets alors que d'autres n'ont sélectionné que les plus gros objets. Certains défauts ne sont donc pas annotés dans la base d'apprentissage, ce qui peut créer des ambiguïtés au niveau de l'entraînement. Dans la partie suivante, nous examinons

les expériences traitant de ce problème.

Fold	Fold 0	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
IOU	0,615	0,567	0,675	0,628	0,672	0,592

	Minimum	Maximum	Moyenne	Écart-Type
IOU	0,567	0,675	**0,625**	0,043

TABLE 3.2 – Tableau des IOU pour les différents tests de segmentation utilisant le réseau U-NET 2D. Les résutats quantitatifs montrent une bonne stabilité d'un bloc à l'autre (écart type de 0,04) et une moyenne de 62,5% sur l'IOU. Ce résultat n'est pas très élevé à cause des problèmes d'annotation discutés précédemment.

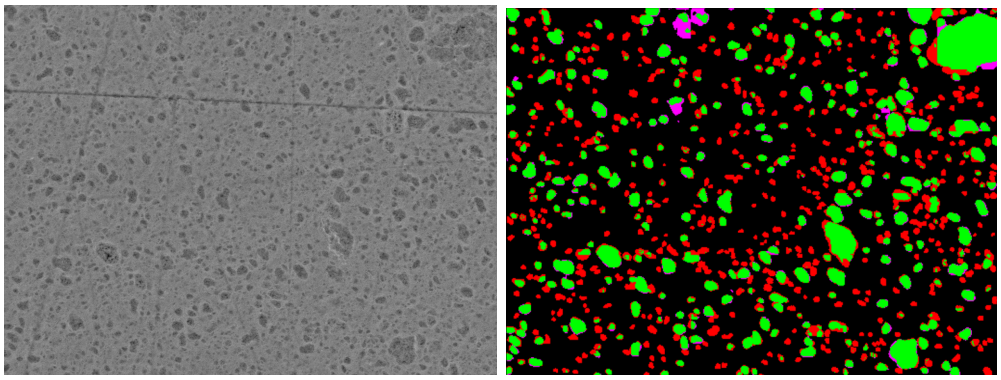


FIGURE 3.15 – Image originale de taille 1024x768 d'une résolution de 0.11165 µm/pixel à gauche, représentation des résultats à droite avec en vert les pixels correctement segmentés, en rouge les pixels manquants par rapport au masque de segmentation (faux négatifs) et en violet les pixels en trop du masque de segmentation (faux positifs). On peut voir que les principaux objets ont été segmentés correctement, cependant de nombreux petits défauts sont manquants. Il y a relativement peu de faux défauts segmentés. De plus, on observe des effets de bord dû au découpage par patchs.

3.5.2 Apport de l'annotation partielle

Cette section montre les résultats d'un réseau de neurone de type U-Net avec un entraînement avec labellisation partielle de la vérité terrain. La même architecture que la partie précédente est conservée. Seule la fonction de perte a été changée. Notons que le masque de labellisation partielle choisi permet d'éviter les erreurs de labellisation du fond pendant l'entraînement. Cependant, lors de l'évaluation, la labellisation complète étant utilisée, le score

obtenu est entaché par des erreurs de labellisation. Il est nécessaire d'évaluer qualitativement la segmentation en complément. Un exemple de segmentation est présenté sur la figure 3.16. Nous pouvons remarquer qualitativement que la segmentation est nettement améliorée. Plus de petits objets ont été segmentés. L'IOU obtenue est de 0.71. Ces expérimentations illustrent que la labellisation partielle peut apporter un gain si couplé avec une architecture de type U-Net. Cette approche a un avantage considérable compte tenu de nos données : il n'est plus nécessaire d'annoter la totalité de nos données pour entraîner un réseau de type U-Net. Cela a pour effet de réduire grandement l'effort d'annotation.

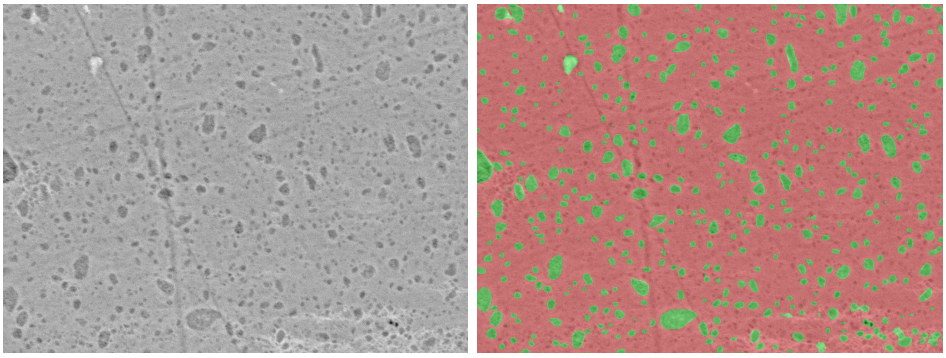


FIGURE 3.16 – Image originale à gauche. À droite, le masque de segmentation superposé à l'image originale avec en vert les pixels catégorisés en objet et en rouge les pixels catégorisés en pixel fond.

3.5.3 Résultat d'un réseau de neurones U-Net sur les données tridimensionnelles de tomographie

Le même dispositif est utilisé sur les images de tomographie électronique. Le réseau U-Net est entraîné avec un nombre N_E de plans d'entraînement $\{X_e\}_{e\in E}$. Sur le reste des plans $\{X_s\}_{s\notin E}$ est calculé le score. L'IOU est toujours utilisé et est adapté dans le contexte 3D en faisant l'union et l'intersection des volumes. On définit l'IOU pour la segmentation de volume :

$$IOU(V) = \frac{\sum_{i=1}^{N} \hat{Y}_i \cap Y_i}{\sum_{i=1}^{N} \hat{Y}_i \cup Y_i} \tag{3.4}$$

Avec Y la vérité terrain du volume et $\hat{Y}$ la prédiction du masque de segmen-

tation.

L'expérience est répétée 100 fois sur un volume de zéolithe pour chaque nombre de plans, afin de réduire l'influence du choix d'un plan. La figure 3.17 montre les résultats obtenus. De très bons résultats sont obtenus dès deux plans sur ce jeu de données (0,78). Cependant, avec un seul plan utilisé à l'entraînement, les résultats sont significativement moins bons. À ce niveau, il faut un changement dans la méthode pour pouvoir essayer d'améliorer ces résultats. Nos efforts vont donc se concentrer dans le cas de figure où très peu de données d'entraînement sont disponibles.

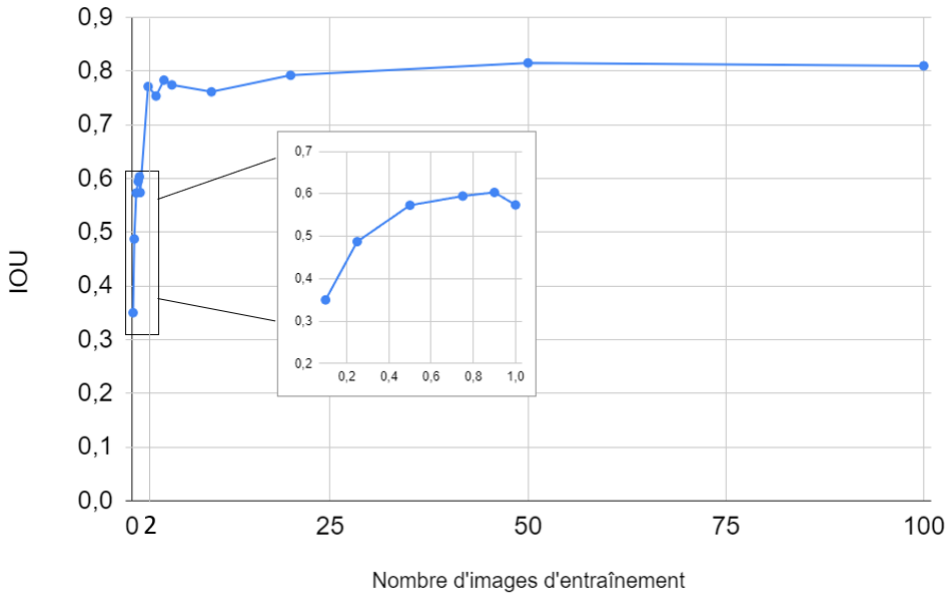


FIGURE 3.17 – Influence du nombre de plans d'entraînement. En abscisses, nombre de plans utilisés à l'entraînement, et en ordonnées, l'IOU correspondant. Plus l'IOU est proche de 1, meilleur est le résultat de la segmentation. Pour les valeurs de nombre d'images d'entraînement inférieur à 1, un seul plan d'entraînement est utilisé avec un taux de labellisation r correspondant à cette valeur.

Avec un seul plan d'entraînement, la sélection de ce plan est d'autant plus importante. On étudie le profil de l'IOU de chaque plan en fonction de la distance du plan d'entraînement. La figure 3.18 montre le résultat de l'expérience. Chaque courbe correspond à un taux r différent. L'IOU est plutôt stable sur tout le volume, jusqu'à une chute des performances abrupte. La chute de l'IOU pour les plans les plus éloignés du plan centrale s'explique

puisqu'il y a de moins en moins de matière au fur et à mesure que l'on s'approche du bord. Cela montre que le choix du plan d'entraînement a peu d'influence à condition de choisir un plan central où la quantité de matière est représentative du reste du volume.

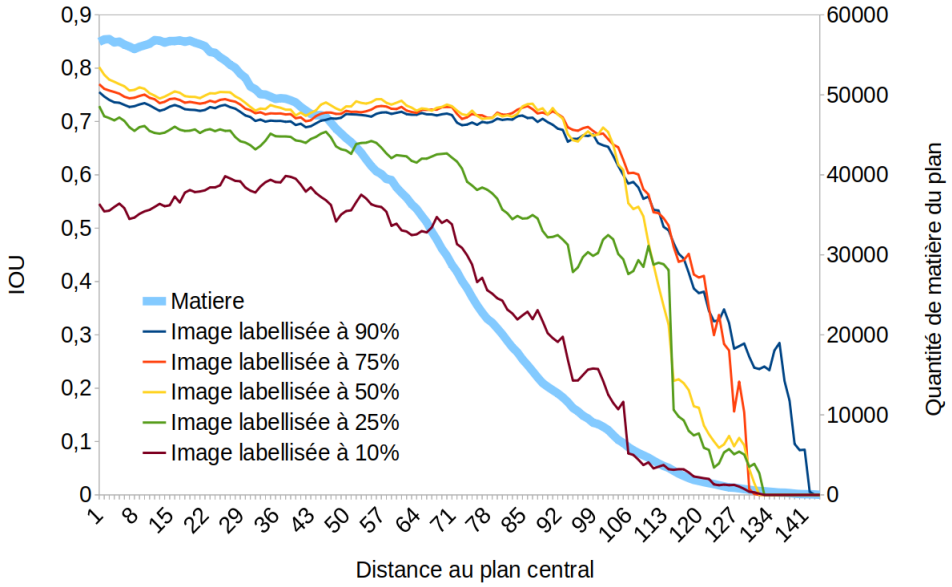


FIGURE 3.18 – IOU de chaque plan en fonction de la distance au plan central et de plusieurs taux de labellisation. En bleu clair, la courbe montrant la quantité de matière.

3.6 Conclusion

Cette première contribution constinue une étude préliminaire des données à notre disposition. De plus, un dispositif a été défini pour répondre à la problématique de la thèse.

En étudiant les images de support de catalyseur alumine issues de la microscopie électronique à balayage, nous avons tiré les conclusions suivantes :

- Un réseau de neurone convolutif de type encodeur-décodeur comme U-Net, souvent utilisé dans la littérature, obtient des résultats prometteurs sur les données. L'inférence par patchs sans précautions particulières peut provoquer des artéfacts de bords lors de l'inférence. Pour la suite, nous utiliserons en entrée systématiquement des images complètes de taille arbitraire.

- Comme le montre la disparité de la précision de l'annotation dans les vérités terrain, notre méthode doit être capable de traiter des annotations imprécises. Nous avons donc introduit une fonction de perte qui tient compte d'un masque permettant de sélectionner les pixels effectivement annotés dans chaque image. L'introduction de cette labellisation partielle améliore la précision de la prédiction de la segmentation. De plus, cette approche permet d'entraîner un réseau de neurone convolutif avec moins de données annotées. Cette remarque est pertinente pour nos travaux puisqu'un nombre réduit d'annotation nécessaire implique un temps de labellisation fortement réduit de la part d'un expert métier.

Concernant les tests sur les données tridimensionnelles issues de la tomographie électronique :

- Nous avons montré qu'une méthode d'apprentissage profond avec un encodeur-décodeur traitant des images bidimensionnelles de type U-Net obtient de très bons résultats, grâce à un entrainement sur quelques plans du volume, puis en segmentant le reste du volume. Ce dispositif s'adapte très bien à notre problématique : on demande à un annotateur de labelliser une fraction du volume, puis le réseau effectue une prédiction sur la totalité du volume, réduisant grandement le temps d'annotation de l'expert. Cependant, le réseau U-Net n'apporte pas de résultats satisfaisants pour un nombre de plans d'entraînement inférieur à 1. Un changement de méthode est alors nécessaire.
- Enfin, nous avons montré que le choix du plan d'entraînement dépendait peu de sa position dans le volume à condition que la quantité de matière du plan d'entraînement soit représentative du reste du volume.

Dans le prochain chapitre, nous reprendrons le dispositif définit pour la segmentation des données tridimensionnelles de tomographie électronique. Une nouvelle fonction de perte sera introduite pour prendre en compte à la fois les données annotées et non annotées.

4

Nouveau modèle basé sur l'apprentissage contrastif

Outline

4.1 Introduction 65
4.1.1 Approches semi-supervisées pour la segmentation sémantique 65
4.1.2 Apprentissage contrastif 66
4.2 Méthode proposée pour la segmentation sémantique à partir d'apprentissage contrastif . 66
4.2.1 Formalisation de la perte contrastive 67
4.2.2 Architecture 69
4.2.3 Gestion des données non annotées 70
4.3 Expérimentations 72
4.3.1 Paramètres expérimentaux 72
4.3.2 Comparaison avec un entraînement conventionnel . 74
4.3.3 Choix du classifieur 75
4.3.4 Apport des fonctions de perte 76
4.4 Conclusion 78

Les expérimentations du chapitre précédent nous ont conduit à examiner la segmentation sémantique d'un volume à partir d'un plan partiellement annoté en entraînement. Cependant, les performances des méthodes n'exploitant que les données annotées ne produisent pas une segmentation satisfaisante. Nous proposons une méthode basée sur l'apprentissage contrastif (*contrastive learning*), qui pourra mieux exploiter les données annotées ainsi que les données non annotées.

