N° d'ordre NNT : 2026ISAL0001

THESE de DOCTORAT DE L'INSA LYON, membre de l'Université de Lyon

Ecole Doctorale N° 160
Électronique, Électrotechnique, Automatique

Spécialité/ discipline de doctorat :
Traitement du Signal et de l'Image

Soutenue publiquement le 19/01/2026, par :

Maylis Jouvencel

Medical image segmentation through learning on unstructured 3D data

Devant le jury composé de :

HEINRICH	Mattias	Professor	University of Lübeck	Rapporteur
NAEGEL	Benoît	PU	Université de Strasbourg	Rapporteur
DESVIGNES	Michel	PE	Université Grenoble Alpes	Examinateur
DIGNE	Julie	DR	Université Claude Bernard Lyon 1	Présidente du Jury
MATEUS	Diana	PU	Centrale Nantes	Examinatrice
VALETTE	Sébastien	CR	INSA Lyon	Directeur de thèse
KÉCHICHIAN	Razmig	MCF	INSA Lyon	Co-directeur de thèse

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Département FEDORA – INSA Lyon - Ecoles Doctorales

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
ED 206 CHIMIE	**CHIMIE DE LYON** **https ://www.edchimie-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL 7, Avenue Jean Capelle 69621 Villeurbanne CEDEX **secretariat@edchimie-lyon.fr**	**M. Stéphane DANIELE** C2P2-CPE LYON-UMR 5265 Bâtiment F308, BP 2077 43 Boulevard du 11 novembre 1918 69616 Villeurbanne CEDEX **directeur@edchimie-lyon.fr**
ED 341 E2M2	**ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION** **http ://e2m2.universite-lyon.fr** Sec. : Bénédicte LANZA Bâtiment Atrium, 43, Boulevard du 11 Novembre 1918 69622 Villeurbanne CEDEX Tél : 04.72.44.83.62 **secretariat.e2m2@univ-lyon1.fr**	**Mme Sandrine CHARLES** Université Claude Bernard Lyon 1 UFR Biosciences, Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69622 Villeurbanne CEDEX **e2m2.codir@listes.univ-lyon1.fr**
ED 205 EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTÉ** **http ://ediss.universite-lyon.fr** Sec. : Bénédicte LANZA Bâtiment Atrium, 43, Boulevard du 11 Novembre 1918 69622 Villeurbanne CEDEX Tél : 04.72.44.83.62 **secretariat.ediss@univ-lyon1.fr**	**M. Jérôme LAMARTINE** Université Claude Bernard Lyon 1 Equipe Intégrité Fonctionnelle du Tissu Cutané LBTI - UMR5305 CNRS IBCP - 7 passage du Vercors, 69367 Lyon CEDEX 07 Tel : 04.72.72.26.66 **jerome.lamartine@univ-lyon1.fr**
ED 34 EDML	**MATÉRIAUX DE LYON** **http ://ed34.universite-lyon.fr** Sec. : Yann DE ORDENANA Tél : 04.72.18.62.44 **yann.de-ordenana@ec-lyon.fr**	**M. Stéphane BENAYOUN** Ecole Centrale de Lyon Laboratoire LTDS 36 avenue Guy de Collongue 69134 Ecully CEDEX Tél : 04.72.18.64.37 **stephane.benayoun@ec-lyon.fr**
ED 160 EEA	**ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE** **https ://edeea.universite-lyon.fr** Sec. : Philomène TRECOURT Bâtiment Charlotte PERRIAND 69621 Villeurbanne CEDEX Tél : 04.72.43.71.70 **secretariat.edeea@insa-lyon.fr**	**M. Philippe DELACHARTRE** INSA LYON Laboratoire CREATIS Bâtiment Blaise Pascal, 7 avenue Jean Capelle 69621 Villeurbanne CEDEX Tél : 04.72.43.88.63 **philippe.delachartre@insa-lyon.fr**
ED 512 INFOMATHS	**INFORMATIQUE ET MATHÉMATIQUES** **http ://edinfomaths.universite-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL 7, Avenue Jean Capelle 69621 Villeurbanne CEDEX **infomaths@univ-lyon1.fr**	**M. Dragos IFTIMIE** Université Claude Bernard Lyon 1 Bâtiment Braconnier 21 Avenue Claude BERNARD 69 622 Villeurbanne CEDEX Tél : 04.72.44.79.58 **direction.infomaths@listes.univ-lyon1.fr**
ED 162 MEGA	**MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE** **http ://edmega.universite-lyon.fr** Sec. : Philomène TRECOURT Bâtiment Charlotte PERRIAND 69621 Villeurbanne CEDEX Tél : 04.72.43.71.70 **mega@insa-lyon.fr**	**M. Etienne PARIZET** INSA Lyon Laboratoire LVA Bâtiment St. Exupéry 25 bis av. Jean Capelle 69621 Villeurbanne CEDEX **etienne.parizet@insa-lyon.fr**
ED 483 ScSo	**ScSo**[1] **https ://edsciencessociales.universite-lyon.fr** Sec. : Mélina FAVETON Tél : 04.78.69.72.76 Université Lyon 2 Campus Berges du Rhône, Bureau BEL 309 18, Quai Claude BERNARD 69365 LYON CEDEX 07 **melina.faveton@univ-lyon2.fr**	**M. Bruno MILLY** Univ. Lyon 2 Campus Berges du Rhône 18, quai Claude Bernard 69365 LYON CEDEX 07 Bureau BEL 319 **bruno.milly@univ-lyon2.fr**

1. ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Contents

List of Figures iii

List of Tables v

Résumé étendu vii

Introduction 1

1 State of the art 5
- I Medical Image Segmentation . . . 6
  - I.1 Introduction . . . 6
  - I.2 Traditional segmentation techniques . . . 7
  - I.3 Deep learning-based methods . . . 8
  - I.4 Efficient Medical Image Segmentation . . . 14
  - I.5 Metrics . . . 18
- II Unstructured 3D data . . . 20
  - II.1 Introduction . . . 20
  - II.2 Point cloud extraction from an image . . . 21
  - II.3 Point cloud processing . . . 26
  - II.4 Implicit Neural Representations . . . 31
- III Conclusion . . . 38

2 SCONet: convolutional occupancy networks for multi-organ segmentation 39
- I Method . . . 40
  - I.1 Point cloud extraction . . . 40
  - I.2 SCONet architecture . . . 41
  - I.3 Training and inference . . . 43
- II Experiments . . . 43
  - II.1 Datasets . . . 43
  - II.2 Implementation . . . 45
  - II.3 Comparison baselines . . . 45
- III Results . . . 45
  - III.1 Segmentation quality . . . 45
  - III.2 Efficiency . . . 49
- IV Artifact reduction through training on random crops . . . 53

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

IV.1 Motivation . . . 53
IV.2 Method . . . 54
IV.3 Results . . . 54
IV.4 Conclusion . . . 56
V Ablation studies . . . 57
V.1 SURF descriptors . . . 57
V.2 Data augmentation . . . 57
V.3 PointNet module . . . 58
VI Conclusion . . . 59

3 Study of the point cloud quality **61**
I Point cloud sampling . . . 62
I.1 Training on a pre-sampled point cloud . . . 62
I.2 Point cloud sampling algorithm . . . 63
II Enhanced point cloud . . . 66
II.1 Reference surface points . . . 66
II.2 Enhanced contour point cloud . . . 69
II.3 Ablation study: PointNet module . . . 73
III Conclusion . . . 75

4 Robustness of the proposed workflow **77**
I Dataset shift . . . 78
I.1 Motivation . . . 78
I.2 Characteristics of the studied datasets . . . 78
I.3 Method . . . 80
I.4 Results and discussion . . . 80
II Cross-modality learning . . . 82
II.1 Motivation . . . 82
II.2 Method . . . 82
II.3 Results . . . 83
II.4 Discussion . . . 84
III Conclusion . . . 86

Conclusion and perspectives **87**

List of Publications **91**

References **93**

A Computing resources **105**

B SCONet-based sampling: detailed per-organ results **107**

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

List of Figures

6 Sample CT volume and its corresponding point cloud extracted from the contour map 2

1.1 Generic architecture of CNNs for image classification . . . 9
1.2 Example of a four-stages 2D UNet architecture . . . 10
1.3 ViT overview . . . 12
1.4 Standard or depthwise separable convolution . . . 15
1.5 Low-rank adpater (LoRA) module . . . 18
1.6 Illustration of the Dice coefficient . . . 18
1.7 Illustration of the Hausdorff Distance . . . 19
1.8 3D representations of four organs . . . 21
1.9 Canny edge detection steps . . . 23
1.10 Second order partial derivative of the Gaussian kernel and second order box filters 24
1.11 Feature descriptor computation for a keypoint k . . . 25
1.12 PointNet architecture . . . 27
1.13 KPConv illustrated on 2D points. . . . 28
1.14 Representation of 3D shapes with a voxel grid and with an implicit function f_θ . . . 32
1.15 DeepSDF overview . . . 33
1.16 Occupancy Network Architecture . . . 34
1.17 Convolutional Occupancy Networks . . . 34
1.18 NeRF workflow . . . 35
1.19 Architecture of ImPulSe . . . 37

2.1 Full segmentation workflow for 3D images . . . 40
2.2 SCONet workflow overview . . . 41
2.3 SCONet detailed PointNet architecture . . . 42
2.4 SCONet decoder detailed architecture . . . 42
2.5 Random linear transform on input and query points . . . 44
2.6 Qualitative results for segmentation of AbdomenCT-1K . . . 46
2.7 Dice results for AbdomenCT-1K . . . 47
2.8 Dice results for AMOS . . . 47
2.9 3D visualization of the pancreas for the reference and the prediction . . . 48
2.10 MISE algorithm illustrated in 2D . . . 52
2.11 Illustration of a segmentation artifact for the AMOS dataset . . . 54
2.12 Dice results for AbdomenCT-1K when training on random crops . . . 55

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

2.13 Dice results for AMOS when training on random crops 55
2.14 Several slices extracted from the feature grid for one PointNet feature 59
2.15 Several slices extracted from the feature grid for one SURF feature 59

3.1 Projection of a point cloud on the original volume (with reference segmentation) for the compared sampling algorithms 64
3.2 Discrepancy between contour and surface points 66
3.3 3D visualization of different types of point clouds 67
3.4 3D visualization of the pancreas predicted by SCONet for contour and surface points 68
3.5 3D visualization of point clouds with corresponding organ labels 69
3.6 3D visualization of the pancreas predicted by SCONet for contour and foreground contour points 71
3.7 Qualitative segmentation results on AbdomenCT-1K for different input point clouds and corresponding contour map 71
3.8 Several slices extracted from the feature grid for one PointNet feature when processing a surface point cloud 74
3.9 Several slices extracted from the feature grid for one PointNet feature when processing a foreground contour point cloud 74

4.1 Organ volume distribution for both datasets 79
4.2 Average Dice score on AbdomenCT-1K and AMOS for models trained on AbdomenCT-1K 81
4.3 Reference and predicted segmentation map when training on AbdomenCT-1K and evaluating on AMOS 81
4.4 Slices from two samples of the AMOS datasets 83
4.5 Average Dice for cross-modality learning 85

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

List of Tables

1.1 Popular abdominal segmentation challenges and datasets 7

2.1 Segmentation results for AbdomenCT-1K 47
2.2 Segmentation results for AMOS 47
2.3 Efficiency results for AbdomenCT-1K. 49
2.4 Network optimization of SCONet 51
2.5 Performance of SCONet when using the MISE algorithm 52
2.6 Inference times for compared baselines. 52
2.7 Evaluation of the segmentation quality on AbdomenCT-1K when training on random crops 55
2.8 Evaluation of the segmentation quality on AMOS when training on random crops . 55
2.9 Efficiency results when training on random crops 56
2.10 Performance of SCONet on AbdomenCT-1K for different input features 57
2.11 Performance of SCONet for different data augmentation strategies 58
2.12 Ablation study of the PointNet module 59

3.1 Performance of SCONet on AbdomenCT-1K when sampling is done online or offline 62
3.2 Results for different point cloud sampling strategies 65
3.3 Performance of SCONet on AbdomenCT-1K for surface and contour point clouds . 68
3.4 Performance of SCONet on AMOS for surface and contour point clouds 68
3.5 Performance of SCONet on AbdomenCT-1K for contour point cloud with and without reference background points 70
3.6 Performance of SCONet on AMOS for contour point cloud with and without reference background points 70
3.7 SCONet performance for different input point clouds 72
3.8 Performance of SCONet trained on foreground contour points when evaluated on different types of points 73
3.9 Ablation study of PointNet on surface and foreground contour points 74

4.1 Detailed characteristics for the AbdomenCT-1K and AMOS datasets 79
4.2 Evaluation on AbdomenCT-1K and AMOS for models trained on AbdomenCT-1K . . 80
4.3 Evaluation on AMOS CT split 85
4.4 Evaluation on AMOS MRI split 85

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

B.1 SCONet performance on AbdomenCT-1K for different input point clouds 107
B.2 Performance of SCONet trained on foreground contour points when evaluated on different types of points (AbdomenCT-1K) 107
B.3 SCONet performance on AMOS for different input point clouds 107
B.4 Performance of SCONet trained on foreground contour points when evaluated on different types of points (AMOS) 107

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Résumé étendu

Introduction

Contexte du travail L'imagerie médicale désigne l'ensemble des techniques permettant de visualiser les structures internes du corps humain, généralement à des fins cliniques. L'imagerie par résonance magnétique (IRM) ou la tomodensitométrie (ou computed tomography, CT) font par exemple partie des nombreuses modalités d'imagerie disponibles qui sont couramment utilisées pour établir un diagnostic ou faire du suivi thérapeutique. Les images médicales peuvent notamment être utilisées pour extraire des informations précises sur la forme d'un organe ou d'une tumeur en passant par la segmentation de cette structure. Cette tâche consiste à subdiviser une image d'entrée en un ensemble de régions qui ne se chevauchent pas, produisant une de segmentation dans laquelle chaque pixel (ou voxel dans le cas d'un volume tridimensionnel) se voit attribuer une étiquette. Cependant, segmenter manuellement un volume médical est une tâche longue et fastidieuse qui peut bénéficier du développement de méthodes automatiques afin d'alléger la charge de travail des cliniciens. Le besoin de tels algorithmes est encore amplifié par l'augmentation significative du nombre d'images médicales acquises (NHS, 2020).

Les premiers outils ayant été proposés pour automatiser la segmentation d'images médicales reposent sur des caractéristiques artisanales ou des approches algorithmiques sensibles à la qualité de l'image et qui nécessitent un réglage manuel de leurs paramètres. En revanche, les approches plus récentes s'appuient sur les réseaux neuronaux profonds, un type d'algorithme qui apprend à accomplir une tâche en optimisant ses paramètres par le traitement d'une grande quantité de données.

Si les modèles les plus récents sont également de plus en plus performants pour la tâche de segmentation, ils nécessitent aussi souvent des ressources en calculs de plus en plus grandes à l'entraînement ou au déploiement. Cela peut poser problème puisque la majeure partie de la production énergétique mondiale repose encore sur des sources émettrices de gaz à effet de serre, facteur principal du changement climatique. Le coût en calcul parfois prohibitif de ces algorithmes peut également compromettre leur déploiement dans des environnements cliniques disposant de ressources en calcul limitées, ce qui soulève des questions d'accessibilité et d'équité.

Un des facteurs qui expliquent le coût particulièrement élevé des algorithmes de segmentation d'images médicales est la taille de ces images. Celles-ci sont en effet généralement représentées sous forme de volumes tridimensionnels (3D) composés de millions de voxels,

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

ce qui entraîne une utilisation importante de la mémoire ainsi que des temps de calcul élevés. Une solution potentielle pour réduire ces coûts consiste à réduire la redondance dans ces images, qui se traduit par une faible variation d'intensité dans le voisinage de chaque voxel et aussi par le fait que de grandes parties du volume sont souvent constituées de voxels de fond, vides. Le traitement des seuls voxels d'intérêt pourrait donc améliorer considérablement le coût en calcul. Cette approche nécessiterait de redéfinir la manière dont les images sont représentées, en remplaçant la grille volumétrique régulière par une représentation sous un format 3D non structuré. Parmi les représentations de données 3D disponibles, les nuages de points sont naturellement parcimonieux. Ceux-ci consistent en une liste de coordonnées 3D et pourraient être une alternative plus légère aux grilles de voxels denses pour permettre de réduire le coût de calcul des algorithmes de segmentation.

Bien que l'idée d'utiliser des nuages de points pour générer des cartes de segmentation ne soit pas nouvelle, rares sont les travaux qui ont complètement supprimé la représentation volumique du processus d'apprentissage.

Objectifs L'objectif principal de cette thèse est d'évaluer si il est possible de réaliser la segmentation d'images médicales en s'appuyant sur une représentation 3D non structurée plutôt que sur l'image d'origine. Pour ce faire, nous souhaitons développer une nouvelle approche de segmentation reposant sur des nuages de points. La chaîne de traitement proposée doit non seulement générer des cartes de segmentation de qualité, mais également être efficace en terme de coût en calcul. Plusieurs défis doivent être relevés pour atteindre cet objectif principal. Tout d'abord, nous devons identifier les points qui doivent être extraits du volume d'entrée: le nuage de points généré doit en effet contenir suffisamment d'informations pour la reconstruction d'une carte de segmentation tout en restant plus léger que le volume. Un autre défi réside dans le développement d'un réseau de neurones qui doit être suffisamment flexible pour pouvoir traiter la représentation parcimonieuse sous forme de nuage de points et produire une carte de segmentation dense.

État de l'art

Nous proposons dans le Chapitre 1 un état de l'art qui permet de contextualiser notre travail.

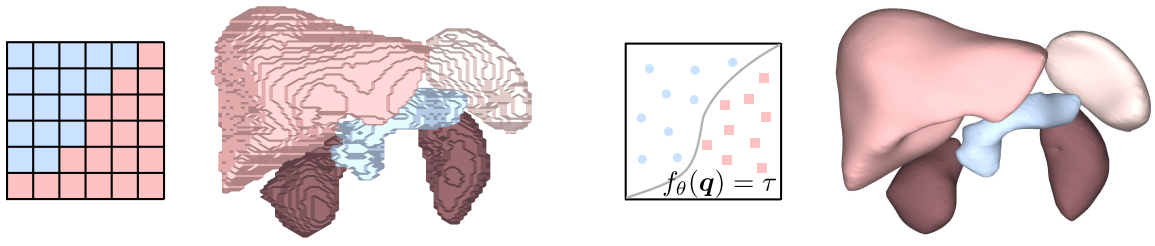
Dans un premier temps, nous présentons les méthodes couramment utilisées pour la segmentation d'images médicales. Le développement de réseaux de neurones convolutifs (CNN), et en particulier les modèles U-Net 2D et 3D (Ronneberger et al., 2015; Çiçek et al., 2016), ont permis de produire des cartes de segmentation de qualité, mais au prix d'un coût en calcul parfois prohibitif. Plus récemment, l'architecture nnU-Net (Isensee et al., 2021) a encore amélioré les résultats de segmentation en adaptant automatiquement la configuration du réseau à chaque base de données. Enfin les réseaux avec une architecture transformers tels que SwinUNETR-V2 (Y. He et al., 2023) ont tiré parti des mécanismes d'auto-attention pour mieux capturer le contexte global des images, ce qui leur a permis d'améliorer la qualité des cartes de segmentation mais en augmentant encore le coût computationnel.

Ensuite, nous étudions les données 3D non structurées, qui, contrairement aux images

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

2D et 3D, ne possèdent pas une structure prédéfinie. Parmi les représentations non structurées, nous nous intéressons plus particulièrement aux nuages de points. Leur extraction à partir d'un volume médical peut se faire notamment en utilisant des cartes de contours comme celles calculées par Canny (Canny, 1986) ou en se basant sur des points-clés, comme les points SURF (Bay et al., 2006). De nombreuses méthodes ont été proposées pour traiter les nuages de points. PointNet (Qi et al., 2016) est notamment un travail crucial s'appuyant sur des perceptrons multi-couches (MLP) et qui peut être utilisé pour la classification ou la segmentation de nuages de points. L'utilisation des nuages de points dans la segmentation d'images médicales a récemment gagné en popularité mais consiste souvent à simplement ajouter un module basé sur un nuage de points afin d'affiner une segmentation produite par un CNN, comme le fait Point-UNet (Ho et al., 2021). Dans la littérature, l'extraction d'un nuage de points à partir d'un volume pré-segmenté nécessite généralement d'échantillonner un point par voxel dans les régions d'intérêt. Lorsqu'on travaille avec des structures d'intérêt de grande taille, comme c'est le cas pour la segmentation multi-organes, cela peut conduire à des nuages de points très volumineux et, par conséquent, à des coûts de calcul élevés.

C'est en étudiant les méthodes capables de reconstruire des surfaces à partir de nuages de points que l'on a commencé à s'intéresser aux Représentations Neuronales Implicites (INR). Ces méthodes s'appuient sur des réseaux de neurones pour apprendre une fonction implicite au lieu de travailler directement sur des représentations explicites, telles que des nuages de points, des maillages ou des volumes voxéliques. Une illustration des deux types de représentations est donnée en Figure 1. Ces réseaux apprennent généralement une fonction de distance, comme DeepSDF (Park et al., 2019), ou une fonction de probabilité d'occupation, comme ONet (Mescheder et al., 2019) et ConvONet (Peng et al., 2020). De tels réseaux ont déjà été utilisés pour des applications médicales dans des tâches telles que la reconstruction d'images ou la segmentation. En particulier, les méthodes utilisant ONet avec un encodeur basé sur un CNN ont donné des résultats prometteurs comme ImPulSe (Kuang et al., 2022) et OSS-Net (Reich et al., 2021), ce qui confirme l'intérêt des INR pour la segmentation d'images médicales.



(a) *Représentation volumique* **(b)** *Représentation implicite*

Figure 1: *Représentation d'une forme 3D avec une grille de voxels et avec une fonction implicite f_θ. La variable cible τ représente ici la valeur d'occupation dans l'espace 3D. Figure adaptée de Kuang et al. (2022).*

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

SCONet: Réseaux d'Occupation Convolutifs pour la Segmentation Multi-Organes

Nous décrivons dans le Chapitre 2 la chaîne de traitement illustrée en Figure 2, qui est la contribution principale de cette thèse. Sa spécificité par rapport aux méthodes proposées dans la littérature provient de sa capacité à exploiter les représentations par nuages de points et les représentations implicites pour réaliser la segmentation de volumes médicaux.

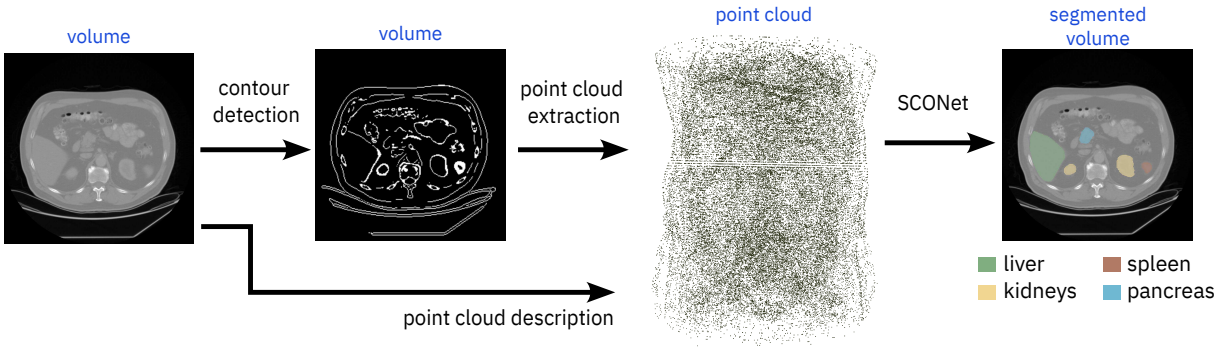


Figure 2: *Chaîne de traitement proposée*

Méthode

Extraction du nuage de point La première étape de notre chaîne de traitement est l'extraction d'un nuage de points de contours. Celle-ci se fait à partir de la carte de contours calculée avec l'algorithme Canny (Canny, 1986). Pour chaque point du nuage, en plus de ses coordonnées 3D, on calcule son descripteur SURF (Bay et al., 2006). Celui-ci encode les informations locales dans un voisinage de 10^3 voxels autour du point.

SCONet Le nuage de points est échantillonné puis utilisé en entrée d'un nouveau réseau implicite que nous nommons SCONet. Ce réseau, adapté de ConvONet (Peng et al., 2020) pour la tache de segmentation multi-organes, est illustré en Figure 3.

Son encodeur génère une grille de caractéristiques à partir du nuage de points de contours. Pour cela, il calcule dans un premier temps des caractéristiques pour chaque point d'entrée avec un module PointNet (Qi et al., 2016). Ces caractéristiques sont ensuite concaténées aux caractéristiques SURF et agrégées dans une grille de résolution r. Cette étape nécessite d'appliquer un moyennage des valeurs de caractéristiques pour tous les points projetés dans la même cellule. Enfin un réseau U-Net 3D traite cette grille et génère une grille latente de caractéristiques.

Le décodeur utilise ensuite cette grille latente pour prédire les probabilités d'occupation d'un point de requête. Pour cela un vecteur latent est d'abord calculé par interpolation trilinéaire sur la grille latente. Ce vecteur est ensuite transmis à un réseau ONet qui génère les probabilités d'occupation pour chaque organe au point interrogé.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

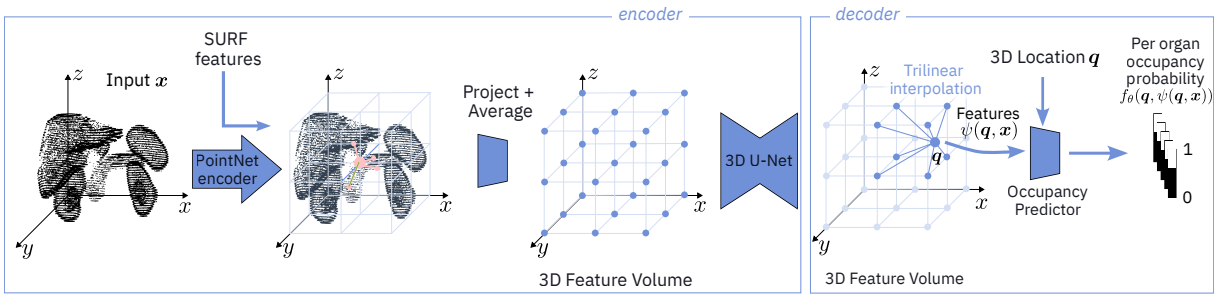


Figure 3: *Aperçu du processus de SCONet. Pour plus de clarté, le nuage de points d'entrée affiché ici n'inclut que les points de contours appartenant aux organes cibles.*

Entraînement et inférence À l'entraînement, nous évaluons SCONet sur 50k points de requête tirés aléatoirement et optimisons une perte combinée Dice et d'entropie croisée. À l'inférence, nous prédisons la carte de segmentation complète en interrogeant les points de requêtes correspondant à tous les voxels de l'image d'origine.

Dispositif expérimental

Nous évaluons notre méthode sur deux bases de données CT abdominales: AbdomenCT-1K (J. Ma et al., 2022b) qui contient les annotations de quatre organes pour 1000 volumes CT et AMOS (Ji et al., 2022) qui contient les annotations de quinze organes pour 300 volumes CT.

Nous nous comparons à des méthodes discrètes: U-Net 2D et 3D servent de base à de nombreuses architectures CNN tandis que SwinUNETR-V2 est un modèle de transformer performants pour la segmentation multi-organe. Pour les méthodes implicites, nous nous comparons à ImPulSe et OSS-Net.

Résultats

Model	AbdomenCT-1K		AMOS		Inf. GPU	#Param	FLOPs	Train.	Inf.
	Dice	HD	Dice	HD	(GB)	(M)	(G)	time (h)	time (s)
2D U-Net	**0.920**	79.26	0.801	69.89	0.77	1.36	960	123	1.26
3D U-Net	**0.919**	46.28	**0.816**	45.68	4.48	4.08	2,980	54	2.16
SwinUNETR-V2	**0.918**	139.16	0.802	176.93	7.59	18.35	2,961	17	4.87
ImPulSe	0.898	21.74	0.669	33.98	16.25	33.28	2,504	13	1.61
OSS-Net	0.774	41.47	0.325	47.39	3.95	9.12	4,908	15	4.14
SCONet ($r = 32$)	0.892	19.12	0.693	**24.74**	0.83	1.26	120	11	0.92
SCONet ($r = 64$)	0.911	**17.94**	0.732	**27.29**	1.27	1.26	218	19	1.00

Table 1: *Performance de SCONet et des différentes méthodes comparées. Sont évalués des métriques de qualité de segmentation et d'efficience.*

Le Tableau 1 résume nos résultats. SCONet atteint une précision de segmentation inférieure à celle des méthodes discrètes, ce qui se traduit par des surfaces trop lissées, comme illustré en Figure 4. Cependant, l'écart avec ces méthodes de référence diminue lorsque la résolution de la grille de caractéristiques augmente. Par ailleurs, SCONet obtient des résultats équivalents voire supérieurs comparé aux deux méthodes implicites de référence. Les organes

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

les plus difficiles à reconstruire pour SCONet sont ceux qui sont de petite taille, ceux dont la forme varie au sein d'un jeu de données, ainsi que ceux qui contiennent peu de points de contours. En ce qui concerne la précision de surface de segmentation, les méthodes implicites ont une meilleure performance.

En ce qui concerne les métriques d'efficacité, SCONet se distingue par une faible empreinte mémoire, une complexité calculatoire réduite et un temps d'inférence bien inférieur à ceux des modèles de référence.

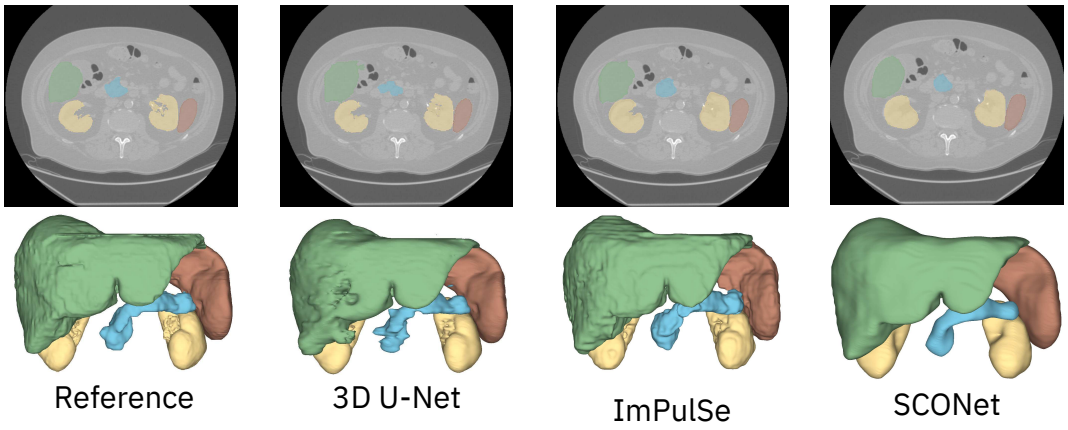


Figure 4: *Comparaison des résultats de segmentation obtenus par SCONet, ImPulSe et U-Net 3D sur une coupe axiale CT (en haut) et en 3D (en bas) pour le foie (vert), les reins (jaune), le pancréas (bleu) et la rate (rouge).*

Optimisation Nous étudions ensuite plusieurs méthodes qui pourraient permettre d'améliorer l'efficacité de SCONet. Nous appliquons d'abord la précision mixte automatique (AMP) qui réduit la précision de certaines couches du réseau. Ensuite, nous appliquons au module UNet deux méthodes couramment utilisées pour améliorer l'efficacité des CNN. Tout d'abord nous remplaçons l'opération de concaténation utilisée dans UNet par une simple addition pour agréger les cartes de caractéristiques précoces de l'encodeur aux caractéristiques plus profondes du décodeur. Ensuite nous remplaçons les convolutions standards par des convolutions séparables en profondeur.

Pour la première étude, utiliser l'AMP à l'inférence permet de réduire l'utilisation de la mémoire et le temps d'inférence pour une qualité de segmentation équivalente. Cependant, les deux autres méthodes étudiées génèrent des cartes de segmentation de moins bonne qualité pour des améliorations moindres de l'utilisation de la mémoire GPU ou du temps d'inférence.

Utilisation de MISE Dans notre chaîne de traitement, générer une carte de segmentation complète nécessite d'appliquer le décodeur pour un point par voxel, ce qui est inefficace. Nous étudions ici l'algorithme MISE, présenté dans la papier décrivant les réseau ONet (Mescheder et al., 2019). Cet algorithme permet de sélectionner des points de requête plus densément dans les régions les plus proches de la surface de l'objet segmenté.

Nos expériences montrent que bien que l'algorithme MISE permette de sélectionner les points de requête de manière plus efficace sans altérer la qualité de la segmentation, il ne

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

parvient pas à accélérer le temps d'inférence de SCONet car son implémentation est trop lente. Cette méthode peut cependant s'avérer utile dans des contextes où les ressources sont limitées et où seul un processeur (CPU) est accessible. Le temps d'inférence de SCONet est alors bien plus court que celui des modèles discrets comparés.

Apprentissage sur des fenêtres extraites du nuage de points

Nous proposons ensuite d'entraîner SCONet sur des fenêtres aléatoires extraites des nuages de points. Cela a pour but de de réduire l'effet de la projection d'un nuage de point hautement résolu dans une grille de basse résolution. Cela pourrait également permettre d'éliminer les artefacts qui apparaissent quand des points sont projetés au-delà des limites de la grille latente. À l'inférence nous utilisons une fenêtre glissante pour reconstruire la carte de segmentation complète.

Nos expériences sur les deux jeux de données montrent que la qualité de segmentation des cartes générées par SCONet est améliorée, en particulier pour les petits organes ou pour les organes à proximité des extrémités du volume et qui contenaient des artefacts. Cependant, la dégradation de la complexité et du temps d'inférence est également non négligeable puisque certains voxels sont décodés plusieurs fois, ce qui peut limiter l'intérêt d'utiliser cette stratégie pour l'apprentissage et l'inférence.

Études d'ablation

Nous proposons dans une dernière partie plusieurs études d'ablation:

- Nous justifions l'utilisation des caractéristiques SURF dans SCONet en montrant une dégradation de la performance quand elles ne sont pas utilisées ou quand elles sont remplacées par les valeurs d'intensité à chaque point du nuage;
- Nous comparons plusieurs stratégies d'augmentation de données, par exemple en ajoutant un bruit aléatoire aux coordonnées de chaque point du nuage. Nous montrons qu'une transformation linéaire aléatoire du nuage d'entrée et des points de requête permet d'obtenir les meilleurs résultats;
- Nous montrons que la suppression du module PointNet n'entraîne pas de perte de performance, ce qui peut s'expliquer par le fait que le nuage de points de contours est trop bruité.

Étude de la qualité du nuage de points

Dans le Chapitre 3, nous proposons une étude approfondie de la représentation sous forme de nuage de points utilisée dans notre chaîne de traitement.

Échantillonnage

Le nuage de points extrait des points de contour a un très grand nombre de points ($\sim 10^6$ points) et il est donc nécessaire de l'échantillonner avant de le mettre en entrée de SCONet.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Pré-échantillonnage Dans une premier temps, nous proposons de comparer les stratégies d'échantillonnage en ligne et hors ligne. La première signifie que le nuage est échantillonné une fois par époque pendant l'entraînement tandis que, pour la deuxième, aussi appelée pré-échantillonnage, le nuage de point est échantillonné une seule fois avant le processus d'entraînement.

Nos expériences sur SCONet ne montrent pas de différences significatives sur la qualité des cartes de segmentation entre les deux stratégies d'échantillonnage. Cependant, le temps d'entraînement double quand l'échantillonnage est réalisé à chaque époque. Cela s'explique notamment par le temps de chargement des nuages de points qui est plus élevé quand ceux-ci contiennent tous les points de contours, avant échantillonnage.

Algorithme d'échantillonnage Nous étudions la performance de SCONet pour trois algorithmes d'échantillonnage: l'échantillonnage aléatoire (utilisé dans le chapitre précédent), le sous-échantillonnage par grille et l'échantillonnage par points les plus éloignés (FPS). Ces deux derniers pourraient permettre de générer un sous-ensemble de points offrant une meilleure couverture du nuage de points d'entrée qu'un simple tirage aléatoire.

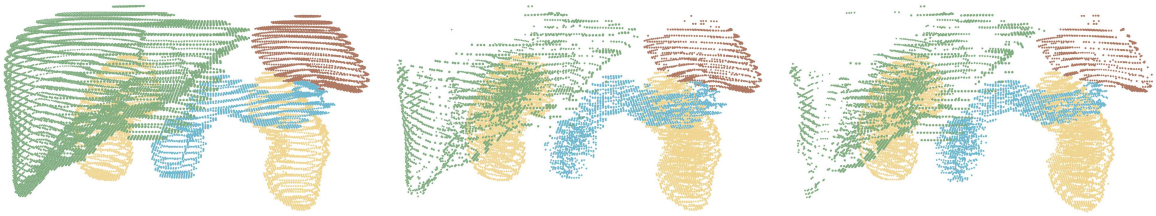
Nos expériences sur SCONet ne montrent pas de différences significatives sur la qualité des cartes de segmentation entre les trois algorithmes d'échantillonnage. Les métriques d'empreinte mémoire et de temps d'inférence sont également équivalentes. Finalement, l'échantillonnage aléatoire reste la stratégie la plus adaptée pour SCONet, car il est le plus rapide et le plus simple à mettre en œuvre. Nous faisons également varier le nombre de points utilisés pour l'échantillonnage aléatoire et montrons que la taille optimale du nuage de points traité par SCONet est 50.000 points.

Amélioration du nuage de points

Nuage de points de surface Nous étudions la capacité de représentation de SCONet et réalisons un test d'intégrité pour évaluer la performance de notre modèle dans la tâche de segmentation volumétrique multi-organes lorsqu'il traite un nuage de points de meilleure qualité. Nous nous concentrons sur ce que nous appelons les points de surface, dont les coordonnées correspondent aux voxels situés à la frontière entre un organe et l'arrière-plan, d'après une carte de segmentation de référence. Nous illustrons ce type de nuage pour AbdomenCT-1K en Figure 5a.

Dans une première expérience, nous entraînons SCONet sur les nuages de points de surface de taille $N = 5k$. Nous trouvons que la qualité des cartes de segmentation est significativement meilleure pour ce type de points que pour le nuage de points de contours, en particulier pour les organes dont la forme varie beaucoup au sein d'un jeu de données. Augmenter la résolution de la grille latente dans SCONet permet même d'atteindre un performance supérieure aux modèles discrets pour AbdomenCT-1K. Cela confirme que la performance de notre réseau est impacté à la fois par la qualité du nuage de points et par la résolution de la grille latente. Nous concluons également que les résultats obtenus ici correspondent à la limite supérieure des performances de notre modèle dans sa configuration standard.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés



(a) *Nuage de points de surface* **(b)** *Nuage de points de contours hors fond (à partir de la carte de référence)* **(c)** *Nuage de points de contours hors fond (à partir de la carte prédite par SCONet)*

Figure 5: *Visualisation 3D des nuages de points avec leurs étiquettes. Les étiquettes des organes sont affichées ici à des fins de visualisation et ne sont pas utilisées comme caractéristique d'entrée de SCONet.*

Dans une deuxième expérience, nous ajoutons des points de contours aux nuages de points de surface. Par rapport à l'expérience où seuls les points de contours sont utilisés, nous observons une amélioration de la qualité des cartes de segmentation, bien que celle-ci soit plus légère qu'en utilisant seulement les points de surface. Nous concluons que bien que les points de contour non surfaciques puissent réduire les performances de SCONet par rapport à l'utilisation exclusive de points de surface de référence, leur impact peut être atténué s'ils contiennent un nombre suffisant de points proches de la surface.

Amélioration du nuage de points de contours Les points de surface peuvent être difficiles à extraire du volume sans s'appuyer sur la carte de segmentation de référence. Nous cherchons plutôt à extraire des points d'intérêt à partir du nuage de points de contours. Dans un premier temps, nous proposons une expérience où nous supprimons les points du fond pour ne conserver que les points de contour situés à l'intérieur des organes cibles, en nous basant sur la carte de segmentation de référence. Le nuage de points résultant sera appelé nuage de points de contour hors fond, dont on donne une illustration en Figure 5b.

