OSS-Net: Memory Efficient High Resolution Semantic Segmentation of 3D Medical Data

32nd British Machine Vision Conference 2021

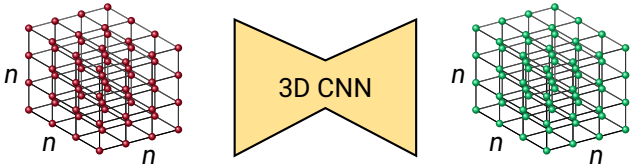
Christoph Reich[‡], **Tim Prangemeier, Özdemir Cetin and Heinz Koeppl**[†]

Centre for Synthetic Biology,
Department of Electrical Engineering
and Information Technology,
Department of Biology,
Technische Universität Darmstadt

[‡]christoph.reich@bcs.tu-darmstadt.de
[†]heinz.koeppl@bcs.tu-darmstadt.de

Introduction

3D Medical Semantic Segmentation

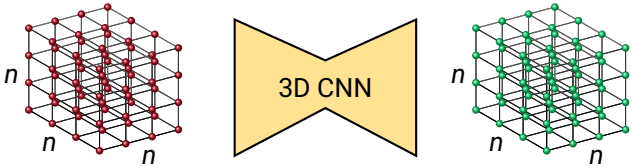


3D CNNs for 3D segmentation

+ Accurate segmentation
− $\mathcal{O}(n^3)$ memory and comp. complexity
+ Fast inference

Introduction

3D Medical Semantic Segmentation



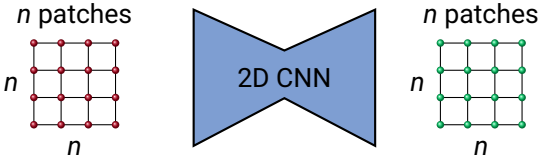
3D CNNs for 3D segmentation

+ Accurate segmentation − $\mathcal{O}(n^3)$ memory and comp. complexity + Fast inference

↯ 3D CNNs do not scale well to high voxel resolutions

Introduction

3D Medical Semantic Segmentation



2D CNNs for 3D segmentation

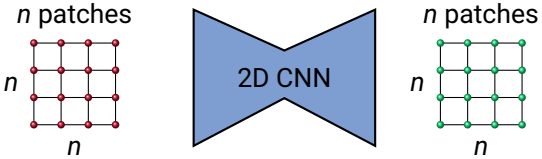
~ Relatively accurate segmentation

+ Memory efficient

~ Relatively fast inference

Introduction

3D Medical Semantic Segmentation



2D CNNs for 3D segmentation

~ Relatively accurate segmentation + Memory efficient ~ Relatively fast inference

↯ 2D CNNs do not model 3D relations

Related Work

Efficient 3D Segmentation Approaches

Efficient 3D CNNs (voxelised)

- RevNet [Gomez *et al.*, NeurIPS 2017]
- Sparse convolutions [Choy *et al.*, CVPR 2019]

Other Representations (Mesh & Point Cloud)

- PointNet++ [Qi *et al.*, NeurIPS 2017]
- Mesh R-CNN [Gkioxari *et al.*, CVPR 2019]

Related Work

Efficient 3D Segmentation Approaches

Efficient 3D CNNs (voxelised)

- RevNet [Gomez *et al.*, NeurIPS 2017]
- Sparse convolutions [Choy *et al.*, CVPR 2019]

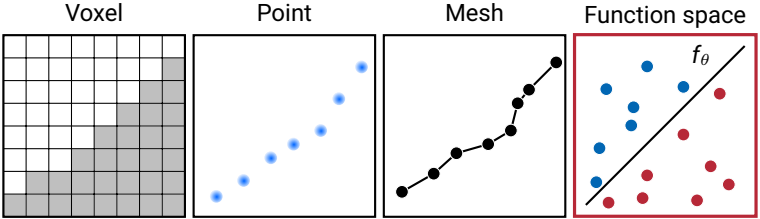
Other Representations (Mesh & Point Cloud)

- PointNet++ [Qi *et al.*, NeurIPS 2017]
- Mesh R-CNN [Gkioxari *et al.*, CVPR 2019]

↯ Not applicable to dense medical data, require heavy post-processing, or sacrifice high memory efficiency.