4.1 Introduction

Dans le chapitre précédent, nous avons montré que les approches traditionnelles de segmentation sémantique ne produisent pas de bons résultats lorsque le taux de labellisation est faible. Lorsqu'il y a très peu de données disponibles pour entraîner une méthode d'apprentissage profond, un changement de méthode est nécessaire. Dans notre configuration, le but est de segmenter un volume à de partir portions du volume annotées. Cette approche se rapproche des méthodes semi-supervisées.

4.1.1 Approches semi-supervisées pour la segmentation sémantique

L'objectif des méthodes semi-supervisées pour la segmentation sémantique est d'optimiser la façon dont les données labellisées et non labellisées sont exploitées ensemble pour apprendre une tâche de segmentation. Des méthodes de segmentation supervisées peuvent introduire des éléments qui ne nécessitent pas de données annotées. C'est le cas de [46], qui utilise des transformations pour produire des paires d'objet de la même classe. Une autre approche consiste à appliquer des réseaux antagonistes, où un réseau générateur et un réseau discriminateur sont entraînés sur des images annotées. Des images non labellisées sont passées dans les deux réseaux et une fonction de similarité est minimisée [70]. Dans [67], un encodeur-décodeur standard est utilisé sur des images annotées. Des images non annotées sont passées dans plusieurs versions légèrement modifiées du décodeur. La sortie de l'ensemble des décodeurs est ensuite comparée à la sortie du décodeur principal. Nous avons décidé d'explorer l'apprentissage contrastif du fait de ses performances remarquables dans le domaine de la classification d'images. De plus, à notre connaissance, aucune méthode d'apprentissage profond avec apprentissage contrastif n'a été appliquée dans le cadre de la segmentation de matériaux mésoporeux issus de tomographie électronique.

4.1.2 Apprentissage contrastif

L'apprentissage contrastif vise à mieux exploiter les annotations, généralement à l'aide de réseaux siamois. L'objectif est de construire un espace latent dans lequel les objets ayant la même étiquette sont proches les uns des autres et les objets n'ayant pas les mêmes étiquettes sont éloignés les uns des autres. Des paires positives et négatives sont formées. Les paires positives sont composées de deux échantillons de la même classe, tandis que les paires négatives sont composées de deux échantillons de classes différentes. Une fonction de perte contrastive est utilisée pendant l'apprentissage pour rapprocher les paires positives et éloigner les paires négatives. L'apprentissage contrastif a donné d'excellents résultats dans la classification d'images [15, 34, 40, 16].

Pour la segmentation sémantique, des paires de pixels peuvent être utilisées [9]. Dans le cas où l'information de classe n'est pas connue, des paires de pixels positives et négatives peuvent tout de même être créées. En transformant une image, il est possible d'attribuer à un pixel, le même pixel issus de l'image transformée. Représentant le même objet, ces deux pixels forment une paire positive. Une paire négative est formée en sélectionnant tout autre pixel de l'image. Il est à noter qu'il est possible d'attribuer un pixel de la même classe pour une paire négative, créant ainsi une ambiguïté lors de l'apprentissage. Nous étudions dans la suite de ce chapitre une méthode afin d'entraîner de manière contrastive un réseau de neurone de type U-Net.

Dans ce chapitre, nous introduisons Une nouvelle méthode d'apprentissage semi-supervisée pour la segmentation de données issues de la tomographie électronique, tirant parti de l'apprentissage contrastif et des principes d'auto-apprentissage pour fournir une segmentation précise du volume en utilisant seulement quelques régions étiquetées d'une ou deux coupes 2D spécifiques.

4.2 Méthode proposée pour la segmentation sémantique à partir d'apprentissage contrastif

L'objectif de la fonction de perte contrastive dans notre contexte est de créer un espace dans lequel les voxels "objet" sont proches entre eux dans l'espace latent ou espace de représentation, mais éloigné des voxels "fond" [17]. Traditionnellement, ce type de représentation est obtenu en utilisant des réseaux siamois [17]. Cette technique permet un contrôle explicite sur la forme de

l'espace de représentation [Figure 4.1]. Cependant, dans le cas d'un réseau siamois, la perte contrastive est habituellement utilisée pour résoudre un problème de classification. Dans notre problème de segmentation sémantique, nous montrons qu'il est possible d'utiliser l'architecture U-Net puis de calculer une perte contrastive en sortie du décodeur pour effectuer une tâche de segmentation sémantique, permettant d'obtenir en sortie la projection de chaque voxel dans l'espace de représentation.

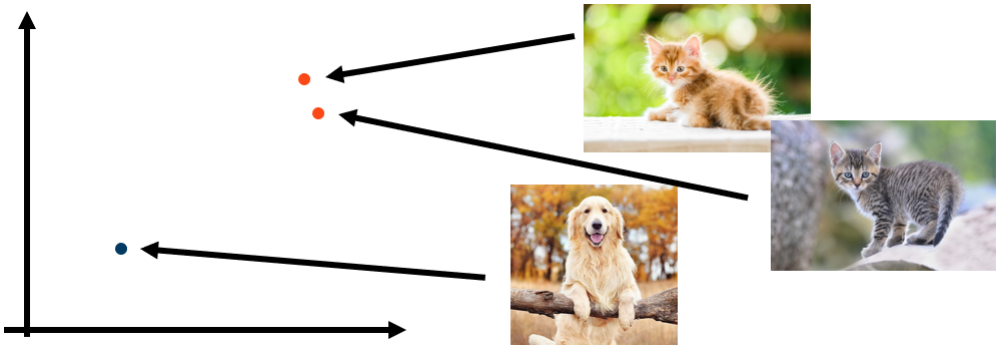


FIGURE 4.1 – Schéma d'un espace latent où chaque point correspond à une image projetée dans cet espace. Les images d'une même classe sont proches, contrairement aux images de classes différentes.

4.2.1 Formalisation de la perte contrastive

Soit le volume d'entrée V, composé d'un ensemble de S plans $\{X_s\}_{s=1,...S}$ tel que chaque plan $X_s \in \mathbb{R}^{W \times H \times 1}$ où W est la longueur du plan, H la largeur du plan. Comme chaque plan est une image en niveaux de gris, la dernière dimension du plan est 1. $x_{s,i}$ est alors le niveau de gris du voxel $i \in \mathcal{I}$ du plan X_s, $\mathcal{I}$ étant le support spatial du plan. Ainsi, $X_s = (x_{s,i})_{i \in \mathcal{I}}$.

La perte contrastive nécessite des paires positives $\mathcal{P}^+_{s,i}$ et des paires négatives $\mathcal{P}^-_{s,i}$. Dans notre cas, pour chaque voxel annoté $x_{s,i}$, $i \in \mathcal{L}_s$, sont construits les paires positives avec un autre voxel de la même classe et les paires négatives avec un voxel d'une classe différente. Soit Y_s la classe du voxel du plan s, alors $Y_s = (y_{s,i})_{i \in \mathcal{I}}$ avec $y_{s,i} \in \{1, 2, ..., C\}$ où $1, 2, .., C$ sont les labels des classes. Il vient :

$$\mathcal{P}^+_{s,i} = \{j \in \mathcal{L}_s,\ i \neq j, y_{s,i} = y_{s,j}\} \tag{4.1}$$

$$\mathcal{P}^-_{s,i} = \{j \in \mathcal{L}_s,\ i \neq j, y_{s,i} \neq y_{s,j}\} \tag{4.2}$$

Soit $\mathcal{P}_{s,i} = \mathcal{P}^+_{s,i} \cup \mathcal{P}^-_{s,i}$ l'ensemble des paires formé par le voxel $z_{s,i}$. La figure 4.2 dénote la stratégie de sélection des paires pour les voxels annotées.

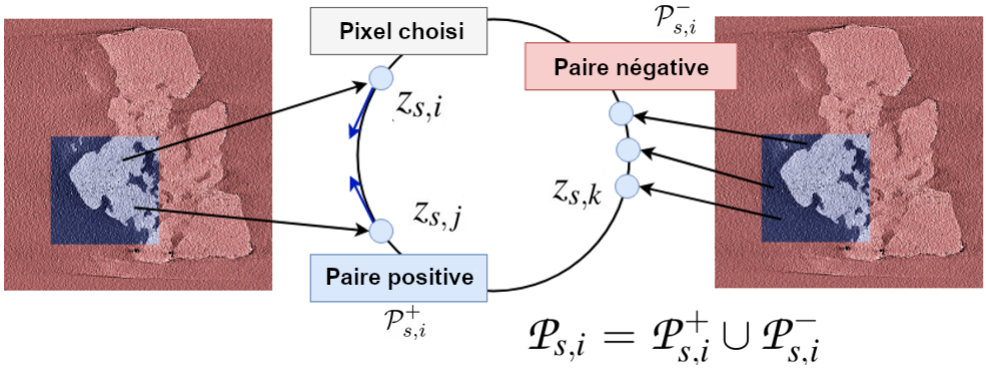


FIGURE 4.2 – Stratégie de sélection de paires pour un voxel annoté $z_{s,i}$.

On note $N_{\mathcal{P}}$ le nombre de paires pour chaque voxel. On définit la fonction de perte contrastive de manière suivante pour un voxel $z_{s,i}$:

$$L(z_{s,i}) = \frac{1}{N_{\mathcal{P}}} \sum_{z_{s,j} \in \mathcal{P}_{s,i}} Y.\text{sim}_1(z_{s,i}, z_{s,j})^2 + (1-Y)\max(1 - \text{sim}_1(z_{s,i}, z_{s,j})^2, 0)^2) \tag{4.3}$$

avec $Y = \begin{cases} 1 & \text{si } z_{s,j} \in P^+_{s,i} \\ 0 & \text{si } z_{s,j} \in P^-_{s,i} \end{cases}$

avec sim_1 une fonction de similarité comme la norme L_2 ou la similarité cosinus. Dans notre cas, nous avons choisi d'utiliser la similarité cosinus, centrée en 0.5, allant de 0 à 1 :

$$\text{sim}_1(u, v) = \frac{1}{2}(1 - \frac{u.v}{||u||.|v||}) \tag{4.4}$$

De plus, il est intéressant de noter qu'il est possible de prendre en compte des données non labellisées avec cette fonction de perte. En effet, les paires positives et négatives peuvent être choisies parmi les voxels annotés $\mathcal{L}_s$. Il est possible de pondérer avec un poids de 0 les voxels non annotés $\mathcal{U}_s$, comme pour l'entropie croisée pondérée. On note la classe "non labellisée" $\emptyset$.

$$\mathcal{L}_s = \{i \in \mathcal{I},\ y_{s,i} \neq \emptyset\}, \quad \mathcal{U}_s = \{i \in \mathcal{I},\ y_{s,i} = \emptyset\} \tag{4.5}$$

Ainsi, la fonction de perte contrastive prenant en compte les voxels non annotés pour un plan Z_s se définit par :

$$L_1(Z_s) = \frac{1}{N_{\mathcal{L}_s}} \sum_{p_{s,i} \in \mathcal{L}_s} L(p_{s,i}) \tag{4.6}$$

Avec $N_{\mathcal{L}_s}$ le cardinal de $\mathcal{L}_s$.

4.2.2 Architecture

Un réseau f de segmentation sémantique est entraîné avec une fonction de perte contrastive au niveau de la sortie. Ainsi, le réseau f est modifié pour avoir en sortie une matrice dans laquelle chaque élément correspond au vecteur du voxel correspondant projeté dans l'espace latent appris. La carte de segmentation finale est obtenue à l'aide d'un classifieur h, lui-même entraîné. Le but de ce classifieur h est de transformer pour chaque voxel le vecteur de l'espace latent en une prédiction de classe. On note $\hat{Y}_s = h(f(X_s))$ la segmentation du plan s. L'encodeur-décodeur f transforme X_s en $Z_s = (z_{s,i})_{i \in I} = f(X_s) \in \mathbb{R}^{W \times H \times D}$ où D est la dimension de l'espace latent. La couche de classification h effectue la prédiction $\hat{Y}_s = h(Z_s)$ tel que $\hat{Y}_s \in \mathbb{R}^{W \times H \times C}$, où C est le nombre de classes [Figure 4.3]. Il est alors nécessaire d'entraîner les deux modules de manière différente. Le choix du classifieur sera discuté dans la partie expérimentale.

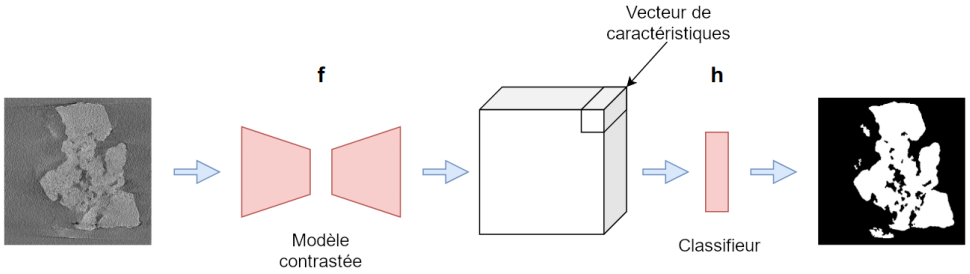


FIGURE 4.3 – Architecture proposée composée d'un réseau f de segmentation et d'un classifieur h. L'image d'entrée est passée dans un réseau de neurones f entraîné avec une fonction de perte contrastive. En sortie, on obtient une carte des caractéristiques dans lesquelles chaque pixel est projeté dans un espace à grande dimension, construit grâce à l'entraînement contrastif. Cette carte des caractéristiques est ensuite passée dans un classifieur h qui permet d'obtenir la carte de segmentation finale.