Nous entraînons donc SCONet sur les nuages de points de contours hors fond et observons une meilleurs qualité de carte de segmentation comparé aux nuages de points de contours. Cependant, on note que cette expérience se concentre uniquement sur les points de contours localisés dans les organes cibles et ignore ainsi les points de contours proches de leur surface externe, ce qui peut produire une baisse de performance pour certains organes.

Dans cette première expérience nous nous sommes appuyés sur les cartes de segmentation de référence auxquelles nous n'aurons pas accès dans la pratique. Ainsi, nous proposons une deuxième étude pour laquelle nous extrayons les points de contours hors fond en se basant sur une prédiction d'un premier réseau SCONet. Un deuxième SCONet est ensuite entraîné sur les points de contour hors fond prédits, dont on donne une illustration en Figure 5c.

Les cartes de segmentation générées pendant cette deuxième expérience sont de qualité inférieure à celles prédites sur les nuages de points de contours d'origine, ce qui s'explique par le fait que la qualité de la prédiction du deuxième réseau dépende fortement de la qualité de la prédiction faite par le premier SCONet.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Ablation du module PointNet Nous concluons ce chapitre par une deuxième étude d'ablation du module PointNet en utilisant les nuages de points étudiés dans cette section. Nous observons ainsi que l'utilisation du module PointNet permet d'améliorer les performances de SCONet quand le nuage d'entrée est de meilleure qualité, en particulier quand les points sont extraits de la surface.

Robustesse de la chaîne de traitement

Dans le Chapitre 4, nous étudions la robustesse de notre approche aux changements de base de données et de modalité.

Changement de base de données Les décalages entre jeux de données représentent un défi majeur pour l'analyse d'images médicales (Guan et al., 2022). En effet, les images acquises en contexte clinique peuvent appartenir à des distributions différentes de celles des différents jeux de données utilisés pour entraîner et évaluer les modèles de segmentation.

Nous entraînons SCONet et les modèles de référence sur le jeu de données AbdomenCT-1K et l'évaluons sur AMOS. Nous avons choisi cette configuration car AbdomenCT-1K contient plus de patients qu'AMOS et car il est plus diversifié en termes de résolutions d'images et de distributions d'intensité.

L'évaluation des différents modèles sur la base AMOS génère des cartes de segmentation de moins bonne qualité que sur la base AbdomenCT-1K d'origine. Cependant, on note aussi que la baisse de performance est réduite notamment pour notre modèle SCONet.

Un des facteurs qui explique la robustesse de notre méthode est son décodeur qui est un MLP, architecture qui a tendance à favoriser l'apprentissage de basses fréquences (Rahaman et al., 2019). Cela est peut-être aussi dû à l'utilisation des caractéristiques SURF, qui sont notamment robustes aux changements d'intensité.

Apprentissage inter-modalité Dans la pratique, les modalités CT et IRM sont toutes les deux couramment utilisées pour visualiser les organes abdominaux ou des anomalies telles que les tumeurs (Valindria et al., 2018). Étant donné la rareté des images médicales annotées disponibles pour l'entraînement des réseaux de neurones profonds, il est pertinent de se demander si les modèles entraînés sur une modalité d'imagerie peuvent se généraliser à une autre, et s'il est même possible de fusionner des images des deux modalités pour constituer un jeu d'entraînement plus large.

Nous réalisons nos expériences sur la base de données AMOS, qui contient 300 images CT et 60 images IRM annotées. Nous entraînons SCONet (et les modèles de référence) dans trois configurations: sur les images CT seulement, sur les images IRM seulement et sur les deux modalités en même temps. Nous évaluons les modèles sur les images CT et les images IRM dans chaque configuration.

Nous remarquons que la baisse de performance de SCONet est moins marquée que celle des autres modèles lorsqu'ils sont confrontés à un changement de modalité CT vers IRM. Dans l'expérience basée uniquement sur l'IRM, où SCONet obtient les meilleures performances

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

malgré un nombre limité d'images d'entraînement, nous concluons que notre approche nécessite moins de données que les méthodes basées sur les voxels. De plus, nous notons que l'intégration d'images IRM dans le jeu d'entraînement CT ne perturbe pas les performances de SCONet, contrairement aux autres modèles. Comme discuté précédemment, nous attribuons cette robustesse accrue non seulement au décodeur MLP de SCONet, mais aussi à l'utilisation des caractéristiques SURF. Nous concluons que cette propriété pourrait rendre notre approche particulièrement adaptée pour travailler avec plusieurs modalités d'imagerie.

Conclusion et perspectives

Pour conclure, notre chaîne de traitement permet de générer des cartes de segmentation cohérentes pour un faible coût en calcul. Notre méthode pourrait être particulièrement adaptée dans des contextes de changement de base de données ou de modalité.

Limites

Bien que notre méthode produise globalement des cartes de segmentation de qualité, notre algorithme d'apprentissage éprouve des difficultés à apprendre les hautes fréquences, ce qui le conduit souvent prédire des surfaces trop lisses. Cela affecte particulièrement les petits organes et les structures dont la forme varie au sein d'un même jeu de données. Nous identifions trois caractéristiques de notre méthode qui limitent sa capacité à générer des cartes de segmentation détaillées.

Qualité du nuage de points de contour Le nuage de points traité par SCONet est extrait à partir de la carte de contour et contient donc des points non informatifs, situés dans des structures non cibles ou résultant du bruit. La présence de ces points peut perturber les performances du réseau, d'autant plus qu'ils peuvent, lors du processus d'échantillonnage, prendre la place de points plus informatifs situés près des surfaces des organes d'intérêt. Cela impacte notamment le module PointNet dans l'encodeur de SCONet, qui échoue à calculer des caractéristiques géométriques complexes par point à partir d'un nuage de points de contours bruité.

Un autre problème provient du manque de points dans les régions où le contraste de l'image est trop faible pour pouvoir détecter des contours. Cela devient particulièrement problématique lorsque ces régions se trouvent près des frontières des structures cibles, forçant le réseau à inférer ou "halluciner" les surfaces des organes en l'absence de données fiables. Nous observons que les points superflus et les points manquants sont deux conséquences de l'utilisation de l'algorithme de Canny standard avec des paramètres choisis de manière empirique pour générer la carte de contour.

Bien que nous ayons montré dans le Chapitre 3 que la priorisation des points de contour proches des surfaces pourrait potentiellement améliorer les performances de SCONet, nous n'avons pas réussi à extraire ces points sans nous appuyer sur les cartes de segmentation de référence. De plus, cette stratégie ne résoudrait pas le problème des points manquants.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Agrégation dans une grille basse résolution Dans l'encodeur SCONet, les points du nuage d'entrée sont d'abord agrégés dans une grille avant d'être traités par le module UNet. Lors de cette étape, toutes les caractéristiques des points projetés dans une même cellule sont moyennées. Cette agrégation et cette moyenne empêchent notre réseau de distinguer les points proches les uns des autres. Par ailleurs, le vecteur latent alimentant le décodeur est calculé par interpolation linéaire de la grille latente apprise par le UNet, ce qui limite intrinsèquement la capacité du décodeur à restituer des hautes fréquences. Cette limite est d'autant plus importante que la grille de caractéristiques a une faible résolution (32^3) comparée à la résolution du nuage de points extrait de la carte de contour de 256^3.

L'augmentation de la résolution de la grille latente ou l'apprentissage sur des fenêtres extraites du nuage de points peuvent atténuer ce problème dans une certaine mesure, comme montré dans le Chapitre 2, bien que ces solutions entraînent également une empreinte mémoire plus importante et/ou un temps de traitement accru.

Réseaux implicites et basses fréquences Enfin, nous avons montré dans le Chapitre 3 que la capacité de représentation de SCONet reste limitée malgré l'augmentation de la résolution de la grille latente ou l'utilisation d'un nuage de points composé exclusivement de points de surface. Nous expliquons cette limite par la nature du décodeur de SCONet qui est un MLP, un type de réseau de neurones qui favorise les basses fréquences pendant l'apprentissage. Il s'agit d'une propriété intrinsèque et bien documentée des réseaux implicites (Tancik et al., 2020).

Perspectives

Les limites discutées précédemment ouvrent plusieurs pistes de recherche prometteuses pour les travaux futurs.

Généralisation à d'autres jeux de données Les expériences menées dans cette thèse se sont principalement concentrées sur des images CT abdominales. À court terme, il serait donc intéressant d'appliquer notre méthode à d'autres tâches de segmentation. Par exemple, la segmentation de tumeurs (comme les tumeurs cérébrales en IRM) présente plusieurs défis. La variabilité de forme et de localisation des tumeurs pourrait poser des difficultés à notre approche. Une autre application exigeante serait la segmentation d'images échographiques, où la détection de contours peut produire des cartes de contours très bruitées en raison des artefacts de speckle, compliquant ainsi le processus de segmentation en aval.

Apprentissage de l'échantillonnage L'une des principales limites de notre approche réside dans la difficulté de SCONet à reconstruire des cartes de segmentation très détaillées à partir de nuages de points de contours bruités. Cependant, nous avons identifié un sous-ensemble de points de contour plus informatifs pour cette tâche. Les travaux futurs pourraient explorer la sélection automatique de ces points pendant le traitement par SCONet. Une solution potentielle consisterait à remplacer le module PointNet par un module d'échantillonnage appris. Nous pourrions, par exemple, nous inspirer des méthodes basées sur les scores, comme

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Nezhadarya et al. (2020), et concevoir un module calculant des poids par point, indiquant leur pertinence pour le traitement en aval. Il serait particulièrement instructif de comparer le nuage de points extrait de l'ensemble des points de contours aux nuages de points étudiés dans le Chapitre 3.

Apprentissage des hautes fréquences L'un des facteurs expliquant la robustesse de notre méthode est son décodeur basé sur un MLP. En effet, sa tendance à favoriser les basses fréquences le rend plus robuste aux changements de distribution des données, comme les décalages entre jeux de données. Cette propriété est à la fois un atout et un inconvénient, car elle explique également pourquoi SCONet génère des cartes de segmentation trop lisses. Dans les travaux futurs, il serait pertinent d'explorer les méthodes proposées dans la littérature pour améliorer la capacité des méthodes implicites à apprendre les hautes fréquences. Une solution existante consiste, par exemple, à projeter les coordonnées des points de requêtes dans un domaine de plus haute dimension (Tancik et al., 2020). Une autre approche potentielle pourrait consister à ajouter un vecteur à chaque point de requête représentant son contexte local, ce qui pourrait permettre au réseau de générer des cartes de segmentation plus détaillées.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Introduction

Context of the work

Medical imaging refers to a set of techniques that enable visualization of the human body's internal structures, generally for clinical purposes. For example, magnetic resonance imaging (MRI), computed tomography (CT) or positron emission tomography (PET) are part of the many available imaging modalities that are commonly used to make a diagnosis or to monitor a treatment. Medical images can notably be used to extract precise shape or volume information about organs or abnormal tissues through the segmentation of these structures. This task consists in subdividing an input image into a set of non-overlapping regions, producing a segmentation map where each pixel (or voxel in the case of a three-dimensional volume) is assigned a specific label. However, manual segmentation is a time-consuming and labor-intensive task, highlighting the urgent need for automated methods to alleviate the workload for clinicians. This need is further amplified by the growing number of medical images being acquired. (NHS, 2020).

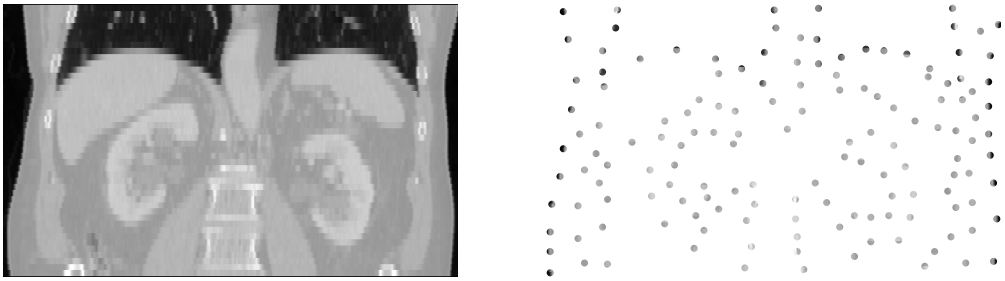
Early tools for automated medical image segmentation relied on handcrafted features and algorithmic approaches such as thresholding or region growing. However, these traditional methods often require to perform manual parameter tuning and are sensitive to the image quality. In contrast, more recent approaches heavily rely on deep neural networks, a type of algorithm that learns to complete a task by optimizing its parameters through processing of a large amount of data. The use of neural networks to segment medical images helped designing more robust algorithms notably based on convolutional layers and attention-based architectures.

Research of increasingly accurate segmentation models yet often fails to address the challenges related to the rising computation needs for these methods. Indeed, the training and deployment of deep learning models generally require substantial energy resources. This can be a problem given that the main part of the global energy production continues to rely on sources that emit greenhouse gases, which are responsible for climate change. Another crucial challenge is that the prohibitive computation cost of these segmentation algorithms can jeopardize their deployment in low-resource clinical settings, which raises questions about accessibility and equity.

The computation resources required to run medical image segmentation algorithms are especially high due to the high dimensionality of the processed images. These images are indeed typically represented as three-dimensional (3D) volumes composed of millions of voxels

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

resulting in high memory usage and long processing times. One potential solution to reduce these costs is to eliminate redundant information from the processed data. Redundancy in medical volumes is evident in two main ways: small neighborhoods around each voxel typically show little intensity variation, and large portions of the image often consist of empty background voxels. Processing only voxels of interest could therefore significantly improve computational efficiency. This approach would require redefining how images are represented, shifting from a regular volumetric grid to an unstructured 3D format. Among the available 3D data representations, point clouds, which consist of sets of 3D coordinates, are naturally sparse and could serve as a lighter alternative to dense voxel grids. Replacing the volumetric grid with a point cloud representation that includes only informative points (and potentially associated features) could help reduce the computational cost of the segmentation algorithms. Figure 6 illustrates a set of points extracted from the volume which all belong to the contour of the different anatomical structures.



(a) *Volume* **(b)** *Point cloud*

Figure 6: *Sample CT volume and its corresponding point cloud extracted from the contour map. Visualization in two dimensions for improved readability.*

While the use of point clouds in segmentation workflows is not new, most previous studies have focused on refining segmentation maps generated by volume-based neural networks. Few works, however, have completely removed the volume representation from the learning process.

Objectives

The main objective of this thesis is to explore whether medical image segmentation can be effectively performed using an unstructured 3D representation instead of the original image volume. To achieve this, we aim to develop a novel segmentation approach based on point clouds. The designed workflow should not only generate accurate segmentation maps of the input volumes but also be computation efficient. There are several challenges that should be addressed in order to complete this main objective. First, we have to identify the points that must be extracted from the input volume: the resulting point cloud should indeed contain sufficient information for the reconstruction of a segmentation map while remaining lighter than the volume. Another challenge lies in the development of a neural network which must be flexible enough to process the sparse point cloud representation extracted from the volume and to output a dense segmentation map.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Organization of the thesis

This thesis is divided into four chapters. Chapter 1 first introduces the state of the art necessary to contextualize this work. The chapter explores two key research areas. First, we explore recent methods for medical image segmentation, with a particular focus on deep learning-based approaches and computationally efficient techniques. Next, we propose an overview of the methods used to process alternative 3D representations beyond volumetric grids such as point clouds and implicit representations.

Chapter 2 introduces a novel medical image segmentation workflow which is the main contribution of this thesis. The proposed approach consists in first extracting a contour point cloud from a medical volume and then using SCONet, an implicit neural network adapted from convolutional occupancy networks (Peng et al., 2020), in order to generate a multi-label segmentation map. The chapter also presents several training and inference strategies which can improve the model's performance in terms of segmentation accuracy or efficiency.

Chapter 3 presents our second study where we provide an in-depth study of the contour point cloud processed by our SCONet model. We notably explore several sampling strategies and identify the points in the set that are more informative for the task of organ segmentation.

Finally, Chapter 4 introduces our third study where we explore the robustness of the proposed workflow with a focus on the challenges of dataset shift and cross-modality learning.

We conclude this manuscript by giving some perspectives on the developed work.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Chapter 1:
State of the art

Contents

I Medical Image Segmentation 6
I.1 Introduction 6
I.2 Traditional segmentation techniques 7
I.3 Deep learning-based methods 8
I.3.1 Convolutional Neural Networks 8
I.3.2 Transformers 11
I.4 Efficient Medical Image Segmentation 14
I.4.1 Motivation 14
I.4.2 Efficient Deep Learning 15
I.4.3 Applications in medical imaging 16
I.5 Metrics 18
I.5.1 Evaluation of the segmentation quality 18
I.5.2 Evaluation of the computation cost 19
II Unstructured 3D data 20
II.1 Introduction 20
II.2 Point cloud extraction from an image 21
II.2.1 Edge point detection 22
II.2.2 Keypoint detection and description 23
II.3 Point cloud processing 26
II.3.1 Methods 26
II.3.2 Point cloud sampling 29
II.3.3 Applications in medical imaging 30
II.4 Implicit Neural Representations 31
II.4.1 Definition 32
II.4.2 3D shape reconstruction with INR 32
II.4.3 Neural Radiance Fields for view synthesis 35
II.4.4 Learning high frequencies 36
II.4.5 Applications in medical imaging 36
III Conclusion 38

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

In this chapter, we explore two areas of research in order to provide a state of the art for our work. We remind the reader that the target task in this thesis is medical image segmentation and that we aim to complete this task more efficiently by taking advantage of sparse and unstructured three-dimensional (3D) representations. We therefore start this chapter by investigating popular methods for medical image segmentation and continue in a second part by describing unstructured 3D data and related processing approaches.

I Medical Image Segmentation

I.1 Introduction

In image processing, segmentation is a task that aims at partitioning an image into non-overlapping regions. This task can also be seen as a classification of the pixels (or voxels when working with a 3D image) since it consists in assigning a label to each pixel. In the medical domain, segmentation can be used to identify pathological tissues or anatomical structures such as blood vessels, bones or organs. Organ segmentation and in particular abdominal organ segmentation is essential for many clinical applications where an organ's volume or shape can be a crucial information. First, it is useful for disease diagnosis. For example, accurate measurement of the volume of the liver can indicate hepatomegaly, an abnormal enlargement of the liver which is characteristic of several conditions (Perez et al., 2022). Surgery planning can also benefit from organ segmentation: knowing the precise volume of an organ is notably important for transplant surgeries (Nakayama et al., 2006). Additionally, this tool is useful for treatment planning prior to radiation therapy (Sykes, 2014). However manual segmentation of medical images is a time-consuming task, especially for the large volumes generated by magnetic resonance imaging (MRI) or computed tomography (CT) which can contain millions of voxels. Therefore, the development of automated tools for the segmentation of medical images is needed.

Evidence of the growing interest in the segmentation of anatomical structures can be found in the large number of datasets and challenges covering this subject that have been published in the last ten years. In this paragraph, we focus on the abdominal multi-organ segmentation task. The BTCV challenge (Landman et al., 2015) is a pioneering work that provided one dataset with annotations of 13 abdominal organs for 50 CT volumes from one medical center. More recent datasets and challenges provide volumes and their annotations for a higher number of cases in order to reduce the bias in the developed methods. They also try to add more variability in their datasets by including images of healthy and sick patients taken from multiple vendors and multiple centers. The AbdomenCT-1K dataset (J. Ma et al., 2022b) is one notable example with 1000 annotated CT volumes from 12 medical centers. While most datasets only provide CT volumes, others include MRI images as they are also used in clinical practice. One early example is the VISCERAL challenge (Jimenez-del-Toro et al., 2016), which gives annotations of 20 abdominal structures for 40 volumes acquired from two CT modalities and 40 volumes from two MRI modalities. The dataset from the CHAOS challenge (Kavur et al., 2021) can also be mentioned: the authors provide annotations of four organs from 40 CT scans

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

and 40 MRI volumes. Additionally, the AMOS challenge (Ji et al., 2022) provides annotations for 300 CT scans and 60 MRI volumes. Finally, we can also cite the FLARE challenge (J. Ma et al., 2022a) for which the authors provide new annotations of existing public abdominal image datasets. The FLARE challenge is especially interesting regarding the task of multi-organ segmentation because its evaluation that takes into consideration the efficiency of the methods proposed by the participants. Table 1.1 gives a summary of the mentionned challenges and datasets.

Dataset	Year	# structures	# annotated scans	Modality	# centers	Evaluation metric
BTCV (Landman et al., 2015)	2015	13	30	CT	1	Dice
VISCERAL (Jimenez-del-Toro et al., 2016)	2015	20	80	CT&MRI	1	Dice, AVD
CHAOS (Kavur et al., 2021)	2020	4	40	CT&MRI	1	Dice, RAVD, ASD, MSD
AbdomenCT-1K (J. Ma et al., 2022b)	2021	4	1000	CT	12	/
FLARE (J. Ma et al., 2022a)	2021	13	411	CT	11	Dice, NSD, Running Time, GPU memory
AMOS (Ji et al., 2022)	2022	15	360	CT&MRI	2	Dice, NSD

Table 1.1: *Popular abdominal segmentation challenges and datasets.*

In this section, we present the state of the art for medical image segmentation, covering both traditional methods and deep learning-based approaches. We also discuss optimization techniques and efficient deep learning algorithms, as one of the main goals of this thesis is to reduce the computation cost of the segmentation process.

I.2 Traditional segmentation techniques

While deep learning has become a preferred approach for image segmentation in recent years, there are many traditional methods that were widely used before (Sharma et al., 2010). In the context of medical image processing, popular methods include :

- **thresholding-based** methods, where the segmentation label of a pixel depends on whether its intensity value is greater or smaller than a threshold. These methods therefore aim to identify one or several optimal thresholds for a specific task and are widely used for medical applications (Jardim et al., 2023).
- **region-based** algorithms. One commonly used method is the region-growing algorithm, which starts with one or several seeds, and expands the segmented areas by aggregating neighboring pixels when they verify a homogeneity criteria. This is often used for tumor segmentation (Biratu et al., 2021).
- **deformable models** (Metaxas, 2012) where an initial contour is deformed by optimizing an energy function so that it fits with the border of an object.
- **atlas-based** methods. This consists in guiding the segmentation process by mapping a reference image with ground-truth segmentation onto a new image (Kéchichian et al., 2018).
- **clustering-based** methods such as the K-means or the Fuzzy C-means algorithm (Ng et al., 2006; Christ et al., 2011). These approaches are unsupervised machine learning algorithms and aim to find natural groupings of the pixels in a multi-dimensional feature space.

One advantage of these traditional methods is their simplicity which makes them easily

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

interpretable. Moreover, most of them are efficient and therefore work well even with limited computation resources. However, they are not easily generalizable and perform poorly on noisy data or heterogeneous datasets, contrary to deep learning-based methods, which are generally more robust and more accurate (Yan Xu et al., 2024).

I.3 Deep learning-based methods

Deep learning-based methods gradually replaced traditional approaches for image processing tasks. These methods use artificial neurons called *perceptrons* to learn from a large amount of data. A perceptron can be defined as a binary classifier algorithm. More specifically, a perceptron applies a weighted sum followed by an activation function to an input signal. The learning mechanism in neural networks is based on the optimization of the perceptron's weights. This is done through backpropagation of the error that comes from comparing the perceptron's output to a reference or ground truth value. The *multi-layer perceptron (MLP)* is the most basic type of neural network and consists in stacking perceptrons into layers, in order to learn more complex patterns. A MLP is characterized by its width, that is to say the number of perceptrons in one layer, and by its depth, which is its number of layers. A simple way to process an input image with a MLP is to process the flattened image. However, as each neuron in one layer is connected to every neuron in the next layer, this means that the network has to learn a separate weight for each pixel, leading to a huge number of parameters, especially for large images. Besides, flattening the image leads to a loss of the spatial relationships between neighboring pixels. MLPs are thus inefficient for image processing and were never popular for this type of applications. They can however be more useful to process data of lower dimension such as points (as we will explain in Section II). In this section we present CNNs and transformers, two more recent architectures compared to the MLP that have produced better results for image processing tasks including segmentation.

I.3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) were originally introduced by LeCun et al. (1989) to perform image classification. They have later become a building block for many image processing algorithms, in particular after AlexNet (Krizhevsky et al., 2012) obtained groundbreaking results on the ImageNet dataset. This section first defines CNNs and presents some key models for medical image segmentation.

a) Definition

CNNs are primarily composed of two key elements, as illustrated in Figure 1.1. First, the *convolution layer*, which is the backbone of CNNs, performs a convolution on the input image. For each pixel, a dot product is computed between that pixel's neighborhood and a similarly sized matrix called filter or kernel. In CNNs, the weights of the kernel are learned during training. There can be one or several convolution layers. An activation function such as ReLU is used after each convolution layer. This function introduces a non linearity to help the network learn more complex patterns. The output image, which is called a feature map is then fed to the the

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

second type of layer of CNNs, which is the *pooling layer*. This layer helps reducing the dimensionality of the data by performing a down-sampling operation. Contrary to convolution layers, pooling layers do not have learnable weights. In a CNN, several convolution layers and pooling layers are stacked. This creates a hierarchical architecture that enables CNNs to learn features with increasingly wider receptive fields. As a final step, a fully-connected MLP layer processes the flattened feature maps to output the class prediction of the input image, in the case of a classification network. For a segmentation network the last layer is usually a convolution layer which outputs a segmentation map with the same size as the input image.

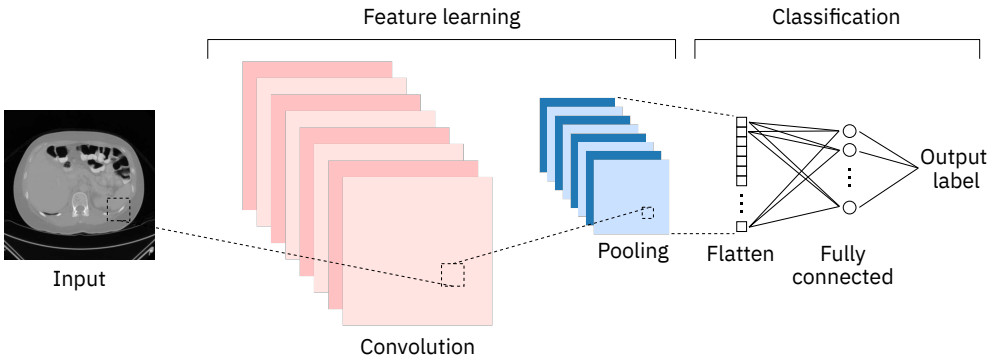


Figure 1.1: *Generic architecture of CNNs for image classification*

For each layer of a CNN, the network only has to learn a small number of parameters as it relies on a small kernel whose weights are reused across the entire image. Therefore, this approach helps reducing significantly the number of parameters learned by the network compared to previous MLP-based methods which use fully connected graphs of hidden layers.

The spread of CNNs for image processing is possible thanks to the use of Graphical Processing Units (GPU) which accelerate the training of deep learning models compared to traditional Central Processing Units (CPU). This increased processing speed is possible thanks to the capacity of GPUs to perform efficiently the parallel matrix operations that are used in CNN training.

b) Popular CNN architectures

After AlexNet, CNN-based models have first succeeded in improving results by increasing the network's depth, that is to say the number of convolution layers. VGG (Simonyan et al., 2014) is a pioneering work that notably proposed to use a small kernel in order to obtain a very deep network. Following works focused on improving the networks' architecture without stacking too many convolution layers as this can lead to the problem of vanishing gradients. This means that the gradient becomes 0 during training which stops the learning process. ResNet (K. He et al., 2016) for example introduced Residual Blocks which connect the output of a layer to further layers by skipping some layers between them. Stacking these blocks to form a residual network improves performance compared to the VGG baseline. Note that while this architectural innovation was introduced for CNN, it can also be used for MLP networks.

These first networks were used for image classification and therefore require to have at least one fully-connected layer to output a classification label. On the contrary Fully Convo-

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

lutional Networks (FCNs), first proposed by Long et al. (2015) for the task of segmentation, replace the last fully-connected layer by a convolution layer. FCNs use upsampling techniques like interpolation or deconvolution to transform an intermediate feature maps with smaller size to the original image size and generate a per-pixel prediction.

While most deep learning networks are designed for natural images, specialized architectures have been proposed to address the unique characteristics of medical images. Specifically, these images can be three-dimensional and have a wider and clinically meaningful intensity range. Moreover, annotated medical image datasets aren't usually as abundant as natural image datasets and they suffer from class imbalance issues (when one class significantly outweighs another class in one image or dataset). A key work for medical image processing is UNet (Ronneberger et al., 2015). Its encoder-decoder architecture with skip connections is shown in Figure 1.2. The encoder forms a contracting path which learns features at different stages or resolutions. The decoder then learns to reconstruct the feature maps to the input resolution. The skip connections propagate the information from the earlier layers in the encoder to the deeper layers in the decoder so as to not lose the spatial information existing in the first layers. The original UNet is designed for 2D processing but a more recent work introduces 3D UNet which can process volumes by performing convolution with 3D kernels (Çiçek et al., 2016). Three-dimensional processing with UNet and UNet-based networks usually requires for memory constraints to divide the input images into smaller sub-volumes called patches. Both 2D or 3D versions of UNet are popular backbones for many newly proposed medical segmentation models. For example, UNet++ (Zhou et al., 2018) starts from a UNet baseline and expands the model by adding several elements. This includes the use of deep supervision where the intermediate feature maps are used to supervise the network. UNet++ also adds deep convolutional layers in the skip connections in order to bridge the semantic gap between the encoder and decoder.

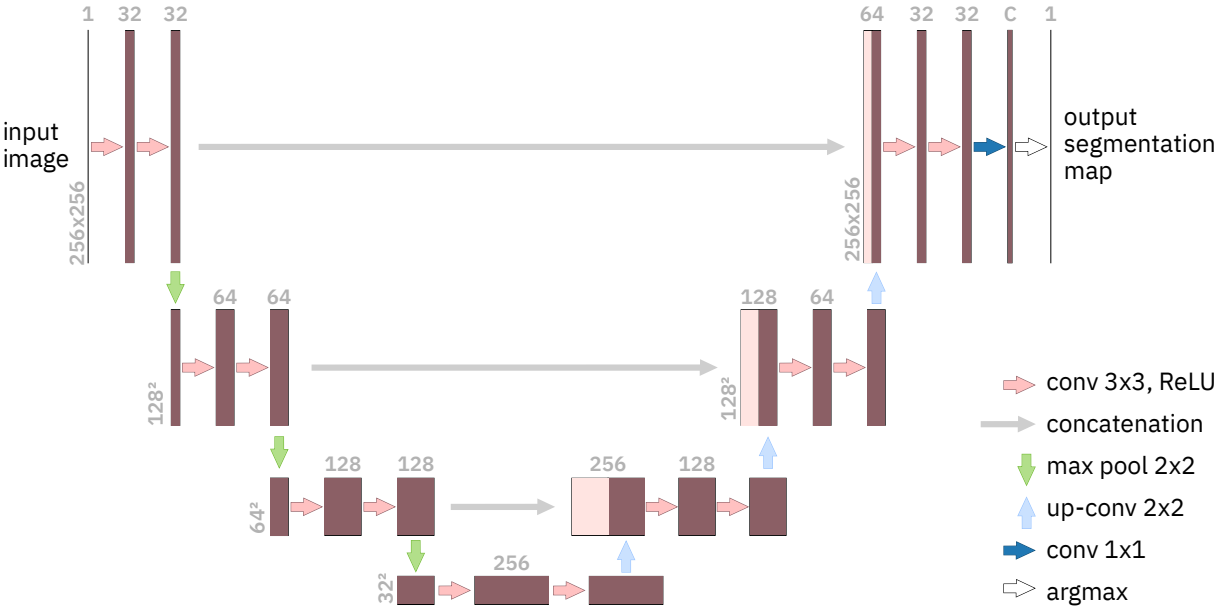


Figure 1.2: *Example of a four-stages 2D UNet architecture with 32 feature maps at the first level. C is the number of target labels. Figure adapted from Ronneberger et al. (2015).*

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

nnU-Net (Isensee et al., 2021) is another important work which presents a new framework based on UNet. Rather than including new architectural designs, the authors propose a framework that can adjust itself to any given dataset with an adaptive network configuration and pre- and post-processing steps. More specifically, nnU-Net first extracts a set of dataset-specific properties which are used for setting data-dependent parameters including the batch size (the number of samples processed by the network during one forward pass) or some pre-processing steps (resampling, normalization and others). Another set of parameters is fixed which includes the loss function, the number of epochs and the data augmentation strategy. Those two sets of parameters are used to train three configurations of UNet. These configurations are a 2D U-Net, a standard 3D U-Net, and a cascade 3D U-Net. This last configuration consists in a coarse UNet followed by a high resolution UNet: the first network outputs a coarse segmentation map and the second network refines it. nnU-Net keeps the best configuration or combination of configurations of these three networks. Post-processing is done if necessary by using the connected components algorithm which consists in removing the smallest segmented elements. The authors explain that using such a framework avoids having to manually tune a network, which often results in a sub-optimal model. The nnU-Net framework has shown promising results for many target structures. For the task of multi-organ segmentation for example, TotalSegmentator (Wasserthal et al., 2023) is a tool based on nnU-Net trained to segment 104 anatomical structures including organs, bones and muscles. One major disadvantage of the nnU-Net framework however is that its exhaustive hyperparameter tuning leads to a costly training process (Selvan et al., 2022).

CNNs have good capabilities in capturing local features. Indeed the inductive bias of the convolution makes them translation invariant. However, they fail to efficiently capture long-range dependencies (X. Wang et al., 2018). Additionally, CNNs also suffer from high computation cost when working on large volumes because the memory used grows cubically when increasing the input resolution.

I.3.2 Transformers

a) Attention-based models

Transformers have become a widely used architecture for many image processing tasks, including medical image segmentation. They are based on the *attention* mechanism, where models learn to focus on the most relevant part of the input data. This involves computing attention weights for each element of the input sequence or image for the given task (Bahdanau et al., 2014). This requires to first turn each part of the input data into vectors (also called token) and compute attention weights based on how similar or how correlated the vectors are. These weights are then used to emphasize the influence of each part of the input sequence for the given task.

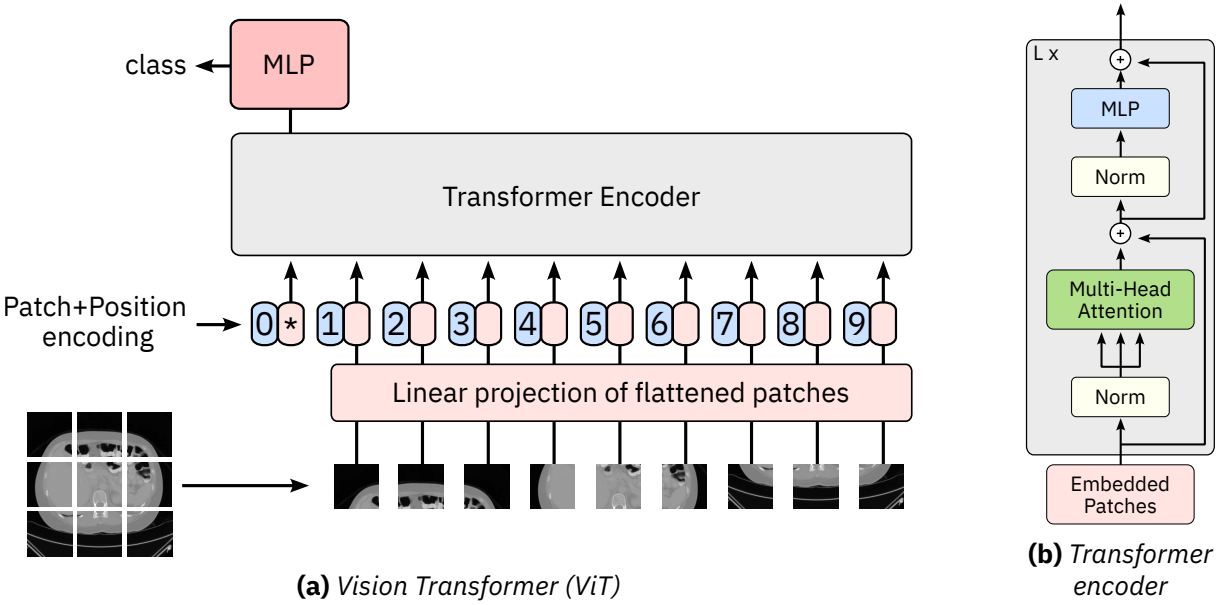
Early use of attention mechanisms for image processing consisted in adding attention modules to CNNs. In the field of medical image segmentation, Attention U-Net is a notable work by Oktay et al. (2018). The authors build on a U-Net model and use an attention gating module in the skip connections. This module prunes the irrelevant parts of the early feature maps before

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

they are concatenated in the deeper layer which helps highlighting important features for the segmentation task in an input image.

b) Transformer models

The transformer architecture is based on the *multi-head self-attention (MSA)* mechanism, as proposed by Vaswani et al. (2017). While standard attention captures relationships between different sequences, *self-attention* can capture relationships between different elements of the same sequence. This allows to take into account the context of the element in the sequence when generating its feature. In the original Transformer paper, self-attention is computed with a dot-product between the different tokens. As for *multi-head attention*, the attention is computed in parallel multiple times and then concatenated. Focusing on different parts of the input sequence at the same time helps learning more complex relationships between the different elements. Another important component of transformers is *positional encoding* which consists in adding the relative position of the element in the sequence to the token. This helps adapting the learned features to the local structure of the input. Architecture-wise, transformers stack alternating layers of MSA blocks and feed-forward blocks as illustrated in Figure 1.3b. The feed-forward block consist in a small MLP and adds non-linearity with the use of the ReLU activation function.



(a) *Vision Transformer (ViT)* **(b)** *Transformer encoder*

Figure 1.3: *ViT overview. Figure taken from (Dosovitskiy et al., 2020).*

While transformers were initially introduced to process text, they have since been adapted for image tasks as well. Vision Transformer (ViT) (Dosovitskiy et al., 2020) is a pioneering work which directly transposes the transformer architecture to process images by first splitting them into 2D patches. Those patches are then flattened, and processed through a transformer encoder, following a similar approach to the original transformer work (Vaswani et al., 2017). Finally, a MLP classification head predicts the class of the image. Figure 1.3 illustrates the architecture of ViT.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

One problem of ViT is that they are computationally expensive because the computational complexity of the self-attention mechanism is quadratic to the input image size. In an effort to solve this issue, Swin transformers (Ze Liu et al., 2021) perform self-attention locally on non-overlapping windows. Since this could lead to the loss of dependencies between the different windows, the authors shift the windows between consecutive self-attention layers. The Swin transformer architecture builds on these blocks and reaches linear computational complexity to the input image size while improving the overall results for various image processing tasks. The authors also introduce a patch merging operation which helps reducing the number of tokens after each Swin block and therefore building hierarchical architecture.

Many methods now perform medical image segmentation with models that combine transformers and CNN blocks. Among them, SwinUNETR (Tang et al., 2022) and its follow-up SwinUNETR-V2 (Y. He et al., 2023) are notable works which both propose a UNet-like architecture with an encoder, a decoder and skip connections that connects the two. However, while the decoder is CNN-based (like UNet), the encoder uses Swin Transformer modules. When compared to SwinUNETR, the V2 version adds convolution layers interleaving the Swin blocks. Another difference between the two works is that SwinUNETR relies on a pre-training of the model on pretext tasks: the network is trained in a non-supervised manner in order to learn features inherent to the modality or dataset. One example of pretext task is in-painting which consists in learning how to reconstruct the masked part of an image. This pre-training is not necessary for SwinUNETR-V2.

Lastly, it is also worth mentioning that transformers are the backbone for many foundation models. The particularity of these models is that they can complete a wide variety of tasks because they have been trained on a large amount of data. They can therefore be fine-tuned to novel tasks. The capacity of transformers to learn strong feature representations during pre-training (Raghu et al., 2021) makes them a tool of choice to build foundation models. Specifically, in the field of image processing, the Segment Anything Model (SAM) (Kirillov et al., 2023) aims to segment any object in an image. Architecture-wise, SAM has two encoders. The first one is a large ViT encoder which processes the image input. The second one encodes a prompt (a text, a bounding box) which is optional and serves as a guide for the segmentation task. The two embeddings are fed to a light ViT decoder to output a segmentation map. SAM is trained on natural images so it does not perform well when applied on medical images. Many works have therefore fine-tuned this model for medical image segmentation like SAMed (K. Zhang et al., 2023). While efficient fine-tuning methods decrease the training time for novel tasks, SAM and in general foundation models remain heavy models with millions of parameters and their original training cost is very high with up to weeks of training time on hundreds of GPUs (Bommasani, 2021).