Related Work

Occupancy Networks (O-Nets)



Occupancy Networks [Mescheder *et al.*, CVPR 2019]

$$f_\theta : \mathbb{R}^3 \times \mathcal{X} \to [0, 1]$$

+ Highly memory efficient
~ Relatively slow inference
− Fail to express fine details

Method

Occupancy Networks for Semantic Segmentation (OSS-Net)

$$f_\theta : \mathbb{R}^3 \times \mathcal{X} \times \mathcal{Z} \rightarrow [0, 1]$$

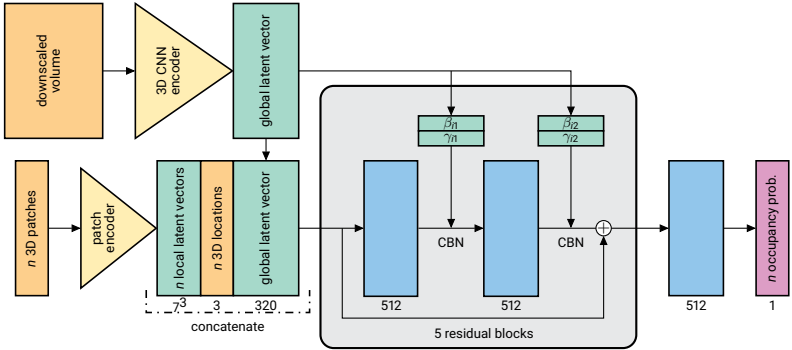
$x \in \mathcal{X}$: downscaled global volume $\quad$ $z \in \mathcal{Z}$: local 3D patches centered at the coordinate $p \in \mathbb{R}^3$

Method

Occupancy Networks for Semantic Segmentation (OSS-Net)

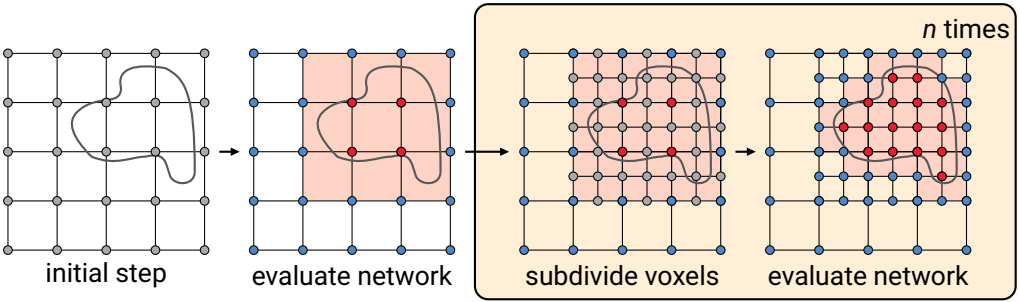
$$f_\theta : \mathbb{R}^3 \times \mathcal{X} \times \mathcal{Z} \to [0,1]$$

$x \in \mathcal{X}$: downscaled global volume $z \in \mathcal{Z}$: local 3D patches centered at the coordinate $p \in \mathbb{R}^3$



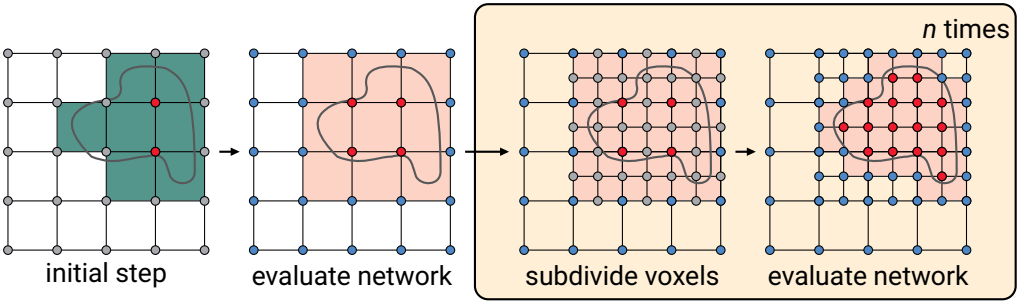
Method

OSS-Net Inference



Method

OSS-Net Inference



- Improved inference (2× speed up) by utilising low-resolution encoder prediction (in green).

