arXiv:1911.01764v1 [cs.LG] 5 Nov 2019

One Network To Segment Them All: A General, Lightweight System for Accurate 3D Medical Image Segmentation

Mathias Perslev[1], Erik Bjørnager Dam[1,2], Akshay Pai[1,2], and Christian Igel[1]

[1] Department of Computer Science, University of Copenhagen, map@di.ku.dk
[2] Cerebriu A/S, Copenhagen, Denmark

Abstract. Many recent medical segmentation systems rely on powerful deep learning models to solve highly specific tasks. To maximize performance, it is standard practice to evaluate numerous pipelines with varying model topologies, optimization parameters, pre- & postprocessing steps, and even model cascades. It is often not clear how the resulting pipeline transfers to different tasks.

We propose a simple and thoroughly evaluated deep learning framework for segmentation of arbitrary medical image volumes. The system requires no task-specific information, no human interaction and is based on a fixed model topology and a fixed hyperparameter set, eliminating the process of model selection and its inherent tendency to cause method-level over-fitting. The system is available in open source and does not require deep learning expertise to use. Without task-specific modifications, the system performed better than or similar to highly specialized deep learning methods across 3 separate segmentation tasks. In addition, it ranked 5-th and 6-th in the first and second round of the 2018 Medical Segmentation Decathlon comprising another 10 tasks.

The system relies on *multi-planar* data augmentation which facilitates the application of a single 2D architecture based on the familiar U-Net. Multi-planar training combines the parameter efficiency of a 2D fully convolutional neural network with a systematic train- and test-time augmentation scheme, which allows the 2D model to learn a representation of the 3D image volume that fosters generalization.

1 Introduction

More and more systems for medical image segmentation rely on deep learning (DL). However, most publications on this topic report performance improvements for a particular segmentation task and imaging modality and use a specialized processing pipeline adapted through hyperparameter tuning. This makes it difficult to generalize the obtained results and bears the risk that the reported

Preprint, final version published in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, LNCS 11765, pp. 30–38, Springer, 2019. The final authenticated publication is available online at https://doi.org/10.1007/978-3-030-32245-8_4

findings are artifacts. In line with the idea behind the 2018 Medical Segmentation Decathlon (MSD)[3] [1], a challenge evaluating the generalisability of machine learning based segmentation algorithms, we argue that new segmentation systems should be evaluated across many different data cohorts and maybe even tasks. This reduces the risk of unintentional method overfitting and may help to gain more general insights about, for example, superior model architectures and learning methods for particular problem classes. This does not only contribute to our basic understanding of the segmentation algorithms, but also to the clinical acceptance and applicability of the systems – even if the generality could come at the cost of not reaching state-of-the-art performance on each individual cohort or task.

A DL segmentation framework that works across a wide range of tasks and in which the individual components and hyperparameters are sufficiently understood allows to automate the task-specific adaptations. This is a prerequisite for being useful for practitioners who are not experts in DL. Big compute clusters offer a way to design systems that provide accurate segmentations for a variety of tasks and do not require tuning by DL experts. If compute resources are not limited, automatic model and hyperparameter selection can be implemented. Given new training data, the systems tests a large variety of segmentation algorithms and, for each algorithm, explores the space of the required hyperparameters. While this approach may produce powerful systems, and was employed to variable extents by top-performing MSD submissions, we argue that it has crucial drawbacks. First, it comes with a risk of automated method overfitting, even if the data is handled carefully. Second, the approach may be prohibitive in clinical practice (and for many scientific institutions) when there is simply no access to sufficient (data regulations compliant) compute resources.

This paper presents an open-source system for medical volume segmentation that addresses all the issues outlined above. It relies on a single neural network of fixed architecture that **1)** showed very good performance across a variety of diverse segmentation tasks, **2)** can be trained efficiently without DL expert knowledge, large amounts of data, and compute clusters, and **3)** does not need large resources when deployed. The system architecture is a 2D U-Net [2,3] variant. The decisive feature of our approach lies in extensive data augmentation, in particular by rotating the input volume before presenting slices to the fully convolutional network. Because of the latter, we refer to our approach as *multi-planar* U-Net training (*MPUnet*). We present a thorough evaluation of our system on a total of 13 different 3D segmentation tasks, including 10 from MSD, on which it obtains high accuracies – often reaching state-of-the-art performance from even highly specialized DL-based methods.