To conclude, the main advantage of transformers is their capacity to capture long range dependencies thanks to the self-attention mechanism. However they remain costly algorithms with a high training and inference cost (Khan et al., 2022).

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

I.4 Efficient Medical Image Segmentation

I.4.1 Motivation

The research for efficient algorithms is often not one of the main goals for works dealing with the task of multi-organ segmentation. Indeed, among the existing challenges introduced in Section I.1, FLARE (J. Ma et al., 2022a) is the only one that takes efficiency into consideration in order to evaluate the methods proposed by the participants. This lack of interest for efficiency compared to segmentation accuracy is a blind spot in regards to the following arguments.

First of all, it has been shown that the use of artificial intelligence (AI) is harmful to the environment notably because of an increased energy usage in the data centers used for storage and processing (Guidi et al., 2024). More specifically in the field of medical imaging, the development of segmentation algorithms has a significant cost in terms of energy consumption (Selvan et al., 2022). This has a direct impact on the algorithms' greenhouse gases emissions which are one of the main contributors to the climate change crisis. Medical images usually have large sizes (CT or MRI volumes contain millions of voxels) and therefore involve a high energy consumption for storage and processing (Buckley et al., 2021). The development of more efficient medical image processing approaches and the use of better data compression techniques could be two ways to address this issue (Doo et al., 2024) [1].

Additionally, designing efficient approaches for medical image segmentation facilitates their usage in clinical settings, where the access to the latest generation GPUs is not guaranteed. Furthermore, working on the efficiency is also a matter of equity as those algorithm could eventually be deployed in regions with limited access to computation resources (J. Chen et al., 2024).

There also has been a growing interest for point-of-care (POC) applications, where the imaging and the processing is done on light devices by the side of the patient. Typical applications include skin lesion segmentation, done on a smartphone, or ultrasound analysis, done on a specialized POC ultrasound device. The use of these light devices requires efficient methods that can for example run on CPUs (Valanarasu et al., 2022).

Finally, the lack of large (annotated) datasets is an important challenge in the field of medical imaging which can be addressed through the development of data-efficient methods. While this aspect of efficiency is not the main focus of this work (we are more interested in memory or time efficient solutions), these methods are also relevant for this section because working on a smaller number of samples ultimately impacts the training cost of the segmentation algorithms.

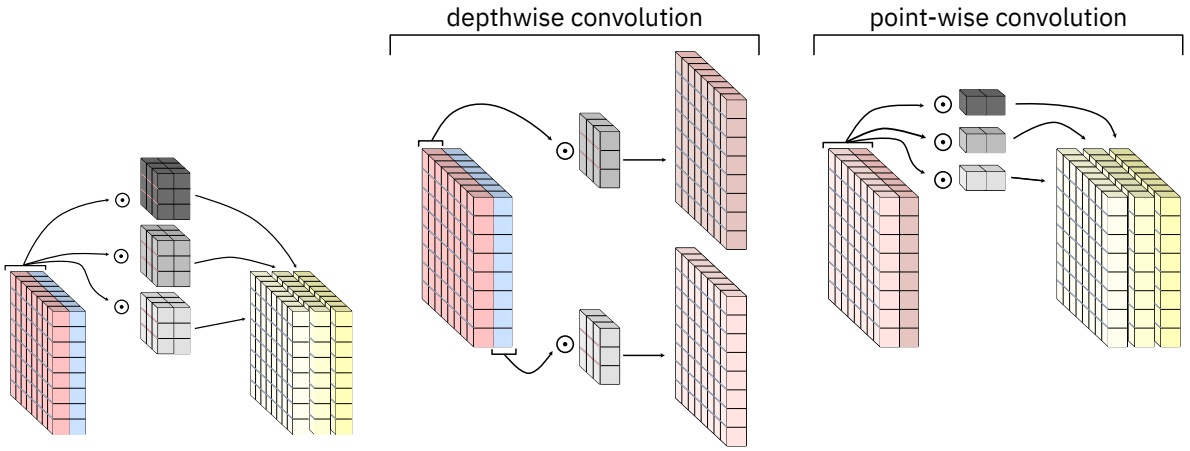
[1] There are two blind spots to this conclusion. First we remind the reader that the high energy consumption of deep learning algorithms is only one aspect of the effect of AI on the environment. Other issues notably include the increased water usage to cool data centers or the the pollution generated by the manufacturing and recycling (or lack thereof) of the hardware used. Secondly the development of efficient deep learning algorithms does not guarantee a decrease in energy consumption. Indeed, the Jevons' paradox explains that it can even have the opposite effect: savings in cost linked to the improved efficiency of a technology can lead an increase in its usage. The interested reader can find more information about those two issues in the work by Luccioni et al. (2025).

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

I.4.2 Efficient Deep Learning

Network optimization The first way to build efficient deep learning algorithms involves using optimization techniques to reduce the size of an existing model or its number of computations. One first approach is *pruning* which consists in removing a subset of the parameters while ensuring that the accuracy of the model remains the same. Network optimization can also be done through *quantization*, which reduces the precision of the model's weights (Liang et al., 2021). A network can also be optimized through *tensor decomposition* techniques where the model's layers are replaced by their low-rank tensor approximations, which reduces the number of parameters (X. Liu et al., 2023). Finally, model compression can also be done using *knowledge distillation*, where a smaller model, called student model, is trained using information from a larger teacher model (Ba et al., 2014). The goal is for the student network to be able to perform the same task as the teacher network with the same accuracy but using less parameters.

Efficient modules In addition to these network optimizations, it is also usual to design more efficient network architectures and modules. For example MobileNets (Howard et al., 2017) use *depthwise (DW) separable convolution* which consists in factorizing a standard convolution into (1) a DW convolution, which applies a single convolutional filter for each input channel and (2) a point-wise convolution which is a 1×1 convolution. Figure 1.4 illustrates the difference between a standard convolution and a DW separable convolution and shows how this leads to a decrease in number of parameters and operations.



(a) *Standard convolution* **(b)** *Depthwise separable convolution*

Figure 1.4: *Standard or depthwise separable convolution. An input image with $C_{in} = 2$ channels is filtered with a 2D kernel with size 3×3 to get $C_{out} = 3$ feature map.*

Automated designs While some works design novel architectures manually, other try to find optimal models automatically through *Neural Architecture Search (NAS)*. NAS models are usually defined by (1) their search space, which corresponds to the space of all the available architectures, and (2) the search algorithm which defines how to explore this space. Early methods

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

succeeded in reaching more computationally efficient models compared to manually designed ones (Zoph et al., 2016). However they usually had to explore a vast search space which lead to high search costs and long training times. Subsequent works proposed more efficient search algorithms through, for example, parameter sharing (Pham et al., 2018). This popular method consists in training a single network called super-net: all the candidate models are sampled from this super-net and therefore do not need additional training.

Hardware Finally, in practice the computation cost of a neural network is also greatly influenced by the hardware used. The running time of processing input data depends for example on the cost to access the part of the memory where this data is stored. A long memory access can be a bottleneck when working on GPUs (N. Ma et al., 2018). There are many methods designed to do infrastructure optimization, which include the search for hardware-aware models with NAS (M. Tan et al., 2019), or the use of acceleration tools, such as Nvidia's TensorRT (Vanholder, 2016).

I.4.3 Applications in medical imaging

Network optimization for medical image segmentation A first approach to design more efficient medical image segmentation methods consists in applying the model optimization techniques mentioned in the previous section to improve a model's efficiency without modifying its architecture. For example Selvan et al. (2022) show that the reduction in the energy consumed during training of nnU-net can reach 30% when using automatic mixed precision. The drop in consumption cost does not lead to a drop in segmentation accuracy. This optimization technique belongs to the quantization category: the precision of the layers is automatically reduced from 32 to 16 bits on some layers.

Efficient architectures Regarding the design of specific architectures for medical image segmentation, one notable example is UNeXt (Valanarasu et al., 2022). This model builds on UNet but replaces some of the standard convolutional blocks by tokenized MLP blocks. In these modules, the feature maps are projected to generate tokens, which are then processed by MLPs. This approach helps to reduce the dimension of the deeper features and therefore the number of parameters of the network. Other techniques such as the use of DW convolution instead on standard convolution or the reduction of the number of feature maps in the convolution stages also help lowering the computation costs. Like many works aiming to build lightweight architectures, UNeXt is particularly well-suited for POC applications.

More recently, EMFFormer (Hao et al., 2024) introduced a lightweight transformer architecture for 2D segmentation with two key modules designed to improve efficiency. First the vector aggregation attention mechanism replaces self-attention. More precisely, an element-wise multiplication replaces the dot product (matrix multiplication) operation used for self-attention, thus reducing the computation complexity of the model. Moreover, a serial multi-head attention module replaces the standard parallel multi-head attention. This reduces the redundancy among features. This change leads to a reduction in memory usage because the attention scores are therefore not computed simultaneously on the different heads. Finally

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

EMFFormer uses DW separable convolution in order to reduce the number of parameters. This operation is slightly changed compared to the original DW separable convolution because the channels are shuffled, encouraging more exchange of information.

Efficiency based on reduced data representations The prohibitive memory usage implied by processing high resolution medical volumes is often addressed by doing patch-based training and sliding-window inference. This technique is commonly used to reduce the size of the input image (and the intermediate feature maps) so that it can fit in GPU memory. Usually Gaussian importance weighting is applied at inference. It consists in giving more importance to voxels located close to the center of the window in order to reduce the stitching artifacts that can occur (Isensee et al., 2021). However, while this method indeed reduces memory usage, it also increases the computation complexity and inference time. To address both aspects of the computation cost of medical image segmentation, it is possible to work with a low resolution input image. F. Zhang et al. (2021) (winner of the FLARE challenge) for example uses a UNet on an under-sampled volume to extract a coarse segmentation map, which is refined in a second network, called EfficientSegNet. This network applies a 3D mixed pyramid pooling to captures global and local context more efficiently than self-attention. Other techniques are added to reduce the computation cost such as separable convolution, and the use of the addition operation instead of the concatenation to reduce the number of feature maps in the decoder. Finally, an other paradigm is to reduce the dimension of the data by extracting a lighter sparse representation from the input dense data. Models specialized in the processing of sparse data are then used to reconstruct a segmentation map. For example Point-Unet (Ho et al., 2021) uses point clouds and achieves better segmentation accuracy for lower memory usage.

Data efficient methods One common data-efficient segmentation method for when some annotations are available is the use of data augmentation to artificially increase the number of samples during training. An other method consists in doing transfer learning to adapt a model previously trained on one dataset to an other (smaller) dataset. Fine-tuning, where a pre-trained network is re-trained on a new set of data, can therefore be used to adapt a model learned on a large dataset of (natural) images to a smaller base of medical images. While it is possible to re-train the full network, it is often better to use parameter-efficient fine-tuning (PEFT), especially when working with large models. PEFT consists in freezing the backbone and training only a subset of its or a new set of parameters. Dutt et al. (2023) has shown that this technique gives the best results for medical image analysis when there is only a small number of images available. A common module used in PEFT for transformer networks is the low-rank adapter (LoRA) (E. J. Hu et al., 2022), illustrated in Figure 1.5. For each transformer block, the frozen pre-trained weights are updated by a trained matrix which can be decomposed in two low-rank matrices. SAMed (K. Zhang et al., 2023), previously described in Section I.3.2, for example uses this method to more efficiently fine-tune a SAM encoder to do medical segmentation.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

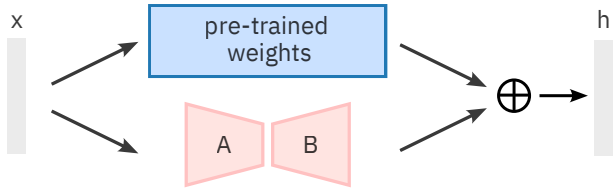


Figure 1.5: *Low-rank adpater (LoRA) module. A and B are two low-rank matrices. The computed feature vector is* $h = (W + AB)x$

I.5 Metrics

I.5.1 Evaluation of the segmentation quality

There are several metrics that are commonly used to evaluate the performance of a segmentation algorithm. For medical image segmentation, it is good practice to use both overlap-based metrics and boundary-based metrics which add complementary information regarding the shape of the segmented maps (Maier-Hein et al., 2024).

Although the metrics presented here are described for the evaluation of a binary segmentation, it is possible to extend them to a multi-class target by averaging each metric over all classes.

a) Dice coefficient

The Dice Similarity Coefficient (DSC or Dice) (Dice, 1945) is one of the most commonly used metrics in medical image segmentation. It is used to measure the overlap between a reference segmentation map R and a predicted segmentation map P, as shown in Figure 1.6. The higher the Dice, the more R and P overlap. $Dice = 1$ when the two maps align perfectly.

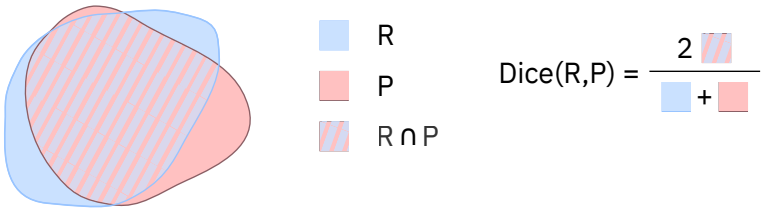


Figure 1.6: *Illustration of the Dice coefficient*

The Dice coefficient is computed as follows:

$$Dice(R, P) = \frac{2 \times |R \cap P|}{|R| + |P|}$$

b) Hausdorff Distance

The Hausdorff Distance (HD) (Huttenlocher et al., 1993) is one of the most commonly used boundary-based metrics for medical image segmentation. For all points from one object boundary to the other, it computes the maximum of all shortest distances, as shown in Figure 1.7. This metric is therefore very sensitive to spatial outliers. The Hausdorff distance is computed as follows:

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

$$HD(P,R) = max(\hat{d}_{HD}(R,P), \hat{d}_{HD}(P,R))$$

$\hat{d}_{HD}$ is the Hausdorff pseudo-distance, which is non-symmetrical and defined as :

$$\hat{d}_{HD}(P,R) = \underset{p \in P}{max}(d(p,R))$$

where $d(p,R) = \underset{r \in R}{min}(d(p,r))$ and $d(p,r)$ is defined as the euclidean distance between p and r.

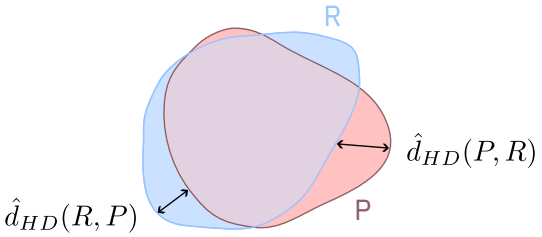


Figure 1.7: *Illustration of the Hausdorff Distance*

I.5.2 Evaluation of the computation cost

Deep learning models require both memory resources and time resources to run. These two space and time factors are commonly used to evaluate a model's efficiency.

a) Memory footprint

During training or inference, deep learning models allocate memory resources in order to store the initial input, operator weights, intermediate outputs, and the final output. As deep learning-based medical image segmentation algorithms are usually run on GPUs, it is common to report the *maximum GPU memory usage* when focusing on the efficiency of those algorithms. This value is usually reported for inference as the final user can work on more limited resources compared to the original training process. In this work, the GPU memory usage is computed using the nvidia-smi tool from the `pynvml` library[2].

The *number of parameters* of the network is another metric commonly used for evaluating networks, especially in the context of point-of-care applications. Indeed mobile medical devices have a limited memory which might not be able to support models with a large number of parameters (Ruan et al., 2022).

b) Time complexity

Time complexity can be evaluated in terms of *training time* or *inference time*. However, depending on the application, a more relevant metric is the number of floating point operations or *FLOPs*. This value represents the number of computations required to process a single input through the model. The main advantage of FLOP over inference time is that FLOP are hardware-independent, whereas inference time may vary depending on the device used.

[2] https://github.com/gpuopenanalytics/pynvml

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

That said, FLOPs are not always straightforward to compute, as the concept isn't precisely defined in all cases. In this work, we use the `fvcore` library[3], which provides an estimation of the number of FLOPs for many deep learning layers. These values should however be interpreted with caution, as they remain approximations. Lastly, it's worth noting that FLOPs are typically reported in the literature for a single input volume and not on the complete test split.

Additionally, in the case of point-of-care applications, some works like UNeXt (Valanarasu et al., 2022) also report the *CPU inference time*, which can be useful as many mobile devices do not have a GPU.

II Unstructured 3D data

II.1 Introduction

There are many ways to represent data in the 3-dimensional space: choosing one representation over another requires to take into consideration the nature of the data itself and its intended application. Figure 1.8 illustrates five commonly used 3D representations. They are:

- **Multi-view 2D projections:** a 3D object is represented through the 2D projections from various viewpoints;
- **Volumetric:** a 3D object is stored as a dense grid of voxels. A voxel is the 3D equivalent of a pixel: each voxel is a 3-dimensional cube which stores a feature value at its location, usually its intensity or RGB value. Note that voxels can be anisotropic, which means that they don't have the same size in the three dimensions. This is often the case for medical volumes;
- **Point cloud:** a 3D object is stored as a set of points described by their coordinates in the 3D space. Each point can also have additional features;
- **Mesh:** the surface of a 3D object is stored as a set of 2D polygons (also called faces) in the 3D space;
- **Graph:** a graph is a structure which consists in a set of nodes or vertices and edges. When used for 3D object representation, a graph can be seen as a point cloud where the points/vertices are connected by the edges;

Multi-view 2D projections and volumetric representations are stored in a regular format, which means that the data lies on a grid. On the contrary, point clouds, meshes and graphs do not have a pre-defined structure and therefore can be described as *unstructured 3D data* representations.

A natural way to process structured data is to use CNNs, which were previously introduced in Section I.3.1. However, while methods based on CNNs have led to good results not only for segmentation but also other tasks such as classification or object detection, they encounter important limitations, especially regarding their computation cost. Indeed, the computation complexity of a CNN for volumetric segmentation grows cubically along with the voxel resolution. This could be improved by processing the volume at a lower resolution, which compro-

[3] https://github.com/facebookresearch/fvcore

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

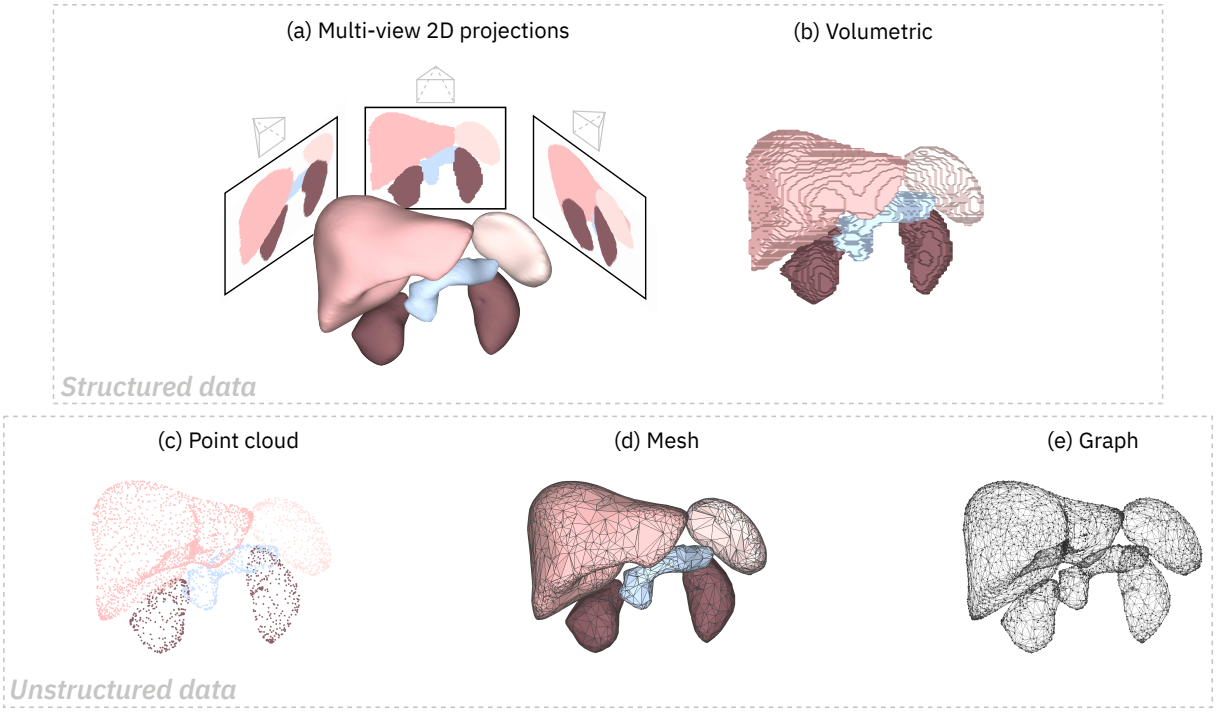


Figure 1.8: *3D representations of four organs*

mises the ability to analyze fine structures. The memory footprint could also be reduced by processing the volume using a sliding window, but at the cost of losing information about the global context (K. Xie et al., 2025).

Point clouds are a naturally sparse representation, which means that they are usually lighter than dense representations such as volumes. This could represent an opportunity to develop methods at a lower computation cost. While meshes and graphs are also sparse representations, we choose not to explore their potential use in this work for different reasons. First, point clouds are a lighter representation compared to graphs. Moreover they are also less complex to learn from than meshes (Qi et al., 2016).

In this section, we first explore different approaches to obtain a point cloud. We then investigate state-of-the-art point cloud processing methods and present several neural networks that learn implicit representations.

II.2 Point cloud extraction from an image

Some imaging modalities naturally generate a point cloud representation. For example LIDAR sensors compute a set of coordinates by measuring the time it takes for a laser to reflect on a surface. In the same say, intra-oral scanners (IOS) provide a point cloud representation of the teeth and gum by detecting reflected light on these structures. However, this work aims to provide a segmentation map of a data that is originally represented as a volume. This raises the question of the type of points that should be extracted in order to best represent the volume for the downstream segmentation task.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

II.2.1 Edge point detection

The *contours* or *edges* of an image can be defined as a sudden variation of intensity. In traditional computer vision approaches (as opposed to more recent deep learning-based computer vision algorithms) edge detection was applied to extract from an image important features that could be then used for higher level algorithms such as object detection. In particular, for the task of segmentation, detecting the contour points can be useful because the surface of the segmentation map usually coincides with at least part of the contour.

Detecting a variation of a signal can be done using its first or second order derivative. In particular, for an image, most edge-detection algorithms are based on the computation of the gradient ∇f. The gradient of a 2D image at location (x, y) contains information about the strength of the edge (through the magnitude) as well as its direction. The gradient is defined as:

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Early methods tried to extract edges by simply using the image gradient that they compute through a convolution with a specialized filter. In general, they apply two filters to compute the gradient along the horizontal x axis and vertical y axis. In a second step, the magnitude of the gradient is computed: the output matrix is the contour map. For example, the Sobel edge detector (Sobel, 1970) uses the following filters:

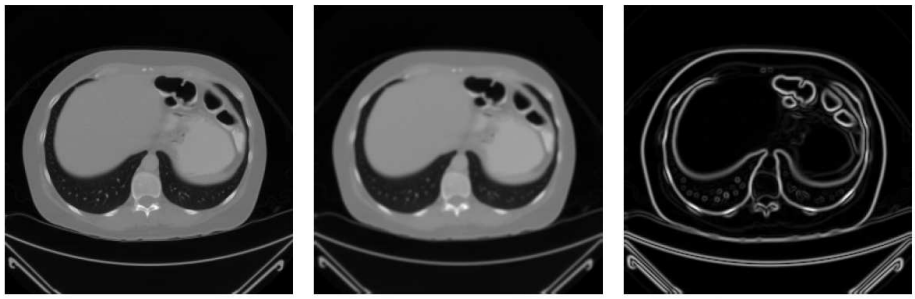
$$H_x = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}, H_y = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

However these methods are very sensitive to noise and do not take into account edge characteristics. To address this challenge, more recent methods have proposed more complex algorithms. For example, the Canny algorithm is able to detect well-localized and thin edges with a low-error rate (Canny, 1986). It consists in four main steps:

1. Noise reduction, using Gaussian low-pass filtering;
2. Gradient magnitude and direction computation using the Sobel operator. The magnitude is computed with $G = \sqrt{g_x^2 + g_y^2}$ and the direction with $\theta = arctan(\frac{g_x}{g_y})$;
3. Non-maxima suppression is applied in order to do get thinner edges. Let's consider a pixel at location (x, y), and its two neighbors along the direction of the gradient θ. If the magnitude at this pixel is inferior to the magnitude of one of its two neighbors, then the pixel is not an edge and its magnitude is set to 0;
4. Hysteresis thresholding is used to remove weak edges that are not connected to strong edges. Let's consider two thresholds T_{low} and T_{high}. For each pixel with position (x, y):
   - if $G(x, y) \leq T_{low}$, this pixel is not an edge;
   - if $G(x, y) \geq T_{high}$, this pixel is a strong edge;
   - if $T_{low} < G(x, y) < T_{low}$, the pixel is a weak edge. It is kept in the final edge map only if it is connected to another pixel already accepted as an edge.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

The tunable parameters for the Canny edge detection algorithm are the two thresholds and the standard deviation of the gaussian distribution which defines the sensitivity of the detector to noise. This algorithm can also be generalized in 3D to be applied on volumes. Example of intermediate results for the Canny edge detector can be found in Figure 1.9.



(a) *Original Image* **(b)** *After Gaussian filter* **(c)** *Gradient magnitude*



(d) *After non-maxima suppression* **(e)** *After thresholding*

Figure 1.9: *Canny edge detection steps*

Learning-based models have later been developed for contour detection. In particular approaches using CNNs (see Section I.3.1 for an introduction to this concept) to output edge maps have shown improvements in this task compared to traditional methods. S. Xie et al. (2015) is a typical work which uses multi-level image features with deep supervision. However the model is more than six times slower than Canny, even when run on a GPU. More recently PiDiNet (Z. Su et al., 2021) proposes a more efficient and accurate model by drawing inspiration from traditional gradient-based methods. In particular, instead of applying the learned convolution layers directly to the input image, they first compute the differences between neighboring pixels in the convolved area.

The extraction of an edge-based point cloud from a volume consists in extracting the coordinates of all the edge pixels. While this is not common, some works have already followed this idea like PointVoxelFormer (M. P. Heinrich, 2024). This approach indeed first extracts a contour-based point cloud in order to generate a segmentation map.

II.2.2 Keypoint detection and description

Keypoints in an image are a set of pixels located in distinctive and informative areas. These points typically correspond to corners, edges or blobs and are usually invariant to transformations in intensity, rotation or scale. The extraction of a set of keypoints builds a sparse representation of the original image which can be used for different tasks including object recogni-

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

tion or image matching. Most keypoint detection algorithms also propose a description step which encodes local information in a feature vector.

The Harris algorithm (Harris et al., 1988) is one of the most popular keypoint detectors. This algorithm aims to detects corners, which can be defined as areas with a high variation of intensity along all directions. Specifically, the Harris detector first computes the gradient in a small window around each pixel. This value is used to build the second moment matrix: the eigenvalues of this matrix are finally are used to classify the pixel as a corner or not. One main problem with the Harris corner detector is that it in not invariant to changes in scale. More recently, the SIFT algorithm introduces a Scale-Invariant Feature Transform (Lowe, 2004). SIFT uses multi-scale analysis to reach scale invariance. Despite its improvements in terms of speed thanks to its approximation of the second derivative, it remains slow for the description step. In contrast, the *Speeded Up Robust Features (SURF)* algorithm (Bay et al., 2006) reaches faster detection and description and is more robust than previous methods. We choose to detail the steps of this algorithm as we use SURF descriptors in the following chapters. The keypoints from an input image are detected as follows:

1. Integral image I_Σ computation: I_Σ at location $\boldsymbol{x} = (x, y)$ represents the sum of all the pixels in the rectangle formed between the origin of the image and the pixel $\boldsymbol{x}$. This allows to compute the sum of all pixels in any rectangular subregion of the image at a constant cost.
2. Hessian matrix $\mathcal{H}(\boldsymbol{x}, \sigma)$ computation: $\mathcal{H}(\boldsymbol{x}, \sigma)$ corresponds to the second order derivative of the image for pixel $\boldsymbol{x}$ at scale σ. $\mathcal{H}$ is usually computed using the convolution with the Gaussian second order derivative. This operator is replaced in SURF by the box filter second order derivative which can be computed very efficiently using an integral image, as illustrated in Figure 1.10. This approximation leads to an increase in speed for the keypoint detection.
3. Multi-scale analysis: $\mathcal{H}$ is computed for several scales of filters s in order to have a scale-invariant detector.

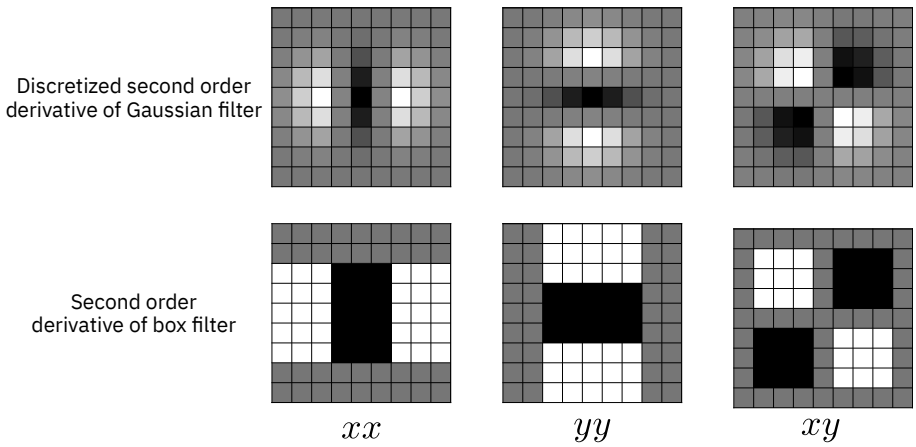


Figure 1.10: *Comparison of second order partial derivative of the Gaussian kernel with their approximated second order box filters. The filters are given for the horizontal x-direction, the vertical y-direction and the diagonal xy-direction. Figure adapted from Bay et al. (2006).*

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

4. Keypoint detection: Keypoints correspond to local extrema of the determinant of $\mathcal{H}$. A keypoint k is characterized by its location (x_k, y_k) as well as its detection scale s_k.

After detection, the authors propose a novel keypoint description algorithm, which is described as follows:

1. Orientation assignment: Haar wavelet responses are computed within a circular neighborhood around each keypoint. The angle for which the sum of responses is the largest corresponds to the dominant orientation of the keypoint. This step makes SURF rotation invariant and is accelerated thanks to the use of I_Σ for computation of the Haar wavelet responses.
2. Feature descriptor computation: For each keypoint k, a region with size $20s_k$ aligned on the orientation is constructed. The region is divided in 16 subregions. In each subregion, the Haar wavelet responses are computed along horizontal dx and vertical dy direction and summed, as illustrated in Figure 1.11. The absolute values of the responses $|dx|$ and $|dy|$ are summed for each subregion as well. This gives a vector with four values for each subregion: $\boldsymbol{v} = (\Sigma dx, \Sigma dy, \Sigma|dx|, \Sigma|dy|)$. This results in a $16 \times 4 = 64$ values feature vector for each keypoint.

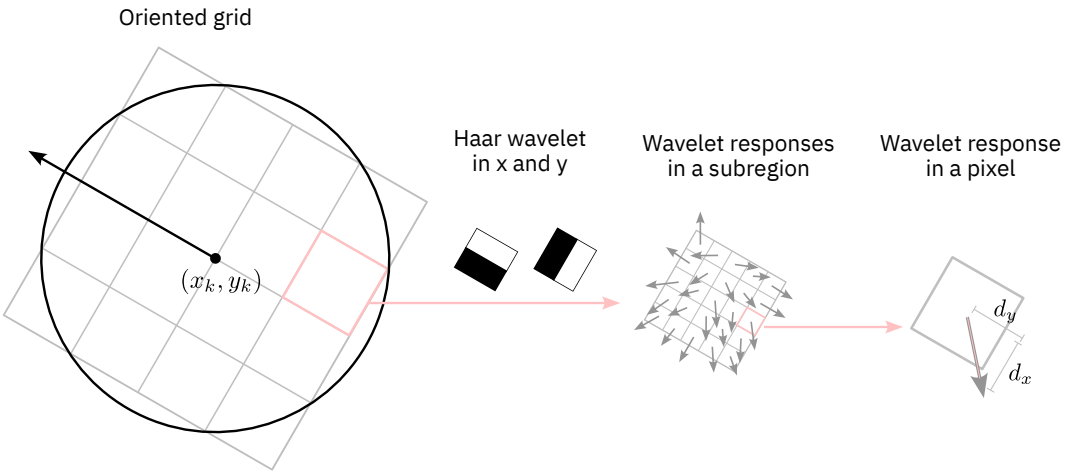


Figure 1.11: *Feature descriptor computation for a keypoint k. Figure adapted from Hassaballah et al. (2018)*

The keypoints extracted from one input image can be used to find a correspondence to other images.

The original SURF algorithm is described in 2D. Agier et al. (2016) proposes a 3D extension for processing of medical volumes. There are some changes made in the algorithm. First of all, the authors do not compute the keypoint descriptors on an oriented grid. Indeed, they argue that the variations in rotation is reduced for medical volumes as all patients are oriented in the same direction. Moreover, the size of the descriptor can be reduced to 48 values in 3D because the third axis introduces more local information.

More recently, promising results from deep neural networks for feature extraction have lead to the development of deep learning-based keypoint detector and descriptors. In particular, CNNs are a popular type of architecture for this task for reasons described in Section

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

I.3.1. Early approaches learn either a detector network or a descriptor network. One popular paradigm is to use a triplet network to learn local patch descriptors like in Kumar BG et al. (2016). More specifically, during training, triplet networks are optimized on three input patches: x, x^+ that belongs to the same class as x and x^- which belongs to an opposite class to x. A triplet loss therefore helps the model to learn to output similar descriptors for matching patches and different descriptors for non-matching patches. More recent methods jointly learn a detector and a descriptor. For example, SuperPoint (DeTone et al., 2018) uses a self-supervised FCN which can jointly detect and describe points of interest using supervision between an image and the same image with a small transformation.

While most methods were initially developed for applications on 2D natural images, keypoint detection and description can also be used to process and analyze medical 3D images. Loiseau–Witon et al. (2022) use the 3D SURF detector from Agier et al. (2016) and a keypoint descriptor based on a triplet network to do volume registration of CT images. Regarding the task of medical image segmentation, Bralet et al. (2021) propose a method based on a keypoint transfer algorithm inspired by Wachinger et al. (2018). Specifically, 3D SURF keypoints are extracted from the target volume and compared with keypoints from several reference volumes. Local annotations around each keypoint from these references are then transferred and weighted according to a similarity criteria.

II.3 Point cloud processing

As seen in Section I.3.1, 3D images are represented as 3D regular grids that can be naturally processed using CNNs. However, CNN-based models might not be optimal for point cloud processing as this data type is naturally unordered and scattered. In this section, we explore the state of the art for point cloud processing, sampling and several applications for medical image segmentation.

II.3.1 Methods

While older standard point cloud analysis algorithms focus on building hand-crafted features (Yue Wang et al., 2019), more recent approaches use deep-learning networks in order to extract richer and more informative representations. This paradigm shift enables more effective performance on downstream tasks such as point cloud classification and segmentation.

Projection-based methods Early deep learning-based methods either project the point cloud on multiple 2D views to learn and fuse view-wise features (H. Su et al., 2015), or aggregate the points in a 3D grid that is then processed through a 2D or 3D CNN (Z. Wu et al., 2015). These methods have several drawbacks. First of all, the projection of a high resolution point cloud onto a lower resolution grid may introduce quantization artifacts. Increasing the resolution of the projection grid would not be an optimal solution because CNNs have a high computation cost when processing high resolution volumes as explained in Section II.1. Moreover, using a CNN on a projected point cloud can be inefficient because of the sparse nature of this type of representation: presumably most voxels in the projection grid will be

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

empty. To overcome this problem, some works use sparse convolutions for which convolution is applied only on non-empty voxels (Graham, 2015; Graham et al., 2018). While this improves the computation complexity of the methods, they remain limited in terms of performance. In a more recent work, Zhijian Liu et al. (2019) introduced the PointVoxel CNN model, which is composed of a low-resolution voxel-based branch processed with a CNN and a high-resolution point-based branch processed with a MLP. By leveraging both representations, this method achieves a smaller memory footprint thanks to the sparse nature of point clouds. It also reaches faster processing thanks to the efficient memory access of regular grids. Building on this idea, PointVoxelFormer (M. P. Heinrich, 2024) proposed to stack voxel-based CNN blocks and point-based MLP layers on order to learn per-point features. This work also introduced a differentiable rasterisation process which aims to reduce the blurring effect of aggregating several points' features into the same grid cell.

MLP-based networks Most recent methods attempt to process the point cloud directly. Among them, PointNet (Qi et al., 2016) is a pioneering work whose ideas have been used as building blocks in many other methods. Essentially, PointNet processes the input point cloud through two transform modules which perform affine transformations of the input and the features using MLPs, as shown in Figure 1.12. The output per-point features are then processed through a max-pooling layer which generates a global feature vector. This layer makes the model permutation invariant. The permutation invariance property is crucial for PointNet, and more generally, for the processing of point clouds, as the points are stored and passed into the network without any specific order. The output global feature can be used either to classify the input point cloud, or, if concatenated with the point-wise local features computed in the earlier layers, can be processed by a MLP to output the segmentation label for each point. Despite its good representation capacity, the network fails to extract powerful local features, as it computes per-point features independently.

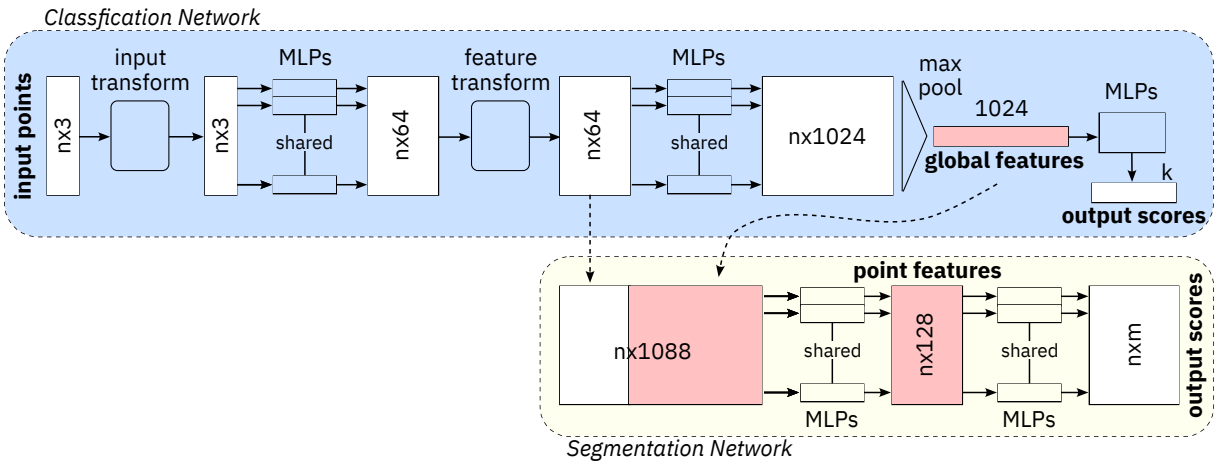


Figure 1.12: *PointNet architecture. n is the number of input points, m the number of segmentation classes and k is the number of classification labels. Figure adapted from Qi et al. (2016).*

PointNet++ (Qi et al., 2017) tries to solve this challenge by applying PointNet on a neighborhood of points. At each stage of the network, the receptive field is increased in order to

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

add more context to the deeper features. More recently, RandLA-Net (Q. Hu et al., 2020) proposes a UNet-like architecture to process large-scale point clouds. The specificity of this work comes from its use of random sampling and a local feature aggregation module in each stage of the encoder. Random sampling is indeed useful for efficiently reducing the size of a large input point cloud while the aggregation module helps retaining key features.

Many models have then been inspired by those works to compute more meaningful point-wise features to perform point cloud classification or segmentation.

Point cloud convolution networks In order to exploit the spatial correlation more efficiently, some methods develop new convolution operators to process the point cloud directly. The kernel used is in general defined in the continuous 3D space instead of on a regular grid. In KPConv (Thomas et al., 2019) the kernel is represented by a set of points, each associated with learnable weights, as shown in Figure 1.13. Unlike grid-based convolution, the neighborhood is defined as all the points within a fixed-radius ball, rather than fixed positions on a regular grid. As a result, there is a varying number of input points which are not aligned with kernel points. To compute the convolution, each input point feature is multiplied by all the kernel weight matrices, with a factor that depends on the point's relative position to the kernel points.