Results

Segmentation

Model/OSS-Net Configuration	BraTS 2020		LiTS	
	Dice ↑	IoU ↑	Dice ↑	IoU ↑
3D residual U-Net (voxelised baseline)	*0.8827*	**0.7995**	**0.7888**	**0.6558**
2D residual U-Net (slicing baseline)	0.8589	0.7658	0.6674	0.5233
O-Net (function space baseline) [Mescheder *et al.*, CVPR 2019]	0.7016	0.5615	0.6506	0.4842
+ patch encoder w/ small patches 7^3 (OSS-Net A)	0.8592	0.7644	0.7127	0.5579
+ avg. pooled intermediate patches 14^3 (OSS-Net B)	0.8541	0.7572	0.7585	0.6154
A + encoder skip-con. & aux. loss (OSS-Net C)	**0.8842**	*0.7991*	*0.7616*	*0.6201*
B + encoder skip-con. & aux. loss (OSS-Net D)	0.8774	0.7876	0.7566	0.6150

- OSS-Nets achieve accurate segmentation results similar to 3D CNNs.

Results

Segmentation

Model/OSS-Net Configuration	BraTS 2020		LiTS	
	Dice ↑	IoU ↑	Dice ↑	IoU ↑
3D residual U-Net (voxelised baseline)	*0.8827*	**0.7995**	**0.7888**	**0.6558**
2D residual U-Net (slicing baseline)	0.8589	0.7658	0.6674	0.5233
O-Net (function space baseline) [Mescheder *et al.*, CVPR 2019]	0.7016	0.5615	0.6506	0.4842
+ patch encoder w/ small patches 7^3 (OSS-Net A)	0.8592	0.7644	0.7127	0.5579
+ avg. pooled intermediate patches 14^3 (OSS-Net B)	0.8541	0.7572	0.7585	0.6154
A + encoder skip-con. & aux. loss (OSS-Net C)	**0.8842**	*0.7991*	*0.7616*	*0.6201*
B + encoder skip-con. & aux. loss (OSS-Net D)	0.8774	0.7876	0.7566	0.6150

- OSS-Nets achieve accurate segmentation results similar to 3D CNNs.

Results

Segmentation

Model/OSS-Net Configuration	BraTS 2020 Dice ↑	BraTS 2020 IoU ↑	LiTS Dice ↑	LiTS IoU ↑
3D residual U-Net (voxelised baseline)	*0.8827*	**0.7995**	**0.7888**	**0.6558**
2D residual U-Net (slicing baseline)	0.8589	0.7658	0.6674	0.5233
O-Net (function space baseline) [Mescheder *et al.*, CVPR 2019]	0.7016	0.5615	0.6506	0.4842
+ patch encoder w/ small patches 7^3 (OSS-Net A)	0.8592	0.7644	0.7127	0.5579
+ avg. pooled intermediate patches 14^3 (OSS-Net B)	0.8541	0.7572	0.7585	0.6154
A + encoder skip-con. & aux. loss (OSS-Net C)	**0.8842**	*0.7991*	*0.7616*	*0.6201*
B + encoder skip-con. & aux. loss (OSS-Net D)	0.8774	0.7876	0.7566	0.6150

- OSS-Nets achieve accurate segmentation results similar to 3D CNNs.