4.2.3 Gestion des données non annotées

Les réseaux siamois peuvent être utilisés dans une approche auto-supervisée. Dans ce cas, on utilise une version transformée d'un échantillon d'apprentissage pour former une paire positive avec l'échantillon original [34, 15]. Des paires positives peuvent tout de même être construites en constituant une paire avec une version transformée de l'image initiale. Ces deux membres de la paire provenant de la même image, ils représentent la même information. Par conséquent, cette paire est une paire positive. Une paire négative peut être formée avec un autre image. Cependant, il existe un risque que ces deux images soient de la même classe [40].

Dans notre cas, il est ainsi possible de calculer une fonction de perte contrastive sur les pixels non annotés. La sortie du réseau correspond à la projection de chaque pixel dans l'espace de représentation. La projection d'un pixel et de ce même pixel issu de la même image légèrement déformée sont utilisées pour calculer la fonction de perte contrastive pour les pixels dont la classe est inconnue. Soit $\mathcal{T}$ la transformation aléatoire appliquée au plan X_s. La sortie Z_s du plan originel et la sortie de la version transformée $\tilde{Z}_s = f(\mathcal{T}(X_s)) = (\tilde{z}_{s,i})_{i \in I}$ sont comparées. La figure 4.4 résume la stratégie pour les voxels non annotés.

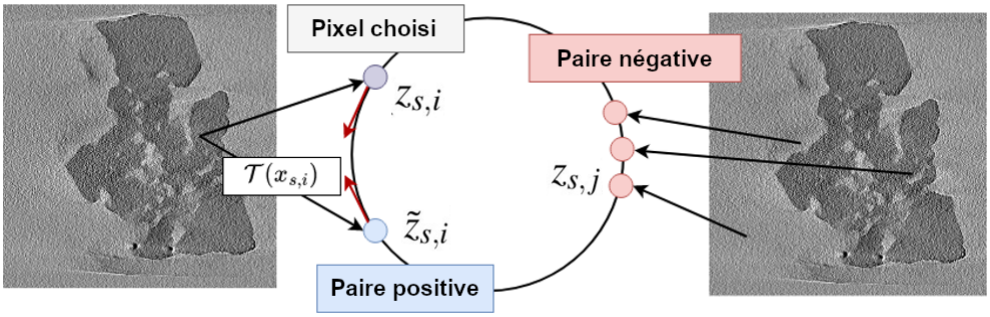


FIGURE 4.4 – Stratégie de sélection de paires pour un voxel non annoté $z_{s,i}$.

La table 4.1 montre les transformations aléatoires utilisées, leur fréquence d'apparition ainsi que leurs paramètres.

Nous avons décidé d'appliquer la fonction de perte utilisée dans les réseaux siamois [34, 15]. On a pour chaque voxel $z_{s,i}$ et sa transformée $\tilde{z}_{s,i}$:

$$L_{\mathcal{U}}(z_{s,i}, \tilde{z}_{s,i}) = -\log \frac{\exp(\text{sim}(z_{s,i}, \tilde{z}_{s,i}))}{\sum_{j \in I, i \neq j} \exp(\text{sim}(z_{s,i}, z_{s,j}))} \tag{4.7}$$

Transformation	Probabilité	Paramètres
Bruit gaussien	1	$\mathcal{N}(0, [0.01; 0.06])$
Décalage des niveaux de gris	0.5	$[-0.01; 0.01]$
Flou gaussien	0.5	$\sigma = [0.5 - 1.5]$

TABLE 4.1 – Table des transformations et de leurs paramètres.

Avec :

$$\text{sim}(u, v) = \frac{u.v}{||u||.||v||} \tag{4.8}$$

Ainsi, une fonction de perte de similarité est calculée pour tous les pixels annotés et non annotés :

$$L_2(Z_s, \tilde{Z}_s) = \frac{1}{N_{\mathcal{L}_s} + N_{\mathcal{U}_s}} \sum_{z_{s,i} \in \mathcal{L}_s \cup \mathcal{U}_s} L_{\mathcal{U}}(z_{s,i}, \tilde{z}_{s,i}) \tag{4.9}$$

Avec $N_{\mathcal{U}_s}$ le cardinal de $\mathcal{U}_s$. Pour que les fonctions de perte soient similaires et plus faciles à combiner, une version modifiée et plus récente de la fonction de perte contrastive dans le contexte supervisé est utilisée [40]. Pour chaque voxel $z_{s,i}$, on a :

$$L_{\mathcal{L}}(z_{s,i}) = -\frac{1}{N_{\mathcal{P}}} \sum_{j \in \mathcal{P}^+_{s,i}} \log \frac{\exp(\text{sim}(z_{s,i}, z_{s,j}))}{\sum\limits_{k \in \mathcal{P}_{s,i}} \exp(\text{sim}(z_{s,i}, z_{s,k}))} \tag{4.10}$$

La fonction de perte pour les voxels annotés pour le plan Z_s s'écrit :

$$L_1(Z_s) = \frac{1}{N_{\mathcal{L}_s}} \sum_{i \in \mathcal{L}_s} L_{\mathcal{L}}(z_{s,i}) \tag{4.11}$$

Il y a donc deux stratégies différentes en fonction du statut d'annotation du voxel. Si l'information de la classe d'un voxel n'est pas disponible, la fonction de perte auto-supervisée (L_2) est utilisée. Pour les voxels dont la classe est connue, les deux fonctions de perte supervisée et auto-supervisée (L_1 et L_2) sont utilisées [Figure 4.5].

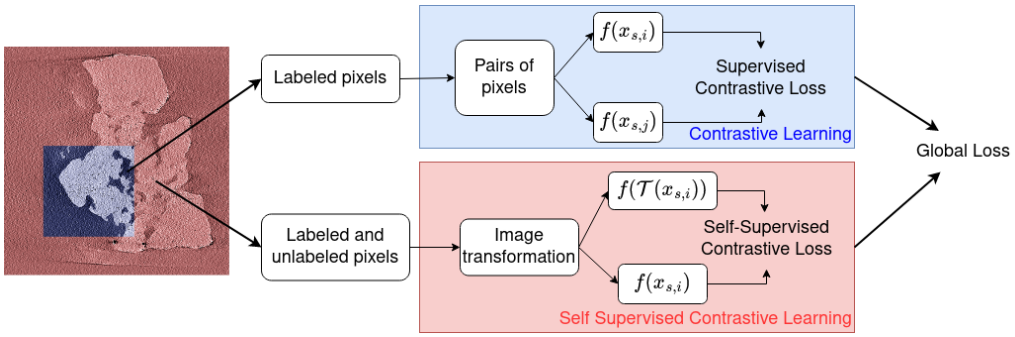


FIGURE 4.5 – Le calcul de la fonction de perte est différent selon la classe des pixels. Pour les pixels labellisés, des paires de pixels sont construites et sont utilisées dans une fonction de perte contrastive supervisée. Pour les pixels labellisés et non labellisé, l'image d'apprentissage et sa version transformée sont utilisées comme paires positives dans une perte contrastive par auto-apprentissage.

Les deux fonctions de perte L_1 (supervisée) et L_2 (auto-supervisée) sont combinées pour calculer la fonction de perte finale pour le plan Z_s :

$$L(Z_s, \tilde{Z}_s) = L_1(Z_s) + L_2(Z_s, \tilde{Z}_s) \tag{4.12}$$

4.3 Expérimentations

Dans cette partie, nous montrons les résultats de notre approche. Nous implémentons notre modèle contrastif en nous basant sur le réseau de segmentation U-Net. Nous considérons le problème de segmentation 3D d'un volume de matériau pour lequel nous ne disposons que d'un petit nombre de voxels annotés, c'est-à-dire moins d'un plan du volume. L'objectif est d'abord d'évaluer les performances de l'approche contrastive semi-supervisée en la comparant avec une approche supervisée classique (entropie croisée). Ensuite, nous étudions l'apport respectif des différents termes de la fonction de perte contrastive. Enfin, nous étudions plusieurs alternatives pour la fonction de classification.

4.3.1 Paramètres expérimentaux

Le réseau de segmentation f utilisé ici est U-Net en raison de sa simplicité et de ses bonnes performances lorsque l'on ne dispose que de peu d'annotations. L'architecture est composée de trois couches de deux convolutions successives

doublant la dimension des vecteurs descripteurs, puis d'une couche de *max pooling* réduisant de moitié la résolution. Le nombre de caractéristiques à la première couche est fixé à 32. Toutes les convolutions sont de taille 3×3. Pour le décodeur, les plans descripteurs sont concaténés au résultat de la couche précédente, suivi d'une convolution. L'augmentation de la résolution est réalisée par une convolution transposée, accompagnée d'une réduction de moitié de la dimension des vecteurs descripteurs. La méthode de référence par entropie croisée comporte une couche soft-max à la sortie du décodeur. Pour l'approche contrastive, la dimension choisie pour l'espace latent est de 16 dimensions, nous modifions ainsi la dernière couche du décodeur et nous supprimons la couche de *softmax* pour obtenir en sortie une projection dans l'espace latent. Le classifieur h est choisi comme étant deux couches convolutives 1x1 suivie d'un *softmax*.

L'entraînement est réalisé avec un seul plan. Néanmoins, nous ne prenons plus systématiquement le plan central pour être certain que ce plan n'apporte pas de biais lors de l'entraînement. Dix plans sont choisis pour être le plan d'entraînement. Les plans ont été sélectionnés entre les plans 200 et 400 du volume, espacés de manière uniforme. Cette façon de sélectionner les plans d'entraînement potentiels permet de garantir que ce ne sont pas les plans du bord, non représentatifs, qui seront sélectionnés pour l'entraînement. On peut également penser qu'un annotateur ne prendra pas un plan du bord pour faire sa labellisation. La validation croisée est utilisée, où chaque plan ne sera pris qu'une fois pour chaque bloc lors de l'entrainement et pendant la validation. Aux termes des cinq blocs de validation croisée, chacune des dix images sélectionnées pour l'entrainement sera utilisé une fois : soit pour l'entraînement, soit pour la validation. Le reste du volume est utilisé pour le test. L'expérience est répétée pour différent taux de labellisation. Chaque expérience est répétée cinq fois et le score d'IOU moyen est calculé.

Les paires sont construites de manière aléatoire. À un pixel objet, 10 paires positives sont tout d'abord construites en sélectionnant aléatoirement 10 autres pixels objets. Puis 10 paires négatives sont construites en sélectionnant 10 pixels fond. L'inverse est effectué pour les pixels fond. Il serait possible de choisir différemment les paires en considérant des pixels plus difficiles à segmenter, par exemple des pixels du fond situés près des bords de l'objet. Nous n'avons pas exploité cette possibilité. Pour calculer la fonction de perte dans le contexte auto-supervisé, seule une seule paire positive peut être choisie. Dix-neuf paires négatives sont sélectionnées.

4.3.2 Comparaison avec un entraînement conventionnel

Nous testons d'abord la méthode pour un volume. Notre méthode produit des résultats satisfaisants, avec un IOU de 0,866 avec seulement 2% d'un plan du volume annoté (soit 7104 voxels sur plus de 200 millions de voxels du volume entier) [Figure 4.6, Table 4.2]. Notre méthode avec une fonction de perte contrastive obtient de meilleurs résultats jusqu'à un taux de labellisation égal à 0,12. À partir de 0,12, les deux méthodes obtiennent des scores similaires pour atteindre un IOU de 0,908 pour un taux de labellisation de 0,25. Cette expérience met en valeur que notre méthode exploite mieux toutes les données disponibles, permettant un temps d'annotation fortement réduit tout en produisant un masque de segmentation précis.

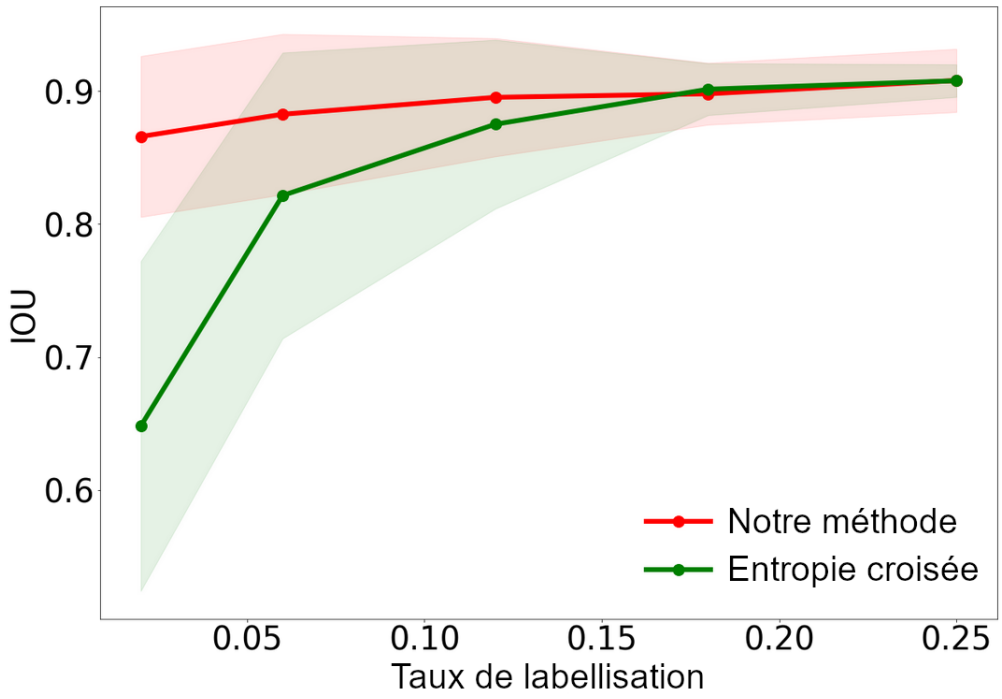


FIGURE 4.6 – Comparaison des IOU pour différents taux de labellisation entre la méthode supervisée par entropie croisée (vert) et notre méthode par fonction de perte contrastive (rouge). Les zones en rouge et vert indiquent l'écart-type mesuré sur l'ensemble des réalisations.

Pour illustrer que cette approche est généralisable, nous avons appliqué notre méthode sur d'autres volumes à notre disposition. La table 4.3 montre les score d'IOU des différents volumes pour différents taux de labellisation ainsi

r	0.02	0.06	0.12	0.18	0.25
Entropie croisée	0.648	0.821	0.875	**0.902**	**0.908**
Contrastive	**0.866**	**0.883**	**0.895**	0.898	**0.908**

TABLE 4.2 – Comparaison des IOU pour différents taux de labellisation entre la méthode supervisée par entropie croisée et notre méthode par fonction de perte contrastive.