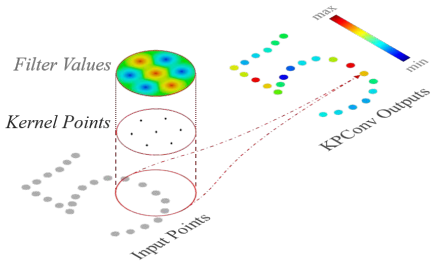


Figure 1.13: *KPConv illustrated on 2D points. Figure taken from Thomas et al. (2019).*

Graph-based networks Some methods view each point in a point cloud as the vertex of a graph and generate edges based on the neighboring points. Their goal is to learn better local features, which was a shortcoming of PointNet-based models. DGCNN (Yue Wang et al., 2019) creates a local graph around each point by determining the point's nearest neighbors. The authors introduce EdgeConv, a convolution-like operation based on a MLP which is used to compute the features for each edge. One particularity of this work is that the graph is computed dynamically: the set of neighbors used to compute the edge features changes after each layer based on the features computed at the previous layer.

Attention-based networks Several works have explored the use of the attention mechanism introduced in Section I.3.2 to learn features from point clouds. Early approaches generally rely on global attention, while more recent methods are based on transformers. The transformer paradigm is well-suited for point cloud processing as the self-attention mechanism is invariant to the permutation of the input elements. Building on this idea, Zhao et al. (2021) design a model based on a point transformer layer that applies self-attention on a neighborhood around

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

each data-point rather than globally. This model, called Point Transformer, also replaces the more common scalar attention mechanism by vector attention for which the attention weights can vary across the different channels. Additionally, it also uses learnable positional encoding. This allows Point Transformers to scale well on large point clouds and therefore to perform segmentation of large scale scenes.

More recent works A recent observation in the field of point cloud analysis is that scaling already existing models can achieve better results than proposing architectural innovations. Following this idea, PointNeXt (Qian et al., 2022) boosts the performances of PointNet++ by improving the training strategies through better data augmentation and model optimization. The minor additions to PointNeXt also allow the authors to scale the model and improve the performances. Similarly, X. Wu et al. (2024) introduce Point Transformer V3 (PTv3), which is an improved version of the original Point Transformer and its successor Point Transformer V2 (X. Wu et al., 2022). This work uses a point cloud serialization method which converts the point cloud to a structured format. This leads to the simplification of the network, notably through the replacement of the k-nearest neighbors algorithm by a more efficient method using the serialized representation. The authors argue that this simplification enables to work with larger models and to ultimately achieve better performances.

II.3.2 Point cloud sampling

Most point cloud processing methods rely on sampling the point cloud in order to reduce its size. This pre-processing step is generally necessary because of computation constraints. Indeed the size of synthetic and real-world point-cloud data can range from thousands to millions of points.

Among the standard sampling methods, we can mention the following algorithms:

- *Random sampling* is the simplest approach and consists in uniformly selecting points from the original point cloud. The complexity of this method does not depend on the number of input points which makes it faster than other methods for the processing of large point clouds (Q. Hu et al., 2020);
- *Farthest point sampling* is an other popular method which selects iteratively the point farthest from the current set of points. While it leads to a good coverage of the original point cloud (Qi et al., 2017), its high complexity makes it difficult to use for large-scale sets of points;
- *Grid subsampling* can help when dealing with point clouds with unbalanced density (Thomas et al., 2019). This method first projects the points into a 3D grid and selects one (or several) point per cell. One disadvantage of this method is that it generates point clouds with varying sizes;
- *Inverse density importance sampling* selects points randomly using weights proportional to the inverse local density (Groh et al., 2018). Therefore, the more dense an area, the less points are sampled in this region. While this method generates a good coverage of the point cloud, it is sensitive to outliers.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

These heuristic methods are simple but may not be optimal for a particular downstream task, which is why some works introduce deep learning models to learn a better sampling strategy. For example Dovrat et al. (2019) is a pioneering work which introduces S-Net. This is a PointNet-based network that can generate a new set of points which is the optimal input for a pre-trained task network. The model is optimized using a combination of two loss terms. First the task loss is used to generate the optimal subset of points for the target task. Note that the weights of the task model are frozen during the training of S-Net. A sampling regularization loss is also used in order to bring the generated points closer to the original points. Despite this second loss term, the generated point cloud is not guaranteed to be a subset of the original point cloud. Therefore, nearest neighbor matching is used at inference to match each new point to its closest point in the original point cloud. A following work by Lang et al. (2020) uses the same idea but makes the matching step differentiable so that is can be used at training time to optimize the network. Their model, called SampleNet, uses a soft projection method that replaces each point in the simplified point set by a weighted average of its nearest neighbors in the original point cloud during training. These methods are sometimes called generator-based because they generate a new set of points from the input point cloud.

Alternatively, score-based methods select points in the original point cloud based on their importance for the downstream task. The Critical Point Layer (CPL) proposed by Nezhadarya et al. (2020) learns to identify and keep the most important points from an input point cloud. The CPL layer processes features computed in previous layers and keeps points with the maximum feature values. This layer can be easily used in a point processing network based on graphs and convolutions. In this approach, the sampling is optimized at the same time as the task network. More recently, APES (C. Wu et al., 2023) uses an attention mechanism to select edge points in the input point cloud. The authors define edge points as points that indicate the outline of the object. More specifically, points are selected as edge points if their local and global attention maps have a higher standard deviation.

A main drawback of learning-based point cloud sampling methods is that they are often only evaluated on small point clouds ($N = 1024$) and might not scale well on a large number of points.

II.3.3 Applications in medical imaging

Numerous studies have focused on the use of point clouds for medical image analysis: in this work, we focus mostly on their application for the segmentation task. Some imaging modalities generate a point cloud directly which makes them suitable for a processing using point-based models. For instance, IOS directly generates a point cloud mapping of the surface of the teeth and gum: segmenting those structures can serve as a useful tool in dentistry (Zanjani et al., 2019). However, we decide to focus on the methods and models whose final target is volume segmentation, whether CT or MRI, as it aligns more closely with the objective of this thesis.

The motivation for using point clouds as a tool for volumetric segmentation stems from the limitations of conventional CNN-based segmentation methods, which often process input volumes in patches or slices due to memory constraints. This patch-wise processing can pre-

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

vent the model from capturing the global structure of the target object. Using a point-based network that can process all target structures simultaneously can be an promising alternative.

Most methods that use point cloud representations for volumetric segmentation of medical images follow a standard workflow. Generally, they start by extracting a point cloud from a coarse segmentation map that was previously generated by a CNN. The extracted point cloud serves as an input to a point cloud segmentation model which works as a refinement module. Point-Unet (Ho et al., 2021) applies this workflow for brain tumor segmentation. From a CNN-generated attention map, the authors propose a context-aware sampling: the point cloud is extracted through sampling of the attention map with the idea that the higher the attention score in a region, the more densely this region must be sampled. The point cloud is then processed by RandLA-Net (Q. Hu et al., 2020) which outputs segmentation labels for each of the generated points.

Application-wise, using a point cloud representation can be especially useful when the target object has a tubular structure like nerves, blood vessels or pulmonary airways. The connectivity of these types of structure is an important characteristic which can be learned by graph-based point processing models. A work which is typical of this idea is the workflow developed by J. He et al. (2020), which learns hybrid representations for 3D vessel centerline extraction. First, a CNN is used on the original image to generate a vessel mask, from which a point cloud is extracted. This point cloud is then processed through a point-based network which outputs labels corresponding to the vessel segments. This segmentation step is necessary for the final post-processing step which connects the different segments before performing centerline extraction. The first CNN-based step is used to learn the local appearance of a vessel while the second point-based step benefits from working on the global geometry. Best performance is reached when the point processing model used is the graph-convolution network DGCNN (Yue Wang et al., 2019).

In the literature, the point cloud extracted from the pre-segmented volume usually requires one point per voxel in the regions of interest. When working with large segmented structures which is the case for multi-organ segmentation, this can lead to very large point clouds and therefore to high computation costs. One possible solution would be to sample the point cloud, but this raises the question of how to interpolate the segmentation map in the voxels that are not sampled.

II.4 Implicit Neural Representations

Generating segmentation maps from a segmented point cloud requires to extract surfaces from this point cloud, which a is non-trivial issue. While methods introduced in previous sections do not specifically address this challenge, surface reconstruction from point clouds is an active area of research in itself. In particular, methods based on implicit neural representations have shown promising results for surface reconstruction as well as other 3D computer vision tasks.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

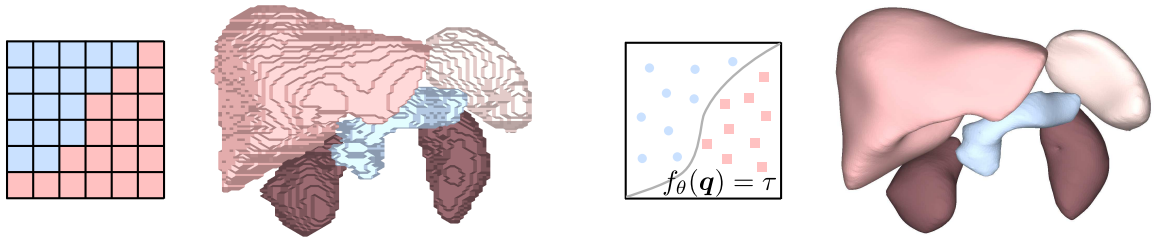
II.4.1 Definition

The methods discussed in the previous sections usually rely on discrete *explicit representations*. Voxel grids, 2D views, point clouds and meshes are some examples of explicit representations commonly used for 3D objects. An alternative approach, called *Implicit Neural Representations (INR)*, has become increasingly popular and consists in representing a signal (for example a 2D image) with a neural network. More specifically, a MLP is used to learn a continuous mapping f_θ between a set of coordinates and a target variable. For example in the case of a 2D image, the target variable may be the image's intensity values.

One interesting property of INRs is their compactness: a signal that was originally explicitly stored with an array of values can be represented in INRs with a smaller number of parameters. For example in the case of a 2D image, instead of storing the intensity values of the $N \times M$ pixels, it is possible to query the intensity value of any pixel with coordinates (x, y) by computing $f_\theta(x, y)$. This is especially useful for high-dimensional data such as 3D objects. INRs can be used for various applications such as shape reconstruction or image generation, which we will describe in this section.

II.4.2 3D shape reconstruction with INR

Explicit representations of a 3D shape described in Section II.1 consist in storing the surface of an object with an array of voxels for the volumetric representation or with a list of polygons for the mesh representation. In contrast, the surface of this object can be extracted with INRs by computing the value of a learned MLP for a set of coordinates in the 3D space, as illustrated in Figure 1.14.



(a) *Volumetric representation* **(b)** *Implicit representation*

Figure 1.14: *Representation of a 3D shape with a voxel grid and with an implicit function f_θ. The target variable τ represents here the occupancy value in the 3D space. Figure adapted from Kuang et al. (2022)*

Generally INRs learned for shape reconstruction do not represent one shape per model but rather are optimized to represent a family of shapes. This is possible because they take as input not only a set of 3D coordinates, called *query points*, but also a vector representing an observation of the object to reconstruct. This vector is usually a latent code which conditions the model to a reconstruct a specific shape.

In this section, we describe two families of INR-based architectures commonly used for 3D shape reconstruction.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

DeepSDF The *Signed Distance Function (SDF)* is a commonly used target value learned by INRs when doing surface reconstruction. SDF-based networks map query points to their distance from the nearest surface. The SDF is zero for surface points, negative for points inside the surface and positive for points outside the surface. A key work in this area is DeepSDF (Park et al., 2019) which uses an auto-decoder architecture to reconstruct the surface of an object based on sampled surface points. Here the surface points, that we also call input points, are the observation used to condition the network. Architecture-wise, contrary to an auto-encoder which decodes a latent code learned from an encoder, an auto-decoder takes directly a latent code as an input, as illustrated in Figure 1.15. At inference, the randomly initialized latent code is optimized through the evaluation of the network on the known surface input points. A surface mesh is then reconstructed using the marching cubes algorithm, a standard computer vision approach that can reconstruct a surface from a 3D scalar field (Lorensen et al., 1998). Inspired by DeepSDF, Chabra et al. (2020) propose DeepLS and divide the space into a regular grid to learn one latent code per grid cell. This makes DeepLS better at reconstructing finer details than the original work.

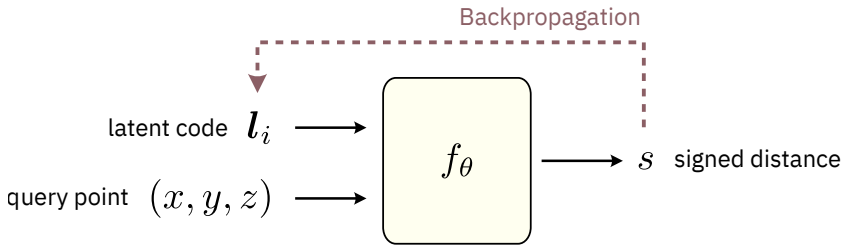


Figure 1.15: *DeepSDF overview. During training, the weights of f_θ and the latent code $\boldsymbol{l}_i$ are optimized. At inference time, f_θ is frozen and $\boldsymbol{l}_i$ is randomly initialized. First, $\boldsymbol{l}_i$ is optimized for known (surface, $s = 0$) query points. Then the signed distance of unknown query points can be computed. Figure adapted from Park et al. (2019).*

Occupancy Networks An alternative approach for surface reconstruction relies on the occupancy function. This function maps any input point p to the occupancy value at its coordinate: this values is equal to 1 when p is inside the object and 0 when p is outside. In practice occupancy-based networks learn a function which outputs the occupancy probability on which a threshold is applied to extract the surface.

Mescheder et al. (2019) is the first work to learn a continuous occupancy function with their Occupancy Network (ONet). In this approach, an encoder generates a latent code from the input data using an architecture adapted to the input type. For example, a lightweight PointNet (Qi et al., 2016) is used for point clouds and a CNN for volumetric data. The latent code is then processed by a decoder as shown in Figure 1.16. This decoder is made up of several fully connected ResNet blocks and is conditioned on the input using conditional batch normalization (CBN). This decoder can estimate the occupancy probability for any 3D query point in space. To make a prediction, the decoder takes both the latent code, which is used to condition the model, and the coordinates of the query point. The surface reconstruction from the input data requires to choose the query points effectively. To do so, the authors introduce the Multiresolution IsoSurface Extraction (MISE) algorithm. MISE begins by generating a low

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

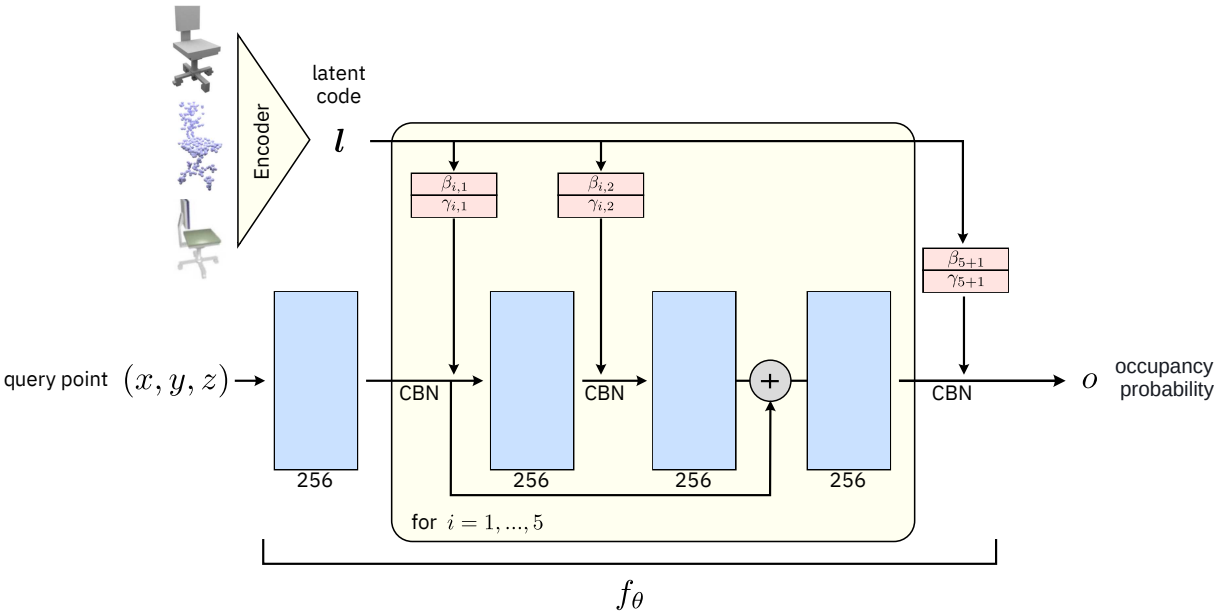
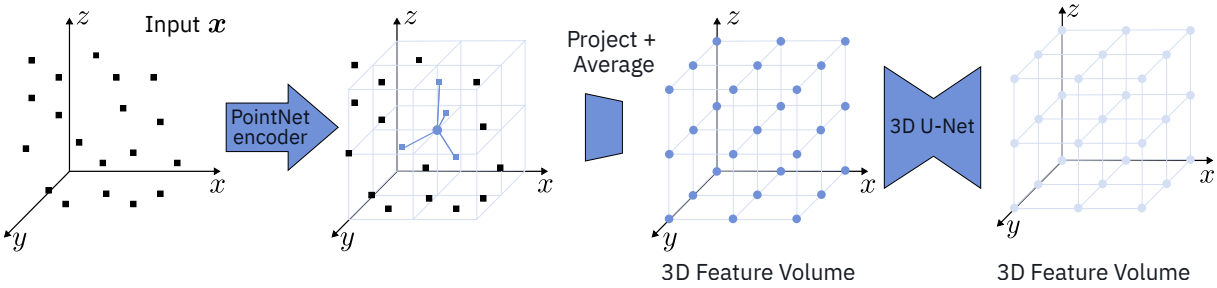


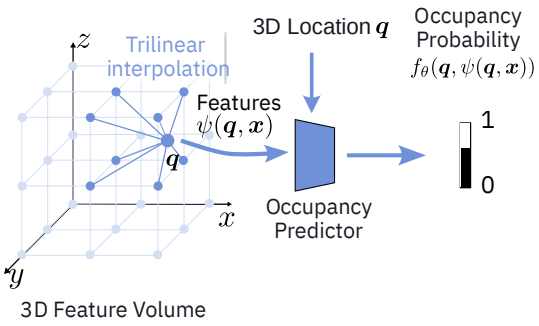
Figure 1.16: *Occupancy Network Architecture. Figure adapted from Mescheder et al. (2019).*

resolution mesh which is then refined by querying more points close to the surface. After selection and occupancy prediction of the query points with the MISE algorithm, the surface mesh is extracted using the marching cubes algorithm. A notable advantage of ONet is that its decoder is lightweight and memory efficient.

The works that build on ONet generally focus on improving the encoder so that it generates a more meaningful latent code. For example, Convolutional Occupancy Networks (ConvONet)



(a) *ConvONet Encoder*



(b) *ConvONet Decoder*

Figure 1.17: *Overview of Convolutional Occupancy Networks. The configuration used is a volume encoder and a volume decoder. Figure adapted from Peng et al. (2020).*

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

(Peng et al., 2020) produce more complex shapes by learning a latent plane or a latent grid instead of the original latent vector. The proposed encoder first computes geometric features by processing the input point cloud through a PointNet module. The enriched point cloud is then projected or aggregated onto a 2D or 3D grid and processed by a small UNet to generate the latent feature grid. The latent code associated to a query point $\boldsymbol{q}$ is then generated by interpolating the latent grid at the location of $\boldsymbol{q}$'. This latent code is then processed by the ONet decoder to output the occupancy probability for $\boldsymbol{q}$. Figure 1.17 illustrates the workflow for the 3D version. More recently, T. Wang et al. (2024) uses sparse convolution in ConvONet to ignore the empty voxels. While this reduces the memory cost of the 3D UNet, it also results in a five-fold increase of the inference time.

II.4.3 Neural Radiance Fields for view synthesis

With the Neural Radiance Fields (NeRF) proposed by Mildenhall et al. (2021), an INR is also used to represent a 3D object. However the originality of this work lies in the fact that NeRF are used to synthesize novel 2D views of this object.

The method for NeRF is illustrated in Figure 1.18. The authors consider a virtual camera that captures a 2D view of the object. A ray is traced from each pixel of the camera view and passes through the object with a certain direction. Several 3D points are sampled along this ray. The network takes the coordinates of these points, as well as the direction of the ray as input. For each point sampled, the network outputs its RGB color and its density. The density value is high when the point belongs to an object, and low when the point is sampled in an empty area. The network used is a MLP that learns to map some 3D coordinates (x, y, z) and 2D viewing direction (θ, ϕ) to a density σ and RGB value: $f_\theta : (x, y, z, \theta, \phi) \rightarrow (RGB, \sigma)$. Classical volume rendering techniques are then used to accumulate the colors and densities predicted along the ray into the 2D image. Without going in too much detail, the pixel color is computed with a weighted sum of the color of the points sampled along the ray, where the weight depends on the density of the points. The network and the volume rendering process are both differentiable so this approach can be optimized by using a rendering loss. At inference, a new 2D view can be queried from any angle.

First experiments on NeRF show that this INR model has poor representation capacity for

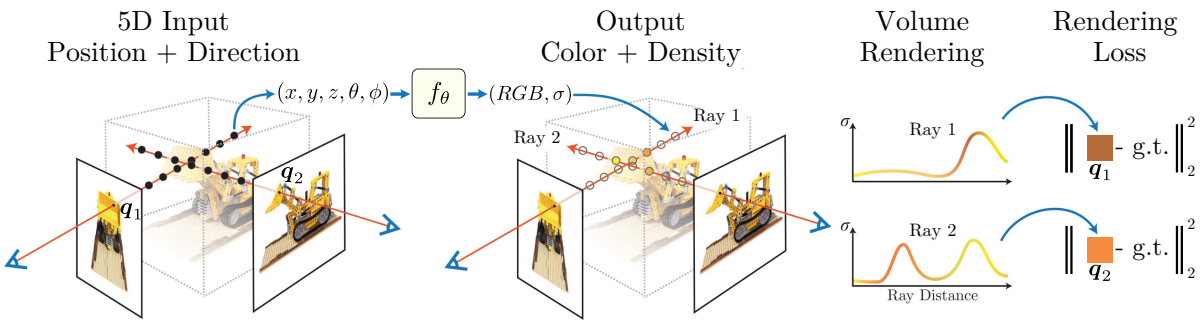


Figure 1.18: *NeRF workflow. $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ are two pixels from two 2D views. RGB values for these pixels are computed by aggregating the RGB values of the points sampled along the rays traced from the viewing camera. The $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ RGB values are used to optimize the network. Figure adapted from Mildenhall et al. (2021).*

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

high frequency details. The authors therefore add a positional encoding which maps the spatial coordinates to a higher dimensional space before feeding it to the network.

II.4.4 Learning high frequencies

One main limitation of INR is that they tend to give very smooth output. A previous work by Rahaman et al. (2019) has indeed shown how ReLU-based MLPs tend to favor low frequencies during learning. To overcome this problem, some works proposed to replace the activation functions and to improve the input through positional encoding.

First, Sitzmann et al. (2020) show that higher frequencies can be learned when replacing the activation functions traditionally used in neural networks such as ReLU by a periodic activation function. The authors use the sine function as an activation function and achieve a better performance on high frequencies across different tasks.

Positional encoding, that we briefly introduced for NeRFs in Section II.4.3, is also used to capture better representations. In NeRF, the positional encoding consists in applying the γ function to all the input coordinates p:

$$\gamma(p) = (sin(2^0\pi p), cos(2^0\pi p), ..., sin(2^{L-1}\pi p), cos(2^{L-1}\pi p))$$

This mapping from $\mathbb{R}$ to $\mathbb{R}^{2L}$ helps learning high frequency details.

II.4.5 Applications in medical imaging

INRs can be especially useful for medical image reconstruction thanks to the capacity of the models to learn and output intensity or occupancy values at arbitrary resolutions. This is especially useful when dealing with sparse or under-sampled data which is a recurrent challenge in medical image datasets. It is indeed common to generate CT or MRI images at a low resolution in order to reduce the imaging time and therefore to limit patients' movements or their exposure to radiations. Models based on INRs can overcome this issue as they can reconstruct high-quality volumes from low-resolution and noisy images. L. Shen et al. (2022) for example introduce NeRP which can reconstruct high resolution CT and MRI images by training an INR on sparsely sampled measurements in the sensor domain (frequency domain for MRI images and projection domain for CT images). The authors also show that initializing the network through training on an image from the same patient can improve the reconstruction capacity. The resolution-independence property of INRs is not only useful for reconstructing images but also 3D shapes and in particular shapes extracted from segmentation maps. For example Amiranashvili et al. (2022) reconstruct the shape of vertebrae at a high resolution by learning on sparse segmentation maps. To do so, they use a network based on an auto-decoder architecture inspired by DeepSDF (Park et al., 2019).

INR-based networks can also be useful for volumetric segmentation. The proposed methods are usually inspired by the 3D shape reconstruction models presented in Section II.4.2. Occupancy networks are a popular baseline for this task as they can be easily adapted to condition the INR model on a volume input. In particular, most approaches focus on generating

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

the best latent code by using a CNN encoder. One typical example of this workflow is OSS-Net (Reich et al., 2021), where a 3D CNN encoder generates a global latent vector to condition the INR decoder. Compared to ONet, OSS-Net also computes a local latent vector around each query point. This vector is concatenated to the query point's coordinates in order to add local intensity information for the computation of the point's segmentation label. A similar work is ImPulSe (Kuang et al., 2022) which aims to generate the segmentation map of pulmonary segments. ImPulSe computes latent grids at multiple resolutions, as shown in Figure 1.19. For each query point, its latent code is computed by concatenating its coordinates with the feature extracted from the latent grid with tri-linear interpolation. Other approaches use a transformer-based encoder in order to process a medical image and extract meaningful features that can be used to condition an INR network. I-MedSAM (Wei et al., 2024) uses a SAM-based encoder and an INR decoder to perform 2D medical image segmentation. More precisely, it uses a pre-trained SAM encoder to output a latent vector: PEFT with LoRA modules (see Section I.4.3 for the definition of these terms) and a frequency adapter are applied to fine-tune the encoder more efficiently. A prompt encoder generates a second latent vector from a bounding box prompt. The final latent vector is fed to the decoder to reconstruct the segmentation map. This latent vector is a concatenation of the image vector, the prompt vector and the coordinates of a query point with positional encoding. Contrary to the previously mentioned approaches, I-MedSAM introduces a novel INR-based decoder which is composed of two INR modules: a first network outputs a coarse segmentation map which is then refined on sample points with higher uncertainty by a refinement INR network. The authors also show that I-MedSAM is more robust than discrete baseline for data shifts in resolution or datasets.

The main limitation of INR-based segmentation networks in terms of efficiency is their encoder. Indeed, despite their use of a light INR MLP decoder, using a CNN or a transformer encoder to process a high resolution 3D volume still encounters the high memory or high computation complexity challenges mentioned in Section I.3.

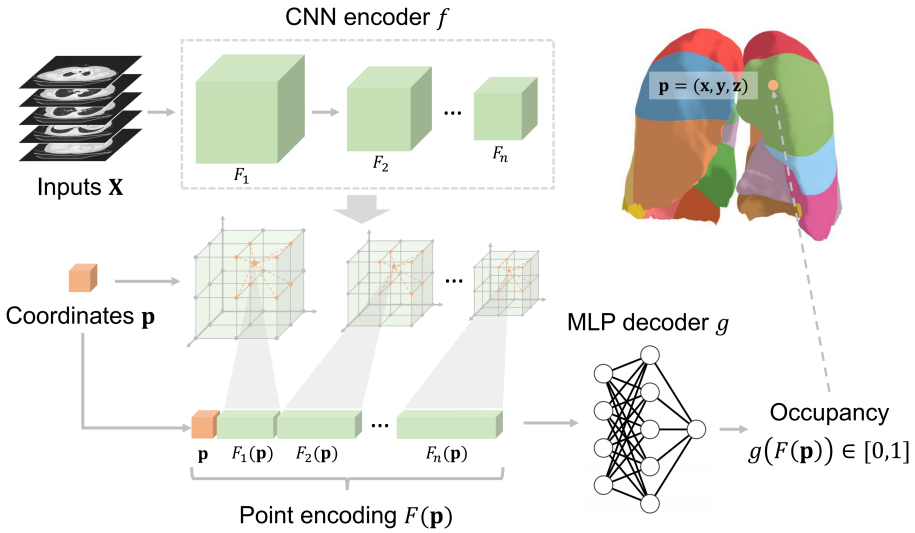


Figure 1.19: *Architecture of ImPulSe. Figure taken from Kuang et al. (2022).*

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

III Conclusion

The current trends for medical image segmentation are highly based on deep learning models and more precisely on models with CNN and transformer architectures. One important limitation of the commonly used methods for this task is their computation cost which is often high due to the processing of large 3D volumes containing thousands of voxels. Among the lightweight approaches introduced to address this issue, some authors focus on using reduced data representations in order to decrease memory usage and computation complexity. It is for example possible to replace the dense voxel grid representation by a point cloud which is a lighter sparse alternative. Although point-based medical volume segmentation often relies on point cloud segmentation models, the generation of a volumetric segmentation map from a segmented point cloud is not straightforward. The use of implicit neural representations can be a solution to tackle this problem thanks to their ability to represent and reconstruct shapes in the continuous 3D space with a lightweight MLP decoder.

Therefore, the following chapters take advantage of both point clouds and implicit representations in order to perform segmentation of large medical volumes.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Chapter 2:

SCONet: convolutional occupancy networks for multi-organ segmentation

Contents

I Method . . . 40
I.1 Point cloud extraction . . . 40
I.2 SCONet architecture . . . 41
I.2.1 Encoder . . . 41
I.2.2 Decoder . . . 42
I.3 Training and inference . . . 43
I.3.1 Query points and optimization . . . 43
I.3.2 Data augmentation . . . 43
II Experiments . . . 43
II.1 Datasets . . . 43
II.1.1 AbdomenCT-1K . . . 44
II.1.2 AMOS . . . 44
II.2 Implementation . . . 45
II.3 Comparison baselines . . . 45
III Results . . . 45
III.1 Segmentation quality . . . 45
III.2 Efficiency . . . 49
III.2.1 Memory footprint . . . 49
III.2.2 Time complexity . . . 49
III.2.3 Network optimization . . . 50
III.2.4 Efficient query point selection with the MISE algorithm . . . 51
IV Artifact reduction through training on random crops . . . 53
IV.1 Motivation . . . 53
IV.2 Method . . . 54
IV.3 Results . . . 54
IV.4 Conclusion . . . 56
V Ablation studies . . . 57
V.1 SURF descriptors . . . 57
V.2 Data augmentation . . . 57
V.3 PointNet module . . . 58
VI Conclusion . . . 59

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

This chapter presents our segmentation workflow which is the main contribution of this thesis. Its specificity compared to those proposed in the literature is that it takes advantage of both point cloud and implicit representations in order to perform medical volume segmentation.

The workflow was first presented at ISBI 2025 (Jouvencel et al., 2025a). This chapter also includes several additional studies.

I Method

This section describes our workflow which first extracts a point cloud from the original volume, and then feeds it to SCONet, an occupancy-based network capable to reconstruct the segmentation map. Figure 2.1 illustrates this process. The implementation of this method is available in a public repository[1].

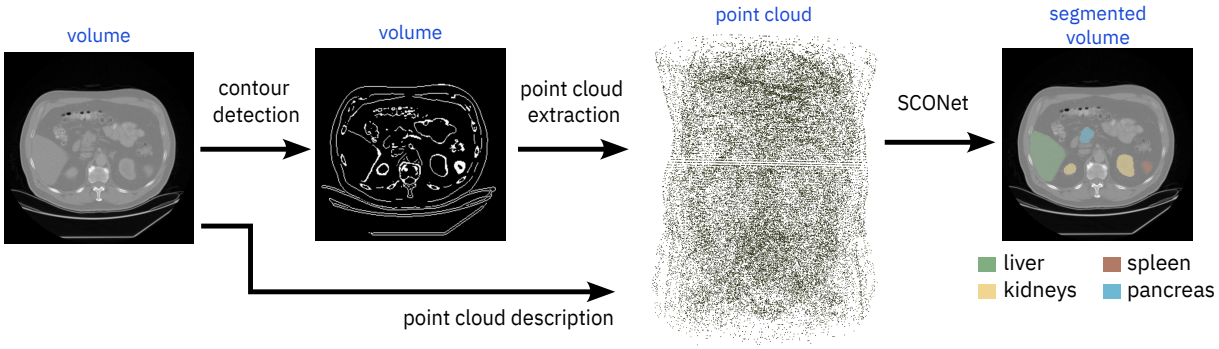


Figure 2.1: *Full segmentation workflow for 3D images.*

I.1 Point cloud extraction

We choose to work on contour points (i.e. points that correspond to a variation in intensity in the image). We indeed assume that they can bring useful information for the surface reconstruction performed in the second stage of the workflow by the occupancy-based network. Therefore, the first step of our workflow consists in a contour point cloud extraction. We use the Canny edge detection algorithm (Canny, 1986) that we introduced in the previous chapter (see Chapter 1, Section II.2.1). This fast standard algorithm has three tunable parameters that are the scale of the low-pass filter used for noise removal and the two thresholds used to reduce the number of false edge points. In our workflow, we choose the values for these parameters empirically to visually ensure that the generated edge maps contain a sufficient number of contour points located near the target organs' surfaces while minimizing the noise level. These parameters are fixed for a given dataset and a given modality. The points are then extracted from the edge volume to form the contour point cloud P.

The point cloud P extracted from the edge map has a large number of points ($N \sim 10^6$). Therefore it is necessary to pre-process P with sampling. We choose to use random sampling for efficiency reasons (Q. Hu et al., 2020).

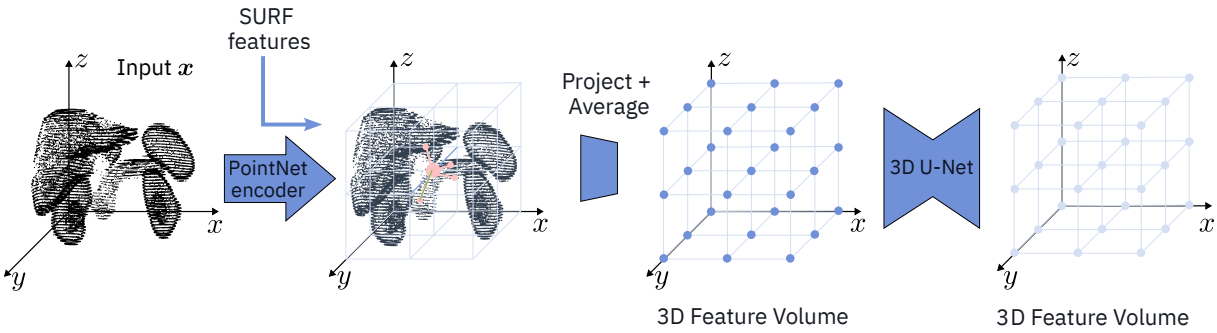
[1] https://github.com/maylis-j/SCONet

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

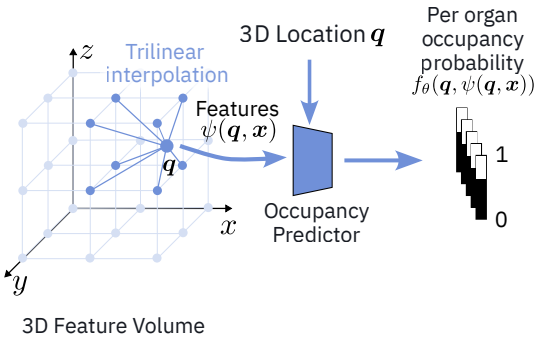
For each point i of the point cloud, in addition to its coordinates (x_i, y_i, z_i), we compute the corresponding SURF descriptors (Bay et al., 2006) to encode local information in a 10^3-voxel neighborhood around the point. As a reminder, Chapter 1, Section II.2.2 gives more details about the description algorithm. This descriptor is computed with the 3D extension (Agier et al., 2016) of the original 2D algorithm and yields a vector with 48 values $(s_i^1, ..., s_i^{48})$ for each point. The resulting point cloud is a $(N \times 51)$ matrix which encodes geometric and local photometric features, where N is the number of points in P. We translate and scale the (x_i, y_i, z_i) coordinates to fit the unit cube. Coordinates located outside the unit cube are clipped to its boundaries. The scaling coefficient used to fit the points to the unit cube is fixed for all patients in a given dataset.

I.2 SCONet architecture

To process this point cloud, we introduce SCONet, a ConvONet-based network adapted from Peng et al. (2020) to the specific task of multi-organ segmentation. The proposed network has the following encoder-decoder architecture.



(a) *SCONet Encoder*



(b) *SCONet Decoder*

Figure 2.2: *SCONet workflow overview. For clarity purposes, the input point cloud displayed here shows only the surface points from each target organ.*

I.2.1 Encoder

The main goal of the encoder is to compute a feature grid from the contour-based input point cloud P. As shown in Figure 2.2a, the coordinates of P are first fed to a lightweight PointNet encoder (Qi et al., 2017) which consists of 5 fully-connected ResNet blocks with max-pooling

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

layers. The max-pooling operation is applied locally in order to learn local features. More precisely, max-pooling is done among points that will be aggregated in the same grid cell in the following steps. The detailed architecture of the PointNet block is shown in Figure 2.3. This module computes per-point geometric features with dimension 32 which are then concatenated to the SURF descriptors. This concatenation of the SURF descriptors is a novelty compared to the original ConvONet method. The PointNet and SURF features are then aggregated on a grid with size r^3 by applying average pooling on each grid cell, resulting in a feature grid with dimension $(r^3 \times 80)$. Finally, this grid is processed by a 3D U-Net (see Chapter 1, Section I.3.1 for a description of the UNet model) with three levels which outputs a learned feature grid with dimension $(r^3 \times 32)$.

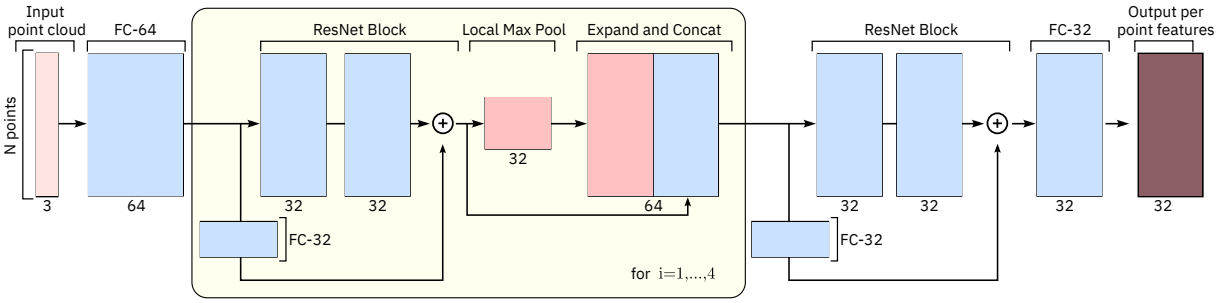


Figure 2.3: *SCONet detailed PointNet architecture*

I.2.2 Decoder

The main role of the decoder is to use the learned feature grid to predict the occupancy probabilities of a query point $\boldsymbol{q}$. As shown in the Figure 2.2b, the query point's latent vector is computed through trilinear interpolation on the learned latent grid at the corresponding coordinates. This feature vector is then fed to a ResNet-based Occupancy Network (ONet) (Niemeyer et al., 2020) followed by a linear layer which outputs logits for all the classes. A final Softmax layer then outputs the query point's occupancy probabilities for each organ. The detailed architecture of the ONet decoder can be found in Figure 2.4.