Results

Memory

Model/ OSS-Net Configuration	BraTS 2020			LiTS		
	Training 2^{14} loc.	Inference 2^{14} locations	Inference 2^{17} locations	Training 2^{14} loc.	Inference 2^{14} locations	Inference 2^{17} locations
3D res. U-Net	14.41GB	3.57GB (dense pred.)		14.41GB*	3.57GB (dense pred.)†	
2D res. U-Net	1.16GB‡	0.46GB (slice pred.)‡		4.29GB‡	1.20GB (slice pred.)‡	
O-Net	2.35GB	0.29GB	1.93GB	5.07GB	1.47GB	1.99GB
OSS-Net A	2.58GB	0.39GB	2.73GB	5.18GB	1.50GB	2.35GB
OSS-Net B	2.76GB	0.48GB	3.45GB	5.21GB	1.53GB	2.53GB
OSS-Net C	2.59GB	0.39GB	2.73GB	5.19GB	1.51GB	2.35GB
OSS-Net D	2.76GB	0.48GB	3.46GB	5.22GB	1.53GB	2.53GB

- OSS-Nets are highly memory efficient during training and inference.

Results

Memory

Model/ OSS-Net Configuration	BraTS 2020			LiTS		
	Training 2^{14} loc.	Inference 2^{14} locations	Inference 2^{17} locations	Training 2^{14} loc.	Inference 2^{14} locations	Inference 2^{17} locations
3D res. U-Net	14.41GB	3.57GB (dense pred.)		14.41GB*	3.57GB (dense pred.)†	
2D res. U-Net	1.16GB‡	0.46GB (slice pred.)‡		4.29GB‡	1.20GB (slice pred.)‡	
O-Net	2.35GB	0.29GB	1.93GB	5.07GB	1.47GB	1.99GB
OSS-Net A	2.58GB	0.39GB	2.73GB	5.18GB	1.50GB	2.35GB
OSS-Net B	2.76GB	0.48GB	3.45GB	5.21GB	1.53GB	2.53GB
OSS-Net C	2.59GB	0.39GB	2.73GB	5.19GB	1.51GB	2.35GB
OSS-Net D	2.76GB	0.48GB	3.46GB	5.22GB	1.53GB	2.53GB

- OSS-Nets are highly memory efficient during training and inference.

Results

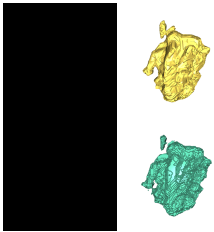
Memory

Model/ OSS-Net Configuration	BraTS 2020			LiTS		
	Training 2^{14} loc.	Inference 2^{14} locations	Inference 2^{17} locations	Training 2^{14} loc.	Inference 2^{14} locations	Inference 2^{17} locations
3D res. U-Net	14.41GB	3.57GB (dense pred.)		14.41GB*	3.57GB (dense pred.)†	
2D res. U-Net	1.16GB‡	0.46GB (slice pred.)‡		4.29GB‡	1.20GB (slice pred.)‡	
O-Net	2.35GB	0.29GB	1.93GB	5.07GB	1.47GB	1.99GB
OSS-Net A	2.58GB	0.39GB	2.73GB	5.18GB	1.50GB	2.35GB
OSS-Net B	2.76GB	0.48GB	3.45GB	5.21GB	1.53GB	2.53GB
OSS-Net C	2.59GB	0.39GB	2.73GB	5.19GB	1.51GB	2.35GB
OSS-Net D	2.76GB	0.48GB	3.46GB	5.22GB	1.53GB	2.53GB

- OSS-Nets are highly memory efficient during training and inference.

Conclusion

- We proposed OSS-Nets for segmenting 3D medical data in function space
  + **Accurate segmentation** - OSS-Nets achieve segmentation results similar to 3D CNNs
  + **Highly memory efficient** - OSS-Nets are highly memory efficient during inference and training
  + **Fast inference** - Improved inference approach yields a 2× speed up



Project Page

https://christophreich1996.github.io/oss_net/

Code & Trained Models

https://github.com/ChristophReich1996/OSS-Net