	Zéolithe 1	Zéolithe 2	Alumine
Entropie croisée $r = 0.06$	0.821	0.219	0.128
Contrastive $r = 0.06$	**0.883**	**0.443**	**0.533**
Entropie croisée $r = 0.25$	**0.908**	0.392	0.196
Contrastive $r = 0.25$	**0.908**	**0.595**	**0.729**

TABLE 4.3 – Résultats sur différents volumes.

qu'une comparaison avec U-Net. La figure 4.10 représente les différentes reconstructions 3D de ces volumes. Les résultats obtenus sont dans le pire des cas équivalent à celui obtenu par un entraînement conventionnel, et en général supérieur, même avec seulement 6% de données annotées sur un seul plan.

4.3.3 Choix du classifieur

Le choix du classifieur a une importance dans notre approche. Nous avons testé deux classifieurs :

- Un classifieur linéaire comme une machine à vecteurs de support (*Support Machine Vector* ou SVM) est généralement utilisée pour discriminer la classe des différentes images [88, 34].
- Deux couches convolutives avec un noyau 1x1. Cette couche transforme la carte des caractéristiques de la dimension de la dernière couche convolutive (64 pour le U-Net original) en une carte des caractéristiques de dimension égale au nombre de classes. Ainsi, grâce à l'entropie croisée [Équation 2.9], pour chaque pixel de l'image, le réseau prédit la probabilité d'appartenir à chaque classe.

Avec les mêmes données d'entrainement, partiellement annoté de manière identique, un modèle composé d'un encodeur-décodeur suivi d'un classifieur SVM a été comparé avec le même encodeur-décodeur avec une couche de classification. Dans les deux cas, l'encodeur-décodeur est entrainé avant le module de classification. Dans le cas du décodeur convolutif concaténé avec

l'encodeur-décodeur, le gradient est stoppé avant l'encodeur-décodeur f, pour ne modifier que le classifieur. La figure 4.7 montre que la version avec une couche de classification est plus stable et apporte de meilleurs résultats lorsqu'il y a peu de pixels annotés. Pour un nombre de données limité, il est plus facile d'entraîner une couche de convolution plutôt qu'un SVM standard.

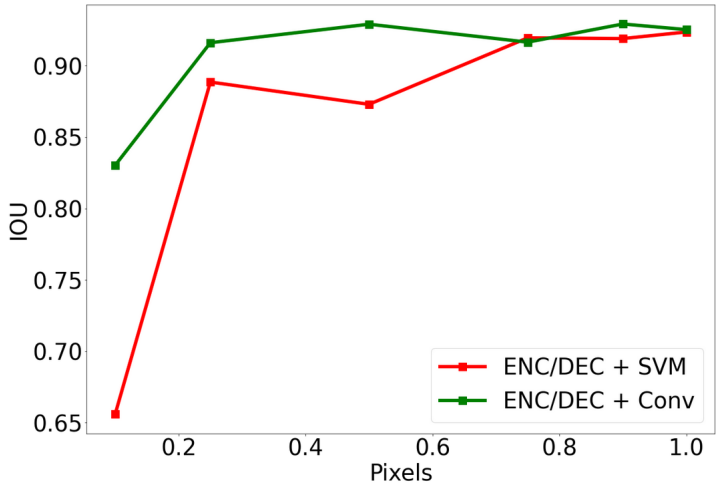


FIGURE 4.7 – Comparaison entre différents modules de classification avec en rouge un classifieur SVM et en vert un classifieur intégré à l'encodeur-décodeur avec une couche de classification. On remarque que pour un nombre de pixels annoté faible, il est plus facile d'entraîner une couche de convolution qu'un SVM.

4.3.4 Apport des fonctions de perte

Nous avons également étudié la contribution de chaque fonction de perte. Les résultats montrent que les deux fonctions de perte apportent un gain notable sur la segmentation obtenue [Figure 4.8].

La même expérience est effectuée avec une fonction de perte différente :

- Uniquement avec la fonction de perte contrastive dans le cadre supervisé (L_1). Les résultats ne sont pas satisfaisants lorsque peu de données sont fournies. Cependant, lorsqu'il y a plus de données étiquetées, la perte supervisée donne de meilleurs résultats grâce à des paires de pixels plus diversifiées.
- Uniquement avec la fonction de perte contrastive dans le cadre auto-supervisé (L_2). Les résultats sont bons, même pour un faible taux de labellisation : l'espace latent est bien appris avec seulement quelques

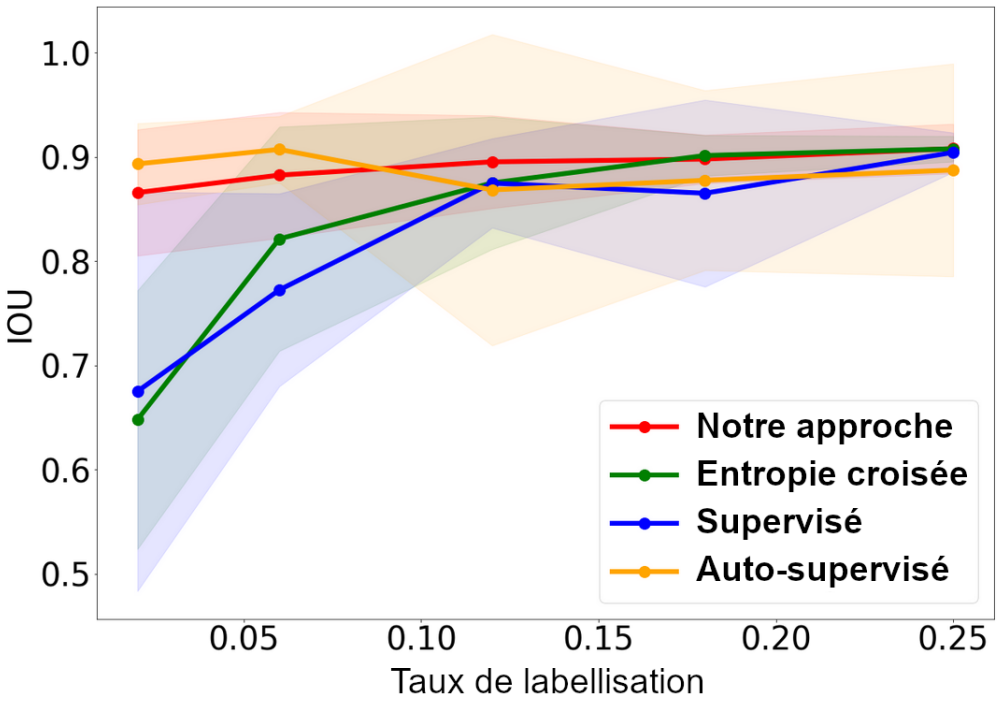


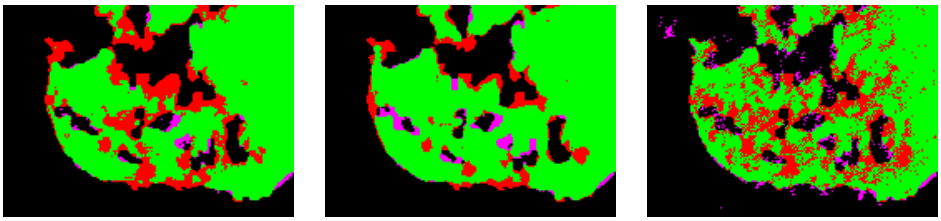
FIGURE 4.8 – Comparaison des IOU avec différentes fonctions de pertes : l'entropie croisée pondérée, la fonction de perte contrastive supervisée, la fonction de perte contrastive auto-supervisée et la combinaison des fonctions de perte supervisée et non supervisée.

pixels annotés. Lorsque le taux de labellisation augmente, seule la couche de classification bénéficie des données supplémentaires. Il en résulte une faible amélioration à mesure que le taux de labellisation augmente.

- Avec les deux fonctions de perte (L_1+L_2), de bons résultats sont obtenus avec un faible taux d'étiquetage, tout en maintenant de meilleurs résultats lorsque la quantité de données étiquetées augmente.

La perte contrastive supervisée et auto-supervisée est nécessaire pour obtenir la meilleure segmentation. De plus, la dispersion des résultats est réduite quand les deux fonctions de perte sont utilisées conjointement. Ce n'est pas le cas lorsqu'une seule des fonctions de perte est utilisée.

La figure 4.9 montre des résultats qualitatifs des différentes méthodes. S'il n'y a que la fonction de perte contrastive dans le cadre supervisé (L_1), le résultat de la segmentation est bruité, mais parvient à segmenter des régions plus difficiles. S'il n'y a que la fonction de perte contrastive dans le cadre



Auto-supervisé Hybride Supervisé

FIGURE 4.9 – Résultats des segmentations pour un taux de labellisation $r = 0.06$ avec différentes fonctions de pertes. Les pixels verts sont les pixels corrects, les pixels rouges sont les pixels manquants à la segmentation et les pixels violets sont les pixels en trop par rapport à la vérité terrain.

auto-supervisé, la carte de segmentation est moins bruitée, mais il y a des zones manquantes au niveau du bord. En combinant les deux méthodes, il y a beaucoup moins de bruit et davantage de zones sont correctement segmentées.

4.4 Conclusion

Nous avons montré dans ce chapitre une nouvelle approche pour la segmentation de volume de matériaux mésoporeux issus de la tomographie électronique. Cette méthode permet la segmentation automatique de volumes entiers de zéolithe avec uniquement une partie infime du volume à annoter pour l'expert, accélérant considérablement le temps d'annotation. La procédure est la suivante :

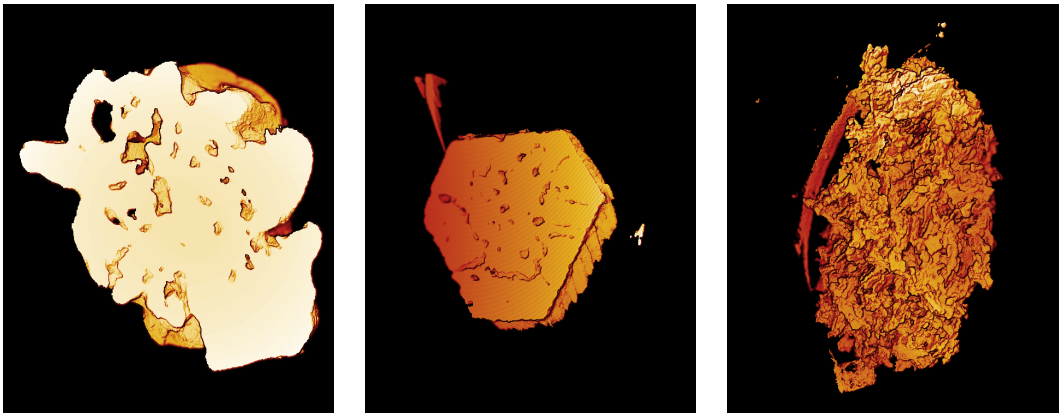
1. L'annotateur annote une petite portion d'un plan.
2. Le réseau de neurones est entraîné avec une approche contrastive avec à la fois les données annotées et non annotées.
3. Le classifieur est entraîné avec uniquement les données annotées.
4. Pour chaque plan du volume, le réseau projette chaque voxel dans l'espace latent appris par la perte contrastive.
5. Le classifieur prédit la classe de chaque voxel.

On obtient ainsi une carte de segmentation pour l'ensemble du volume avec très peu de données annotées grâce notre stratégie d'entraînement. La stratégie diffère pour les voxels labellisés et les voxels non labellisés :

- Pour les voxels annotés, des paires positives et négatives sont construites grâce à l'information apportée par l'annotation. Une fonction de perte contrastive est calculée sur ces voxels.

- Pour les voxels annotés et non annotés, le plan d'entraînement est transformé et une paire positive est construite avec un voxel du plan originel et celui du plan transformé. Le reste des paires négatives sont construites avec des voxels provenant du reste du volume. Une fonction de perte contrastive est calculée sur ces voxels.

La somme de ces deux fonctions de perte est minimisée. Cette approche permet d'obtenir de très bons résultats sur les données testées par rapport à la littérature pour de très faible taux de labellisations (inférieur à 10% d'un plan). Les travaux présents dans ce chapitre ont été présentés à la conférence internationale VISAPP [45].



Zéolithe 1 Zéolithe 2 Alumine

FIGURE 4.10 – Visualisation des reconstructions 3D de différents volumes.

5

Hijacked-STM : un réseau de segmentation de vidéo détourné pour de la segmentation semi-supervisé de matériaux

Outline

5.1 Introduction 82
5.1.1 Potentiels de la segmentation d'objet vidéo pour les données volumiques 83
5.1.2 Segmentation d'objet vidéo semi-supervisé 84
5.1.3 Segmentation interactive 85
5.2 Détournement d'un réseau à mémoire 86
5.2.1 Procédure de propagation des annotations 86
5.2.2 Encodage Clé-Valeur 87
5.2.3 Lecture partielle de la mémoire 89
5.3 Expérimentation 91
5.3.1 Dispositif expérimental 91
5.3.2 Comparaison avec STM 92
5.3.3 Propagation de la segmentation 93
5.3.4 Comparaison avec des méthodes de segmentation . 94
5.4 Conclusion 94

Dans le chapitre précédent, nous avons mis en place une méthode pour segmenter un volume de matériaux issu de tomographie électronique. Un entraînement contrastif a été mis en place. Cette étape d'entraînement peut prendre un temps non négligeable. Ce chapitre introduit une méthode permettant la segmentation sémantique de volumes issus de tomographie électronique, sans la nécessité d'une étape d'entraînement. Ce gain de performance permet de mettre en place des méthodes de segmentation interactives. Les approches de segmentation interactives permettent un dialogue entre une méthode de segmentation et un annotateur, notamment avec des demandes de correction. C'est pourquoi une méthode nécessitant un temps de segmentation court est important.

Nous étudions le domaine de la segmentation vidéo, où de nombreuses méthodes de segmentation sémantique interactive ont fait leurs preuves. Dans un premier temps, nous nous intéressons à l'analogie qu'il y a entre une vidéo et un volume. Puis, nous détaillons l'architecture des réseaux à mémoire qui seront utilisés ainsi que l'adaptation effectuée dans le cadre de la segmentation sémantique de volume partiellement annoté. Enfin, les expérimentations réalisées seront présentées.