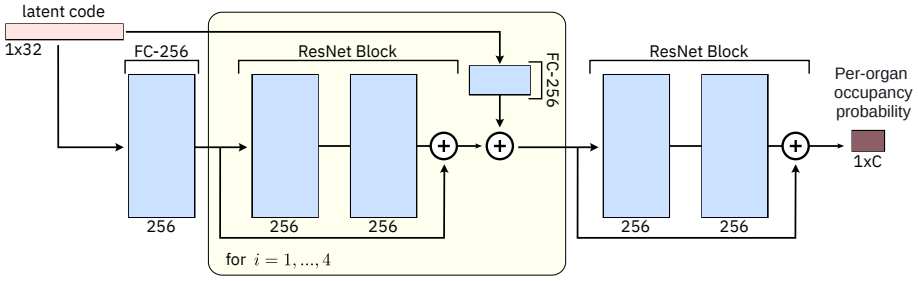


Figure 2.4: *SCONet decoder detailed architecture. The latent vector generated by interpolation of the latent feature grid at the query point's location is fed to the decoder, which outputs the per-organ occupancy probability.*

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

I.3 Training and inference

I.3.1 Query points and optimization

At training time we optimize the network by predicting the occupancy values of N_q random query points. The loss function used is the combination of the Dice loss (Milletari et al., 2016) and the cross-entropy loss. The Dice loss is a continuous and derivable relaxation of the discrete Dice metric introduced in Chapter 1, Section I.5. The full loss is given in the following equation:

$$\mathcal{L} = \mathcal{L}_{Dice} + \mathcal{L}_{CE} = (1 - \frac{2\sum_q^{N_q} y_q \hat{y}_q}{\sum_q^{N_q} y_q^2 + \sum_q^{N_q} \hat{y}_q^2}) + (-\sum_q^{N_q}\sum_c^{C} y_{c,q} log(\hat{y}_{c,q})) \tag{2.1}$$

In Equation 2.1 y is the reference label, $\hat{y}$ is the predicted label probability and C is the number of classes.

During inference, we predict the segmentation map for the entire volume by querying points corresponding to the coordinates of all the voxels in the original volume. An additional study in Section III.2.4 explores a different strategy based on the MISE algorithm to select a reduced number of query points.

I.3.2 Data augmentation

We apply a data augmentation strategy based on the *random linear transform*, which is a novelty compared to the original ConvONet work. This transform is applied on the point's coordinates only and is defined by the following equation:

$$\boldsymbol{x_t} = (I + \lambda M)\boldsymbol{x} + \lambda \boldsymbol{t} \tag{2.2}$$

In Equation 2.2, $\boldsymbol{x_t}$ corresponds to the transformed coordinates of the input point $\boldsymbol{x}$. $I \in \mathbb{R}^{3\times3}$ is the identity matrix, $M \in \mathbb{R}^{3\times3}$ is a matrix with coefficients randomly sampled on a normal distribution and $\boldsymbol{t} \in \mathbb{R}^3$ is a translation vector with coefficients randomly sampled on a normal distribution. We set $\lambda = 0.1$ in our experiments.

The same transform is applied on the input point cloud and the query point cloud. Each random linear transform can be described as a combination of several affine transformations which have different effects: translation, scaling, rotation and shear. Figure 2.5 illustrates the effect of this data augmentation on a 2D point cloud. An ablation study justifying the use of this data augmentation strategy can be found in Section V.2.

II Experiments

II.1 Datasets

We evaluate our method on two abdominal CT datasets that provide annotations for abdominal organs.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

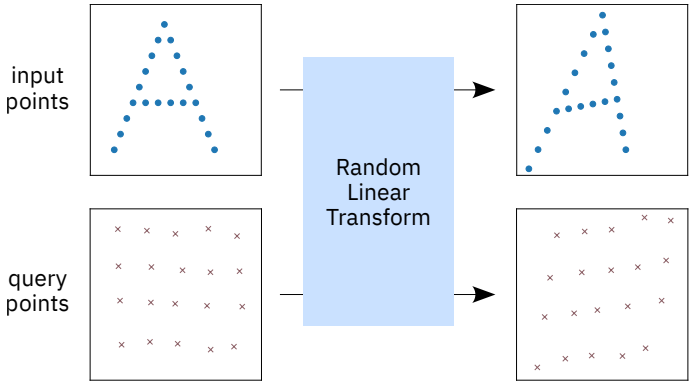


Figure 2.5: *Random linear transform on input and query points. This illustration is given for a 2D point cloud.*

II.1.1 AbdomenCT-1K

The AbdomenCT-1K dataset (J. Ma et al., 2022b) consists of 1000 abdominal CT scans from 12 medical centers, with reference annotations for four abdominal organs: liver (LIV), kidneys (KID), spleen (SPL) and pancreas (PAN). The size of the volumes is $(512 \times 512 \times z)$ with z that varies between 31 and 1026 depending on the patient. The spacing (size of the voxels, given in millimeters) varies between 0.52mm and 1.04mm for the x and y axes and between 0.45mm and 8mm for the z axis.

For our experiments, we resample the original images along the x and y axes to obtain volumes of size $(256 \times 256 \times z)$. This pre-processing helps achieving reasonable training times for voxel-based comparison baselines despite the large dataset and image sizes. As the spacing along the z axis can be much larger than those of the two other axes, we keep the original spacing along this axis in order to avoid possible deterioration of the results.

We use a custom data split train/val/test with 806/91/100 images.

II.1.2 AMOS

The AMOS dataset (Ji et al., 2022), as introduced for the Multi-Modality Abdominal Multi-Organ Segmentation Challenge 2022, consists of 500 CT scans and 100 MRI scans. The challenge includes two tasks with (1) the segmentation of abdominal organs on CT only and (2) the segmentation of abdominal organs on CT and MRI. In this section we focus on the first task only. The authors published annotations of 15 organs for 300 CT scans. The annotated organs are: spleen (SPL), right kidney (RKI), left kidney (LKI), gallbladder (GBL), esophagus (ESO), liver (LIV), stomach (STO), aorta (AOR), inferior vena cava (IVC), pancreas (PAN), right adrenal gland (RAG), left adrenal gland (LAG), duodenum (DUO), bladder (BLA) and prostate/uterus (PRO/UTE).

The size of the volumes varies between 512 and 768 for the x and y axes and between 68 and 353 for the z axis. The spacing varies between 0.45mm and 1.07mm for the x and y axes and between 1.25mm and 5mm for the z axis. We resample the original images along the x and y axes and obtain volumes of size $(256 \times 256 \times z)$ to achieve reasonable training times for voxel-based comparison baselines.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

The dataset provides an official data split with 200 training CT volumes and 100 testing CT volumes. We split the training set into training and validation to get a final data split with 180/20/100 volumes.

II.2 Implementation

The extraction of the point cloud and computation of the SURF features is done before training, as a pre-processing step.

We implement SCONet with PyTorch (Paszke et al., 2019), a Python package which provides tools to design and run deep learning models. Hardware-wise, we use Jean-Zay supercomputer resources[2] and perform all experiments on a NVIDIA Tesla V100-SXM2-32GB GPU. We train the network during 300 epochs with the AdamW optimizer and a learning rate of $l_r = 10^{-4}$. The mini-batch size is set to one input point cloud. During training, the number of query points is set to $N_q = 50k$ for training and $N_q = 400k$ for validation. We also set the size of the input point cloud to $N = 50k$ for training and inference.

II.3 Comparison baselines

To evaluate the advantages of working with a lighter point cloud representation and an implicit network, we compare our performances with both discrete and implicit methods.

We choose 2D and 3D U-Net (Ronneberger et al., 2015; Çiçek et al., 2016) as CNN baselines because U-Net remains the backbone for many state-of-the-art methods. For both networks, we implement a four-level architecture that we train with the Dice loss during 50 epochs. The mini-batch sizes used are 64 and 2 for 2D U-Net and 3D U-Net, respectively. Moreover we train the 3D U-Net on image patches with size (256x256x30). Our 2D and 3D U-Nets are based on the PyTorch implementation from Wolny et al. (2020). SwinUNETR-V2 (Y. He et al., 2023) serves as a transformer baseline, and is trained for 100 epochs with the weighted combination of cross-entropy loss and Dice loss.

We also compare our network with two implicit baselines that use a CNN encoder and ONet decoder. Those baselines are ImPulSe (Kuang et al., 2022) that we train for 50 epochs with the weighted combination of cross-entropy loss and Dice loss from the original paper, and OSS-Net (Reich et al., 2021), that we train for 100 epochs with a cross-entropy loss.

III Results

III.1 Segmentation quality

The first criterion to evaluate our method is the quality of the predicted segmentation maps. A visualization in 2D and 3D of the segmentation algorithms for one sample of AbdomenCT-1K is presented in Figure 2.6. These qualitative results show that SCONet can produce coherent segmentation maps, achieving overall good performance over the four abdominal organs. Another observation is that our method as well as ImPulSe produce segmentation maps that

[2] http://www.idris.fr/eng/jean-zay/jean-zay-presentation-eng.html

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

are noticeably less noisy than those of the three discrete baselines. This smoothness property is a noteworthy advantage of SCONet even though noise can often be reduced through post-processing. However, smoothness also has a disadvantage as our method sometimes struggles to capture fine details. This occasionally results in imprecise and overly smoothed boundaries, in particular around the kidneys. A final observation is that OSS-Net fails to produce meaningful segmentation maps: there is noise near the predicted surfaces and the organs close to one another are sometimes fused like the liver and the right kidney .

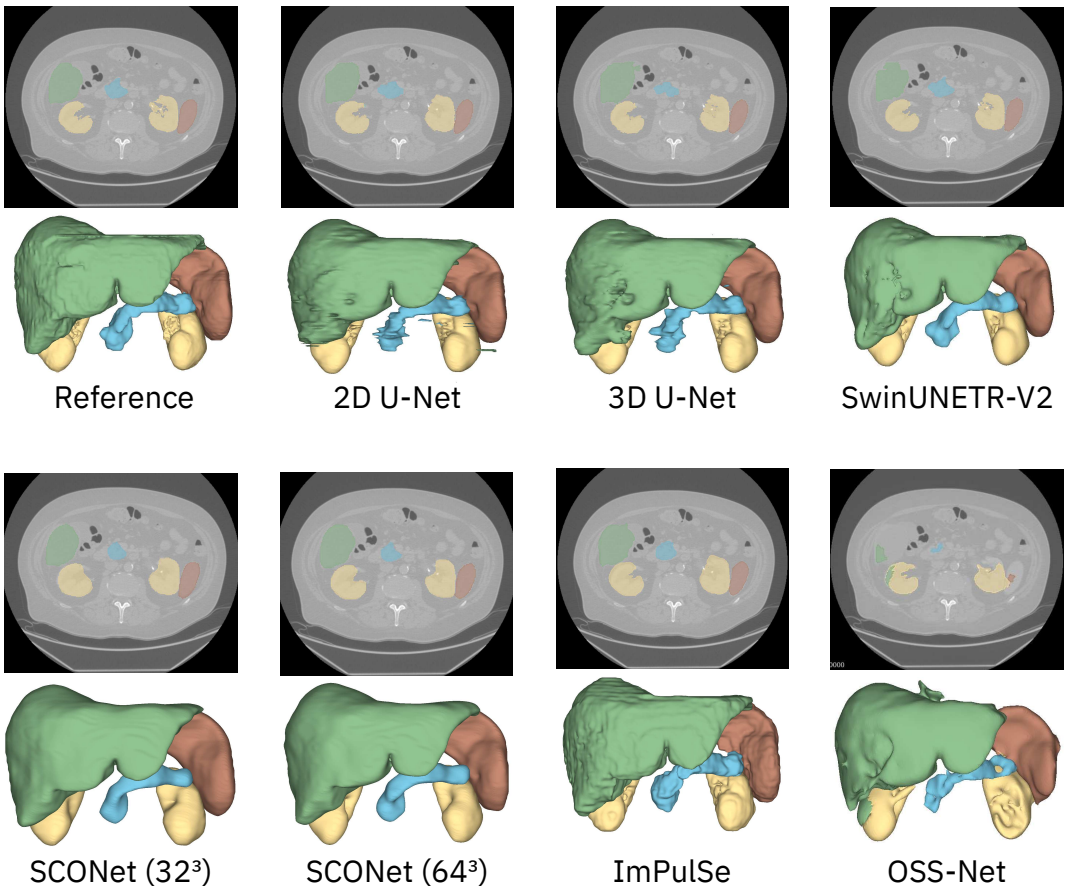


Figure 2.6: *Qualitative results for segmentation of AbdomenCT-1K. The segmentation maps are represented in 2D with an axial CT slice and in 3D for the liver (green), kidneys (yellow), pancreas (blue) and spleen (red).*

Results of quantitative evaluation and comparison according to the Dice coefficient and the Hausdorff Distance (HD) are reported in Table 2.1 (AbdomenCT-1K) and Table 2.2 (AMOS). We compare SCONet at two grid resolutions ($r = 32$ and $r = 64$) with discrete and implicit baselines. We highlight in bold the results for the average Dice and HD that are significantly better than those of other methods, based on the Wilcoxon test. Several bold figures in the same column indicate that there is no significant difference between these methods but that they are significantly better than the other non-bold methods. Figure 2.7 and Figure 2.8 report per-organ metrics with a boxplot representation in order to have a better visualization of the distribution of the results for the different organs.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Model	Dice					HD
	LIV	*KID*	*SPL*	*PAN*	AVG	AVG
2D U-Net	0.966	0.948	0.950	0.817	**0.920**	79.26
3D U-Net	0.961	0.946	0.946	0.823	**0.919**	46.28
SwinUNETR-V2	0.950	0.936	0.954	0.830	**0.918**	139.16
ImPulSe	0.956	0.926	0.939	0.772	0.898	21.74
OSS-Net	0.915	0.836	0.851	0.493	0.774	41.47
SCONet (32^3)	0.958	0.916	0.938	0.755	0.892	19.12
SCONet (64^3)	0.965	0.929	0.949	0.801	0.911	**17.94**

Table 2.1: *Segmentation results for AbdomenCT-1K*

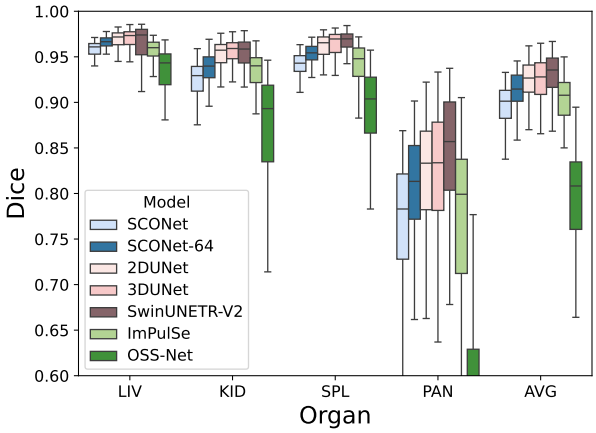


Figure 2.7: *Dice results for AbdomenCT-1K*

Model	Dice																HD
	SPL	*RKI*	*LKI*	*GBL*	*ESO*	*LIV*	*STO*	*AOR*	*IVC*	*PAN*	*RAG*	*LAG*	*DUO*	*BLA*	*PRO/UTE*	AVG	AVG
2D U-Net	0.926	0.918	0.937	0.701	0.776	0.945	0.806	0.915	0.813	0.745	0.679	0.679	0.665	0.788	0.725	0.801	69.89
3D U-Net	0.931	0.922	0.940	0.718	0.781	0.953	0.838	0.925	0.851	0.782	0.703	0.687	0.708	0.801	0.707	**0.816**	45.68
SwinUNETR-V2	0.840	0.921	0.924	0.693	0.759	0.900	0.791	0.898	0.852	0.804	0.695	0.669	0.710	0.794	0.780	0.802	176.93
ImPulSe	0.883	0.856	0.868	0.580	0.528	0.925	0.772	0.826	0.720	0.655	0.278	0.325	0.532	0.687	0.594	0.669	33.98
OSS-Net	0.701	0.647	0.691	0.033	0.000	0.816	0.478	0.357	0.276	0.291	0.000	0.000	0.067	0.283	0.243	0.325	47.39
SCONet (32^3)	0.884	0.849	0.870	0.577	0.588	0.924	0.771	0.795	0.733	0.667	0.488	0.437	0.557	0.642	0.612	0.693	**24.74**
SCONet (64^3)	0.909	0.880	0.905	0.623	0.640	0.938	0.804	0.848	0.776	0.717	0.523	0.493	0.601	0.673	0.655	0.732	**27.29**

Table 2.2: *Segmentation results for AMOS*

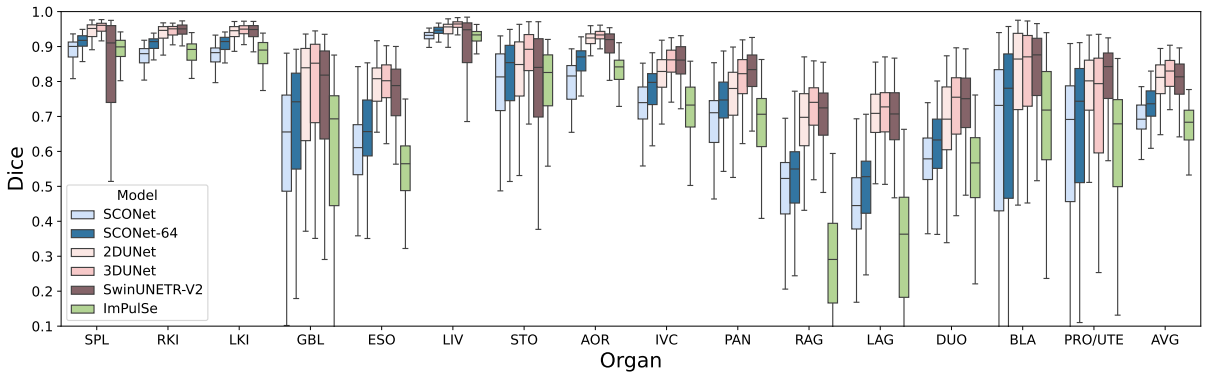
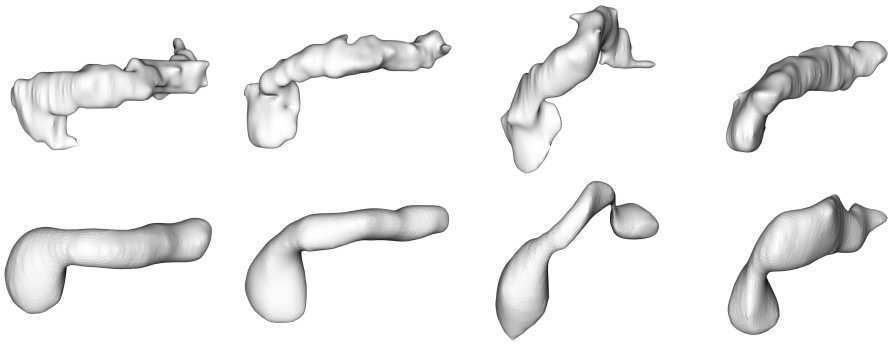


Figure 2.8: *Dice results for AMOS. Results for OSS-Net are not included to improve readability.*

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

In terms of segmentation surface precision, INR-based methods produce better HD, with SCONet having the best results. This low HD comes from a low number of outlier pixels in the segmentation maps generated with these methods. The larger HD from discrete strategies can be explained by their restriction to work on slices or patches, which can generate stitching artifacts and thus noise.

Dice scores in both datasets confirm the visual observation that SCONet achieves a lower segmentation accuracy compared to discrete methods. However, the gap with these baselines decreases when using a larger resolution of the feature grid. Additionally, SCONet has equivalent or better results compared to the two implicit baselines. An in-depth analysis of the results from SCONet shows that the pancreas (PAN) is particularly difficult to segment in both datasets compared to the other organs. We explain this result not only by the high diversity of shapes for this organ as shown in Figure 2.9, but also by its local texture which results in poorly defined contours in the extracted point clouds, making the reconstruction more difficult. In AMOS, the duodenum (DUO) is also poorly segmented for the same reasons. Additionally, the prostate/uterus (PRO/UTE) is difficult to segment due to the ambiguity of the label which represents two different anatomical structures depending on the sex of the patient. Moreover, organs with the lowest Dice scores are also the smallest ones: the adrenal grands (LAG, RAG), the gallbladder (GBL) or the esophagus (ESO). Other organs like the bladder (BLA) have low Dice scores because of missing points near the reference surface.



(a) *Dice=0.792* **(b)** *Dice=0.773* **(c)** *Dice=0.593* **(d)** *Dice=0.483*

Figure 2.9: *3D visualization of the pancreas for the reference (top) and the prediction with SCONet (bottom) for two samples with a high Dice score (a and b) and two samples with a low Dice score (c and d)*

We can define two limiting factors for the segmentation accuracy of SCONet. First, the projection of the point cloud into a low-resolution grid leads to a loss of precision from the high-resolution point cloud, especially when a lot of points are projected in the same grid cell. This transformation into a lower resolution explains the difficulty of our method to reconstruct the small organs or organs that have an irregular, bumpy or winding shape. The second limiting factor is the input point cloud itself because there are missing points near organs that have low contrast compared to their background. In this case SCONet interpolates a surface that might not represent accurately the ground truth, especially for organs whose shape varies a lot depending on the patient. We explore this limitation in Chapter 3. Modifying the parameters for the contour detection might help recovering these surface points but the additional noise in the overall point cloud could lead to a deterioration of results for other organs.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

III.2 Efficiency

We also report the computation cost of all compared methods in Table 2.3. We study the memory and time related metrics described in Section I.5.

Model	Inf. GPU (GB)	#Param (M)	FLOPs (G)	Train. time (h)	Inf. time (s)
2D U-Net	0.77	1.36	960	123	1.26
3D U-Net	4.48	4.08	2,980	54	2.16
SwinUNETR-V2	7.59	18.35	2,961	17	4.87
ImPulSe	16.25	33.28	2,504	13	1.61
OSS-Net	3.95	9.12	4,908	15	4.14
SCONet (32^3)	0.83	1.26	120	11	0.92
SCONet (64^3)	1.27	1.26	218	19	1.00

Table 2.3: *Efficiency results for AbdomenCT-1K.*

III.2.1 Memory footprint

In terms of memory footprint, we report the maximum GPU memory usage at inference and the number of trainable parameters for the models compared. SCONet and 2D U-Net have the lowest memory usage: both models are for example four times more memory efficient than 3D U-Net or OSS-Net and more than ten times more memory efficient than ImPulSe or SwinUNETR-V2. The memory efficiency of SCONet comes from two key design choices. First, its encoder uses a small CNN (only three levels) to process a low resolution grid (32^3 or 64^3), while the other voxel-based models process high resolution volumes (256^3) with large CNNs. SCONet's memory efficiency also comes from its decoder which consists of a small MLP. This light INR decoder enables SCONet to output a segmentation map at any resolution with a constant memory cost, which is an advantage compared to non-implicit baselines.

SCONet and 2D U-Net also have the lowest number of parameters. In SCONet, most of these parameters (1.21 million out of the total 1.26 million) belong to the UNet module.

III.2.2 Time complexity

To evaluate the computational complexity of the different models, the number of FLOPs is reported for a fixed output size of (256x256x100). We find that SCONet is the fastest network (for both grid resolutions tested) because it requires at least four times fewer FLOPs compared to the baselines. We also note that SCONet's encoder has a fixed number of computations for a given input size which is 15 GFLOPs for $r = 32$ and 113 GFLOPs for $r = 64$. Therefore, the number of FLOPs for SCONet depends mostly on the number of query operations necessary to reconstruct the segmentation map.

We also report the average inference time for the testing set from the AbdomenCT-1K dataset. We find that SCONet has the fastest inference speed compared to the other baselines, closely followed by 2D U-Net and ImPulse. This inference time however does not include the

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

pre-processing steps: for SCONet, it does not include the extraction of the point cloud and SURF features. This pre-processing requires on average 2.90s to run on one input volume. To be more precise, the contour detection step takes 0.65s to run and the SURF feature computation requires 2.25s. Therefore, while the SCONet network is more time efficient than the other networks, our final workflow is slightly slower than most of our competitors with an average total inference time equal to 3.82 seconds.

We also report the training times, though these should be interpreted with caution. The number of epochs was indeed chosen empirically to ensure convergence of each approach so it varies across models. Most networks require a training time within the same order of magnitude except for the 2D and 3D U-Net. These two models require longer training times due to memory constraint: since the full datasets do not fit inside the RAM, volumes must be loaded once or several times per epoch. This issue is aggravated by the fact that the Jean Zay computing resources on which we run our experiments has non-negligible loading times: loading a single (256×256×100) volume for example takes approximately 0.23 seconds.

III.2.3 Network optimization

Motivation and method In this section we investigate several optimization tools or efficient modules introduced in Chapter 1, Section I.4 in order to study whether SCONet's efficiency can be improved even further.

We first apply automatic mixed precision (AMP), which reduces the precision of certain layers, originally encoded in 32 bits, down to 16 bits. This precision reduction is for example automatically applied on convolution and linear layers which are in result significantly accelerated. While AMP can be applied during both training and inference, our experiments focus specifically on inference.

We then explore two approaches that are commonly used to improve the efficiency of CNNs and apply them to the UNet module contained in the encoder. The goal is to reduce the computation cost associated with UNet. This would allow us to increase the resolution of the latent grid and therefore improve the segmentation accuracy, without adding to the overall computational burden. We re-train SCONet by using the following approaches:

- we replace the original concatenation operation used in UNet to aggregate the early feature maps from the encoder to the deeper decoder features. Instead, we apply the sum operation in order to reduce the number of intermediate feature maps and therefore the memory used;
- we use depthwise (DW) separable convolutions to reduce UNet's number of parameters as well as its complexity. This method consists in factorizing a regular convolution into one DW convolution and one point-wise convolution.

Results and discussion Table 2.4 presents the segmentation and efficiency performance for these three experiments on the AbdomenCT-1K dataset. We highlight in bold the results for the average Dice and HD that are significantly better than those of other methods, based on the Wilcoxon test. Several bold figures in the same column indicate that there is no significant

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

difference between these methods but that they are significantly better than the other non-bold methods.

Configuration	Dice	HD	Inf. GPU (GB)	#Param (M)	FLOPs (G)	Inf. time (s)
SCONet baseline	**0.892**	**19.12**	0.83	1.26	120	0.92
AMP (inference)	**0.892**	**19.26**	0.80	1.26	120	0.52
Sum instead of concatenation	**0.887**	21.20	0.82	1.26	118	0.93
DW separable convolution	0.871	23.17	0.86	0.11	107	1.04

Table 2.4: *Network optimization of SCONet*

Our first observation is that using AMP at inference leads to an improved efficiency with no drop in segmentation accuracy. More precisely, inference is almost twice as fast and there is a slight reduction of the GPU memory usage because the layers' weights are encoded on a smaller number of bits.

Regarding the modifications applied to the UNet module, we first find that replacing the concatenation by a sum does reduce the GPU memory usage and the number FLOPs, although this improvement is quite small. For the second experiment, using DW separable convolution requires more than ten times less parameters to run and also slightly reduces the computation complexity. However, it also increases the number of intermediate feature maps and therefore the GPU memory usage. Finally, while both methods do improve slightly either the memory usage or the computation complexity, they also degrade the segmentation accuracy, which makes them unsuitable to our final goal.

In summary, AMP effectively reduces both the memory usage and the inference cost. Further memory reductions through optimization of the UNet module is however challenging, as the results so far have been unsatisfactory. In contrast, there is leeway to improve the computation complexity. Indeed, we can question the decoding process which is computationally expensive due to the querying of one point per voxel to reconstruct the segmentation map. The selection of a more optimal subset of query points could reduce the computation complexity at inference.

III.2.4 Efficient query point selection with the MISE algorithm

Motivation and method As explained previously, when generating a segmentation map, SCONet applies the decoder for one point per voxel which is inefficient. In this section we study the MISE algorithm, which was proposed in the original ONet paper (Mescheder et al., 2019). It helps selecting query points more efficiently by sampling points more densely in areas that are close to the surface of the segmented object.

The Multiresolution IsoSurface Extraction (MISE) algorithm is illustrated in two dimensions in Figure 2.10. We first evaluate the occupancy values for the corners on a low resolution grid (we note this resolution r_0). The grid voxels that have both occupied and unoccupied corners are classified as active (shown in pink) and subdivided into eight subvoxels (four subpixels in the 2D illustration). This subdivison introduces new grid points whose occupancy values

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

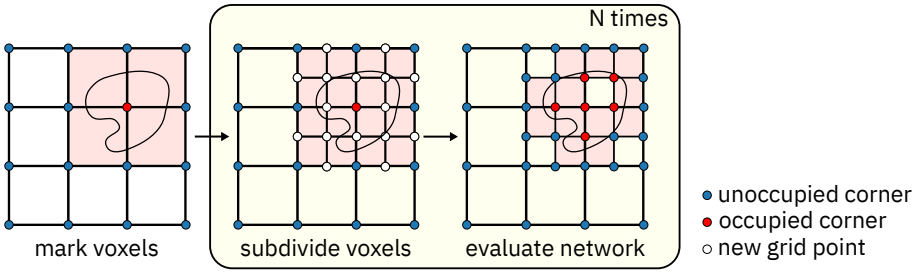


Figure 2.10: *MISE algorithm illustrated in 2D. Figure adapted from Mescheder et al. (2019).*

are then evaluated. The two previous steps are repeated until the desired output resolution is reached. In the original paper, the authors propose to use the marching cubes algorithm (Lorensen et al., 1998) in order to extract a surface mesh. In contrast, in our implementation, we reconstruct a segmented volume by labeling all voxels with at least one occupied corner as part of the reconstructed object. Since we reconstruct C distinct organs, we apply the MISE algorithm C times, once for each organ.

The MISE algorithm is implemented with Cython[3]: this library translates Python code to C/C++ code in order to optimize it and speed up its execution. We use ONet's original implementation and adapt it in order to deal with anisotropic output volumes.

Results and discussion The results are given in Table 2.5 for the AbdomenCT-1K dataset. We compare our SCONet baseline (with a latent grid resolution $r = 32$) where we query one point per voxel to the MISE algorithm for two initial resolutions $r_0 = 32$ and $r_0 = 64$. The MISE algorithm is applied until the full resolution is reached (which corresponds to the input CT volume dimension).

Configuration	DICE	HD	FLOPs (G)	Inference time GPU (s)	Inference time CPU (s)	#Query points
One point per voxel	0.892	19.12	120	0.92	26.38	10,067 K
MISE ($r_0 = 64$)	0.892	18.63	21	1.85	3.06	509 K
MISE ($r_0 = 32$)	0.892	18.63	20	1.66	2.57	330 K

Table 2.5: *Performance of SCONet when using the MISE algorithm for two initial resolutions. The DICE and HD values are given for AbdomeCT-1K.*

Model	Inference time (s)	Inference time CPU (s)
2D U-Net	1.26	89.22
3D U-Net	2.16	18.52
SwinUNETR-V2	4.87	82.17
ImPulSe	1.61	540.40
OSS-Net	4.14	54.60

Table 2.6: *Inference times for compared baselines.*

[3] https://github.com/cython/cython

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

In terms of segmentation quality, there are small differences between the segmentation maps which slightly impacts the HD, although not significantly. Indeed, MISE helps removing small segmentation artifacts that are considered outliers: these are lost during the initial low resolution inference. Those differences however do not impact the Dice score.

Efficiency-wise, using the MISE algorithm leads to a significant decrease of the number of FLOPs. For the record, in the original SCONet, the encoder only required 15 GFLOPs, while the decoder required 105 GFLOPs. With MISE, for the lowest initial resolution $r_0 = 32$, the encoder has not changed but the decoder requires only 5 GFLOPs, thanks to a much smaller number of query points. One major drawback is the inference time which increases because the algorithm is time-costly, despite its implementation in Cython. This increase in inference time was not a problem in the original paper as the structures were less detailed and therefore required fewer query points.

Using the MISE algorithm could however be useful in environments where no GPUs are available to run the SCONet model. Indeed MISE runs almost ten times faster than the standard inference configuration when using a CPU thanks to a reduced number of query points. We find that running the discrete and implicit baselines on a CPU also leads to a large increase in inference time, as presented in Table 2.6. In comparison, our full workflow is at least three times faster than the fastest network, which is the 3D U-Net, even when including the preprocessing steps. This low inference time can be an advantage when working in low-resource settings.

To conclude, while MISE helps selecting query points more efficiently without a decrease in the segmentation quality, it fails to speed up the inference time of the SCONet method due to its time-costly implementation. This algorithm can however be of use in low-resource settings because it ends up being much faster than the compared methods when working on a CPU. We however recommend to be careful when setting the size of the initial resolution grid because smaller isolated structures can be missed for the smallest values of r_0.

IV Artifact reduction through training on random crops

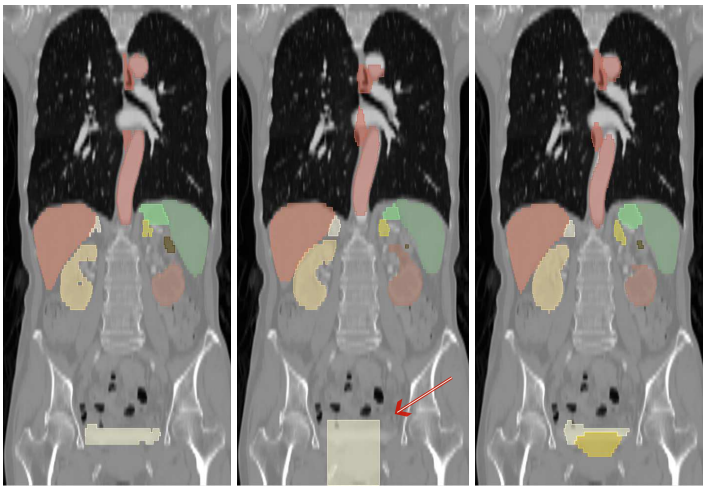
IV.1 Motivation

The aggregation of the input points into the low-resolution grid in the SCONet encoder can lead to a decreased representation capacity because features from multiple points are averaged within a single cell. One way to address this is to increase the resolution of the grid. However, results in Section III showed that doubling the resolution improved performance but also doubled the GPU memory usage. Additionally, another challenge arises when processing volumes with a large field of view along the z-axis. Some points from the input contour point cloud may be projected outside the unit cube. In practice we solve this by clipping the coordinates of the points affected by this issue to the unit cube's boundaries. However, it eventually leads to the type of artifacts shown in Figure 2.11. One way to address this issue is to increase the scaling coefficient mentioned in Section I.1 to ensure that all volumes are projected inside the feature grid. However this would degrade even more the capacity for SCONet to represent

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

high frequency details.

In this section, we propose to train SCONet on random crops of the point clouds in order to both reduce the effect of the projection into a low resolution grid and to remove artifacts. This work was presented in GRETSI 2025 as an ablation study of SCONet (Jouvencel et al., 2025b).



(a) *Reference* **(b)** *Standard SCONet* **(c)** *SCONet trained on random crops*

Figure 2.11: *Illustration of a segmentation artifact for the AMOS dataset. We compare (a) the reference map, (b) the result of a standard SCONet baseline, with an artifact (red arrow) resulting from the aggregation of points that were projected outside the unit cube and (c) the corrected results from SCONet trained on random crops.*

IV.2 Method

For a given cropping window, all the points from the input point cloud whose coordinates are outside its boundaries are first removed. The coordinates of the remaining points are then normalized to fit within the unit cube. The same process is applied to the query point cloud. During training, the cropping window is randomly located in the unit cube. Only one crop per image is processed at each epoch, with the crop location changing between epochs. At inference time, the full point cloud is processed in a sliding window fashion by running several forward passes. Images in each dataset have varying fields of view, requiring a different number of forward passes to generate the full segmentation map. The window size remains the same during both training and inference of SCONet.

IV.3 Results

The size of the cropping window used in our experiments is $(0.8 \times 0.8 \times 0.8)$[4]. For this experiment, we report the detailed Dice and HD for both resolutions of SCONet in Table 2.7 (AbdomenCT-1K) and Table 2.8 (AMOS). We highlight in bold the results for the average Dice and HD that are significantly better than for the other training configuration for a given latent

[4] In comparison, the size of the original unit cube is $(1 \times 1 \times 1)$.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

r	Random crops	Dice					HD
		LIV	*KID*	*SPL*	*PAN*	AVG	AVG
32	No	0.958	0.916	0.938	0.755	0.892	19.12
	Yes	0.962	0.929	0.930	0.781	**0.900**	**18.98**
64	No	0.965	0.929	0.949	0.801	0.911	17.94
	Yes	0.966	0.933	0.948	0.805	0.913	21.43

Table 2.7: *Evaluation of the segmentation quality on AbdomenCT-1K when training on random crops. Standard SCONet is compared with SCONet trained with random crops for both resolutions.*

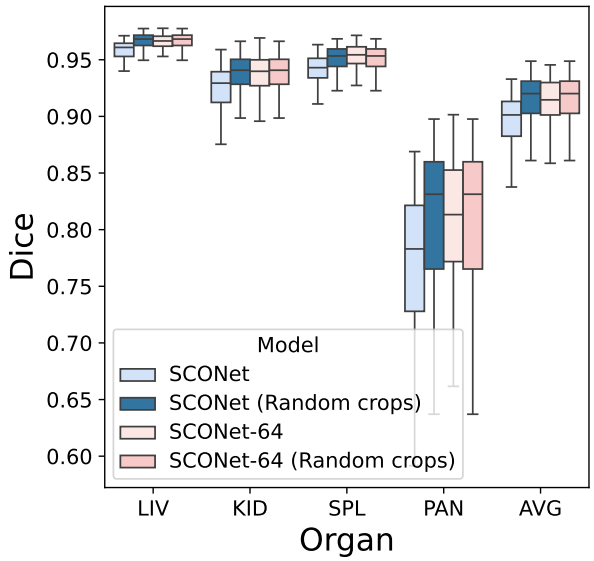


Figure 2.12: *Dice results for AbdomenCT-1K when training on random crops*

r	Random crops	Dice																HD
		SPL	*RKI*	*LKI*	*GBL*	*ESO*	*LIV*	*STO*	*AOR*	*IVC*	*PAN*	*RAG*	*LAG*	*DUO*	*BLA*	*PRO/UTE*	AVG	AVG
32	No	0.884	0.849	0.870	0.577	0.588	0.924	0.771	0.795	0.733	0.667	0.488	0.437	0.557	0.642	0.612	0.693	**24.74**
	Yes	0.884	0.874	0.880	0.588	0.592	0.931	0.796	0.839	0.753	0.682	0.523	0.457	0.580	0.749	0.671	**0.720**	30.79
64	No	0.909	0.880	0.905	0.623	0.640	0.938	0.804	0.848	0.776	0.717	0.523	0.493	0.601	0.673	0.655	0.732	**27.29**
	Yes	0.907	0.884	0.906	0.617	0.658	0.941	0.805	0.870	0.787	0.722	0.531	0.498	0.604	0.793	0.719	**0.750**	35.45

Table 2.8: *Evaluation of the segmentation quality on AMOS when training on random crops. Standard SCONet is compared with SCONet trained with random crops, for both resolutions.*

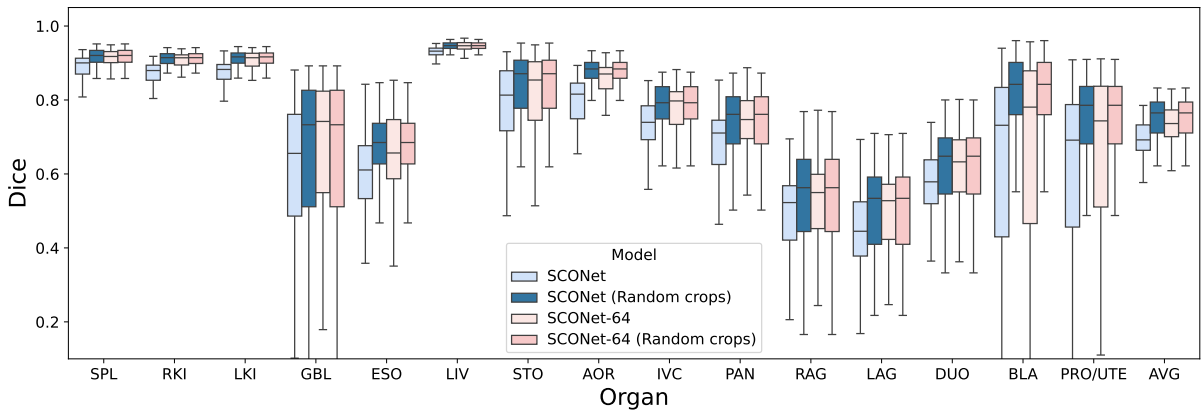


Figure 2.13: *Dice results for AMOS when using a sliding window*

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

grid resolution, based on the Wilcoxon test. Figure 2.12 and Figure 2.13 report the metrics per organ with a boxplot representation in order to have a better visualization of the distribution of the results for the different organs. Table 2.9 reports the memory footprint, number of FLOPs and inference time for the AbdomenCT-1K dataset.