2 Method

At the heart of our system lies a 2D U-net [2] modified slightly to **1)** include batch normalization layers [4] intervening each double convolution- and up-convolution

[3] `http://medicaldecathlon.com`

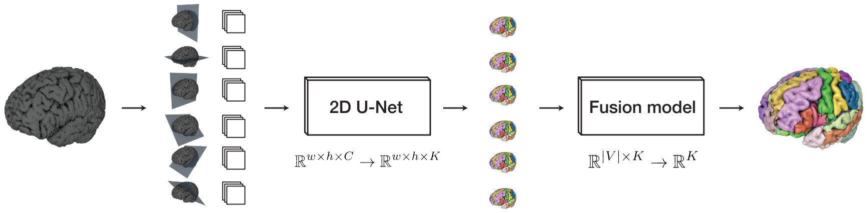
2D U-Net
$\mathbb{R}^{w\times h\times C} \to \mathbb{R}^{w\times h\times K}$
Fusion model
$\mathbb{R}^{|V|\times K} \to \mathbb{R}^{K}$

Fig. 1. Model overview. In the inference phase, the input volume (left) is sampled on 2D isotropic grids along multiple view axes. The model predicts a full volume along each axis and maps the predictions into the original image space. A fusion model combines the 6 proposed segmentation volumes into a single final segmentation.

block and **2)** use nearest-neighbor up-sampling followed by convolution to implement up-convolutions [5]. Basic network topology and hyperparameters can bet set to their default choices as done in all experiments in this paper, see Table S.1 in the supplementary material for an overview. Compared to [2], the number of filters has been increased by a factor of $\sqrt{2}$, see supplementary Table S.6 for details. As a result, the model has ≈ 62 million parameters. While one would assume that the size of the model is a crucial hyperparameter, we kept the model architecture the same for all tasks. For each task, only the filters in the first layer were resized according to the number C of input channels and the number of output units was set to the number of classes K.

The decisive feature of our multi-planar U-Net training (*MPUnet*) is the generation of the inputs at training and test time, which is done by sampling from multiple planes of random orientation spanning the image volume. That is, the network must learn to segment the input seen from different views, see Fig. 1.

The model $f(x;\theta)$ takes as input multi-channel 2D image slices of size $w \times h$, $x \in \mathbb{R}^{w\times h\times C}$, and outputs a probabilistic segmentation map $P \in \mathbb{R}^{w\times h\times K}$ for K classes. Prior to training we define a set $V = \{v_1, v_2, ..., v_i\}$ of i randomly sampled unit vectors in $\mathbb{R}^3$. The set defines the axes through the image volume along which we sample 2D inputs to the model, visualized in Fig. 2. We re-sample the set V until all pairs of vectors have an angle of at least 60 deg between them. A sampled set of planar axes is shown in Fig. 2(a). Note that the model could also be fit using a set of fixed, predefined planes, but we found no performance gain in doing so, even if the fixed set included the standard planes. We use $i = 6$ for all reported evaluations. This number was chosen based on prior experiments in which we observed monotonically improving performance with the inclusion of additional planes and $i = 6$ providing a good balance between accuracy and computation, see supplementary Table S.2.

During training, the model is provided batches of images randomly sampled from the i planes in V without supplying information about the corresponding axis. During inference, the model predicts along each plane producing a set of i segmentation volumes $\mathbf{P} = \{P_v \in \mathbb{R}^{w\times h\times d\times K} \mid v \in V\}$. Each P_v is mapped to