Dans ce chapitre :

- Une nouvelle méthode semi-supervisée de segmentation de volume est développée, réutilisant un réseau STM de segmentation d'objet pré-entraîné. Cette approche ne nécessite aucun entraînement supplémentaire.
- Un module de lecture de la mémoire a été mis en place pour fournir une image d'interrogation partiellement segmentée lors de l'étape d'inférence.
- Des expérimentations détaillées sur plusieurs données de tomographie électronique ont été effectuées, montrant qu'une segmentation précise est possible avec un seul plan requête annoté avec seulement une très petite quantité des pixels.

5.1 Introduction

Le domaine de la segmentation sémantique de vidéo est proche de celui de la segmentation sémantique de volumes. Dans ce domaine, de nombreuses méthodes permettent la segmentation interactive de vidéos, qui n'ont pas encore été mises en place dans le domaine de la segmentation sémantique volumique. Les données vidéos et volumiques partagent des similarités. Il est alors possible d'étudier les méthodes utilisées en segmentation vidéo et

en segmentation interactive pour mettre en place une nouvelle approche de segmentation sémantique de volumes issus de la tomographie électronique.

5.1.1 Potentiels de la segmentation d'objet vidéo pour les données volumiques

Les vidéos et les données volumiques ont de nombreux points communs :

- Une vidéo est une suite d'images animées. Chaque vidéo se décompose alors en une succession d'images bidimensionnelles. Un volume comporte des données sur trois dimensions. En choisissant deux dimensions, le volume peut être décomposé en plans bidimensionnels. Vidéos et volumes peuvent être découpés en pile d'images 2D [Figure 5.1].
- Entre deux images d'une vidéo et deux plans d'un volume, il existe une continuité entre les images voisines. Les variations entre deux plans sont minimes et il est possible d'extrapoler des informations d'une image à l'autre.
- De nouvelles caractéristiques peuvent éventuellement apparaître au cours d'une vidéo ou du volume.

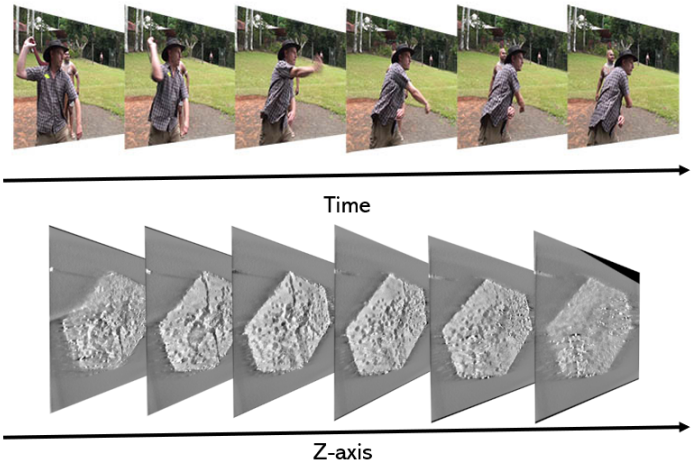


FIGURE 5.1 – Similarités entre vidéos et volumes. Les deux types de données peuvent être interprétés comme une pile d'images successives avec une continuité d'une image à l'autre.

Tous ces points communs montrent qu'une adaptation d'une méthode implémentée pour la segmentation sémantique de vidéos peut également être appliquée pour la segmentation volumique, comme le montrent de récents travaux [91, 92, 51]. Ce champ d'étude est appelé segmentation d'objet vidéo.

Cependant, les méthodes vidéos concernent des scènes de la vie courante. Ce type de données est très différent de données volumiques qui sont plus spécifiques. Dans la prochaine partie, une étude sur les méthodes de segmentation classique sera effectuée.

5.1.2 Segmentation d'objet vidéo semi-supervisé

Dans le domaine de la segmentation d'objet vidéo, des progrès significatifs ont été réalisés en utilisant des réseaux à mémoire [64, 87, 19]. Dans une configuration semi-supervisée, l'utilisateur fournit en entrée une image requête et son masque de segmentation puis le système segmente l'ensemble de la vidéo. Une image requête annotée est passée au système et fournit une segmentation complète de la vidéo correspondante. L'image requête est généralement la première image de la vidéo, segmentée manuellement par un annotateur. Cette configuration présente l'avantage d'être indépendante de la classe (*class-agnostic*), c'est-à-dire que le système n'a pas appris spécifiquement la classe de l'objet. C'est le masque fournit avec l'image requête qui guide la segmentation. Les méthodes *class-agnostic* ne nécessitent aucun apprentissage supplémentaire pour de nouvelles images. Les réseaux à mémoire encodent l'image annotée dans un module de mémoire et segmentent les autres images en utilisant cette mémoire [82]. En général, les images de la mémoire sont encodées sous la forme d'une carte de clés et d'une carte de valeurs. Les cartes de clés et valeurs sont des vecteurs caractéristiques permettant respectivement d'identifier un patch de l'image pour la clé et de guider la segmentation pour la valeur. Les réseaux à mémoire de segmentation tels que STM [64], SwiftNet [87] ou STCN [19] encodent la première image vidéo, annotée par l'utilisateur, dans le module de mémoire. L'image suivante (requête) est codée à son tour sous la forme d'une carte de clés et d'une carte de valeurs. Le vecteur clé de chaque patch de la requête est comparé à tous les patchs issus des images de la mémoire. Un vecteur de valeurs est ensuite calculé en combinant les valeurs des patchs de la mémoire ressemblant le plus au patch de la requête. Ce vecteur valeur est ensuite utilisé pour segmenter l'objet sur cette image. Le module de mémoire est alors complété avec la nouvelle clé et la nouvelle valeur [Figure 5.3]. Cette technique est souvent utilisée dans la segmentation vidéo, car l'objet à segmenter, dont les formes changent au fil du temps, est constamment ajouté à la mémoire, fournissant plusieurs exemples pour aider à la segmentation [64].

Dans notre cas, plusieurs questions se posent pour pouvoir utiliser un réseau mémoire sur nos images volumiques. Premièrement, est-ce qu'un réseau pré-entraîné pour la vidéo sera efficace sur nos images de matériaux ?

Deuxièmement, comment prendre en compte une première image qui ne serait que partiellement segmentée ?

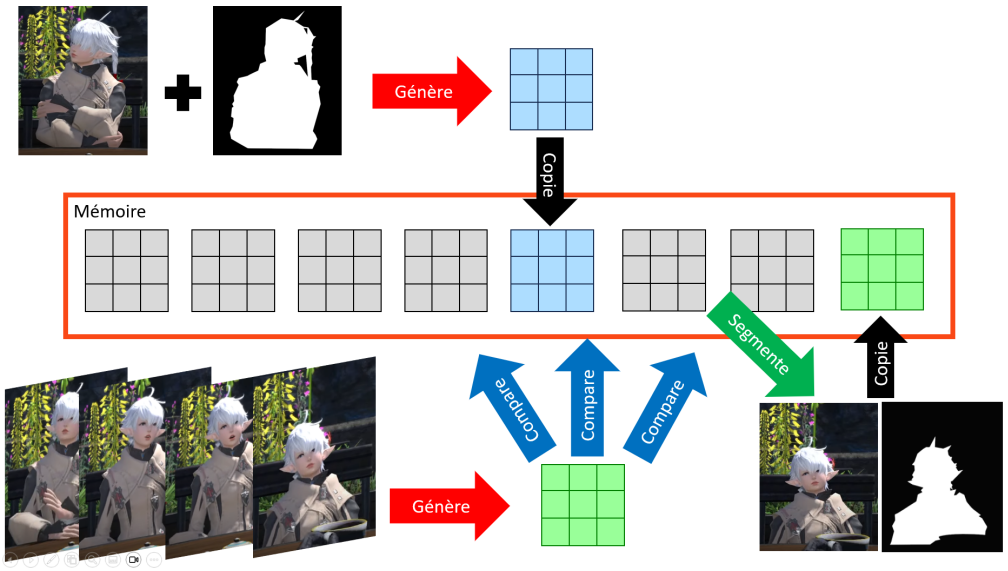


FIGURE 5.2 – Principe d'un réseau à mémoire pour la segmentation sémantique de vidéo. Une image et son masque sont encodés dans la mémoire. L'image requête est encodée puis comparée aux données dans la mémoire pour produire une prédiction de masque. Le résultat est stocké dans la mémoire.

5.1.3 Segmentation interactive

Les réseaux à mémoire sont efficaces, mais nécessitent la segmentation de l'ensemble de la première image. Une boucle d'interaction peut être ajoutée, dans laquelle l'utilisateur est invité à segmenter la première image par des clics ou des gribouillages afin d'annoter l'objet d'intérêt pour le réseau [18]. L'utilisateur apporte des corrections jusqu'à ce qu'il soit satisfait du résultat. Les réseaux pour la segmentation interactive sont des modèles de segmentation sémantique standards formés avec un canal image, un canal masque et un canal interaction [56]. En combinant le réseau de segmentation interactive pour la première image et le réseau de segmentation d'objets à mémoire pour propager le masque, des travaux récents produisent un masque de segmentation pour l'ensemble de la vidéo avec une contribution minimale de l'utilisateur [18]. Des méthodes similaires ont été appliquées à la segmentation volumique [91, 92, 51]. Cependant, pour des volumes poreux complexes imagés par tomographie électronique, les méthodes interactives standard peinent à segmen-

ter correctement ces images. L'adaptation d'un modèle interactif nécessite des données d'entraînement composées de nombreux volumes segmentés qui ne sont pas disponibles pour les images de tomographie électronique. Nous proposons une approche similaire aux méthodes interactives en utilisant un plan partiellement segmentée. Notre méthode ne nécessite pas d'entraînement préalable.

FIGURE 5.3 – Différentes étapes d'une étape de segmentation interactive. Tout d'abord, une prédiction de segmentation est produite. L'annotateur indique des corrections, qui seront effectuées par la méthode de segmentation interactive [16].

5.2 Détournement d'un réseau à mémoire

Dans ce chapitre, nous proposons de détourner un réseau à mémoire utilisé dans le domaine vidéo (*Space-Time Memory network* ou STM), pré-entraîné pour la segmentation sémantique de vidéos, afin de l'utiliser pour segmenter des volumes de tomographie électronique avec seulement quelques voxels annotés d'un plan du volume. La structure du réseau est modifiée pour ne prendre que quelques voxels d'un plan comme requête à l'étape de l'inférence et ne nécessite aucun entraînement. À notre connaissance, c'est la première fois que ce type de réseau de segmentation d'objets vidéo pré-entraîné est utilisé pour segmenter des images de tomographie électronique.

5.2.1 Procédure de propagation des annotations

Notre méthode utilise le même modèle pour reconstruire l'image partiellement annotée et le volume entier. Les images et les masques dans la mémoire sont stockés sous forme de matrices de vecteurs de caractéristiques clé et valeur. Le vecteur clé encode une représentation visuelle d'un patch, de sorte que les patchs ayant des vecteurs clés similaires ont des formes et des textures similaires. Le vecteur valeur contient des informations pour le décodeur sur la segmentation du patch.

Nous demandons à un expert d'annoter une petite partie d'une tranche A_s du volume V et l'objectif est d'obtenir la segmentation $\hat{Y}$ du volume entier. À partir des annotations données par l'expert, un masque binaire d'étiquetage M_s est construit où les voxels annotés et non annotés sont dénotés. L'image et les annotations sont codées en vecteurs clé et valeur $\{k^M, v^M\}$ stockées dans la mémoire. L'annotation est alors propagée dans le reste du volume.

Il existe deux façons de propager l'annotation dans l'ensemble du volume.

- **Propagation par patchs** La carte clés et les valeurs du plan requête partiellement annotée V_s est insérée en mémoire. La mémoire est directement utilisée pour segmenter tous les patchs issus des autres plans du volume. Dans ce cas, uniquement les voxels annotés sont mis en mémoire [Algorithme 1 et Figure 5.4].

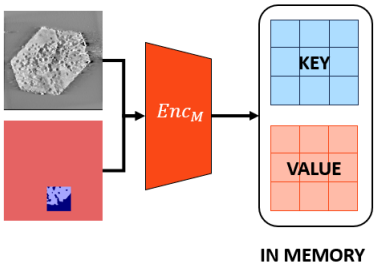


FIGURE 5.4 – Propagation par patchs : seuls les voxels annotés contribuent à la mémoire.

- **Propagation par frame** Le plan requête partiellement annoté V_s est transmis au réseau pour obtenir des pseudo-annotations de la tranche entière $\hat{Y}_s$. La tranche entière et le masque de pseudo-étiquettes nouvellement acquis sont ensuite codés en vecteurs clé et une valeur $\{k^S, v^S\}$ pour segmenter d'autres plans $V_i, i \in [1, N]$ avec N le nombre de plans dans le volume. Le plan entier reconstruit est mis en mémoire [Algorithme 2 et Figure 5.5].

5.2.2 Encodage Clé-Valeur

L'encodage de la mémoire et de la requête sont légèrement différents [Figure 5.6]. Nous considérons, pour un ensemble d'images $I \in \mathbb{R}^{H \times W}$ et leurs annotations $A \in \mathbb{R}^{H \times W}$, l'encodeur mémoire Enc_M composé d'un réseau convolutif, suivi de deux couches convolutives parallèles, le vecteur clé mémoire $k^M \in \mathbb{R}^{H/8 \times W/8 \times C/8}$ et le vecteur valeur mémoire $v^M \in \mathbb{R}^{H/8 \times W/8 \times C/2}$ tel que :

Algorithm 1 Procédure de segmentation de l'ensemble du volume avec une portion d'un seul plan annoté avec une propagation par patchs.

Require: V, s, A_s, M_s, N
Ensure: $\hat{Y}$
  $\{k^M, v^M\} \leftarrow Enc_M(V_s, A_s)$
  while $i \in \{1, ..., N\}$ **do**
    $\{k^Q, v^Q\} \leftarrow Enc_Q(V_i)$
    $f \leftarrow PartialMemoryRead(M_s, k^M, k^Q, v^M, v^Q)$
    $\hat{Y}_i \leftarrow Decoder(V_i, f)$
  end while

Algorithm 2 Procédure de segmentation de l'ensemble du volume avec une portion d'un seul plan annoté avec une propagation par frame.