On the AbdomenCT-1K dataset, the Dice scores show a slight improvement on all organs only for the low-resolution configuration. On AMOS, the gains are more substantial across both grid resolutions. The organs with the greatest Dice increase fall into two categories. (1) Some small organs benefit from the "zoom effect" of using a cropping window. For example, the right adrenal gland (RAG) shows an improvement of nearly four Dice points. This effect has however less impact for the higher resolution configuration ($r = 64$) because a plateau is reached when increasing the resolution. (2) Organs that are generally located at the extremities of the volume and that suffered from artifacts mentioned previously also see significant improvements. These include the aorta (AOR), the bladder (BLA) and the prostate/uterus (PRO/UTE). The boxplot representation illustrates that using random crops for training and inference helps reducing the number of outlier volumes with poor segmentation results, especially for these boundary organs. Additionally, Figure 2.11c shows how inference in a sliding window fashion corrects the artifact issues seen in the standard SCONet. Other tests not included here with a smaller cropping window $(0.6 \times 0.6 \times 0.6)$ show a a similar improvement in Dice score but reduce the HD significantly.

r	Random crops	Inf. GPU (GB)	FLOPs (G)	Inf. time (s)
32	No	0.83	120	0.92
	Yes	0.83	409	6.68
64	No	1.27	218	1.00
	Yes	1.27	800	6.81

Table 2.9: *Efficiency results when when training on random crops. Standard SCONet is compared with SCONet trained on random crops, for both resolutions.*

Training on random crops however impacts the computation complexity. The number of FLOPs is indeed multiplied by two: inference using a sliding window requires to infer on some voxels several times as the windows generally overlap. The inference time is also largely impacted: it is more than six times larger when using random crops. This also leads to a small increase in training time, which is due to the additional pre-processing step of cropping the point cloud. It is however notable that the memory usage at inference remains constant, which is one advantage of using this method over simply working at a higher resolution.

IV.4 Conclusion

In conclusion, using random crops to train the network can increase overall performances of SCONet. However the computation complexity can become prohibitive, especially when considering the full workflow. This training strategy could however be of interest when working on datasets that have a size along one axis which is much larger than along the others, which is the case for the AMOS dataset.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

V Ablation studies

V.1 SURF descriptors

Motivation and method In this first study, we explore the relevance of using SURF descriptors as input features for SCONet. These descriptors are concatenated with the PointNet features before being aggregated in the grid. We compare three configurations: (1) using no additional features, (2) using intensity features and (3) using the SURF features. Intensity features are simply the per-voxel intensity values from the original volume at the coordinates of each input point. The version with no additional features corresponds to the original ConvONet which aggregates PointNet features only.

Results and discussion The results for this study are reported for the AbdomenCT-1K dataset in Table 2.10. We highlight in bold the results for the average Dice and HD that are significantly better than for the other training configurations, based on the Wilcoxon test.

Input features	Dice	HD	Inf. GPU (GB)	#Param (M)	FLOPs (G)
None	0.805	24.17	0.78	1.07	114
Intensity	0.834	23.25	0.78	1.07	114
SURF	**0.892**	**19.12**	0.83	1.26	120

Table 2.10: *Performance of SCONet on AbdomenCT-1K for different input features*

We show that enriching points with photometric features improves the segmentation quality significantly with only a small loss in computation efficiency. In particular, using SURF features improves both Dice and HD compared to using only voxel intensities. Indeed, the SURF descriptor contains rich information about a point's neighborhood since it is computed on a 10^3 cube centered on the point. Additional tests that are not displayed here also show that using the additional features (intensity or SURF) as an input to PointNet lead to slightly lower segmentation quality, which is why we chose to concatenate the features at a later stage.

V.2 Data augmentation

Method We evaluate SCONet for different data augmentation strategies:

- Random jittering: we add independent noise to the coordinates of each point in the input point cloud only;
- Random translation: we add a random translation coefficient to the coordinates of the input and query points;
- Random scaling: we multiply the coordinates of the input points and the query points by a random scaling factor in [0.9,1.1];
- Random linear transform: we apply the transformation described in Section I.3.2 on the input and the query points.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

We compare our random linear transform with random jittering, translation and scaling because these three data augmentation strategies are typically used in point cloud processing workflows, notably in PointNeXt (Qian et al., 2022).

Results and discussion Table 2.11 reports the segmentation accuracy on the AbomdenCT-1K and AMOS datasets for the four data augmentation strategies. We highlight in bold the results of average Dice and HD scores that are significantly better than those of other methods, based on the Wilcoxon test. Several bold figures in the same column indicate that there is no significant difference between these methods but that they are significantly better than the other non-bold methods.

Data augmentation	AbdomenCT-1K		AMOS	
	Dice	HD	Dice	HD
No data augmentation	0.881	23.75	0.661	31.04
Random jittering	0.874	27.36	0.643	32.22
Random translation	**0.889**	**20.18**	**0.684**	26.02
Random scaling	0.884	25.38	0.670	30.08
Random linear transform	**0.892**	**19.12**	**0.693**	**24.74**

Table 2.11: *Performance of SCONet for different data augmentation strategies on AbdomenCT-1K and AMOS.*

Random jittering reduces the Dice score of SCONet for both datasets. This drop in performance can be attributed to the fact that the point cloud extracted from the edge map is already naturally noisy. Adding more noise with jittering thus only worsens the results. Regarding random scaling and translation, while they lead to an improvement in segmentation quality compared to the baseline, it is best combining their effect with the random linear transform. Indeed, the proposed transform combines translation, scaling, rotation and shear and gives best results in Dice and HD on both datasets.

V.3 PointNet module

Motivation and method In this final section, we focus on the PointNet module from the SCONet encoder which is used to learn geometric per-point features. Our investigation was motivated by the following observation: when visualizing the learned PointNet features we noticed that they do not seem to capture complex patterns. Instead, the feature values appear to be only correlated to the points' spatial coordinates. For example Figure 2.14 shows several slices from the latent grid corresponding to the same subject and to one PointNet feature (after aggregation). For comparison, Figure 2.15 shows several slices from the latent grid corresponding to the same subject and to one SURF feature (after aggregation). Therefore, to better understand the role of the PointNet module, we decide to remove it entirely and analyze the impact on the model's performance.

Results and discussion Table 2.12 reports the performance of the ablation of the PointNet module in terms of segmentation accuracy and efficiency for AbdomenCT-1K and AMOS.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

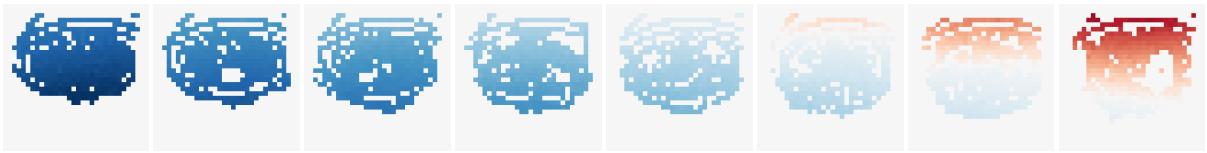


Figure 2.14: *Several slices extracted from the feature grid for one PointNet feature (for one subject of the AMOS dataset)*



Figure 2.15: *Several slices extracted from the feature grid for one SURF feature (for one subject of the AMOS dataset)*

PointNet	AbdomenCT-1K		AMOS		Inf. GPU	#Param	FLOPs	Inf.
	Dice	HD	Dice	HD	(GB)	(M)	(G)	time (s)
Yes	0.892	19.12	0.693	24.74	0.83	1.26	120	0.92
No	0.892	20.95	0.690	24.48	0.82	1.23	119	0.96

Table 2.12: *Ablation study of the PointNet module*

We find that removing the PointNet module does not impact the segmentation accuracy: both configurations do not a show significant difference in the Dice and HD scores. It is also relevant to note that the ablation of this module reduces the memory cost of SCONet. This reduction comes from not only the removal of the PointNet module itself (and therefore the weights and intermediate outputs) but also because the dimension of the feature grid is reduced to aggregate only the SURF features. Improvements in computation complexity are negligible because this metric is mainly influcenced by the decoding process.

We explain the lack of deterioration in the segmentation performance when removing the PointNet module by the nature of the point cloud processed. Indeed, compared to the original work, the point cloud extracted from the contour map is not only much larger (even after sampling) but it also contains a lot of non-surface points which can be considered as noise. In comparison, in ConvONet, the input point cloud contained 2048 surface points. Further experiments regarding PointNet can be found in Chapter 3, Section II.3.

In conclusion, removing PointNet from the SCONet encoder is more suitable as it leads to a lower computation cost compared to the standard baseline. Moreover it makes SCONet's memory usage independent to the number of input points. However, since this finding emerged late in the thesis process, all experiments on SCONet still include the PointNet module unless explicitly stated otherwise.

VI Conclusion

This chapter introduced a novel workflow to generate a segmentation map by using a point cloud representation extracted from the input volume. The point cloud extraction includes

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

the computation of a contour map using the Canny edge detection algorithm, as well as the computation of photometric features for each contour point using the SURF algorithm. We also introduced SCONet, a lightweight implicit network based on ConvONet (Peng et al., 2020), which can generate a segmentation map at an arbitrary resolution by processing a point cloud enriched with photometric features.

We evaluated the proposed workflow on two abdominal CT datasets and compared its performance with both discrete and implicit baselines. SCONet shows good performances on both datasets on large organs with well-defined contours. The evaluation of the computation cost shows that SCONet has a competitive memory usage and particularly low computational complexity, which is useful with limited GPU resources.

We identified two main factors that limit the performance of our segmentation workflow. First, segmentation accuracy suffers from the loss of high frequency details that comes with projecting a high-resolution point cloud into a low-resolution grid in the SCONet encoder, which we addressed by training on random crops of the input point cloud. Additionally, the computation efficiency of SCONet is limited at inference due to the need to query one point per voxel to reconstruct the full segmentation map. One potential solution in low-resource settings is to use the MISE algorithm.

Another challenge that was not solved here is the deterioration of SCONet's performance due to missing points or noisy points. We investigate the issue of the quality of the input point cloud in the following chapter.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Chapter 3:
Study of the point cloud quality

Contents

I Point cloud sampling 62
I.1 Training on a pre-sampled point cloud 62
I.1.1 Motivation and method 62
I.1.2 Results and discussion 62
I.2 Point cloud sampling algorithm 63
I.2.1 Motivation and method 63
I.2.2 Results and discussion 65
II Enhanced point cloud 66
II.1 Reference surface points 66
II.1.1 Motivation and method 66
II.1.2 Results and discussion 67
II.2 Enhanced contour point cloud 69
II.2.1 Reference contour points 70
II.2.2 SCONet-based sampling 71
II.3 Ablation study: PointNet module 73
III Conclusion 75

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

One of the key elements of the segmentation workflow presented in the previous chapter is the point cloud extracted from the contour map. Until now, we have assumed that the information contained in the contour points (and their corresponding features) is sufficient to reconstruct a high-quality segmentation map, but it might not be the case. In this chapter, we present an in-depth study of the point cloud representation used in our approach.

I Point cloud sampling

In our segmentation workflow, the point cloud extracted from the contour map has a very large size (the average size of the point cloud is 1.2 million points for AbdomenCT-1K and 870,000 points for AMOS) and therefore needs to be sampled in order to use it as an input for SCONet. This raises several questions: at what stage should we apply sampling? Which algorithm should we use? How many points should we sample? This section explores these different challenges.

I.1 Training on a pre-sampled point cloud

I.1.1 Motivation and method

The experiments presented in Chapter 2 were done by sampling N points from the full contour point cloud at each epoch. This sampling approach is sometimes referred to as point resampling in the literature and used as a data augmentation step to improve point cloud processing models (Qian et al., 2022). An alternative to this approach is to do offline sampling which consists in generating one subset of points that will be re-used during training. In this section, we compare the online (during training) and offline sampling strategies.

We conduct our experiments on the AbdomenCT-1K dataset with the low-resolution SCONet (with a latent grid size $r = 32$). $N = 50k$ points are sampled for both configurations.

I.1.2 Results and discussion

Table 3.1 compares online and offline sampling of the input point cloud for the AbdomenCT-1K dataset. Experiments on the AMOS dataset show similar trends so we chose to not include them in this thesis.

Configuration	**Dice**					**HD**	**Training**
	LIV	*KID*	*SPL*	*PAN*	AVG	AVG	**time (h)**
Online	0.958	0.916	0.938	0.755	0.892	19.12	11
Offline	0.957	0.913	0.935	0.755	0.890	21.08	6

Table 3.1: *Performance of SCONet on AbdomenCT-1K when sampling is done online or offline*

There is no significant difference in segmentation accuracy between the two configurations. This finding is unexpected because it was hypothesized that re-sampling the point cloud at every epoch would reduce over-fitting and thus improve the Dice and HD scores.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

The aggregation of the input point cloud into a low resolution grid might explain why online sampling does not improve the segmentation accuracy. In particular, features of points aggregated into the same grid cell are averaged so latent grids generated from different point clouds sampled from the same input may remain similar because they have comparable local averages.

In order to evaluate the efficiency of the two sampling configurations, we report their total training time. We show that the training time is almost doubled when re-sampling the point cloud at each epoch. The main factor that explains this increase is that loading a larger point cloud requires more time. More precisely, loading a pre-sampled point cloud containing 50,000 points (occupying approximately 3 MB in memory) takes around 36 ms, whereas loading the full point cloud containing one million points (occupying approximately 60 MB in memory) takes around 91 ms. This 55 ms difference in loading time leads to an increase of 50 seconds for each epoch when training on the AbdomenCT-1K dataset. In total 4.1 hours are thus added to the total training time. Note that the loading time is highly dependent on the hardware used. In our case, experiments are done on the Jean Zay supercomputer resources and therefore the data we process is not stored on the same device as the one we use to train our models. In contrast, using the same device for storage and training would probably reduce the loading time and ultimately the difference in training time.

The other factor that explains the increase in training time is the sampling operation itself. This factor has however a lower impact on the increase compared to the loading time. Indeed sampling 50,000 points from an input point cloud containing one million points requires around 2 ms. This leads to an increase in 1.8s for each epoch and around 9 minutes to the total training time.

To conclude, using a pre-sampled point cloud decreases the total training time of SCONet while maintaining the same level of performance. This makes pre-sampling more suitable for future works.

I.2 Point cloud sampling algorithm

I.2.1 Motivation and method

In this section we study the performance of SCONet for three sampling algorithms. More precisely, we compare random sampling that we used for our main experiments with grid subsampling and farthest point sampling (FPS) (see Section II.3.2 for more information). Experiments in this section are applied on the low-resolution SCONet (with the latent grid with size $r = 32$).

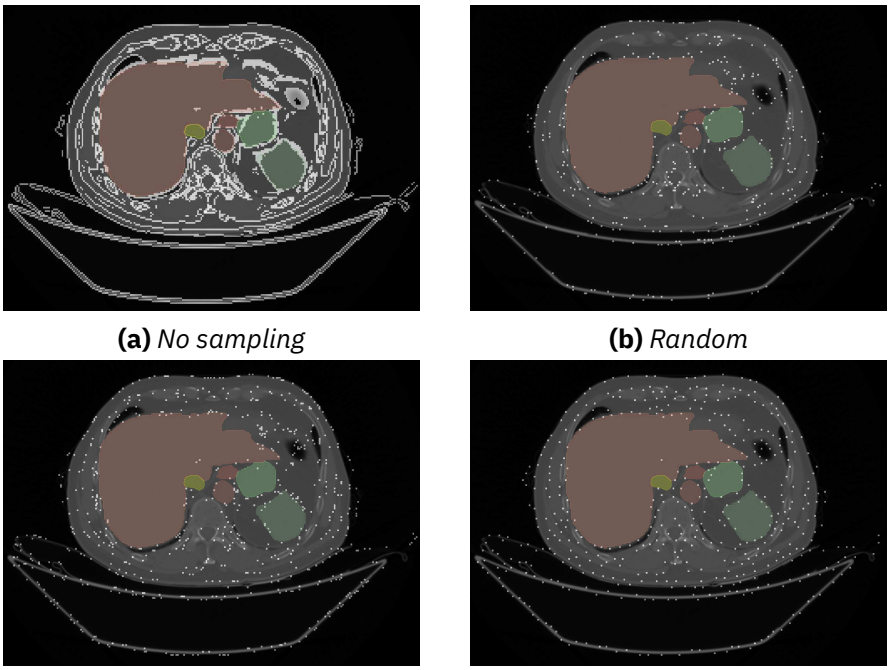
Grid subsampling We choose to investigate the grid subsampling algorithm in order to generate sampled point clouds with more uniform local densities. Our implementation consists in first projecting the point cloud into the 3D grid and then applying weighted random sampling. The weight for each point is computed based on the number of points that are projected on the same grid cell. Empty grid cells are ignored. The parameters of this algorithm are r the resolution of the grid and K the number of points to sample per grid cell. In practice, we set $K = 5$ points in order to have a point cloud with a size within the same order of magnitude

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

as our main experiments. We also use the same grid resolution $r = 32$ as the one used in the SCONet encoder. The resulting point clouds have a varying number of points: in average the size is 61,428 points for AbdomenCT-1K and 57,297 points for AMOS.

Farthest Point Sampling (FPS) We also explore FPS because this algorithm can generate a subset of points that has a good coverage of the input point cloud. Figure 3.1 illustrates this property: compared to random and grid-based sampling, the points generated with FPS are distributed more evenly. The main problem of FPS is that it is slow when dealing with large scale point clouds. For example three minutes are necessary to sample 50,000 points from an input point cloud with one million points. We therefore use the FPS implementation from the `fpsample` library[1], which is based on the bucket based FPS algorithm proposed by Han et al. (2023). This implementation significantly decreases the processing time of this algorithm: sampling 50,000 points from an input set of one million points eventually requires 0.67 seconds.

Random sampling for different sampling sizes Random sampling is the algorithm used for experiments performed in Chapter 2. Additionally to the comparison with the other sampling algorithms, we want to study the influence of the number of points sampled from the input point cloud on the performance of SCONet. We therefore compare three sampling sizes: $N = 10k$, $N = 50k$ and $N = 200k$ points.



(a) *No sampling* **(b)** *Random*

(c) *Grid subsampling* **(d)** *FPS*

Figure 3.1: *Projection of a (subsampled) point cloud on the original volume (with reference segmentation) for the compared sampling algorithms*

[1] https://github.com/leonardodalinky/fpsample

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

I.2.2 Results and discussion

Table 3.2 reports the results for the three sampling strategies on both the AbdomenCT-1K and the AMOS datasets, with an evaluation of the segmentation accuracy and the efficiency. We also report the time necessary to perform sampling of an input point cloud that contains one million points. We highlight in bold the results of average Dice and HD scores that are significantly better than those of other methods, based on the Wilcoxon test. Several bold figures in the same column indicate that there is no significant difference between these methods but that they are significantly better than the other non-bold methods.

Sampling	**AbdomenCT-1K**		**AMOS**		**Inf. GPU**	**FLOPs**	**Inf.**	**Sampling**
strategy	**Dice**	**HD**	**Dice**	**HD**	**(GB)**	**(G)**	**time (s)**	**time (ms)**
None	**0.896**	**18.71**	**0.696**	26.58	16.02	140	1.20	/
FPS	0.895	**18.67**	**0.696**	28.08	0.83	120	0.89	672.1
Grid subsampling	0.889	**19.53**	**0.695**	26.37	0.99	120	0.88	93.6
Random (10k)	0.882	**19.15**	0.677	26.68	0.82	119	1.04	0.4
Random (50k)	0.892	**19.12**	**0.693**	**24.74**	0.83	120	0.92	1.7
Random (200k)	0.894	**20.29**	**0.695**	27.14	1.02	124	1.01	5.2

Table 3.2: *Results for different point cloud sampling strategies*

We first observe that sampling is necessary because of memory constraints. Indeed, using the full point cloud as input for SCONet leads a prohibitive GPU memory usage with only a slight improvement in the Dice and HD scores for the AbdomenCT-1K dataset. This large memory usage comes from to the variation in input sizes across the testing set where some point clouds contain more than one million points, requiring additional memory allocation for both the extra points and the first layer of the PointNet module. The impact of the increased input size on the computation complexity and inference time is however negligible because the decoder dominates these metrics.

Sampling algorithm When comparing the three sampling strategies for an equivalent number of input points we observe no significant difference in segmentation accuracy. In terms of efficiency, random sampling and FPS have the same performance in terms of GPU memory usage, number of FLOPs and inference time. This is expected because the input point clouds are sampled to the same size. In contrast the variation in terms of number of sampled points with grid subsampling leads to a slight increase in memory usage and computation complexity. To conclude, random sampling remains the most suitable strategy to use in SCONet as it is the fastest and the most simple to implement and has same performance in segmentation accuracy as the more complex algorithms.

Sampling size Our experiments on the variation of the number of sampled points show that using too few points to train SCONet leads to a decrease in performance, for a negligible reduction in GPU memory usage and computation complexity. We also find that increasing the number of sampled points to $N = 200k$ does not improve the segmentation accuracy. Simi-

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

larly to what we explained for grid subsampling, processing a larger point cloud with SCONet also leads to a higher GPU memory usage. On a side note, the small variations in inference time for all experiments in this section are probably due to hardware-based fluctuations. To conclude, this experiment confirms that using $N = 50k$ sampled points is the most suitable configuration for SCONet.

II Enhanced point cloud

One of the main assumptions in our work is that contour points are appropriate to recover the surface of the target organs. While there is indeed some overlap between contour points and the reference surface, we also notice two challenges that we illustrate in Figure 3.2. We first observe that are there many points in the cloud which do not lie on the surface and can therefore disturb the model. Besides, critical points are also missing from the contour point cloud, which can lead SCONet to hallucinate some surfaces. In this section, we study how the quality of the point cloud influences SCONet's segmentation performance.

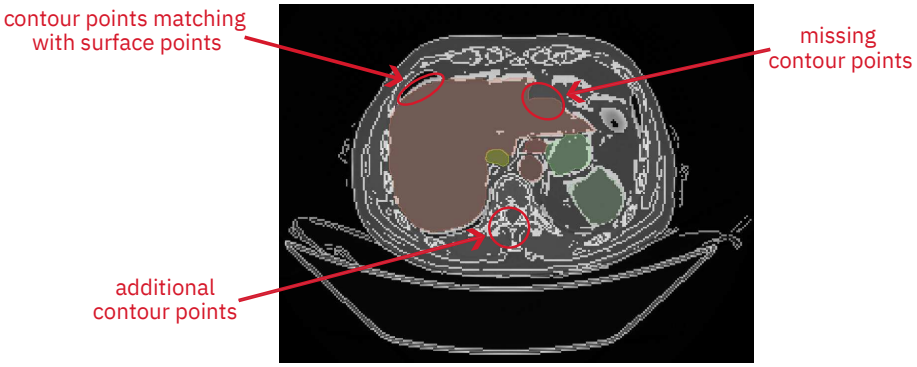


Figure 3.2: *Discrepancy between contour and surface points. This figure shows the original CT slice extracted from the volume, the reference segmentation map and the detected contour points (in white).*

II.1 Reference surface points

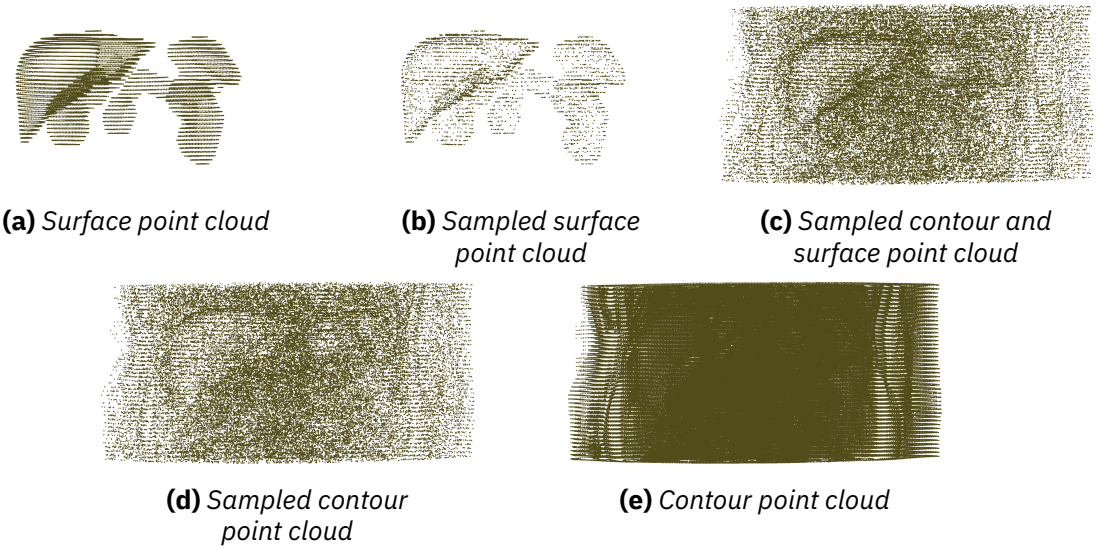
II.1.1 Motivation and method

There is a difference between the point cloud processed in the original ConvONet work (Peng et al., 2020) and the one we use in our method. Indeed, the set of points extracted from the contour map has a larger scale (even after sampling) compared to the one used in ConvONet (which contains 2048 points in the proposed experiments). It also contains many points that are located far away from the surface and which could be considered as noise. In this section, we investigate the representation capacity of SCONet and perform a sanity check to evaluate how effective our model is for the task of multi-label volumetric segmentation when processing a higher quality point cloud. We focus on what we refer to as **surface points**[2]: these are the points whose coordinates correspond to voxels located at the boundary between an organ and

[2] Not to be confused with what we refer to as **contour points** which are extracted from the edge map and therefore correspond to voxels that have a high variation in intensity in the original volume.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

the background, based on a segmentation map. An illustration of the full surface point cloud extracted from the reference segmentation map can be found in Figure 3.3a.



(a) *Surface point cloud*

(b) *Sampled surface point cloud*

(c) *Sampled contour and surface point cloud*

(d) *Sampled contour point cloud*

(e) *Contour point cloud*

Figure 3.3: *3D visualization of different types of point clouds*

Our first experiment therefore consists in applying SCONet on the reference surface points. Pre-processing includes the extraction of the point cloud using the reference segmentation map and the computation of the corresponding SURF features, following the same method described in the previous chapter. Implementation-wise, we set the input size of SCONet to $N = 5k$, as illustrated in Figure 3.3b. This input size is smaller than that of the input size in previous experiments. Indeed the contour point cloud is inherently much larger than the surface point cloud and thus requires a larger input size to ensure that a sufficient number of points are sampled near the surface. We conduct this experiment for the two resolutions of latent grid ($r = 32$ and $r = 64$) in order to evaluate which limitations of SCONet come from the resolution of the grid and which limitations come from the quality of the input point cloud.

We perform a second experiment where we merge contour points and surface points. The goal in this study is to investigate how points far from the surface can influence the performance of our model. We set the total input size to $N = 50k$ in order to compare the results with the SCONet baseline more easily. The input point cloud is composed of 5,000 surface points (the same size as for the first experiment in this section) and 45,000 contour points. An illustration of such a point cloud can be found in Figure 3.3c.

II.1.2 Results and discussion

Segmentation performance for these experiments are reported in Table 3.3 (AbdomenCT-1K) and Table 3.4 (AMOS). We highlight in bold the results of average Dice and HD scores that are significantly better than those of other methods for a given grid resolution, based on the Wilcoxon test.

For our first experiment, we compare the performance of SCONet for surface points and for contour points. We find that both the Dice and the HD score are improved significantly when

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

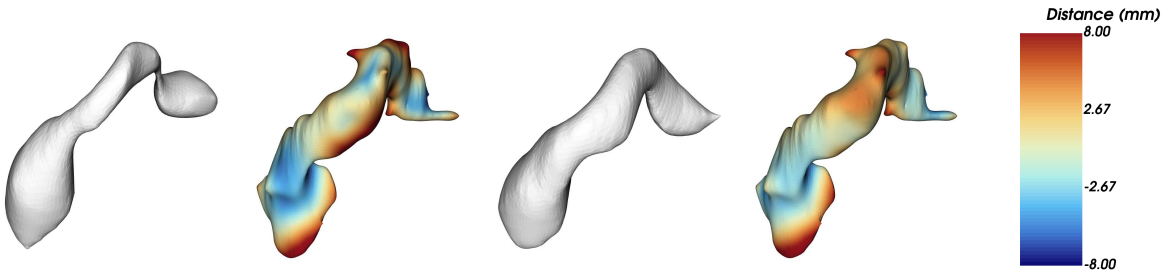
r	#Contour points	#Surface points	Dice					HD
			LIV	*KID*	*SPL*	*PAN*	AVG	AVG
32	50k	0	0.958	0.916	0.938	0.755	0.892	19.12
	45k	5k	0.961	0.918	0.929	0.773	0.895	18.60
	0	5k	0.965	0.933	0.933	0.842	**0.918**	**13.50**
64	50k	0	0.965	0.929	0.949	0.801	0.911	17.94
	0	5k	0.974	0.944	0.944	0.880	**0.936**	**12.74**

Table 3.3: *Performance of SCONet on AbdomenCT-1K for surface and contour point clouds*

r	#Contour points	#Surface points	Dice															HD
			SPL	*RKI*	*LKI*	*GBL*	*ESO*	*LIV*	*STO*	*AOR*	*IVC*	*PAN*	*RAG*	*LAG*	*DUO*	*BLA*	*PRO/UTE*	AVG	AVG
32	50k	0	0.884	0.849	0.870	0.577	0.588	0.924	0.771	0.795	0.733	0.667	0.488	0.437	0.557	0.642	0.612	0.693	24.74
	45k	5k	0.891	0.865	0.875	0.624	0.600	0.932	0.822	0.807	0.756	0.706	0.508	0.463	0.621	0.709	0.692	0.725	23.15
	0	5k	0.897	0.879	0.884	0.633	0.641	0.936	0.845	0.817	0.771	0.724	0.531	0.483	0.660	0.737	0.724	**0.744**	**20.54**
64	50k	0	0.909	0.880	0.905	0.623	0.640	0.938	0.804	0.848	0.776	0.717	0.523	0.493	0.601	0.673	0.655	0.732	27.29
	0	5k	0.929	0.905	0.914	0.707	0.705	0.952	0.887	0.863	0.832	0.792	0.564	0.551	0.739	0.766	0.762	**0.791**	**21.00**

Table 3.4: *Performance of SCONet on AMOS for surface and contour point clouds*

using surface points. Dice scores are particularly increased for organs whose shape varies among patients such as the pancreas (PAN) that has a 0.087 increase for AbdomenCT-1K and a 0.057 increase for AMOS, or the stomach (STO) that has a 0.074 increase for AMOS. To illustrate the performance of SCONet when processing a surface point cloud, Figure 3.4 depicts how the overall shape of the output pancreas label aligns more closely with the reference label (with overall smaller distances between the two labels). We however note that the prediction is still overly smooth compared to the reference. Organs with missing contour points are also significantly improved, like the bladder (BLA) that has a 0.095 increase in Dice score. However, although SCONet processes reference points, our results for the lower gid resolution ($r = 32$) are equivalent or inferior to those of the best discrete baselines from the previous chapter reported in Table 2.1 (AbdomenCT-1K) and Table 2.2 (AMOS). Increasing the resolution to $r = 64$ shows improved results for surface points with significantly better Dice scores compared to the baselines for the AbdomenCT-1K dataset. This experiment confirms that the performance of SCONet is impacted by both the quality of the input point cloud and the resolution of the learned latent grid.



(a) *Contour point cloud (Dice=0.593)*

(b) *Surface point cloud (Dice=0.687)*

Figure 3.4: *3D visualization of the pancreas predicted by SCONet for contour and surface points. For each point cloud, we show the predicted mesh extracted from the segmentation map (left) and the reference mesh colored by signed distance to the predicted surface (right)*

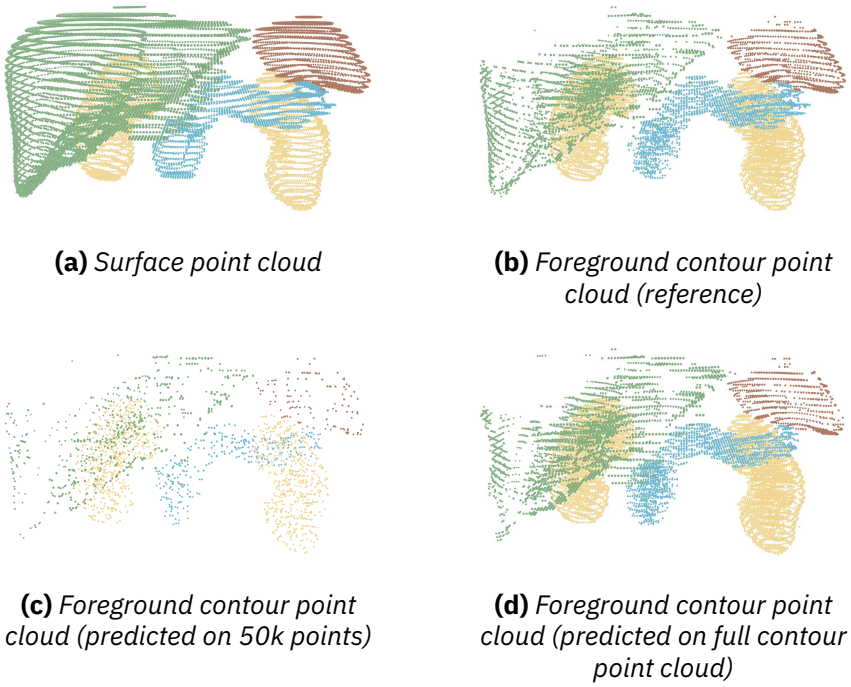
Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

In our second experiment, we find that there is a significant improvement in Dice and HD scores when adding surface points to the original contour points. However, we also observe that this improvement is still much lower than when using only surface points. Therefore, although non-surface contour points can influence SCONet and decrease the segmentation performance compared to using only reference surface points, the impact can be lowered if there is a sufficient number of points close to the surface.

The conclusion to both experiments is that there is room for improvement for the performance of SCONet if we succeed to improve the quality of the point cloud used by removing uninformative noisy points far from the surface. We however admit that it might be difficult to obtain a point cloud with this level of quality since the processed surface points were extracted using the reference segmentation map. Additionally, we can also conclude that the results obtained with the surface point clouds correspond to the upper bound for our model's performance in the standard configuration.

II.2 Enhanced contour point cloud

In the previous section, we found that the quality of the input point cloud significantly impacts SCONet's ability to generate accurate segmentation maps. In this section, we investigate whether it is possible to obtain similar performance by extracting an enhanced point cloud from the contour points used in the main contribution.



(a) *Surface point cloud*

(b) *Foreground contour point cloud (reference)*

(c) *Foreground contour point cloud (predicted on 50k points)*

(d) *Foreground contour point cloud (predicted on full contour point cloud)*

Figure 3.5: *3D visualization of point clouds with corresponding organ labels. Note that the organ labels are displayed here for visualization purposes and are not used as an input feature of SCONet.*

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

II.2.1 Reference contour points

Motivation and method Processing surface points with SCONet like we did in Section II.1 does improve the segmentation accuracy of our workflow. However, these points might be difficult to extract from the volume without relying on the reference segmentation map. Instead, we focus on the contour point cloud used in our main contribution because we assume that some of its points may provide more useful information than others for the reconstruction of the target organs. In particular, points that might be of interest for our task are points that are closer to the surface of the segmentation labels.

We propose an experiment where we remove background points to keep only contour points that are inside the target organs, based on the reference segmentation map. The resulting point cloud will be referred to as the **foreground contour point cloud**. Figure 3.5b gives an illustration of such a point cloud for one sample of the AbdomenCT-1K dataset.

Results and discussion Segmentation performance for this experiment are reported in Table 3.5 (AbdomenCT-1K) and Table 3.6 (AMOS). We highlight in bold the results of average Dice and HD scores that are significantly better than those of other methods, based on the Wilcoxon test.

Point cloud	Dice					HD
	LIV	*KID*	*SPL*	*PAN*	AVG	AVG
Contour	0.958	0.916	0.938	0.755	0.892	19.12
Foreground contour	0.956	0.936	0.941	0.847	**0.920**	**14.37**

Table 3.5: *Performance of SCONet on AbdomenCT-1K for contour point cloud with and without reference background points*

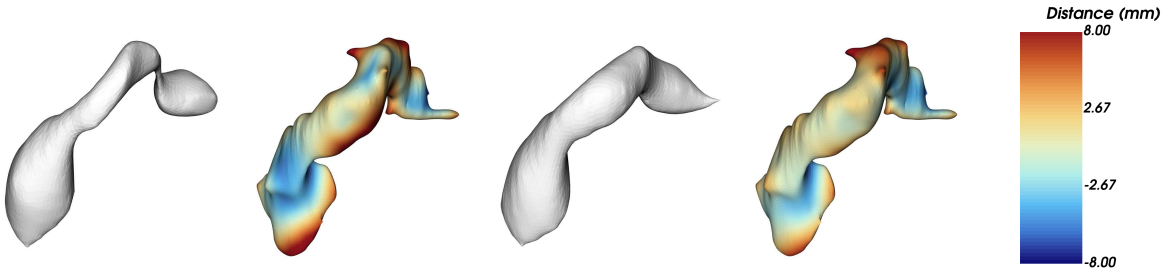
Point cloud	Dice																HD
	SPL	*RKI*	*LKI*	*GBL*	*ESO*	*LIV*	*STO*	*AOR*	*IVC*	*PAN*	*RAG*	*LAG*	*DUO*	*BLA*	*PRO/UTE*	AVG	AVG
Contour	0.884	0.849	0.870	0.577	0.588	0.924	0.771	0.795	0.733	0.667	0.488	0.437	0.557	0.642	0.612	0.693	24.74
Foreground contour	0.879	0.876	0.881	0.548	0.611	0.924	0.833	0.815	0.744	0.695	0.507	0.472	0.629	0.699	0.654	**0.718**	24.59

Table 3.6: *Performance of SCONet on AMOS for contour point cloud with and without reference background points*

For both datasets, using only foreground contour points leads to a significant rise in Dice score. Organs with the largest Dice increase are those whose shape varies across the dataset: for example the pancreas (PAN) has a 0.092 increase for AbdomenCT-1K and a 0.028 increase for AMOS, or the stomach (STO) that has a 0.062 increase for AMOS. Indeed, it is easier for SCONet to reconstruct these structures when the contour points belonging to the neighboring organs are removed. To illustrate the performance of SCONet when processing a foreground contour point cloud, Figure 3.6 depicts how the overall shape of the output pancreas label aligns more closely with the reference label (with overall smaller distances between the two labels). However we observe that the prediction is still overly smooth compared to the reference.

On the other hand, we also observe a decrease or stagnation in Dice score for certain organs such as the liver (LIV). Missing points outside the surface explain the stagnation in Dice

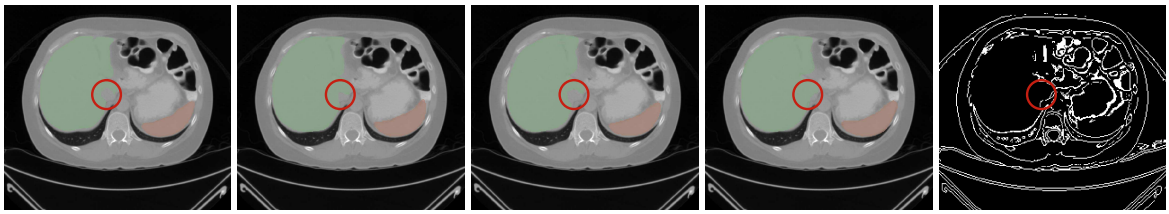
Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés



(a) *Contour point cloud (Dice=0.593)* **(b)** *Foreground contour point cloud (Dice=0.669)*

Figure 3.6: *3D visualization of the pancreas predicted by SCONet for contour and foreground contour points. For each type of point, we show the predicted mesh (left) and the reference mesh colored by signed distance to the predicted surface (right)*

score. For example, Figure 3.7 illustrates how the liver prediction is impacted in a region (in the red circle) where surface points are missing from the contour map. As a result the label map predicted by SCONet when processing the contour point cloud or the surface point cloud align more closely with the reference surface than when processing the foreground contour point cloud.