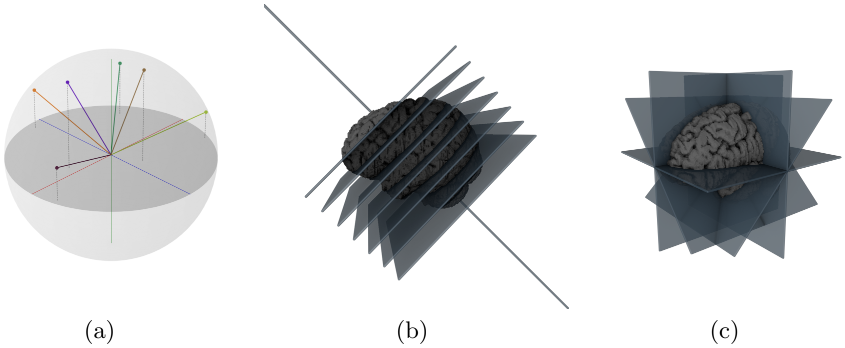

Fig. 2. **(a)** Visualization of a set V of sampled view axis unit vectors. **(b)** Illustration of images sampled along one view. **(c)** Illustration of multiple images sampled along multiple unique views.

the input image space to obtain point correspondence by assigning to each voxel in the input image the value of its nearest predicted point in P_v. Distances are computed in physical coordinates.

At test-time, the learned invariance to orientation is exploited by segmenting the entire volume from each view. This results in several candidate segmentations for each subject, which are combined by a linear fusion model, see Fig. 1. We map $\mathbf{P}$ to a single probabilistic segmentation by a weighted sum of the per-class and per-view softmax-scores. For all $w \cdot h \cdot d$ voxels x in $\mathbf{P}$ and each class $k \in \{1, ..., K\}$, the *fusion model* $f_{\text{fusion}} : \mathbb{R}^{|V| \times K} \rightarrow \mathbb{R}^K$ calculates $z(x)_k = \sum_{n=1}^{|V|} W_{n,k} \cdot p_{n,x,k} + \beta_k$. Here $p_{n,x,k}$ denotes the probability of class k at voxel x as predicted by segmentation P_n. The $W \in \mathbb{R}^{|V| \times K}$ weighs the probabilities of each class as predicted from each view and $\beta \in \mathbb{R}^K$ are bias parameters, which can adjust the overall tendency to predict a given class. The parameters of f_{fusion} are learned from the validation data. The model scales the predictions according to which views do well on each class, motivated by the fact that different target classes may appear in different shapes and levels of recognizability when seen from the different directions in V.

Isotropic Image Sampling. Interpolation is needed to sample image planes not aligned with the original voxel grid. We use tri-linear and nearest-neighbour interpolation to sample the image and label map, respectively. We take advantage of the necessity for interpolation by sampling images on isotropic grids in the physical scanner space, oriented according to the patient's position in the scanner. This ensures that the model always operates on images in which the shapes of anatomical structures are maintained across scanners and acquisition protocols. Note that this approach may lead to over- or under-sampling along some axes, which may lead to loss of image information or interpolation artefacts. Empirically, however, we found that the benefit of maintaining isotropy outweighed potential drawbacks of interpolation.

We must define a set of parameters restricting the sampling. Specifically, we are free to choose **1)** the pixel dimensions, $q \in \mathbb{Z}^+$ (the number of pixels to sample for each image), **2)** the real-space extent of the image (in mm), $m \in \mathbb{R}^+$, and **3)** the real-space distance between consecutive voxels, $r \in \mathbb{R}^+$. Note that two of these parameters define the third. We restrict our sampling to equal q, m and r for both image dimensions producing squared images. We sample images within a sphere of diameter m centered at the origin of the scanner coordinate system. We employ a simple heuristic that attempts to pick q, m and r so that **1)** the training is computable on our GPUs with batch sizes of at least 8, **2)** r approximately matches the resolution of the images along their highest resolution axis and **3)** the sampled images span the entirety of the relevant volume of all images in the dataset. When this is not possible, the requirements are prioritized in the given order, with 1 having highest priority. Note that 3 becomes less important with increasing numbers of planes as voxels missed in one plane are likely to be included in some of the others.

Augmentation. Processing the input image from different views has the the same effect as applying affine transformations to the 3D input and presenting the transformed images to a (single-view) network. Thus, at the heart the MPUnet is a U-Net with extensive, systematic affine data augmentation. On top of the multi-view sampling, we also employ non-linear transformations to further augment the training data. We apply the Random Elastic Deformations algorithm [6] to each sampled image in a batch with a probability of 1/3. The elasticity constants σ and deformation intensity multipliers α are sampled uniformly from $[20, 30]$ and $[100, 500]$, respectively. This generates augmented images with high variability in terms of both deformation strength and smoothness.