Require: V, s, A_s, M_s, N
Ensure: $\hat{Y}$
  $\{k^M, v^M\} \leftarrow Enc_M(V_s, A_s)$ ▷ Reconstruction du premier plan
  $\{k^Q, v^Q\} \leftarrow Enc_Q(V_s)$
  $f \leftarrow PartialMemoryRead(M_s, k^M, k^Q, v^M, v^Q)$
  $\hat{Y}_s \leftarrow Decoder(V_s, f)$
  $\{k^S, v^S\} \leftarrow Enc_M(V_s, \hat{Y}_s)$
  while $i \in \{1, ..., s-1\} \cup \{s+1, ..., N\}$ **do** ▷ Propagation dans tout le volume
    $\{k^Q, v^Q\} \leftarrow Enc_Q(V_i)$
    $f \leftarrow MemoryRead(k^S, k^Q, v^S, v^Q)$
    $\hat{Y}_i \leftarrow Decoder(V_i, f)$
  end while

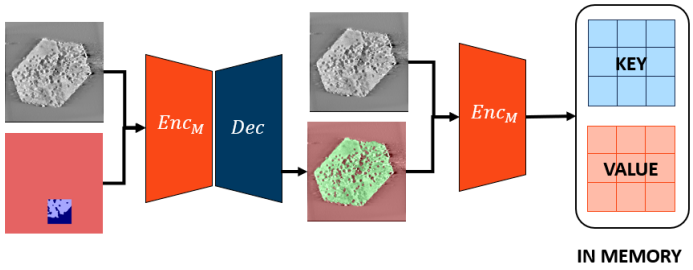


FIGURE 5.5 – Propagation par frame : le plan reconstruit est encodé dans la mémoire.

$$Enc_M(I, A) = \{k^M, v^M\} \tag{5.1}$$

où W et H sont la taille de l'image et C le nombre de dimensions des vecteurs de caractéristiques à la sortie de l'encodeur mémoire.

L'encodeur de requêtes Enc_Q partage la même architecture que l'encodeur mémoire Enc_M avec des poids différents, mais comme le masque de l'image n'est pas disponible, seul le plan est utilisé par l'encodeur de requêtes :

$$Enc_Q(I) = \{k^Q, v^Q\} \tag{5.2}$$

avec $k^Q \in \mathbb{R}^{H/8 \times W/8 \times C/8}$ le vecteur clé requête et $v^Q \in \mathbb{R}^{H/8 \times W/8 \times C/2}$ le vecteur valeur requête.

5.2.3 Lecture partielle de la mémoire

Notre intuition est que si nous désactivons les zones contenant des données non annotées dans la mémoire, nous pouvons encoder des informations utiles pour segmenter le volume entier, même avec une petite portion d'un plan annoté. La méthode standard de lecture de la mémoire du réseau STM est modifiée pour prendre en compte des images partiellement annotées. Soit $M \in \mathbb{R}^{H \times W}$ le masque d'annotation :

$$M_{i,j} = \begin{cases} 0 & \text{si } A_{i,j} \text{ n'est pas annoté} \\ 1 & \text{si } A_{i,j} \text{ est annoté} \end{cases} \tag{5.3}$$

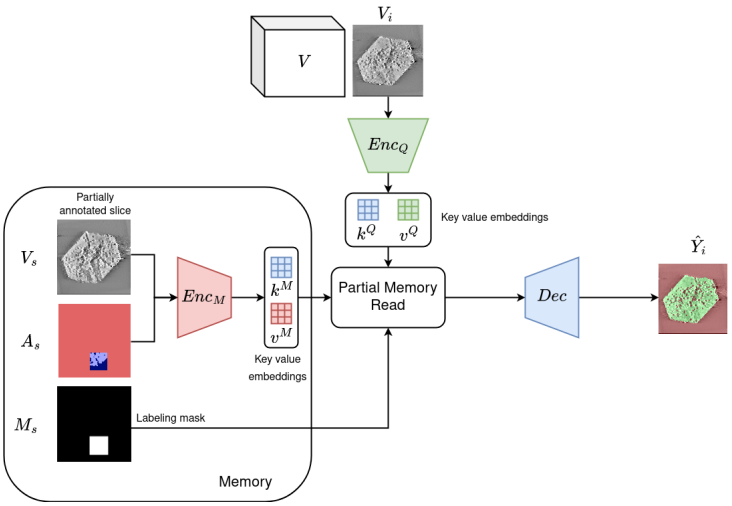


FIGURE 5.6 – Architecture de la méthode. Un encodeur mémoire et requête produisent des vecteurs clé et valeur qui sont utilisés pour la lecture de la mémoire et permettent de guider le décodeur.

Le masque M est ensuite sous-échantillonné pour être compatible avec les vecteurs clé de la mémoire :

$$M^D = Downsample(M, 8) \in \mathbb{R}^{H/8 \times W/8} \tag{5.4}$$

Une interpolation bilinéaire est utilisée pour lisser les bords. Ensuite, la carte de similarité $S \in \mathbb{R}^{HW/16 \times HW/16}$ est calculée entre le vecteur clé mémoire remodelés $k_r^M \in \mathbb{R}^{HW/16 \times C/8}$ et le vecteur clé requête $k_r^Q \in \mathbb{R}^{HW/16 \times C/8}$:

$$S = k_r^Q \times k_r^{M^T} \tag{5.5}$$

où $\times$ est le produit matriciel. Un softmax est ensuite appliqué à S. Cependant, nous masquons S avec M^D pour annuler la contribution de la clé mémoire k^M des zones non annotées. Étant donné que S est le produit matriciel de k^Q et k^M, pour masquer correctement k^M, M^D est remodelé en $M^L \in \mathbb{R}^{HW/16}$. Nous multiplions ensuite chaque ligne de S par M^L :

$$R_{i,j} = \frac{\exp(S_{i,j}) M_i^L}{\sum_{k,l} \exp(S_{k,l}) M_k^L} \tag{5.6}$$

La matrice résultante $R \in \mathbb{R}^{HW/16 \times HW/16}$ est la similarité entre chaque zone de k^M et k^Q sans la contribution des zones non annotées. La clé de segmentation $f \in \mathbb{R}^{H/8 \times W/8 \times C}$ est obtenue en concaténant le vecteur valeur de la requête et le vecteur valeur de la mémoire, pondérés par R [Figure 5.7].

$$f = [v^Q, R \times v^M] \tag{5.7}$$

où $\times$ désigne le produit matriciel.

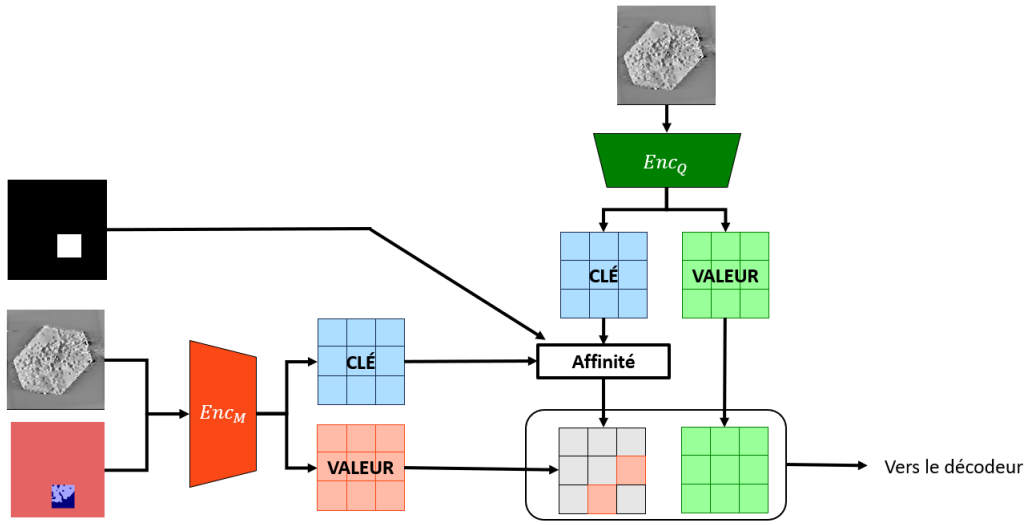


FIGURE 5.7 – Schéma de la lecture partielle de la mémoire.

5.3 Expérimentation

Dans cette section, l'objectif est d'évaluer la méthode sur les données de tomographie électronique avec les différents types de propagation. Tout d'abord, le dispositif expérimental sera détaillé, puis les résultats seront présentés. Une étude comparant les différents types de propagation (propagation par patch ou propagation par frame) sera également menée. Enfin, une comparaison avec les méthodes du chapitre 4 sera effectuée.

5.3.1 Dispositif expérimental

Nous utilisons l'architecture du réseau STM [64] ainsi que les poids proposés par les auteurs de la méthode. Les différents encodeurs sont composés

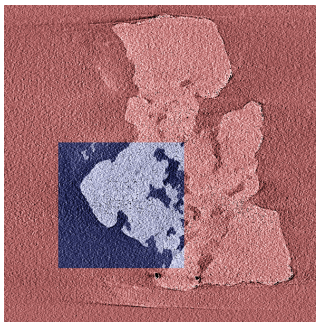


FIGURE 5.8 – Un plan partiellement annoté d'une zéolithe (1 nm/pixel). Une fenêtre d'une surface A_w est considérée comme annotée, alors que la classe des voxels à l'extérieur de cette fenêtre est inconnue. Le centre de la fenêtre est choisi au hasard près de la frontière entre l'objet et l'arrière-plan afin d'inclure des pixels des deux classes.

d'un réseau ResNet [36], entraîné pour une tâche de segmentation vidéo avec Youtube-VOS [89] et DAVIS [71]. Le décodeur produit une carte de probabilité qui est 1/4 de la taille initiale de l'entrée, ce qui dégrade les résultats d'images de tomographie électronique où des détails fins sur les zones poreuses sont nécessaires. Une opération de sur-échantillonnage est effectuée sur le plan d'entrée. Le plan d'entrée est sur-échantillonné d'un facteur deux, ce qui constitue un compromis entre la consommation de mémoire et la finesse des détails.

Les plans sont automatiquement partiellement annotés pour simuler les données du monde réel. Seule une fenêtre rectangulaire de la surface A_w est annotée. Le reste du plan n'est pas considéré comme annoté. Le centre de la fenêtre est choisi aléatoirement sur la frontière entre l'objet et l'arrière-plan pour être certain d'annoter le fond ainsi que l'objet [Figure 5.8]. La fenêtre est ajustée pour s'adapter aux bords de l'image tout en conservant sa surface. Nous définissons le taux de labellisation comme $r = \frac{A_w}{H \times W}$.

Tous les résultats sont évalués avec l'IOU sur l'ensemble du volume V. Pour chaque volume, nous effectuons chaque expérience sur les mêmes cinq plans sélectionnés au hasard. La moyenne des IOU des trois volumes est affichée.

5.3.2 Comparaison avec STM

Nous comparons d'abord notre méthode à un réseau STM non modifié pour plusieurs taux de labellisation r. Nous donnons les mêmes plans partiellement annotés pour le réseau STM et notre méthode. Les résultats sont présentés

TABLE 5.1 – Moyenne des IOU sur nos volumes pour notre méthode et un réseau STM non modifié. La modification de notre approche permet à un modèle de type STM de produire une bonne segmentation.

r	0.02	0.06	0.12	0.18	0.25
Ours - Zéolithe 1	**0.447**	**0.487**	**0.594**	**0.666**	**0.744**
STM - Zéolithe 1	0.002	0.003	0.006	0.018	0.068
Ours - Zéolithe 2	**0.578**	**0.672**	**0.706**	**0.714**	**0.723**
STM - Zéolithe 2	0.015	0.032	0.096	0.215	0.391
Ours - Zéolithe 3	**0.626**	**0.717**	**0.779**	**0.794**	**0.809**
STM - Zéolithe 3	0.021	0.038	0.108	0.218	0.396
Ours - Moyenne	**0.551**	**0.625**	**0.693**	**0.725**	**0.758**
STM - Moyenne	0.013	0.024	0.070	0.150	0.285

dans le tableau 5.1. Notre méthode atteint rapidement un IOU de 0,65 tandis que la méthode originale échoue complètement et peine à atteindre un IOU de 0,3. Le réseau STM ne peut pas traiter la tranche partiellement étiquetée, car il n'a pas été conçu pour traiter ce type de données et les portions non annotées sont traitées comme fond. Par conséquent, le réseau STM n'a aucun moyen de différencier les pixels étiquetés et non étiquetés, ce qui entraîne une mauvaise segmentation. Notre masquage des clés permet au réseau STM d'obtenir de bien meilleurs résultats avec une segmentation précise.

5.3.3 Propagation de la segmentation

Nous étudions ensuite les différentes méthodes de propagations. Nous avons testé la méthode avec seulement les parties annotées dans la mémoire (propagation par patchs, algorithme 1) et la méthode avec les pseudo-labels du premier plan en mémoire (propagation par frame, algorithme 2). Les résultats du tableau 5.2 démontrent que de meilleurs résultats sont obtenus avec seulement la portion annotée dans la mémoire. Pour un nombre très faible de données annotées, la propagation par frame obtient de légèrement meilleurs scores que la propagation par patchs. Cependant, la propagation par patchs atteint de meilleurs scores à partir d'un taux de labellisation de 0,06. Le réseau STM est plus performant avec des données précises plutôt qu'avec une plus grande variété de données en mémoire. Nous retenons la propagation par patch.