(a) *Reference* **(b)** *Contour point cloud* **(c)** *Surface point cloud* **(d)** *Foreground contour point cloud* **(e)** *Contour map*

Figure 3.7: *Qualitative segmentation results on AbdomenCT-1K for different input point clouds (a-d) and corresponding contour map (e)*

In conclusion, we found that the contour points located inside the organs are more informative for the reconstruction of the segmentation map. To nuance the obtained results, this experiment focused on contour points located inside the target organs and thus ignored the points near the outer surface boundary. To address this limitation, future work could explore extracting the contour points that are located inside a dilated segmentation label or those that are inside a narrow strip around the organ surfaces.

II.2.2 SCONet-based sampling

Motivation and method In the previous section, the extraction of the foreground contour point cloud was based on the reference segmentation map, which we would not have access to for an unknown subject. However, we have access to a predicted segmentation map thanks to our already trained SCONet model.

In this section, we use SCONet to perform a point cloud segmentation task rather than volume segmentation. Instead of querying one point per voxel, we query the label value for all the points in the input point cloud. It is then possible to filter out all the background points

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

from the contour point cloud. We call the obtained set of points the **predicted foreground contour point cloud**. Figure 3.5c illustrates this point cloud for one patient of the AbdomenCT-1K dataset.

In a first experiment, we start by generating the predicted foreground contour point cloud for every subject in a dataset with our SCONet baseline. We then train a second SCONet model on these point clouds. When using the standard configuration for the initial SCONet (with an input size of 50,000 points) the number of predicted foreground contour points is, on average, over ten times smaller (1,889 points for AbdomenCT-1K) compared to the number of reference foreground contour points (35,704 points for AbdomenCT-1K). We therefore add a comparison where the initial SCONet is evaluated on the original (unsampled) point cloud: in this case the number of predicted foreground contour points has the same order of magnitude as the reference (34,678 points on average for AbdomenCT-1K). Figure 3.5d illustrates this point cloud for one patient of the AbdomenCT-1K dataset.

In a second experiment, we investigate whether training the second SCONet on reference foreground contour points could lead to higher segmentation performance. Indeed, we hypothesize that a model trained on higher quality point clouds would learn better representations of the target organs. We therefore use the SCONet model trained on reference foreground contour points from the previous section and evaluate it on the predicted foreground contour points.

Results and discussion Segmentation performance on both datasets is reported in Table 3.7 for the first experiment and in Table 3.8 for the second experiment. Detailed per-organ results can be found in Appendix B. *Predicted foreground contour* corresponds to the standard configuration where 50,000 points are sampled from the original contour point cloud and *Predicted foreground contour (full)* corresponds to the alternative configuration where no sampling is applied. We highlight in bold the results of average Dice and HD scores that are significantly better than those of other methods, based on the Wilcoxon test. Several bold figures in the same column indicate that there is no significant difference between these methods but that they are significantly better than the other non-bold methods.

Point cloud	AbdomenCT-1K		AMOS	
	Dice	*HD*	*Dice*	*HD*
Contour	**0.892**	**19.12**	**0.693**	**24.74**
Predicted foreground contour	0.874	19.85	0.672	**25.42**
Predicted foreground contour (full)	0.885	19.60	0.686	26.32

Table 3.7: *SCONet performance for different input point clouds.*

In our first experiment, where we trained and evaluated a new SCONet model on foreground contour points predicted by an initial SCONet, we observe that both the Dice and HD scores are deteriorated. The results are worse than those obtained with reference foreground contour points and also compared to the standard baseline performance, even when the initial SCONet is evaluated on the full (unsampled) contour point cloud. In our second experiment

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Inference	AbdomenCT-1K		AMOS	
point cloud	*Dice*	*HD*	*Dice*	*HD*
Reference foreground contour	**0.920**	**14.37**	**0.718**	**24.59**
Predicted foreground contour	0.817	23.37	0.626	28.26
Predicted foreground contour (full)	0.874	19.93	0.670	26.88

Table 3.8: *Performance of SCONet trained on foreground contour points when evaluated on different types of points.*

where we trained SCONet on reference foreground contour points and evaluated on predicted foreground contour points, we found that the deterioration of the Dice and HD scores is even worse for both datasets and both input sizes.

The decrease in performance observed in both experiments can be attributed to the high dependency of this learning strategy on the quality of the initial SCONet predictions. This notably aggravates the issue mentioned in Section II.2.1: points close to the outer boundary of a label can be filtered out even though they are relevant in regards to the reconstruction task. Therefore, if the initial prediction suffers from significant under-segmentation, the performance after processing with the second SCONet may be deteriorated even further.

Another limitation of both workflows presented in this section is their efficiency. Indeed, it is necessary to train two separate SCONet models. Generating the final segmentation map also requires to perform two inferences, although computing the segmentation label for the contour point cloud is much faster than generating a full segmentation map because the number of query operations necessary is reduced. For instance, on average for the AbdomenCT-1K dataset, generating the foreground contour point cloud takes 0.06s with the standard configuration while generating a segmentation map requires 0.92s.

Several approaches could be explored in future works in order to improve the extraction of more informative subsets of points from the contour point cloud. A natural follow up of this experiment is to jointly train the two SCONet models: the first one would focus on learning the best points from the contour point cloud and the second would learn the reconstruction of the segmentation map. Another potential solution is to add a learned sampling module to the SCONet model to process the point cloud before it is fed to the PointNet module.

II.3 Ablation study: PointNet module

Motivation and method This section aims to further explore the findings from Section V.3. We remind the reader that the earlier ablation study of the PointNet module showed no significant effect on the segmentation accuracy of SCONet. We attributed this result to the presence of many contour points which are far from the target organ surfaces and can potentially reduce the module's performance. In contrast, the point clouds used in this chapter contain less noisy points: this could potentially allow PointNet to learn finer details. An illustration of one PointNet feature (after aggregation in the feature grid) can be found for two types of points in Figure 3.8 and Figure 3.9 and shows that PointNet seems to learn richer patterns (at least when processing surface points).

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

In this section, we investigate whether removing the PointNet module still has no impact on the performance of the SCONet model when processing an enhanced point cloud. To be precise, we perform this study for the surface points and for the reference foreground contour points.



Figure 3.8: *Several slices extracted from the feature grid for one PointNet feature when processing a surface point cloud (for one subject of the AMOS dataset)*

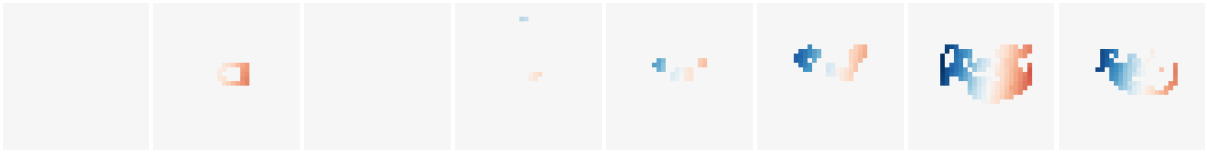


Figure 3.9: *Several slices extracted from the feature grid for one PointNet feature when processing a foreground contour point cloud (for one subject of the AMOS dataset)*

Results and discussion The performance of SCONet for this experiment is reported for the AbdomenCT-1K and AMOS datasets in Table 3.9. We highlight in bold the results of average Dice and HD scores that are significantly better than those of the other method for a given type of point cloud, based on the Wilcoxon test.

Point cloud	PointNet	AbdomenCT-1K		AMOS	
		Dice	*HD*	*Dice*	*HD*
Surface	yes	**0.918**	**13.50**	**0.744**	**20.54**
	no	0.915	14.08	0.733	22.50
Foreground contour	yes	**0.920**	**14.37**	0.718	24.59
	no	0.912	14.92	0.721	24.87

Table 3.9: *Ablation study of PointNet on surface and foreground contour points*

Removing the PointNet module leads to a slight but significant deterioration of the Dice and HD score on both datasets when processing reference surface points. However, this trend does not hold when using the reference foreground contour points as input: the decrease in performance is only observed for the AbdomenCT-1K dataset and not for the AMOS dataset. This finding could be explained by the fact that the AMOS dataset contains a higher number of annotated organs, which results in a higher number of non-surface points extracted from the contour map which may disturb the model.

The results of this study support the earlier assumption: the PointNet module's performance may decrease as the proportion of non-surface points increases. However, these findings also nuance the conclusion of the ablation study from Section V.3 where we explained that

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

it could be more suitable to remove the PointNet module altogether. Indeed the new observations show that the module could improve SCONet's performance if the input point clouds align more closely with the surface of the target object.

III Conclusion

In this chapter we focused on the point cloud representation used in our SCONet model. We first investigated several sampling strategies and notably found that the best configuration is to apply random sampling to extract an optimal number of 50,000 points from the contour point cloud. In a second part, we demonstrated that SCONet's performance could be further improved by enhancing the quality of the input point cloud. In particular, identifying the contour points that are closer to the surface of the target organs could boost the performance of SCONet, although we have not been able to do so without relying on the reference segmentation map.

Beyond the challenges related to the point cloud representation, our current evaluation has focused on a standard segmentation task. We hypothesize that SCONet's ability to learn low frequencies can be an advantage in terms of robustness across different data distributions which we will explore in the following chapter.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Chapter 4:
Robustness of the proposed workflow

Contents

I Dataset shift 78
I.1 Motivation 78
I.2 Characteristics of the studied datasets 78
I.3 Method 80
I.4 Results and discussion 80
II Cross-modality learning 82
II.1 Motivation 82
II.2 Method 82
II.3 Results 83
II.4 Discussion 84
III Conclusion 86

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

In the previous chapters, we studied the performance of our workflow for a standard segmentation task, where training and evaluation were conducted for a given dataset and image modality. However, deploying medical image segmentation algorithms in real-world settings requires to also examine their performance under domain shifts.

A domain shift occurs when the source data (used for training) and the target data (used for evaluation) follow different distributions. In this chapter, we investigate the robustness of our workflow when faced with such shifts.

I Dataset shift

I.1 Motivation

Dataset shifts are a major challenge for medical image analysis (Guan et al., 2022), in particular for the task of segmentation. Indeed, images that make up various datasets used to train and evaluate segmentation models may have different data distributions compared to images acquired in a clinical setting. This holds even when the target modality is the same and when the target structure is the same. For example, images provided by different hospitals may have been acquired following different protocols on devices from different vendors. These images could have also been acquired from healthy patients or from patients who suffer from various diseases. Therefore, there is no guarantee that an image acquired in a clinical setting that falls outside of the training distribution will be accurately segmented. For example J. Ma et al. (2022b) show how the segmentation performance of a neural network can be impacted when evaluated on images from new medical centers where data distributions differ from the training set, or on images from patients that suffer from pathologies that are underrepresented in the training set. While training on larger and more diverse datasets may help mitigate the effects of this dataset shift, it is still relevant to examine how developed segmentation approaches perform in this context. In this section, our objective is to evaluate the robustness our our workflow when faced with a dataset shift.

I.2 Characteristics of the studied datasets

In the previous chapters we have trained and evaluated our workflow on the two datasets of interest in a separate way. We however note that both AbdomenCT-1K and AMOS are comprised of abdominal CT volumes, and that they also contain annotations of the same target organs with the liver, the spleen, the kidneys (although the right and left kidneys have separate labels in the AMOS dataset), and the pancreas. In this subsection, we extract several characteristics from the studied datasets which can be relevant in order to better understand the results obtained from the dataset shift experiment performed in the following subsection.

As a reminder, we note that the AbdomenCT-1K dataset (J. Ma et al., 2022b) is composed of 1000 annotated CT volumes. It comes from the merging of five existing public datasets with a new set of 50 CT images. In total, the dataset contains images from twelve medical centers, acquired with both portal and arterial phases on different imaging devices. As for the AMOS

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

dataset (Ji et al., 2022), it contains annotations for 300 CT images acquired from two medical centers and three imaging devices.

We study the diversity of AbdomenCT-1K and AMOS by examining some statistics related to the dimension of the images and corresponding annotations, and to their intensity distributions. We first observe in Table 4.1 that both datasets contain images with a varied number of slices and slice thicknesses, with AbdomenCT-1K containing the largest heterogeneity. This indicates the diversity of these datasets in terms of image resolution and field of view. Additionally, the same table reports the average intensity values within the target organs, measured in Hounsfield Units (HU) which show that the range of intensity in a given organ is quite wide for each dataset. We also notice that AbdomenCT-1K has slightly higher average intensity values compared to AMOS which might be the result of the difference in acquisition protocols in both datasets. Finally, Figure 4.1 illustrates the diversity of organ volumes in both datasets (for example, in AbdomenCT-1K, the liver can occupy between 0.5L and 4.5L). It also shows an overlap for the different structures across both datasets which indicates that they share a similar distribution of organ sizes.

Finally, it is also relevant to note that both datasets include images from patients who suffer from various diseases, although they do not provide per-image details. From the available information, we know that AbdomenCT-1K includes both healthy and non-healthy patients, whereas AMOS consists exclusively of patients with abdominal tumors or abnormalities.

Dataset	**# Slice**	**Slice spacing (mm)**	**HU mean (min,max)**			
	mean (min,max)	mean (min,max)	LIV	KID	SPL	PAN
AbdomenCT-1K	201(31,1026)	2.7(0.45,8)	103.6 (-2.1,192.4)	137.1 (16.1,268.7)	109.9 (29.7,229.9)	75.1 (-19.3,176.6)
AMOS	138(68,353)	4.2(1.25,5)	70.8 (19.1,170.6)	84.5 (14.1,252.5)	76.1 (30.3,195.1)	62.4 (-10.7,159.7)

Table 4.1: *Detailed characteristics for the AbdomenCT-1K and AMOS datasets. For the AMOS dataset, we only include organs that are also in AbdomenCT-1K and we fuse the two kidneys' annotations.*

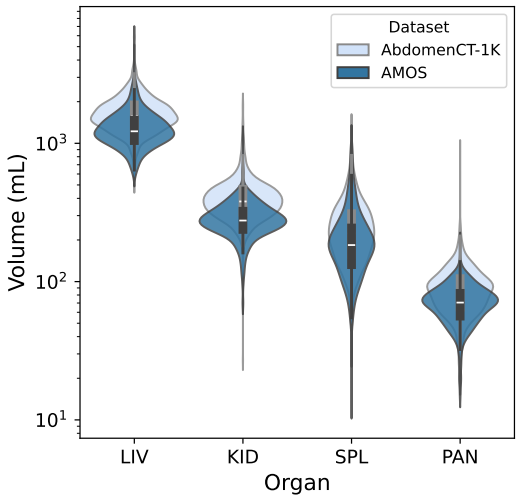


Figure 4.1: *Organ volume distribution for both datasets. For the AMOS dataset, we only include organs that are also in AbdomenCT-1K and we combine the two kidneys' annotations.*

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

I.3 Method

We investigate the dataset shift from AbdomenCT-1K to AMOS: we train SCONet on the AbdomenCT-1K dataset and evaluate it on the AMOS dataset. We chose this configuration of dataset shift because not only is AbdomenCT-1K larger than AMOS but it is also more diverse in terms of image resolutions or intensity distributions. We perform this study for both resolutions of SCONet ($r = 32$ and $r = 64$) and for the discrete and implicit baselines introduced in Chapter 2, Section II.3.

I.4 Results and discussion

Table 4.2 reports the performance of the compared models on both datasets when trained on AbdomenCT-1K. We also include a boxplot of the average Dice values in Figure 4.2 in order to have a better visualization of the distribution of the results for the different evaluation datasets.

Model	**Test dataset**	**Dice**					**HD**
		LIV	*KID*	*SPL*	*PAN*	AVG	AVG
2D U-Net	AbdomenCT-1K	0.966	0.948	0.950	0.817	0.920*	79.26
	AMOS	0.911	0.845	0.882	0.674	0.828	93.20
3D U-Net	AbdomenCT-1K	0.961	0.946	0.946	0.823	0.919*	46.28
	AMOS	0.916	0.878	0.891	0.669	0.839	58.34
SwinUNETR-V2	AbdomenCT-1K	0.950	0.936	0.954	0.830	0.918*	139.16
	AMOS	0.878	0.833	0.907	0.726	0.836	209.61
ImPulSe	AbdomenCT-1K	0.956	0.926	0.939	0.772	0.898	21.74
	AMOS	0.936	0.866	0.902	0.667	0.843	31.72
OSS-Net	AbdomenCT-1K	0.915	0.836	0.851	0.493	0.774	41.47
	AMOS	0.863	0.720	0.750	0.325	0.664	44.53
SCONet (32^3)	AbdomenCT-1K	0.958	0.916	0.938	0.755	0.892	19.12
	AMOS	0.926	0.869	0.908	0.698	0.850	25.02
SCONet (64^3)	AbdomenCT-1K	0.965	0.929	0.949	0.801	0.911	17.94*
	AMOS	0.934	0.879	0.915	0.721	0.862†	24.15†

Table 4.2: *Evaluation on AbdomenCT-1K and AMOS for models trained on AbdomenCT-1K. * and † indicate the best average Dice and HD score when the evaluation is performed on AbdomenCT-1K and AMOS respectively, according to the Wilcoxon test.*

The evaluation the different approaches on the AMOS dataset results in poorer segmentation accuracy for all the models tested. The drop in performance occurs for all four organs but it is even larger for the pancreas (PAN) which is the most challenging organ in the AbdomenCT-1K dataset. However, we also find that the overall drop in performance is lower for both SCONet configurations and for ImPulSe compared to the other baselines. For example SCONet (32) has only a 0.042 drop in Dice score while 3D U-Net has a 0.08 drop. SCONet (64^3) notably has a significantly higher Dice score and lower HD. The boxplot also indicates that SCONet has a smaller number of outlier volumes with poor segmentation results as seen with the model's

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

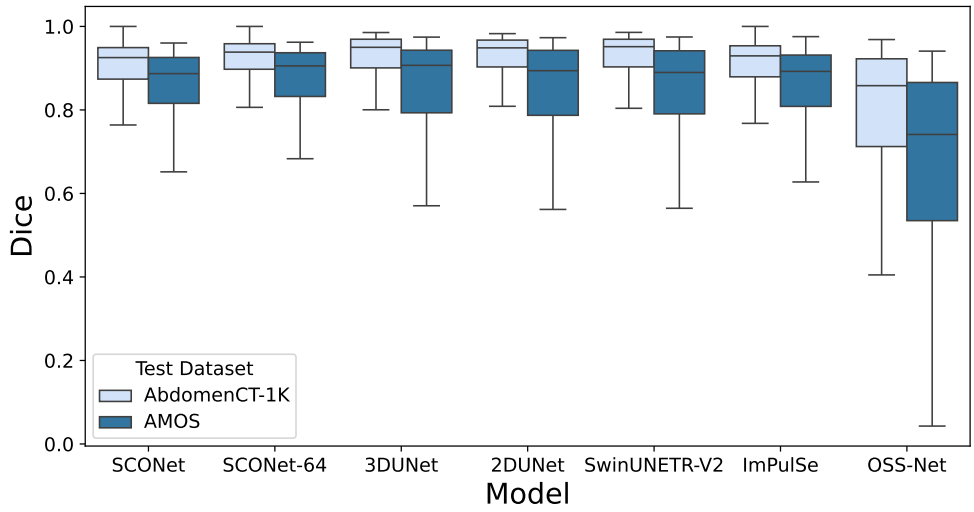
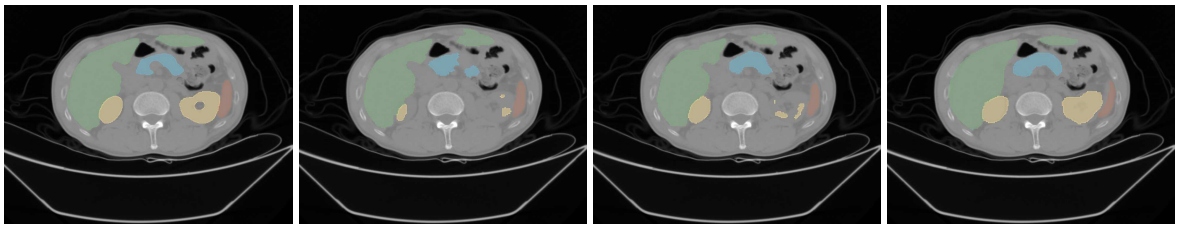


Figure 4.2: *Average Dice score on AbdomenCT-1K and AMOS for models trained on AbdomenCT-1K*

smaller interquartile range and distance between the whiskers compared to the baselines, including ImPulSe.

Figure 4.3 illustrates a challenging case where two discrete baselines, 3D U-Net and SwinUNETR-V2 both fail to segment the kidneys accurately, probably because the device on which the image was acquired differs from the ones used to build the AbdomenCT-1K dataset. In contrast, SCONet generates a coherent segmentation map, although its labels' boundaries remain over-smoothed compared to the reference.



(a) *Reference* **(b)** *3D U-Net* **(c)** *SwinUNETR-V2* **(d)** *SCONet*

Figure 4.3: *Reference and predicted segmentation map when training on AbdomenCT-1K and evaluating on AMOS. This example is overall well-segmented for SCONet but challenging for the discrete baselines.*

To conclude, training a deep learning model on a large and diverse set does not prevent a significant drop in performance when evaluating it on a different dataset, even when both share similar intensity and annotation distributions. The other observation from this experiment is that two of the three evaluated implicit baselines are more robust than the discrete baselines when confronted to such a dataset shift. This shows the benefits of using implicit representations when facing this issue. One hypothesis for the improved robustness of INRs is that their decoder usually consists in a MLP, a type of network known to favor the lower frequencies during learning (Rahaman et al., 2019). However it also leads to the over-smoothness previously mentioned. Another factor contributing to SCONet's slightly better robustness could be that it does not rely directly on intensity values contrary to voxel-based models. Instead, it processes

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Canny-extracted contour points enriched with SURF features, which may be less sensitive to minor changes in intensity distributions across images.

II Cross-modality learning

II.1 Motivation

While our previous experiments have solely focused on CT volumes, in practice both CT and MRI images are commonly acquired to visualize anatomical structures such as abdominal organs or abnormalities like tumors (Valindria et al., 2018). These modalities capture complementary information: CT provides clear visualization of the bones or blood vessels (depending on the contrast agents used) whereas MRI delivers highly detailed images of soft tissues and organs (FDA, 2017). Given the scarcity of available annotated medical images for training deep neural networks, it is relevant to explore whether models trained on one imaging modality can generalize to another, or if it can even be possible to merge images from both modalities to build a larger training set. This challenge is known as cross-modality learning and consists in learning across multiple modalities using unpaired images. This last term indicates that the subsets from different modalities do not contain images from the same patients.

In the previous section, we have found that SCONet is more robust than discrete baselines to dataset shifts. In this section, we want to examine whether this result can also hold in case of what we will refer to as a modality shift. We therefore evaluate the performance of our approach in the context of cross-modality learning.

II.2 Method

In this experiment, we study the three following configurations:

- train on CT images only, evaluate on CT and MR images;
- train on MR images only, evaluate MR images. There is a high imbalance between CT and MR images and we choose not to evaluate the configuration on the other modality since the CT testing split is larger than the MRI training split;
- train on CT and MR images, evaluate en CT and MR images.

We conduct this experiment for both resolutions of SCONet and for the discrete and implicit baselines introduced in Chapter 2, Section II.3.

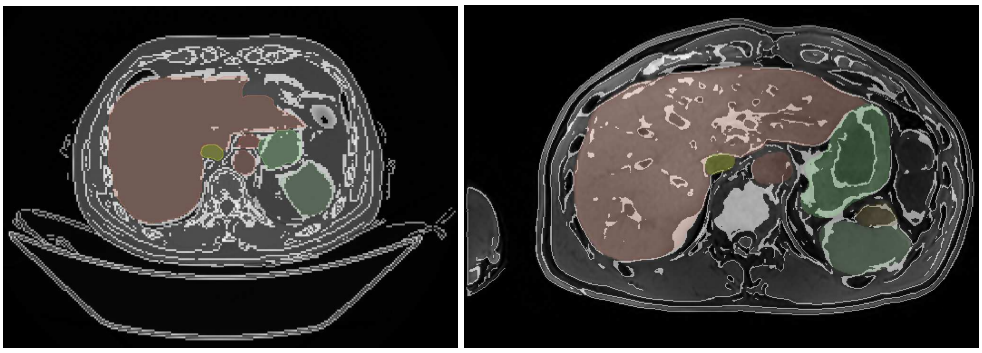
Dataset We perform this experiment on the AMOS dataset since it contains unpaired CT and MR images. More precisely, the authors provide annotations for 300 CT images and 60 MR images.

We set the train/val/test splits for these experiments to 180/20/100 for the CT subset (similarly to the previous studies) and 36/4/20 for the MRI subset. We apply a pre-processing step similar to that used on the CT images: the MR images are resampled to $(256 \times 256 \times z)$ so that we can jointly train the models on both modalities. Additionally, we notice that MR images usually have narrower fields of view compared to CT images and consequently do not contain

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

the bladder or the prostate/uterus. We therefore decide to remove these labels from the experiments in order to avoid possible issues that could stem from missing organs in the entire MRI dataset.

Contour edge detection for MRI images In our approach, the extraction of the point cloud from the original volume requires to generate the edge map with the Canny contour detection algorithm (Canny, 1986). While we used the same set of parameters to generate contour maps for the AbdomenCT-1K dataset and for the CT split of the AMOS dataset, we had to determine a new set of parameters for the MRI images. Similarly to what we explained in Chapter 2, Section I.1 we found these values empirically, by visually ensuring that the generated maps contained a high enough number of contour voxels located near the surface of target organs while minimizing excessive noise. On a side note, we notice that the difference in contrast between CT and MRI images leads to a slight difference in characteristics between the corresponding edge maps. This is illustrated in Figure 4.4 where the contours of some organs' boundaries are more pronounced in the MRI volume. For example missing contour points close to the surface of the liver (in brown) in the CT image are visible in the MRI image.



(a) *CT* **(b)** *MRI*

Figure 4.4: *Slices from two samples of the AMOS datasets. This figure shows the original slice extracted from the volume, the reference segmentation map and the detected contour points (in white).*

II.3 Results

Segmentation performance on the CT and MRI splits are reported in Table 4.3 and Table 4.4 respectively. We also include a boxplot of the the average Dice scores in Figure 4.5 in order to provide a better visualization of the distribution of the results for the different models and learning configurations.

Training on CT Results of the modality shift for which we used the models trained on CT images to generate the segmentation maps of the MRI images show a large drop in performance for all organs and for both metrics. We however also notice that the decrease in Dice score is smaller for SwinUNETR-v2 that has a 0.225 drop and even more for both SCONet configurations (-0.175 for $r = 32$ and -0.192 for $r = 64$).

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Training on MRI When training and evaluating the models exclusively on MRI volumes, most baseline models produce segmentation maps with a much lower accuracy compared to the main CT-based experiments. SwinUNETR-V2, in particular, performs poorly, struggling to effectively learn the smaller and more challenging organs. In contrast, SCONet achieves significantly higher Dice scores, with the largest grid size configuration yielding the best results.

Training on CT and MRI In the joint training configuration, evaluation of the trained models on the MRI split reported in Table 4.4 show an improved segmentation accuracy, with the best performance achieved by SCONet (64^3) and SwinUNETR-V2. As for the evaluation of the CT images reported in Table 4.3, we find that all models except SCONet experience a decline in Dice score. Observation of the boxplot whiskers also show that SCONet produces fewer outlier volumes with poor segmentation results. However, we also note that the discrete baselines still outperform our approach.

II.4 Discussion

The results from the three experiments confirm that SCONet outperforms the baseline models in terms of robustness. Specifically, its performance drop is less pronounced than that of the other models when faced with a modality shift. From the MRI-only experiment where SCONet achieves the highest performance despite being trained on a limited number of images, we conclude that our approach is much less data-hungry than the voxel-based methods. Additionally, incorporating MRI images into the CT training set does not disrupt SCONet's performance, unlike the other models. As previously discussed, we attribute this enhanced robustness not only to SCONet's implicit MLP decoder but also to its use of SURF features, which are known for their strong robustness. This could make our approach more suitable when working from multiple image modalities.

However, one of the main limitations of the experiments conducted in this section is the size of the available MRI subset, which is much smaller than the CT subset. This imbalance likely explains the poor segmentation performance of the compared baselines, especially for SwinUNETR-V2. As a transformer-based architecture, SwinUNETR-V2 typically requires large amounts of training data to learn how to capture long-range dependencies with the self-attention mechanism. Regarding the improvement in Dice score when CT volumes are included, this gain is probably more attributable to the increased size of the training set than to the models effectively learning complementary information from both modalities. In would be relevant for future works to explore further this subject by using a dataset with a higher number of MRI volumes, such as TotalSegmentator (D'Antonoli et al., 2024) which provides annotations of most major anatomical structures for 616 MRI volumes.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Model	Train. set	Dice														HD
		SPL	*RKI*	*LKI*	*GBL*	*ESO*	*LIV*	*STO*	*AOR*	*IVC*	*PAN*	*RAG*	*LAG*	*DUO*	AVG	AVG
2D U-Net	CT	0.926	0.918	0.937	0.701	0.776	0.945	0.806	0.915	0.813	0.745	0.679	0.679	0.665	0.808	69.89
	CT+MRI	0.913	0.913	0.930	0.687	0.772	0.939	0.784	0.908	0.797	0.720	0.648	0.614	0.641	0.790	64.20
3D U-Net	CT	0.931	0.922	0.940	0.718	0.781	0.953	0.838	0.925	0.851	0.782	0.703	0.687	0.708	0.826*	45.68
	CT+MRI	0.931	0.911	0.940	0.701	0.770	0.941	0.822	0.918	0.846	0.763	0.680	0.689	0.685	0.815†	51.97
SwinUNETR-V2	CT	0.840	0.921	0.924	0.693	0.759	0.900	0.791	0.898	0.852	0.804	0.695	0.669	0.710	0.804	176.93
	CT+MRI	0.899	0.919	0.887	0.706	0.733	0.897	0.703	0.850	0.807	0.729	0.478	0.631	0.685	0.763	193.69
ImPulSe	CT	0.883	0.856	0.868	0.580	0.528	0.925	0.772	0.826	0.720	0.655	0.278	0.325	0.532	0.673	33.98
	CT+MRI	0.891	0.861	0.874	0.555	0.529	0.921	0.773	0.824	0.726	0.663	0.209	0.278	0.537	0.665	33.54
OSS-Net	CT	0.701	0.647	0.691	0.033	0.000	0.816	0.478	0.357	0.276	0.291	0.000	0.000	0.067	0.335	47.39
	CT+MRI	0.637	0.579	0.638	0.070	0.000	0.796	0.464	0.346	0.267	0.196	0.000	0.000	0.016	0.308	52.35
SCONet (32^3)	CT	0.884	0.849	0.870	0.577	0.588	0.924	0.771	0.795	0.733	0.667	0.488	0.437	0.557	0.703	24.74*
	CT+MRI	0.887	0.859	0.866	0.569	0.583	0.927	0.783	0.798	0.729	0.672	0.490	0.435	0.575	0.706	25.97†
SCONet (64^3)	CT	0.909	0.880	0.905	0.623	0.640	0.938	0.804	0.848	0.776	0.717	0.523	0.493	0.601	0.743	27.29*
	CT+MRI	0.911	0.883	0.904	0.625	0.635	0.940	0.820	0.853	0.788	0.724	0.524	0.481	0.617	0.746	25.82†

Table 4.3: *Evaluation on AMOS CT split. * and † indicate the best average Dice and HD score when the training is performed on CT and CT+MRI respectively, according to the Wilcoxon test.*

Model	Train. set	Dice														HD
		SPL	*RKI*	*LKI*	*GBL*	*ESO*	*LIV*	*STO*	*AOR*	*IVC*	*PAN*	*RAG*	*LAG*	*DUO*	AVG	AVG
2D U-Net	MRI	0.887	0.905	0.910	0.372	0.605	0.922	0.713	0.363	0.563	0.644	0.457	0.411	0.421	0.629	74.19
	CT	0.301	0.223	0.240	0.070	0.174	0.560	0.086	0.248	0.123	0.110	0.000	0.015	0.105	0.174	146.52
	CT+MRI	0.896	0.923	0.912	0.540	0.715	0.913	0.686	0.869	0.723	0.682	0.458	0.519	0.482	0.717	70.35
3D U-Net	MRI	0.929	0.934	0.926	0.590	0.088	0.928	0.727	0.903	0.821	0.718	0.051	0.530	0.158	0.639	44.88
	CT	0.192	0.212	0.226	0.067	0.199	0.329	0.064	0.263	0.187	0.091	0.000	0.000	0.044	0.144	118.78
	CT+MRI	0.913	0.902	0.903	0.580	0.679	0.932	0.763	0.909	0.815	0.746	0.548	0.585	0.543	0.755	46.45
SwinUNETR-V2	MRI	0.888	0.850	0.855	0.050	0.000	0.915	0.690	0.010	0.000	0.567	0.000	0.000	0.000	0.371	90.82
	CT	0.810	0.766	0.783	0.334	0.517	0.858	0.579	0.684	0.612	0.595	0.179	0.320	0.489	0.579*	95.95
	CT+MRI	0.935	0.936	0.936	0.657	0.668	0.952	0.843	0.877	0.810	0.777	0.518	0.564	0.630	0.777†	25.14
ImPulSe	MRI	0.904	0.880	0.861	0.404	0.393	0.925	0.672	0.817	0.673	0.565	0.000	0.000	0.335	0.571	36.30
	CT	0.000	0.000	0.000	0.050	0.000	0.000	0.000	0.154	0.000	0.000	0.000	0.000	0.000	0.016	157.50
	CT+MRI	0.563	0.677	0.487	0.296	0.253	0.332	0.339	0.682	0.451	0.315	0.089	0.014	0.150	0.358	136.84
OSS-Net	MRI	0.365	0.454	0.470	0.050	0.000	0.720	0.000	0.025	0.000	0.000	0.000	0.000	0.000	0.160	88.17
	CT	0.000	0.000	0.000	0.050	0.000	0.024	0.033	0.015	0.000	0.000	0.000	0.000	0.000	0.009	109.87
	CT+MRI	0.731	0.745	0.691	0.090	0.007	0.863	0.421	0.575	0.384	0.394	0.000	0.000	0.108	0.385	44.52
SCONet (32^3)	MRI	0.877	0.873	0.857	0.383	0.573	0.925	0.720	0.792	0.725	0.677	0.377	0.386	0.518	0.668	21.48‡
	CT	0.773	0.760	0.792	0.243	0.328	0.798	0.596	0.550	0.506	0.564	0.318	0.341	0.301	0.528*	59.53*
	CT+MRI	0.896	0.888	0.883	0.523	0.623	0.937	0.789	0.818	0.745	0.707	0.457	0.443	0.556	0.713	20.05†
SCONet (64^3)	MRI	0.922	0.920	0.922	0.567	0.626	0.949	0.816	0.883	0.785	0.764	0.459	0.404	0.548	0.736‡	22.57‡
	CT	0.802	0.741	0.718	0.354	0.566	0.815	0.675	0.503	0.516	0.572	0.239	0.280	0.375	0.551*	77.48
	CT+MRI	0.932	0.939	0.938	0.684	0.666	0.955	0.855	0.892	0.813	0.783	0.503	0.506	0.606	0.775†	19.08†

Table 4.4: *Evaluation on AMOS MRI split. ‡, * and † indicate the best average Dice and HD score when training is performed on MRI, CT and CT+MRI respectively, according to the Wilcoxon test.*

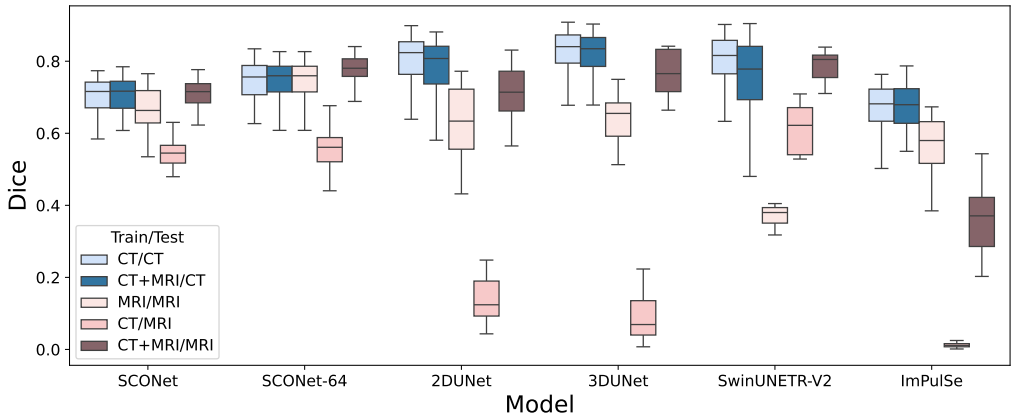


Figure 4.5: *Average Dice for cross-modality learning. Performance of OSS-Net is much lower than that of the other models and is not included for better readability.*

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

III Conclusion

In this chapter we explored the robustness of our workflow to domain shifts. In particular we found that our approach is more robust to dataset shifts than the voxel-based methods. Our experiments on cross-modality learning also showed how its robustness could be an advantage to learn from images from multiple modalities.

Ultimately, SCONet's robustness and its tendency to produce overly smooth segmentation maps both stem from the implicit MLP decoder's bias toward lower frequencies. In future works, it would be valuable to explore methods for enhancing high-frequency learning while preserving this robustness.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Conclusion and perspectives

General conclusion

Our goal in this work was to explore the use of unstructured 3D representations for accurate and efficient medical image segmentation. We notably aimed to design a novel segmentation workflow which extracts a point cloud from the original medical 3D image and uses it to generate a segmentation map.

After introducing the context of our work and our objectives, we provided in Chapter 1 an overview of the state of the art related to both the task of medical image segmentation and the use of alternative 3D data representations such as point clouds and implicit representations.

We then introduced in Chapter 2 the main contribution of our work: a new medical image segmentation workflow which uses point clouds as an intermediate representation. Our workflow consists in two main steps which are the point cloud extraction and the generation of a segmentation map using a deep neural network. More precisely, we proposed to extract the point cloud from a contour map computed with Canny which is a standard contour detection algorithm, and to then enrich it with SURF features computed from the volume. For the second step, we introduced SCONet, an INR-based network adapted from ConvONet (Peng et al., 2020) to generate a multi-label segmentation map from the enriched contour point cloud. Thanks to its implicit decoder, SCONet can predict the segmentation label for any point in the 3D space, enabling it to output segmentation maps at arbitrary resolutions while maintaining a constant memory cost. We evaluated the proposed workflow on two abdominal CT datasets and showed that our approach produces accurate segmentation maps for a reduced computation cost compared to a set of volume-based methods traditionally used for this task. This chapter also included additional experiments: for example the MISE algorithm (Mescheder et al., 2019) was applied to reduce the number of query operations necessary to generate a segmentation map. We showed that this approach could be useful to reduce inference time in low-resource settings where there are no GPUs available. We also investigated the use of random crops not only to correct artifacts that appeared on some images but also to generally improve the quality of predicted segmentation maps.

Following studies in Chapter 3 focused on the point cloud representation. We notably investigated various sampling strategies and identified the optimal size for the contour point cloud input to SCONet, balancing segmentation accuracy and processing time. This chapter also explored the influence of the quality of the point cloud representation on the accuracy of the segmentation maps generated by SCONet. We notably found that points closer to the

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

surface of the target organs are generally more useful for the segmentation task although identifying these points can be tricky without using the reference segmentation map.

Finally, we investigated the robustness of our workflow in Chapter 4. In particular, we showed that our method is more robust than volume-based models when confronted to dataset shifts. We also proposed an experiment on cross-modality learning where we found that SCONet's ability to learn low-frequency features may help it effectively handle images from multiple modalities, such as CT and MRI. This last experiment also showed that our workflow is less data-hungry than the voxel-based baselines which can be a noteworthy advantage in contexts where the number of training images is limited.

Limitations

Although our workflow produces overall accurate segmentation maps, our learning algorithm actually struggles to learn high frequencies, generally resulting in overly smooth predicted surfaces. This notably affects smaller organs and structures whose shape varies across a dataset. We identify three characteristics of our workflow which hinder its capacity to effectively output detailed segmentation maps.

Quality of the contour point cloud In the proposed workflow, the point cloud processed by SCONet is extracted from the contour map and therefore contains uninformative points located in non-target structures or resulting from noise. The presence of these points can disturb the network's performance, all the more so since they can take the place of more informative points located near the target organ's surfaces during the sampling process. This notably impacts the PointNet module from the SCONet encoder, which fails to compute complex per-point geometric features from a noisy contour point cloud, as shown in Chapter 2, Section V.3 and Chapter 3, Section II.3. Additionally, another issue stems from a lack of points in regions where the image contrast is too low for any present contours to be detected. This becomes particularly problematic when such regions are near the boundaries of target structures, since the network is then forced to infer or "hallucinate" organ surfaces in the absence of reliable data. We observe that both the extra points and the missing points are consequences of the use of the standard Canny algorithm with empirically-chosen parameters to generate the contour map.

While we showed in Chapter 3 that prioritizing contour points that are closer to the surfaces could potentially improve the performance of SCONet, we failed to identify such points without relying on the reference segmentation maps. Besides, this strategy would not solve the issue of missing points.

Aggregation in a low-resolution grid In the SCONet encoder, points from the input point cloud are first aggregated into a grid before being processed by the UNet module. During this step, all features of points projected within the same cell are averaged. This aggregation and averaging prevents our network from distinguishing between closely located points. In addition, the latent vector fed into the decoder is computed through linear interpolation of the

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

UNet-learned latent grid, which inherently limits the capacity of the decoder to output high frequencies. This is all the more relevant given that the feature grid has a low resolution (32^3) compared to the resolution of the point cloud extracted from the 256^3 edge map.

Increasing the resolution of the latent grid or learning on point cloud crops can help mitigate this issue to a certain extent as shown in Chapter 2, although both solutions also exhibit a larger memory footprint and/or a higher processing time.

Implicit networks and low frequencies Lastly, we showed in Chapter 3 that SCONet's representation capacity is still limited even when increasing the resolution of the latent grid and when processing a point cloud composed exclusively of surface points. The performance of our method indeed remained equivalent or even inferior to that of the discrete baselines, although we used perfect points. We explain this limitation by the fact that SCONet's decoder is an MLP, a type of neural network which tends to favor low frequencies during learning. This is a well-documented inherent property of implicit networks (Tancik et al., 2020).

Perspectives

The limitations discussed in the previous section highlight several promising avenues for future research.

Generalization to other datasets Experiments in this thesis mainly studied one modality (CT, with one study done on MRI) and one anatomical region with a focus on abdominal organs. In the short term, it would be therefore interesting to apply our method to other segmentation tasks. For example, tumor segmentation (for example brain tumor segmentation of MR images) presents several challenges. The variability in shape and location of the tumors could notably pose difficulties for our approach. Another challenging application could be ultrasound image segmentation. Here, contour detection might produce highly noisy edge maps due to speckle artifacts, further complicating the downstream segmentation process.

Learning-based sampling One of the main limitations of our approach is SCONet's difficulty to reconstruct highly detailed segmentation maps from noisy contour point clouds. However, we identified a subset of contour points that are more informative for this task. Future works could explore automatically selecting these points during processing with SCONet. One potential solution could be to replace the PointNet module by a learnable sampling module. We could for example draw inspiration from score-based methods like Nezhadarya et al. (2020) and design a module which computes per-point weights that indicate the points' relevance for downstream processing. It would be particularly insightful to investigate whether the learned point cloud extracted from the set of contour points aligns with the foreground contour point cloud described in Chapter 3.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Learning high frequencies One of the factors that explain the robustness of our method is its MLP-based decoder. Indeed its tendency to favor low frequencies makes it more robust to changes in data distribution such as dataset shifts. This property is both a blessing and a curse because it is also one of the reasons why SCONet generates overly-smooth segmentation maps. In future works, it would be relevant to explore the methods proposed in the literature to improve the capacities of implicit methods to learn high frequencies. One existing solution is for example to use a Fourier features mapping of the query points' coordinates (Tancik et al., 2020). Another potential approach could be to incorporate to each query point additional local context that may help the network generate higher-frequency details.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

List of Publications

International conference with proceedings

Maylis Jouvencel, Razmig Kéchichian, Julie Digne, Sébastien Valette (2025) “SCONet: Convolutional Occupancy Networks for Multi-Organ Segmentation”. In: *2025 IEEE 22nd International Symposium on Biomedical Imaging (ISBI) [Oral presentation]*

National conference with proceedings

Maylis Jouvencel, Razmig Kéchichian, Julie Digne, Sébastien Valette (2025) “SCONet: Réseaux d’Occupation Convolutifs pour la Segmentation Multi-Organes”. In: *30° Colloque sur le traitement du signal et des images*. 2025-1664. Strasbourg: GRETSI - Groupe de Recherche en Traitement du Signal et des Images, p. 841–844. *[Poster]*

National conference

Maylis Jouvencel, Razmig Kéchichian, Sébastien Valette (2024) ”Des réseaux d’occupation convolutifs pour la segmentation multi-organes”. In *Colloque Français d’Intelligence Artificielle en Imagerie Biomédicale (IABM 2024) [Poster]*

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Bibliography

Agier et al., 2016: Agier, Rémi, Sébastien Valette, Laurent Fanton, Pierre Croisille, et al. (2016). “Hubless 3D medical image bundle registration”. In: *VISAPP 2016 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. Vol. 3, pp. 265–272.

Amiranashvili et al., 2022: Amiranashvili, Tamaz, David Lüdke, Hongwei Bran Li, Bjoern Menze, et al. (2022). “Learning shape reconstruction from sparse measurements with neural implicit functions”. In: *International Conference on Medical Imaging with Deep Learning*. PMLR, pp. 22–34.

Ba et al., 2014: Ba, Jimmy and Rich Caruana (2014). “Do deep nets really need to be deep?” In: *Advances in neural information processing systems* 27.

Bahdanau et al., 2014: Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473*.

Bay et al., 2006: Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool (2006). “Surf: Speeded up robust features”. In: *European conference on computer vision*. Springer, pp. 404–417.

Biratu et al., 2021: Biratu, Erena Siyoum, Friedhelm Schwenker, Taye Girma Debelee, Samuel Rahimeto Kebede, et al. (2021). “Enhanced region growing for brain tumor MR image segmentation”. In: *Journal of Imaging* 7.2, p. 22.

Bommasani, 2021: Bommasani, Rishi (2021). “On the opportunities and risks of foundation models”. In: *arXiv preprint arXiv:2108.07258*.

Bralet et al., 2021: Bralet, Antoine, Razmig Kéchichian, and Sebastien Valette (2021). “Local surf-based keypoint transfer segmentation”. In: *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*. IEEE, pp. 1390–1393.

Buckley et al., 2021: Buckley, Bryan W and Peter J MacMahon (2021). “Radiology and the climate crisis: opportunities and challenges—radiology in training”. In: *Radiology* 300.3, E339–E341.

Canny, 1986: Canny, John (1986). “A Computational Approach to Edge Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6, pp. 679–698. DOI: 10.1109/TPAMI.1986.4767851.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Chabra et al., 2020: Chabra, Rohan, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, et al. (2020). "Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction". In: *European Conference on Computer Vision*. URL: https://api.semanticscholar.org/CorpusID:214623267.

J. Chen et al., 2024: Chen, Junren, Rui Chen, Wei Wang, Junlong Cheng, et al. (2024). "Tinyu-net: Lighter yet better u-net with cascaded multi-receptive fields". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 626–635.

Christ et al., 2011: Christ, M. C. Jobin and R. M. S. Parvathi (2011). "Fuzzy c-means algorithm for medical image segmentation". In: *2011 3rd International Conference on Electronics Computer Technology*. Vol. 4, pp. 33–36. DOI: 10.1109/ICECTECH.2011.5941851.

Çiçek et al., 2016: Çiçek, Özgün, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, et al. (2016). "3D U-Net: learning dense volumetric segmentation from sparse annotation". In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*. Springer, pp. 424–432.

D'Antonoli et al., 2024: D'Antonoli, Tugba Akinci, Lucas K Berger, Ashraya K Indrakanti, Nathan Vishwanathan, et al. (2024). "TotalSegmentator MRI: Robust sequence-independent segmentation of multiple anatomic structures in MRI". In: *arXiv preprint arXiv:2405.19492*.

DeTone et al., 2018: DeTone, Daniel, Tomasz Malisiewicz, and Andrew Rabinovich (2018). "Superpoint: Self-supervised interest point detection and description". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 224–236.

Dice, 1945: Dice, Lee R (1945). "Measures of the amount of ecologic association between species". In: *Ecology* 26.3, pp. 297–302.

Doo et al., 2024: Doo, Florence X, Jan Vosshenrich, Tessa S Cook, Linda Moy, et al. (2024). "Environmental sustainability and AI in radiology: a double-edged sword". In: *Radiology* 310.2, e232030.

Dosovitskiy et al., 2020: Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, et al. (2020). "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929*.

Dovrat et al., 2019: Dovrat, Oren, Itai Lang, and Shai Avidan (2019). "Learning to sample". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2760–2769.

Dutt et al., 2023: Dutt, Raman, Linus Ericsson, Pedro Sanchez, Sotirios A Tsaftaris, et al. (2023). "Parameter-efficient fine-tuning for medical image analysis: The missed opportunity". In: *arXiv preprint arXiv:2305.08252*.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

FDA, 2017: FDA (Sept. 2017). *MRI (Magnetic Resonance Imaging): Benefits and risks*. en. https://www.fda.gov/radiation-emitting-products/mri-magnetic-resonance-imaging/benefits-and-risks. Accessed: 2025-10-30.

Graham, 2015: Graham, Benjamin (2015). "Sparse 3D convolutional neural networks". In: *ArXiv* abs/1505.02890. URL: https://api.semanticscholar.org/CorpusID:13884358.

Graham et al., 2018: Graham, Benjamin, Martin Engelcke, and Laurens van der Maaten (2018). "3D Semantic Segmentation with Submanifold Sparse Convolutional Networks". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9224–9232. DOI: 10.1109/CVPR.2018.00961.

Groh et al., 2018: Groh, Fabian, Patrick Wieschollek, and Hendrik PA Lensch (2018). "Flex-convolution: Million-scale point-cloud learning beyond grid-worlds". In: *Asian Conference on Computer Vision*. Springer, pp. 105–122.

Guan et al., 2022: Guan, Hao and Mingxia Liu (Feb. 2022). "Domain Adaptation for Medical Image Analysis: A Survey". en. In: *IEEE Trans Biomed Eng* 69.3, pp. 1173–1185.

Guidi et al., 2024: Guidi, Gianluca, Francesca Dominici, Jonathan Gilmour, Kevin Butler, et al. (2024). "Environmental burden of United States data centers in the artificial intelligence era". In: *arXiv preprint arXiv:2411.09786*.

Han et al., 2023: Han, Meng, Liang Wang, Limin Xiao, Hao Zhang, et al. (2023). "QuickFPS: Architecture and Algorithm Co-Design for Farthest Point Sampling in Large-Scale Point Clouds". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

Hao et al., 2024: Hao, Zhaoquan, Hongyan Quan, and Yinbin Lu (2024). "Emf-former: An efficient and memory-friendly transformer for medical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 231–241.

Harris et al., 1988: Harris, Chris, Mike Stephens, et al. (1988). "A combined corner and edge detector". In: *Alvey vision conference*. Vol. 15. 50. Manchester, UK, pp. 10–5244.

Hassaballah et al., 2018: Hassaballah, Mahmoud, Hammam A Alshazly, and Abdelmgeid A Ali (2018). "Analysis and evaluation of keypoint descriptors for image matching". In: *Recent Advances in Computer Vision: Theories and Applications*. Springer, pp. 113–140.

J. He et al., 2020: He, Jiafa, Chengwei Pan, Can Yang, Ming Zhang, et al. (2020). "Learning hybrid representations for automatic 3D vessel centerline extraction". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VI 23*. Springer, pp. 24–34.

K. He et al., 2016: He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Y. He et al., 2023: He, Yufan, Vishwesh Nath, Dong Yang, Yucheng Tang, et al. (2023). “Swinunetr-v2: Stronger swin transformers with stagewise convolutions for 3d medical image segmentation”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 416–426.

M. P. Heinrich, 2024: Heinrich, Mattias Paul (2024). “PointVoxelFormer–Reviving point cloud networks for 3D medical imaging”. In: *arXiv preprint arXiv:2412.17390*.

Ho et al., 2021: Ho, Ngoc-Vuong, Tan Nguyen, Gia-Han Diep, Ngan Le, et al. (2021). “Point-Unet: A Context-Aware Point-Based Neural Network for Volumetric Segmentation”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 644–655.

Howard et al., 2017: Howard, Andrew G, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, et al. (2017). “Mobilenets: Efficient convolutional neural networks for mobile vision applications”. In: *arXiv preprint arXiv:1704.04861*.

E. J. Hu et al., 2022: Hu, Edward J, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, et al. (2022). “Lora: Low-rank adaptation of large language models.” In: *ICLR* 1.2, p. 3.

Q. Hu et al., 2020: Hu, Qingyong, Bo Yang, Linhai Xie, Stefano Rosa, et al. (2020). “RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Huttenlocher et al., 1993: Huttenlocher, D.P., G.A. Klanderman, and W.J. Rucklidge (1993). “Comparing images using the Hausdorff distance”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.9, pp. 850–863. DOI: 10.1109/34.232073.

Isensee et al., 2021: Isensee, Fabian, Paul F Jaeger, Simon AA Kohl, Jens Petersen, et al. (2021). “nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation”. In: *Nature methods* 18.2, pp. 203–211.

Jardim et al., 2023: Jardim, Sandra, João António, and Carlos Mora (2023). “Image thresholding approaches for medical image segmentation-short literature review”. In: *Procedia Computer Science* 219, pp. 1485–1492.

Ji et al., 2022: Ji, Yuanfeng, Haotian Bai, Chongjian Ge, Jie Yang, et al. (2022). “Amos: A large-scale abdominal multi-organ benchmark for versatile medical image segmentation”. In: *Advances in neural information processing systems* 35, pp. 36722–36732.

Jimenez-del-Toro et al., 2016: Jimenez-del-Toro, Oscar, Henning Müller, Markus Krenn, Katharina Gruenberg, et al. (2016). “Cloud-based evaluation of anatomical structure segmentation and landmark detection algorithms: VISCERAL anatomy benchmarks”. In: *IEEE transactions on medical imaging* 35.11, pp. 2459–2475.

Jouvencel et al., 2025a: Jouvencel, Maylis, Razmig Kéchichian, Julie Digne, and Sébastien Valette (2025a). “SCONet: Convolutional Occupancy Networks for Multi-Organ Segmentation”. In: *2025 IEEE 22nd International Symposium on Biomedical Imaging (ISBI)*. IEEE, pp. 1–5.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Jouvencel et al., 2025b: Jouvencel, Maylis, Razmig Kéchichian, Julie Digne, and Sébastien Valette (Aug. 2025b). “SCONet: Réseaux d’Occupation Convolutifs pour la Segmentation Multi-Organes”. In: *30° Colloque sur le traitement du signal et des images*. 2025-1664. Strasbourg: GRETSI - Groupe de Recherche en Traitement du Signal et des Images, p. 841–844.

Kavur et al., 2021: Kavur, A. Emre, N. Sinem Gezer, Mustafa Barış, Sinem Aslan, et al. (Apr. 2021). “CHAOS Challenge - combined (CT-MR) healthy abdominal organ segmentation”. In: *Medical Image Analysis* 69, p. 101950. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2020.101950. URL: http://www.sciencedirect.com/science/article/pii/S1361841520303145.

Kéchichian et al., 2018: Kéchichian, Razmig, Sébastien Valette, and Michel Desvignes (2018). “Automatic multiorgan segmentation via multiscale registration and graph cut”. In: *IEEE transactions on medical imaging* 37.12, pp. 2739–2749.

Khan et al., 2022: Khan, Salman, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, et al. (2022). “Transformers in vision: A survey”. In: *ACM computing surveys (CSUR)* 54.10s, pp. 1–41.

Kirillov et al., 2023: Kirillov, Alexander, Eric Mintun, Nikhila Ravi, Hanzi Mao, et al. (2023). “Segment anything”. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026.

Krizhevsky et al., 2012: Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25.

Kuang et al., 2022: Kuang, Kaiming, Li Zhang, Jingyu Li, Hongwei Li, et al. (2022). “What makes for automatic reconstruction of pulmonary segments”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 495–505.

Kumar BG et al., 2016: Kumar BG, Vijay, Gustavo Carneiro, and Ian Reid (2016). “Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5385–5394.

Landman et al., 2015: Landman, Bennett, Zhoubing Xu, Juan Igelsias, Martin Styner, et al. (2015). “Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge”. In: *Proc. MICCAI multi-atlas labeling beyond cranial vault—workshop challenge*. Vol. 5. Munich, Germany, p. 12.

Lang et al., 2020: Lang, Itai, Asaf Manor, and Shai Avidan (2020). “Samplenet: Differentiable point cloud sampling”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7578–7588.

LeCun et al., 1989: LeCun, Yann, Bernhard Boser, John S Denker, Donnie Henderson, et al. (1989). “Backpropagation applied to handwritten zip code recognition”. In: *Neural computation* 1.4, pp. 541–551.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Liang et al., 2021: Liang, Tailin, John Glossner, Lei Wang, Shaobo Shi, et al. (2021). “Pruning and quantization for deep neural network acceleration: A survey”. In: *Neurocomputing* 461, pp. 370–403.

X. Liu et al., 2023: Liu, Xingyi and Keshab K Parhi (2023). “Tensor decomposition for model reduction in neural networks: A review [feature]”. In: *IEEE Circuits and Systems Magazine* 23.2, pp. 8–28.

Ze Liu et al., 2021: Liu, Ze, Yutong Lin, Yue Cao, Han Hu, et al. (2021). “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022.

Zhijian Liu et al., 2019: Liu, Zhijian, Haotian Tang, Yujun Lin, and Song Han (2019). “Point-Voxel CNN for Efficient 3D Deep Learning”. In: *Advances in Neural Information Processing Systems*.

Loiseau–Witon et al., 2022: Loiseau–Witon, Nicolas, Razmig Kéchichian, Sebastien Valette, and Adrien Bartoli (2022). “Learning 3D medical image keypoint descriptors with the triplet loss”. In: *International Journal of Computer Assisted Radiology and Surgery* 17.1, pp. 141–146.

Long et al., 2015: Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Lorensen et al., 1998: Lorensen, William E and Harvey E Cline (1998). “Marching cubes: A high resolution 3D surface construction algorithm”. In: *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353.

Lowe, 2004: Lowe, David G (2004). “Distinctive image features from scale-invariant keypoints”. In: *International journal of computer vision* 60.2, pp. 91–110.

Luccioni et al., 2025: Luccioni, Alexandra Sasha, Emma Strubell, and Kate Crawford (2025). “From efficiency gains to rebound effects: The problem of jevons’ paradox in AI’s polarized environmental debate”. In: *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, pp. 76–88.

J. Ma et al., 2022a: Ma, Jun, Yao Zhang, Song Gu, Xingle An, et al. (2022a). “Fast and low-GPU-memory abdomen CT organ segmentation: the flare challenge”. In: *Medical Image Analysis* 82, p. 102616.

J. Ma et al., 2022b: Ma, Jun, Yao Zhang, Song Gu, Cheng Zhu, et al. (2022b). “AbdomenCT-1K: Is Abdominal Organ Segmentation a Solved Problem?” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10, pp. 6695–6714. DOI: 10.1109/TPAMI.2021.3100536.

N. Ma et al., 2018: Ma, Ningning, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun (2018). “Shufflenet v2: Practical guidelines for efficient cnn architecture design”. In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Maier-Hein et al., 2024: Maier-Hein, Lena, Annika Reinke, Patrick Godau, Minu D Tizabi, et al. (2024). “Metrics reloaded: recommendations for image analysis validation”. In: *Nature methods* 21.2, pp. 195–212.

Mescheder et al., 2019: Mescheder, Lars, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, et al. (2019). “Occupancy networks: Learning 3d reconstruction in function space”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4460–4470.

Metaxas, 2012: Metaxas, Dimitris N (2012). *Physics-based deformable models: applications to computer vision, graphics and medical imaging*. Vol. 389. Springer Science & Business Media.

Mildenhall et al., 2021: Mildenhall, Ben, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, et al. (2021). “Nerf: Representing scenes as neural radiance fields for view synthesis”. In: *Communications of the ACM* 65.1, pp. 99–106.

Milletari et al., 2016: Milletari, Fausto, Nassir Navab, and Seyed-Ahmad Ahmadi (2016). “V-net: Fully convolutional neural networks for volumetric medical image segmentation”. In: *2016 fourth international conference on 3D vision (3DV)*. Ieee, pp. 565–571.

Nakayama et al., 2006: Nakayama, Yoshiharu, Qiang Li, Shigehiko Katsuragawa, Ryuji Ikeda, et al. (Sept. 2006). “Automated hepatic volumetry for living related liver transplantation at multisection CT”. eng. In: *Radiology* 240.3, pp. 743–748. ISSN: 0033-8419. DOI: 10.1148/radiol.2403050850.

Nezhadarya et al., 2020: Nezhadarya, Ehsan, Ehsan Taghavi, Ryan Razani, Bingbing Liu, et al. (2020). “Adaptive hierarchical down-sampling for point cloud classification”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12956–12964.

Ng et al., 2006: Ng, H.P., S.H. Ong, K.W.C. Foong, P.S. Goh, et al. (2006). “Medical Image Segmentation Using K-Means Clustering and Improved Watershed Algorithm”. In: *2006 IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 61–65. DOI: 10.1109/SSIAI.2006.1633722.

NHS, 2020: NHS, England (Dec. 2020). *Diagnostics: recovery and renewal–report of the independent review of diagnostic services for NHS England, 2020*. https://www.england.nhs.uk/publication/diagnostics-recovery-and-renewal-report-of-the-independent-review-of-diagnostic-services-for-nhs-england.

Niemeyer et al., 2020: Niemeyer, Michael, Lars Mescheder, Michael Oechsle, and Andreas Geiger (2020). “Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3504–3515.

Oktay et al., 2018: Oktay, Ozan, Jo Schlemper, Loic Le Folgoc, Matthew Lee, et al. (2018). “Attention u-net: Learning where to look for the pancreas”. In: *arXiv preprint arXiv:1804.03999*.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Park et al., 2019: Park, Jeong Joon, Peter Florence, Julian Straub, Richard Newcombe, et al. (2019). “DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Paszke et al., 2019: Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, et al. (2019). “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32.

Peng et al., 2020: Peng, Songyou, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, et al. (2020). “Convolutional occupancy networks”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, pp. 523–540.

Perez et al., 2022: Perez, Alberto A., Victoria Noe-Kim, Meghan G. Lubner, Peter M. Graffy, et al. (Feb. 2022). “Deep Learning CT-based Quantitative Visualization Tool for Liver Volume Estimation: Defining Normal and Hepatomegaly”. In: *Radiology* 302.2, pp. 336–342. ISSN: 0033-8419. DOI: 10.1148/radiol.2021210531.

Pham et al., 2018: Pham, Hieu, Melody Guan, Barret Zoph, Quoc Le, et al. (2018). “Efficient neural architecture search via parameters sharing”. In: *International conference on machine learning*. PMLR, pp. 4095–4104.

Qi et al., 2016: Qi, Charles R, Hao Su, Kaichun Mo, and Leonidas J Guibas (2016). “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”. In: *arXiv preprint arXiv:1612.00593*.

Qi et al., 2017: Qi, Charles R, Li Yi, Hao Su, and Leonidas J Guibas (2017). “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space”. In: *arXiv preprint arXiv:1706.02413*.

Qian et al., 2022: Qian, Guocheng, Yuchen Li, Houwen Peng, Jinjie Mai, et al. (2022). “PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies”. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Raghu et al., 2021: Raghu, Maithra, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, et al. (2021). “Do vision transformers see like convolutional neural networks?” In: *Advances in neural information processing systems* 34, pp. 12116–12128.

Rahaman et al., 2019: Rahaman, Nasim, Aristide Baratin, Devansh Arpit, Felix Draxler, et al. (2019). “On the spectral bias of neural networks”. In: *International conference on machine learning*. PMLR, pp. 5301–5310.

Reich et al., 2021: Reich, Christoph, Tim Prangemeier, Özdemir Cetin, and Heinz Koeppl (2021). “OSS-Net: Memory Efficient High Resolution Semantic Segmentation of 3D Medical Data”. In: *British Machine Vision Conference (BMVC)*. British Machine Vision Association.

Ronneberger et al., 2015: Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, pp. 234–241.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Ruan et al., 2022: Ruan, Jiacheng, Suncheng Xiang, Mingye Xie, Ting Liu, et al. (2022). “Malunet: A multi-attention and light-weight unet for skin lesion segmentation”. In: *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, pp. 1150–1156.

Selvan et al., 2022: Selvan, Raghavendra, Nikhil Bhagwat, Lasse F Wolff Anthony, Benjamin Kanding, et al. (2022). “Carbon footprint of selecting and training deep learning models for medical image analysis”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 506–516.

Sharma et al., 2010: Sharma, Neeraj and Lalit M Aggarwal (2010). “Automated medical image segmentation techniques”. In: *Journal of medical physics* 35.1, pp. 3–14.

L. Shen et al., 2022: Shen, Liyue, John Pauly, and Lei Xing (2022). “NeRP: implicit neural representation learning with prior embedding for sparsely sampled image reconstruction”. In: *IEEE Transactions on Neural Networks and Learning Systems* 35.1, pp. 770–782.

Simonyan et al., 2014: Simonyan, Karen and Andrew Zisserman (2014). “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556*.

Sitzmann et al., 2020: Sitzmann, Vincent, Julien Martel, Alexander Bergman, David Lindell, et al. (2020). “Implicit neural representations with periodic activation functions”. In: *Advances in neural information processing systems* 33, pp. 7462–7473.

Sobel, 1970: Sobel, Irwin Edward (1970). *Camera models and machine perception*. stanford university.

H. Su et al., 2015: Su, Hang, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller (2015). “Multi-view convolutional neural networks for 3d shape recognition”. In: *Proceedings of the IEEE international conference on computer vision*, pp. 945–953.

Z. Su et al., 2021: Su, Zhuo, Wenzhe Liu, Zitong Yu, Dewen Hu, et al. (2021). “Pixel difference networks for efficient edge detection”. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5117–5127.

Sykes, 2014: Sykes, Jonathan (Sept. 2014). “Reflections on the current status of commercial automated segmentation systems in clinical practice”. In: *Journal of Medical Radiation Sciences* 61.3, pp. 131–134. ISSN: 2051-3895. DOI: 10.1002/jmrs.65.

M. Tan et al., 2019: Tan, Mingxing, Bo Chen, Ruoming Pang, Vijay Vasudevan, et al. (2019). “Mnasnet: Platform-aware neural architecture search for mobile”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2820–2828.

Tancik et al., 2020: Tancik, Matthew, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, et al. (2020). “Fourier features let networks learn high frequency functions in low dimensional domains”. In: *Advances in neural information processing systems* 33, pp. 7537–7547.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Tang et al., 2022: Tang, Yucheng, Dong Yang, Wenqi Li, Holger R Roth, et al. (2022). “Self-supervised pre-training of swin transformers for 3d medical image analysis”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20730–20740.

Thomas et al., 2019: Thomas, Hugues, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, et al. (2019). “KPConv: Flexible and Deformable Convolution for Point Clouds”. In: *Proceedings of the IEEE International Conference on Computer Vision*.

Valanarasu et al., 2022: Valanarasu, Jeya Maria Jose and Vishal M Patel (2022). “Unext: Mlp-based rapid medical image segmentation network”. In: *International conference on medical image computing and computer-assisted intervention*. Springer, pp. 23–33.

Valindria et al., 2018: Valindria, Vanya V, Nick Pawlowski, Martin Rajchl, Ioannis Lavdas, et al. (2018). “Multi-modal learning from unpaired images: Application to multi-organ segmentation in CT and MRI”. In: *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, pp. 547–556.

Vanholder, 2016: Vanholder, Han (2016). “Efficient inference with tensorrt”. In: *GPU Technology Conference*. Vol. 1. 2, pp. 1–24.

Vaswani et al., 2017: Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, et al. (2017). “Attention is all you need”. In: *Advances in neural information processing systems* 30.

Wachinger et al., 2018: Wachinger, Christian, Matthew Toews, Georg Langs, William Wells, et al. (2018). “Keypoint transfer for fast whole-body segmentation”. In: *IEEE transactions on medical imaging* 39.2, pp. 273–282.

T. Wang et al., 2024: Wang, Tao, Jing Wu, Ze Ji, and Yu-Kun Lai (2024). “Sparse convolutional networks for surface reconstruction from noisy point clouds”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3212–3221.

X. Wang et al., 2018: Wang, Xiaolong, Ross Girshick, Abhinav Gupta, and Kaiming He (2018). “Non-local neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803.

Yue Wang et al., 2019: Wang, Yue, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, et al. (2019). “Dynamic Graph CNN for Learning on Point Clouds”. In: *ACM Transactions on Graphics (TOG)*.

Wasserthal et al., 2023: Wasserthal, Jakob, Hanns-Christian Breit, Manfred T Meyer, Maurice Pradella, et al. (2023). “TotalSegmentator: robust segmentation of 104 anatomic structures in CT images”. In: *Radiology: Artificial Intelligence* 5.5, e230024.

Wei et al., 2024: Wei, Xiaobao, Jiajun Cao, Yizhu Jin, Ming Lu, et al. (2024). “I-medsam: Implicit medical image segmentation with segment anything”. In: *European Conference on Computer Vision*. Springer, pp. 90–107.

Wolny et al., 2020: Wolny, Adrian, Lorenzo Cerrone, Athul Vijayan, Rachele Tofanelli, et al. (July 2020). “Accurate and versatile 3D segmentation of plant tissues at cellular resolution”. In: *eLife* 9. Ed. by Christian S Hardtke, Dominique C Bergmann, Dominique C Bergmann, and Moritz

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Graeff, e57613. ISSN: 2050-084X. DOI: 10.7554/eLife.57613. URL: https://doi.org/10.7554/eLife.57613.

C. Wu et al., 2023: Wu, Chengzhi, Junwei Zheng, Julius Pfrommer, and Jürgen Beyerer (2023). "Attention-based point cloud edge sampling". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5333–5343.

X. Wu et al., 2024: Wu, Xiaoyang, Li Jiang, Peng-Shuai Wang, Zhijian Liu, et al. (2024). "Point Transformer V3: Simpler, Faster, Stronger". In: *CVPR*.

X. Wu et al., 2022: Wu, Xiaoyang, Yixing Lao, Li Jiang, Xihui Liu, et al. (2022). "Point transformer v2: Grouped vector attention and partition-based pooling". In: *Advances in Neural Information Processing Systems* 35, pp. 33330–33342.

Z. Wu et al., 2015: Wu, Zhirong, Shuran Song, Aditya Khosla, Fisher Yu, et al. (2015). "3d shapenets: A deep representation for volumetric shapes". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920.

K. Xie et al., 2025: Xie, Kangxian, Jiancheng Yang, Donglai Wei, Ziqiao Weng, et al. (Jan. 2025). "Efficient anatomical labeling of pulmonary tree structures via deep point-graph representation-based implicit fields". In: 99, p. 103367. ISSN: 1361-8415. DOI: 10.1016/j.media.2024.103367.

S. Xie et al., 2015: Xie, Saining and Zhuowen Tu (2015). "Holistically-nested edge detection". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1395–1403.

Yan Xu et al., 2024: Xu, Yan, Rixiang Quan, Weiting Xu, Yi Huang, et al. (2024). "Advances in medical image segmentation: A comprehensive review of traditional, deep learning and hybrid approaches". In: *Bioengineering* 11.10, p. 1034.

Zanjani et al., 2019: Zanjani, Farhad Ghazvinian, David Anssari Moin, Bas Verheij, Frank Claessen, et al. (2019). "Deep learning approach to semantic segmentation in 3D point cloud intra-oral scans of teeth". In: *International Conference on Medical Imaging with Deep Learning*. PMLR, pp. 557–571.

F. Zhang et al., 2021: Zhang, Fan, Yu Wang, and Hua Yang (2021). "Efficient context-aware network for abdominal multi-organ segmentation". In: *arXiv preprint arXiv:2109.10601*.

K. Zhang et al., 2023: Zhang, Kaidong and Dong Liu (2023). "Customized segment anything model for medical image segmentation". In: *arXiv preprint arXiv:2304.13785*.

Zhao et al., 2021: Zhao, Hengshuang, Li Jiang, Jiaya Jia, Philip HS Torr, et al. (2021). "Point transformer". In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16259–16268.

Zhou et al., 2018: Zhou, Zongwei, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang (2018). "Unet++: A nested u-net architecture for medical image segmentation". In: *Deep learning in medical image analysis and multimodal learning for clinical decision support: 4th international workshop, DLMIA 2018, and 8th international workshop, ML-CDS 2018,*

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

held in conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, proceedings 4. Springer, pp. 3–11.

Zoph et al., 2016: Zoph, Barret and Quoc V Le (2016). “Neural architecture search with reinforcement learning”. In: *arXiv preprint arXiv:1611.01578*.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Appendix A:
Computing resources

This work was granted access to the HPC resources of IDRIS under the allocations 2022-AD011013983, 2023-AD011013983R1 and 2024-AD011013983R2 made by GENCI.

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Appendix B:
SCONet-based sampling: detailed per-organ results

Point cloud	Dice					HD
	LIV	*KID*	*SPL*	*PAN*	AVG	AVG
Contour	0.958	0.916	0.938	0.755	**0.892**	**19.12**
Pred. contour	0.942	0.905	0.910	0.741	0.874	19.85
Pred. contour (full)	0.945	0.910	0.930	0.754	0.885	19.60

Table B.1: *SCONet performance on AbdomenCT-1K for different input point clouds.*

Inference point cloud	Dice					HD
	LIV	*KID*	*SPL*	*PAN*	AVG	AVG
Reference contour	0.956	0.936	0.941	0.847	**0.920**	**14.37**
Pred. contour	0.920	0.829	0.882	0.635	0.817	23.37
Pred. contour (full)	0.933	0.906	0.923	0.735	0.874	19.93

Table B.2: *Performance of SCONet trained on foreground contour points when evaluated on different types of points (AbdomenCT-1K).*

Point cloud	Dice																HD
	SPL	*RKI*	*LKI*	*GBL*	*ESO*	*LIV*	*STO*	*AOR*	*IVC*	*PAN*	*RAG*	*LAG*	*DUO*	*BLA*	*PRO/UTE*	AVG	AVG
Contour	0.884	0.849	0.870	0.577	0.588	0.924	0.771	0.795	0.733	0.667	0.488	0.437	0.557	0.642	0.612	**0.693**	**24.74**
Pred. contour	0.865	0.840	0.857	0.519	0.584	0.914	0.766	0.780	0.717	0.657	0.477	0.415	0.558	0.599	0.539	0.672	**25.42**
Pred. contour (full)	0.871	0.851	0.870	0.559	0.582	0.917	0.780	0.797	0.729	0.672	0.466	0.448	0.561	0.638	0.556	0.686	26.32

Table B.3: *SCONet performance on AMOS for different input point clouds.*

Inference point cloud	Dice																HD
	SPL	*RKI*	*LKI*	*GBL*	*ESO*	*LIV*	*STO*	*AOR*	*IVC*	*PAN*	*RAG*	*LAG*	*DUO*	*BLA*	*PRO/UTE*	AVG	AVG
Reference contour	0.879	0.876	0.881	0.548	0.611	0.924	0.833	0.815	0.744	0.695	0.507	0.472	0.629	0.699	0.654	**0.718**	**24.59**
Pred. contour	0.819	0.794	0.809	0.443	0.548	0.899	0.716	0.728	0.670	0.596	0.464	0.421	0.497	0.515	0.472	0.626	28.26
Pred. contour (full)	0.849	0.844	0.861	0.493	0.578	0.906	0.760	0.786	0.713	0.657	0.487	0.448	0.550	0.580	0.541	0.670	26.88

Table B.4: *Performance of SCONet trained on foreground contour points when evaluated on different types of points (AMOS).*

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Département de la Formation par la Recherche et des Études Doctorales (**FEDORA**)

Bâtiment Charlotte Perriand 3e étage
17, av. des Arts
69621 Villeurbanne Cédex
fedora@insa-lyon.fr

Référence : TH1321_JOUVENCEL Maylis

L'INSA Lyon a mis en place une procédure de contrôle systématique via un outil de détection de similitudes (logiciel Compilatio). Après le dépôt du manuscrit de thèse, celui-ci est analysé par l'outil. Pour tout taux de similarité supérieur à 10%, le manuscrit est vérifié par l'équipe de FEDORA. Il s'agit notamment d'exclure les auto-citations, à condition qu'elles soient correctement référencées avec citation expresse dans le manuscrit.

Par ce document, il est attesté que ce manuscrit, dans la forme communiquée par la personne doctorante à l'INSA Lyon, satisfait aux exigences de l'Etablissement concernant le taux maximal de similitude admissible.

INSA LYON
Campus LyonTech La Doua
20, avenue Albert Einstein - 69621 Villeurbanne cedex - France
Tél. +33 (0)4 72 43 83 83 - Fax +33 (0)4 72 43 85 00
www.insa-lyon.fr

membre de UNIVERSITÉ DE LYON

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés

FOLIO ADMINISTRATIF

THESE DE L'INSA LYON, MEMBRE DE L'UNIVERSITE DE LYON

NOM : JOUVENCEL DATE de SOUTENANCE : 19/01/2026

Prénoms : Maylis, Amélie

TITRE (ENG) : Medical image segmentation through learning on unstructured 3D data
TITRE (FR) : Segmentation d'images médicales par apprentissage sur des données 3D non structurées

NATURE : Doctorat Numéro d'ordre : 2026ISAL0001

École Doctorale : Électronique, Électrotechnique et Automatique de Lyon

Spécialité : Traitement du signal et de l'image

RÉSUMÉ (ENG) : Medical image segmentation is an essential task for diagnosis and treatment monitoring. However, deep learning methods used to perform this task generally have a relatively high computation cost, particularly when the processed images are large volumes containing millions of voxels. This PhD thesis therefore proposes to replace the original voxel representation by a lighter point cloud representation in order to develop more efficient segmentation algorithms. In the first part of this thesis, we propose a workflow that extracts a contour point cloud from the original image and feeds it to SCONet, our implicit network adapted from convolutional occupancy networks for the task of multi-organ segmentation. We evaluate our method on two abdominal CT datasets. In the second part, we provide an in-depth study of the influence of the point cloud quality on SCONet's performance and identify which points are more useful for segmentation. In the third part, we evaluate the robustness of our method to domain shifts and find that SCONet performs well when confronted to dataset shifts and to changes in imaging modality.

RÉSUMÉ (FR) : La segmentation d'images médicales est une tâche essentielle pour le diagnostique et le suivi thérapeutique. Cependant les méthodes d'apprentissage profond utilisées pour la réaliser ont en général un coût en calcul assez élevé, en particulier quand les images à traiter sont des volumes de grande taille contenant des millions de voxels. Ainsi, cette thèse de doctorat propose de remplacer la représentation voxelique d'origine par une représentation plus légère sous forme de nuage de points, ce qui pourrait permettre de développer des algorithmes de segmentation moins coûteux. Dans la première partie de cette thèse, nous proposons une chaîne de traitement qui extrait un nuage de points de contours à partir de l'image d'origine et le fournit à SCONet, notre réseau implicit adapté des réseaux d'occupation convolutifs pour la tâche de segmentation multi-organes. Nous évaluons notre méthode sur deux bases de données d'images CT abdominales. Dans une deuxième partie, nous étudions plus finement l'influence de la qualité du nuage de points sur la performance de SCONet et identifions les points les plus utiles pour la segmentation. Dans une troisième partie, nous évaluons la robustesse de notre méthode aux changements de domaine et trouvons que SCONet est performant quand confronté à un changement de base de données ou de modalité d'imagerie.

MOTS-CLÉS (ENG) : Deep learning, Segmentation, Medical imaging, Point cloud
MOTS-CLÉS (FR) : Apprentissage profond, Segmentation, Imagerie médicale, Nuage de points

Laboratoire(s) de recherche : CREATIS – Centre de Recherche en Acquisition et Traitement de l'Image pour la Santé

Directeur de thèse : Sébastien Valette
Co-directeur de thèse : Razmig Kéchichian

Présidente du Jury : Julie Digne

Composition du Jury :

HEINRICH Mattias, Professor, University of Lübeck, Rapporteur
NAEGEL Benoît, PU, Université de Strasbourg, Rapporteur
DESVIGNES Michel, PE, Université Grenoble Alpes, Examinateur
MATEUS Diana, PU, Centrale Nantes, Examinatrice
VALETTE Sébastien, CR, INSA Lyon, Directeur de thèse
KÉCHICHIAN Razmig, MCF, INSA Lyon, Co-directeur de thèse

Thèse accessible à l'adresse : https://theses.insa-lyon.fr/publication/2026ISAL0001/these.pdf © [M. Jouvencel], [2026], INSA Lyon, tous droits réservés