TABLE 5.2 – IOU moyen sur nos volumes pour notre méthode avec seulement les parties étiquetées dans la mémoire (propagation par patchs) et notre méthode avec les pseudo-labels du premier plan dans la mémoire (propagation par frame). Notre approche n'utilise que les zones annotées en mémoire.

r	0.02	0.06	0.12	0.18	0.25
Propagation par patchs - Zéolithe 1	**0.447**	**0.487**	**0.594**	**0.666**	**0.744**
Propagation par frame - Zéolithe 1	0.202	0.315	0.389	0.341	0.328
Propagation par patchs - Zéolithe 2	**0.578**	**0.672**	**0.706**	**0.714**	**0.723**
Propagation par frame - Zéolithe 2	0.392	0.577	0.595	0.686	0.681
Propagation par patchs - Zéolithe 3	0.626	**0.717**	**0.779**	**0.794**	**0.809**
Propagation par frame - Zéolithe 3	**0.650**	0.696	0.651	0.725	0.727
Propagation par patchs - Moyenne	0.550	**0.625**	**0.693**	**0.725**	**0.758**
Propagation par frame - Moyenne	**0.564**	0.588	0.650	0.671	0.710

5.3.4 Comparaison avec des méthodes de segmentation

Enfin, nous comparons nos méthodes avec les approches du chapitre 4 qui peuvent également traiter des plans partiellement annotés. Nous étudions les performances de notre méthode par rapport à un réseau de neurones U-Net entraîné uniquement sur les zones étiquetées avec une entropie croisée pondérée et un réseau de neurones U-Net qui utilise l'apprentissage contrastif pour exploiter les zones non étiquetées similaire à la méthode proposée dans le chapitre 4. Les deux méthodes nécessitent une phase d'entraînement. Les résultats sont présentés dans le tableau 5.3. Notre méthode est plus performante que la méthode U-Net standard entraîné avec une entropie croisée pondérée, mais reste en retrait par rapport à la méthode U-Net avec un entrainement contrastif. Néanmoins, notre méthode sans entraînement montre des résultats prometteurs avec des scores proches des méthodes qui nécessitent une étape d'entraînement. La figure 5.9 montre tous les résultats des expériences mentionnées précédemment.

5.4 Conclusion

Dans ce chapitre, nous avons établi une méthode de segmentation sémantique de volumes issus de la tomographie électronique. Notre méthode s'approprie une approche de segmentation vidéo, les réseaux à mémoire :

- Les réseaux à mémoire fonctionnent avec des encodeurs de clés et de

TABLE 5.3 – IOU moyen sur nos volumes pour notre méthode, un U-Net adapté aux zones partiellement segmentées, et un U-Net utilisant une fonction de perte contrastive pour exploiter à la fois les zones annotées et non annotées. La méthode que nous proposons obtient des résultats proches de ces méthodes malgré l'absence de phase d'entraînement.

r	0.02	0.06	0.12	0.18	0.25
Notre méthode - Zéolithe 1	0.447	0.487	0.594	0.666	0.744
U-Net - Zéolithe 1	0.651	0.821	0.875	**0.902**	**0.908**
U-Net contrastif - Zéolithe 1	**0.839**	**0.883**	**0.895**	0.898	**0.908**
Notre méthode - Zéolithe 2	0.578	**0.672**	0.706	**0.714**	0.723
U-Net - Zéolithe 2	0.357	0.363	0.442	0.446	**0.861**
U-Net contrastif - Zéolithe 2	**0.589**	0.636	**0.710**	0.710	0.692
Notre méthode - Zéolithe 3	0.626	0.717	0.779	0.794	0.809
U-Net - Zéolithe 3	0.623	0.617	0.696	0.738	0.747
U-Net contrastif - Zéolithe 3	**0.782**	**0.786**	**0.773**	**0.832**	**0.847**
Notre méthode - Moyenne	0.551	0.625	0.693	0.725	0.758
U-Net - Moyenne	0.544	0.600	0.671	0.695	**0.839**
U-Net contrastif - Moyenne	**0.737**	**0.768**	**0.793**	**0.813**	0.815

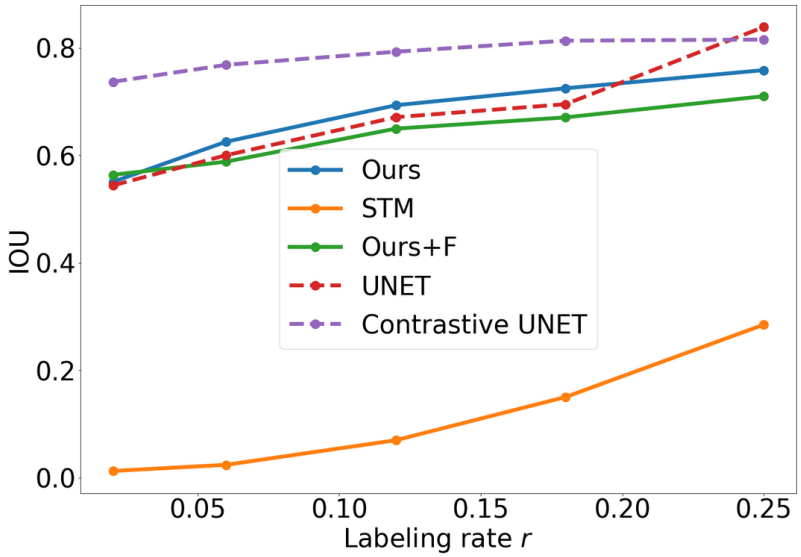


FIGURE 5.9 – Moyenne des IOU pour plusieurs taux de labellisation r. Toutes les méthodes ne nécessitent pas de procédure d'apprentissage supplémentaire, à l'exception de U-Net et de la version contrastive de U-Net. [45].

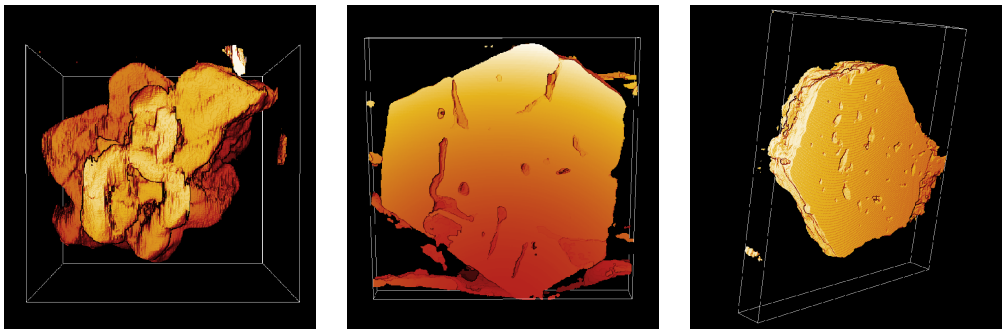


FIGURE 5.10 – Visualisation 3D de volumes segmentés de zéolithes NaX Siliporite G5. Une fenêtre aléatoire de 6% d'une tranche a été annotée. Les segmentations sont fournies en utilisant notre approche (Algorithme 1).

valeurs. Les clés permettent une reconnaissance des différents patterns présents dans l'image tandis que les valeurs sont fournies au décodeur pour guider la segmentation.

- Lorsqu'une nouvelle image requête doit être segmentée, les vecteurs de caractéristiques clé et valeur sont construits. Le vecteur clé est comparé avec tous les vecteurs clé présents dans la mémoire et une clé de segmentation est produite à partir des vecteurs valeur de la requête et de la mémoire, en fonction de la similarité avec les différents vecteurs clé.
- La mémoire est mise à jour régulièrement au fur et à mesure que le volume est segmenté. Ainsi, la mémoire reflète les changements du volume.

Cette méthode a été modifiée pour prendre en compte des plans partiellement annotés comme image requête. Un masque est utilisé lors de la lecture de la mémoire pour ne prendre en compte que les voxels qui ont une annotation. Cette modification permet d'obtenir de très bons résultats de segmentation sur notre jeu de données, même avec très peu de données annotées. De plus, cette approche ne nécessite aucun entraînement, accélérant le temps de segmentation de la méthode. Cette approche est un premier pas vers une méthode interactive de segmentation de volume de tomographie électronique. Les résultats présentés dans ce chapitre ont fait l'objet d'une publication [44].

6

Conclusions et perspectives

Outline

6.1 Contributions 98
6.1.1 Évaluation des données de tomographie électronique 98
6.1.2 Apprentissage contrastif pour la segmentation de volumes 99
6.1.3 Segmentation semi-supervisée utilisant des réseaux à mémoire 99
6.2 Perspectives 100
6.2.1 Évaluation sur d'autres types de données 100
6.2.2 Adaptation de contexte pour les vecteurs clés . . . 100
6.2.3 Implantation d'une méthode interactive de segmentation 101

Dans le contexte de la production de biocarburants, la catalyse est un processus important. La microstructure des catalyseurs utilisés a un impact direct sur les performances de catalyse. Les techniques de tomographie électronique permettent d'imager la structure interne à l'échelle nanoscopique et d'observer la mésoporosité de ces matériaux. Afin de caractériser la performance de différents catalyseurs, il est nécessaire d'obtenir une carte de segmentation. Annoter manuellement le volume du catalyseur est une tâche fastidieuse. C'est pourquoi des méthodes automatiques ont été développées, notamment dans le domaine de l'apprentissage profond pour la vision par ordinateur. Dans ce document, nous avons mis en place plusieurs méthodes d'apprentissage profond dans le but d'assister un expert lors de l'annotation de nouveaux volumes.

6.1 Contributions

6.1.1 Évaluation des données de tomographie électronique

Dans le chapitre 3, après une étude avancée des données de tomographie électronique à notre disposition, nous avons montré qu'un encodeur-décodeur classique de type U-Net était capable de fournir de bons résultats sur les données disponibles. Les erreurs d'annotations de la vérité terrain peuvent être prises en compte dans une méthode d'entraînement que nous avons introduit, utilisant seulement des images partiellement labellisées. Cette expérience a montré que U-Net peut être entraîné avec des données partiellement annotées. Entraîner une méthode de segmentation sémantique de cette façon nécessite beaucoup moins de données préalablement annotées.

Concernant les tests menés sur les données volumiques de notre base de données, nous montrons qu'un réseau U-Net entraîné sur plusieurs plans d'un volume fournit des résultats de segmentation satisfaisants sur les autres plans de ce volume. Par exemple, en annotant seulement deux plans d'un volume de taille $592 \times 600 \times 623$, nous obtenons un IOU de 0,77. Cette configuration s'adapte très bien à la problématique. Un expert voulant annoter un nouveau volume fournit quelques annotations manuelles du volume. Puis, un encodeur-décodeur est entraîné sur les données disponibles, afin de fournir un masque de segmentation du volume entier. Cependant, lorsque moins d'un plan est labellisé, cette approche ne donne pas de bons résultats. Il est alors nécessaire de changer de méthode afin d'obtenir une segmentation satisfaisante.

6.1.2 Apprentissage contrastif pour la segmentation de volumes

Dans le chapitre 4, nous avons mis en place une méthode semi-supervisée de segmentation volumique de matériaux issus de la tomographie électronique. La méthode comprend deux modules. Le premier module projette les voxels d'une image dans un espace latent construit avec un apprentissage contrastif. Le deuxième module, entraîné avec une fonction de perte de type entropie croisée pondérée, classifie chaque voxel en voxel fond ou voxel objet.

La fonction de perte associée à chaque voxel doit pouvoir prendre en compte les voxels annotés et non annotés. Dans le cas des voxels non labellisés, l'information de classe n'est pas disponible. La fonction de perte contrastive dans le domaine auto-supervisé est utilisée, à partir de paires de voxels. Des paires positives sont formées à partir d'un voxel et de sa version transformée. Des paires négatives sont formées par association avec d'autres voxels du plan. Dans le cas où le voxel est labellisé, sa classe d'appartenance est disponible. La fonction de perte contrastive dans le domaine supervisé est utilisée en conjonction avec la fonction de perte contrastive dans le domaine auto-supervisé. Des paires positives et négatives sont formées avec d'autres voxels dont l'information de classe est disponible.

Cette manière de procéder permet d'exploiter toutes les données disponibles et la méthode fournit de très bons résultats de segmentation, même avec très peu de données annotées en entraînement. Par exemple, en annotant seulement 0,06% d'un plan d'un volume de taille $592 \times 600 \times 623$, nous obtenons un IOU de 0,883. Les travaux présents dans ce chapitre ont été présentés à la conférence internationale VISAPP [45]

6.1.3 Segmentation semi-supervisée utilisant des réseaux à mémoire

Dans le chapitre 5, nous présentons une méthode semi-supervisé de segmentation sémantique de tomogrames inspirée des approches de segmentation d'objet vidéo. Un réseau mémoire est déployé. Ce type d'approche s'appuie sur un mécanisme de mémoire où est sauvegardée une référence de l'objet à segmenter. Pour chaque plan à segmenter, une comparaison avec la mémoire est effectuée pour guider la segmentation. Au fur et à mesure que les plans du volume sont segmentés, la mémoire est mise à jour pour tenir compte des différences entre les plans du volume.

Nous avons proposé une amélioration portant sur la lecture de la mémoire

pour gérer correctement les plans partiellement annotés. Ainsi, pour l'annotateur, il n'est plus nécessaire de labelliser entièrement un premier plan à insérer dans la mémoire. Cette méthode s'appuie sur un réseau pré-entraîné et ne nécessite aucun entraînement supplémentaire. Elle permet d'obtenir un masque de segmentation précis avec seulement une infime partie d'un plan ($\leq$ 10% d'un plan) annoté et mis en mémoire. Par exemple, en annotant seulement 0,06% d'un plan, nous obtenons une moyenne de 0,625 d'IOU sur l'ensemble des volumes disponibles. Les résultats présentés dans ce chapitre ont fait l'objet d'une publication dans le congrès ACIVS [44].

6.2 Perspectives

Bien que nous ayons présenté et discuté des méthodes de segmentation développées pour des images issues de tomographie électronique, il existe des axes d'améliorations qui peuvent faire l'objet d'études plus approfondies.

6.2.1 Évaluation sur d'autres types de données

Nos méthodes ont été conçues pour des images issues de la tomographie électronique. Cependant, d'autre type de données pourraient bénéficier d'une telle architecture comme en imagerie médicale ou pour la segmentation de vidéos par exemple. De nombreuses bases de données publiques peuvent être utilisées pour tester notre méthode dans un autre contexte comme le challenge de segmentation de tumeur de cerveau BraTS [1]. Nous pouvons également utiliser nos méthodes de labellisation partielle sur des bases de données de segmentation classique comme PASCAL VOC 2012 [28].

6.2.2 Adaptation de contexte pour les vecteurs clés

Notre méthode de segmentation à l'aide d'un réseau à mémoire reprend les poids fournis par les auteurs du réseau originel [64]. Ce réseau est entraîné sur une base de données vidéo très importante. Le contenu de ces vidéos est très différent des images volumiques issues de la tomographie électronique. Des approches plus récentes proposent une étape d'adaptation rapide du module d'encodage des vecteurs clés [27]. Un court entraînement sur le premier plan fourni permet d'adapter les encodeurs du réseau pour mieux encoder l'objet à segmenter. Dans notre cas où les images d'entraînement et les images à segmenter sont extrêmement différents, cette étape de raffinement permettrait une adaptation du réseau sur des données de tomographie électronique.

6.2.3 Implantation d'une méthode interactive de segmentation

Bien que nos méthodes reprennent la configuration de la segmentation interactive, elles ne sont pas véritablement interactives. Actuellement, un annotateur peut fournir la labellisation d'une petite partie du volume pour obtenir une segmentation complète, mais il n'est pas possible ensuite de suggérer des corrections au modèle. Dans le domaine vidéo, les modules d'interaction que nous avons testés [56, 18] fournissent de mauvais résultats sur des images volumiques de catalyseurs. De plus, ces modèles nécessitent un nombre important de données d'entraînement [56] et il est difficile d'apprendre ce type de module sur des données de tomographie électronique. Ce domaine reste ouvert au développement d'une méthode de segmentation interactive.

Bibliographie

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution : Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[2] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning : Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8(1) :1–74, 2021.

[3] Nicolas Audebert, Alexandre Boulch, Bertrand Le Saux, and Sébastien Lefèvre. Segmentation sémantique profonde par régression sur cartes de distances signées. In *Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP)*, Marne-la-Vallée, France, June 2018.

[4] Jeanne Ayache, Luc Beaunier, Jacqueline Pottu-Boumendil, Gabrielle Ehret, Danièle Laub, and Stéphane Mottin. *Guide de préparation des échantillons pour la microscopie électronique en transmission, tome1.* Number 9 in Intégrations [des savoirs et des savoir-faire]. MRCT-CNRS, 2007. pour le champs 'auteurs', les Directeurs de publications apparaissent en premier puis le Directeur de collection.

[5] Vijay Badrinarayanan, Fabio Galasso, and Roberto Cipolla. Label propagation in video sequences. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3265–3272. IEEE, 2010.

[6] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec : A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning*, pages 1298–1312. PMLR, 2022.

[7] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11700–11709, 2019.

[8] E Bousias Alexakis and C Armenakis. Evaluation of unet and unet++ architectures in high resolution image change detection applications. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43 :1507–1514, 2020.

[9] Krishna Chaitanya, Ertunc Erdil, Neerav Karani, and Ender Konukoglu. Contrastive learning of global and local features for medical image segmentation with limited annotations. *Advances in Neural Information Processing Systems*, 33 :12546–12558, 2020.

[10] Bike Chen, Chen Gong, and Jian Yang. Importance-aware semantic segmentation for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 20(1) :137–148, 2019.

[11] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab : Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4) :834–848, 2018.

[12] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. 06 2017.

[13] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan Yuille. Attention to scale : Scale-aware semantic image segmentation. 11 2015.

[14] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 833–851, Cham, 2018. Springer International Publishing.

[15] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. 02 2020.

[16] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758, June 2021.

[17] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758, June 2021.

[18] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation : Interaction-to-mask, propagation and difference-aware fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5559–5568, 2021.

[19] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in Neural Information Processing Systems*, 34 :11781–11794, 2021.

[20] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net : Learning dense volumetric segmentation from sparse annotation. In Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pages 424–432, Cham, 2016. Springer International Publishing.

[21] Anthony Cioppa, Adrien Deliège, and Marc Droogenbroeck. A bottom-up approach based on semantics for the interpretation of the main camera stream in soccer games. 06 2018.

[22] Stanislas Dehaene. *Apprendre ! : les talents du cerveau, le défi des machines*. Odile Jacob, 2018.

[23] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6) :141–142, 2012.

[24] Emily Denton, Sam Gross, and Rob Fergus. Semi-supervised learning with context-conditional generative adversarial networks. *arXiv preprint arXiv :1611.06430*, 2016.

[25] Carl Doersch, Abhinav Gupta, and Alexei Efros. Unsupervised visual representation learning by context prediction. 05 2015.

[26] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words : Transformers for image recognition at scale. *arXiv preprint arXiv :2010.11929*, 2020.

[27] Isidore Dubuisson, Damien Muselet, Christophe Ducottet, and Jochen Lang. Fast context adaptation for video object segmentation. In *International Conference on Computer Analysis of Images and Patterns*, pages 273–283. Springer, 2023.

[28] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http ://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[29] Lingling Fang, Tianshuang Qiu, Hongyang Zhao, and Fang Lv. A hybrid active contour model based on global and local information for medical image segmentation. *Multidimensional Systems and Signal Processing*, 30 :689–703, 2019.

[30] Damien Fourure, Rémi Emonet, Elisa Fromont, Damien Muselet, Alain Tremeau, and Christian Wolf. Segmentation de scènes extérieures à partir d'ensembles d'étiquettes à granularité et sémantique variables. In *RFIA 2016*, 2016.

[31] Damien Fourure, Rémi Emonet, Elisa Fromont, Damien Muselet, Alain Tremeau, and Christian Wolf. Residual Conv-Deconv Grid Network for Semantic Segmentation. In *BMVC 2017*, Londre, United Kingdom, September 2017.

[32] Golnaz Ghiasi and Charless Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. volume 9907, pages 519–534, 10 2016.

[33] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[34] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot,

Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap Your Own Latent : A new approach to self-supervised learning. In *Neural Information Processing Systems*, Montréal, Canada, 2020.

[35] Adam Hammoumi, Maxime Moreaud, Christophe Ducottet, and Sylvain Desroziers. Adding geodesic information and stochastic patch-wise image prediction for small dataset learning. *Neurocomputing*, 456 :481–491, 2021.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[37] Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, and Jian Wu. Unet 3+ : A full-scale connected unet for medical image segmentation. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 1055–1059. IEEE, 2020.

[38] Avinash C Kak and Malcolm Slaney. *Principles of computerized tomographic imaging*. SIAM, 2001.

[39] Daria Kern, Ulrich Klauck, Timo Ropinski, and André Mastmeyer. 2D vs. 3D U-Net abdominal organ segmentation in CT data using organ bounds. In Thomas M. Deserno and Brian J. Park, editors, *Medical Imaging 2021 : Imaging Informatics for Healthcare, Research, and Applications*, volume 11601, pages 192 – 200. International Society for Optics and Photonics, SPIE, 2021.

[40] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33 :18661–18673, 2020.

[41] Navid Alemi Koohbanani, Mostafa Jahanifar, Neda Zamani Tajadin, and Nasir Rajpoot. Nuclick : a deep learning framework for interactive segmentation of microscopic images. *Medical Image Analysis*, 65 :101771, 2020.

[42] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[43] T.H.N. Le, Gia Quach, Khoa Luu, Chi Nhan Duong, and Marios Savvides. Reformulating level sets as deep recurrent neural network approach to semantic segmentation. *IEEE Transactions on Image Processing*, PP :1–1, 01 2018.

[44] Cyril Li, Christophe Ducottet, Sylvain Desroziers, and Maxime Moreaud. Less-than-one shot 3d segmentation hijacking a pre-trained space-time memory network. In *Advanced Concepts for Intelligent Vision Systems, ACIVS 2023*, 2023.

[45] Cyril Li, Christophe Ducottet, Sylvain Desroziers, and Maxime Moreaud. Toward few pixel annotations for 3D segmentation of material from electron tomography. In *International Conference on Computer Vision Theory and Applications, VISAPP 2023*, 2023.

[46] Xiaomeng Li, Lequan Yu, Hao Chen, Chi-Wing Fu, and Pheng-Ann Heng. Semi-supervised skin lesion segmentation via transformation consistent self-ensembling model. *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 63, 2018.

[47] Sheng Lian, Zhiming Luo, Zhun Zhong, Xiang Lin, Songzhi Su, and Shaozi Li. Attention guided u-net for accurate iris segmentation. *Journal of Visual Communication and Image Representation*, 56 :296–304, 2018.

[48] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco : Common objects in context. In *Computer Vision–ECCV 2014 : 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[49] Zheng Lin, Zheng-Peng Duan, Zhao Zhang, Chun-Le Guo, and Ming-Ming Cheng. Focuscut : Diving into a focus view in interactive segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2637–2646, 2022.

[50] Geert Litjens, Robert Toth, Wendy Van De Ven, Caroline Hoeks, Sjoerd Kerkstra, Bram Van Ginneken, Graham Vincent, Gwenael Guillard, Neil Birbeck, Jindang Zhang, et al. Evaluation of prostate segmentation algorithms for mri : the promise12 challenge. *Medical image analysis*, 18(2) :359–373, 2014.

[51] Qin Liu, Zhenlin Xu, Yining Jiao, and Marc Niethammer. iSegFormer : Interactive segmentation via transformers with application to 3D

knee MR images. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022 : 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part V*, pages 464–474. Springer, 2022.

[52] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8759–8768, Los Alamitos, CA, USA, jun 2018. IEEE Computer Society.

[53] Wei Liu, Andrew Rabinovich, and Alexander Berg. Parsenet : Looking wider to see better. 06 2015.

[54] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. pages 3431–3440, 06 2015.

[55] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic Segmentation using Adversarial Networks. In *NIPS Workshop on Adversarial Training*, Barcelona, Spain, December 2016.

[56] Sabarinath Mahadevan, Paul Voigtlaender, and Bastian Leibe. Iteratively trained interactive segmentation. In *British Machine Vision Conference (BMVC)*, 2018.

[57] Satoru Masubuchi, Eisuke Watanabe, Yuta Seo, Shota Okazaki, Takao Sasagawa, Kenji Watanabe, Takashi Taniguchi, and Tomoki Machida. Deep-learning-based image segmentation integrated with optical microscopy for automatically searching for two-dimensional materials. *npj 2D Materials and Applications*, 4(1) :1–9, 2020.

[58] Joe McGlinchy, Brian Johnson, Brian Muller, Maxwell Joseph, and Jeremy Diaz. Application of unet fully convolutional neural network to impervious surface segmentation in urban environment from high resolution satellite imagery. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 3915–3918. IEEE, 2019.

[59] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net : Fully convolutional neural networks for volumetric medical image segmentation. pages 565–571, 10 2016.

[60] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning : A survey. 2020.

[61] Werner Nagel. Image analysis and mathematical morphology. volume 2 : Theoretical advances. edited by jean serra. *Journal of Microscopy*, 152 :597–597.

[62] David Nilsson and Cristian Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6819–6828, 2018.

[63] Richard Nock and Frank Nielsen. Statistical region merging. *IEEE transactions on pattern analysis and machine intelligence*, 26 :1452–8, 12 2004.

[64] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019.

[65] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1) :62–66, 1979.

[66] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1) :62–66, 1979.

[67] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12674–12684, 2020.

[68] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders : Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[69] Jizong Peng, Guillermo Estrada, Marco Pedersoli, and Christian Desrosiers. Deep co-training for semi-supervised image segmentation. *Pattern Recognition*, 107 :107269, 2020.

[70] Jizong Peng, Guillermo Estrada, Marco Pedersoli, and Christian Desrosiers. Deep co-training for semi-supervised image segmentation. *Pattern Recognition*, 107 :107269, 2020.

[71] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016.

[72] Nils Plath, Marc Toussaint, and Shinichi Nakajima. Multi-class image segmentation using conditional random fields and global classification. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 817–824, New York, NY, USA, 2009. Association for Computing Machinery.

[73] Johann Radon. Ueber die bestimmung von funktionen durch ihre integralwerte langs gewisser mannigfaltigkeiten. *Mathematisch-Physische Klasse*, 69 :262–277, 1917.

[74] Waleed Ragheb, Mehdi Mirzapour, Ali Delfardi, Hélène Jacquenet, and Lawrence Carbon. Emotional speech recognition with pre-trained deep visual models, 04 2022.

[75] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alexei Efros, and Sergey Levine. Few-shot segmentation propagation with guided networks. 05 2018.

[76] Terry Regier. The emergence of words : Attentional learning in form and meaning. *Cognitive Science*, 29(6) :819–865, 2005.

[77] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn : Towards real-time object detection with region proposal networks. 2016.

[78] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net : Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[79] Andrej Sali, Robert Glaeser, Thomas Earnest, and Wolfgang Baumeister. From words to literature in structural proteomics. *Nature*, 422(6928) :216–225, 2003.

[80] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*, 2014.

[81] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose : a generalist algorithm for cellular segmentation. *Nature Methods*, 18(1) :100–106, 2021.

[82] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. *Advances in neural information processing systems*, 28, 2015.

[83] Zheng Tong, Philippe Xu, and Thierry Denœux. Evidential fully convolutional network for semantic segmentation. *Applied Intelligence*, 51, 09 2021.

[84] Viet Dung Tran. *Reconstruction et segmentation d'image 3D de tomographie électronique par approche" problème inverse"*. PhD thesis, Université Jean Monnet-Saint-Etienne, 2013.

[85] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. Video segmentation via object flow. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3899–3908, 2016.

[86] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[87] Haochen Wang, Xiaolong Jiang, Haibing Ren, Yao Hu, and Song Bai. Swiftnet : Real-time video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1296–1305, 2021.

[88] Jianxin Wu. Efficient hik svm learning for image classification. *IEEE Transactions on Image Processing*, 21(10) :4442–4453, 2012.

[89] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas S. Huang. Youtube-vos : A large-scale video object segmentation benchmark. *CoRR*, abs/1809.03327, 2018.

[90] Yading Yuan, Ming Chao, and Yeh-Chi Lo. Automatic skin lesion segmentation using deep fully convolutional networks with jaccard distance. *IEEE transactions on medical imaging*, 36(9) :1876–1886, 2017.

[91] Tianfei Zhou, Liulei Li, Gustav Bredell, Jianwu Li, and Ender Konukoglu. Quality-aware memory network for interactive volumetric image

segmentation. In Marleen de Bruijne, Philippe C. Cattin, Stéphane Cotin, Nicolas Padoy, Stefanie Speidel, Yefeng Zheng, and Caroline Essert, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, pages 560–570, Cham, 2021. Springer International Publishing.

[92] Tianfei Zhou, Liulei Li, Gustav Bredell, Jianwu Li, Jan Unkelbach, and Ender Konukoglu. Volumetric memory network for interactive medical image segmentation. *Medical Image Analysis*, 83 :102599, 2023.