The augmented images do not always display anatomically plausible structures. Yet, they often significantly improve the generalization especially when training on small datasets or tasks involving pathologies of highly variable shape. However, we weigh the loss-contribution from augmented images by 1/3 in order to optimize primarily over true images.

Pre- and post-processing. Our model uses a minimum of image processing outside of the network itself. We restrain from applying any post-processing of the model's output, because post-processing is typically highly task-specific. We only apply an image- and channel-wise outlier-robust pre-possessing that scales intensity values according to the median and inter-quartile range computed over all non-background voxels. Background voxels are defined by having intensities less than or equal to the first percentile of the intensity distribution.

Implementation. The MPUnet is available as open-source. The fully autonomous implementation makes the MPUnet applicable also for users with limited deep learning expertise and/or compute resources. A command line interface supports fixed split or cross-validation training and evaluation on arbitrary images. Any non-constant hyperparameter can automatically be inferred from the

Table 1. Performance of the MPUnet across thirteen segmentation tasks. The shown F1 (dice) scores are mean values computed across all non-background per-class F1 scores. For the 10 MSD datasets evaluation was performed by the challenge organisers on non-publicly available test-sets. For MICCAI and HarP, evaluation was performed over three trials. Five fold cross-validation was used for OAI. The 'Classes' column include the background class, which is not included when computing the F1 scores. The 'Size' column gives the total dataset size. Note that the F1 standard deviations for tasks 8, 9 & 10 are not yet published by the challenge organizers. We refer to `http://medicaldecathlon.com/results.html` for a detailed comparison of our results (team CerebriuDIKU) with those of other challenge participants.

	Dataset	Modality	Segmentation Target(s)	Classes	Size	F1 Score
	MICCAI	MRI	Whole-Brain	135	35	0.74 ± 0.03
	HarP	MRI	L+R Hippocampus	3	135	0.85 ± 0.03
	OAI	MRI	Knee Cartilages	7	176	0.87 ± 0.06
2018 Medical Segmentation Decathlon	Task 1	MRI	Brain Tumours	4	750	0.60 ± 0.24
	Task 2	MRI	Cardiac, Left Atrium	2	30	0.89 ± 0.09
	Task 3	CT	Liver & Tumour	2	201	0.76 ± 0.18
	Task 4	MRI	Hippocampus ROI.	2	394	0.89 ± 0.04
	Task 5	MRI	Prostate	3	48	0.78 ± 0.10
	Task 6	CT	Lung Tumours	2	96	0.59 ± 0.23
	Task 7	CT	Pancreas & Tumour	3	420	0.48 ± 0.21
	Task 8	CT	Hepatic Ves. & Tumour	3	443	0.49
	Task 9	CT	Spleen	2	61	0.95
	Task 10	CT	Colon Cancer	2	190	0.28

training data. See the GitHub repository at `https://github.com/perslev/MultiPlanarUNet` for a user guide.

3 Experiments and Results

We applied the MPUNet without task-specific modifications to a total of 13 segmentation tasks. Ten of those datasets were part of the 2018 MSD challenge, described in detail and sourced on the challenge's website. The remaining three datasets were the MICCAI 2012 Multi-Atlas Challenge (MICCAI) dataset [7], the EADC-ADNI Harmonized Hippocampal Protocol (HarP) dataset [8] and a knee MRI dataset from the Osteoarthritis Initiative (OAI) [9]. The evaluation covers healthy and pathological anatomical structures, mono- and multi-modal MR and CT, and various acquisition protocols. The mean per-class F1 (dice) scores of the MPUNet are reported in Table 1. Note that in MSD tumour segmentation tasks 3 & 7 both organ and tumour are segmented, and the mean F1 for those tasks is lifted by the performance on the organ and decreased by the performance on the tumour. We refer to the supplementary Table S.4 for detailed per-class scores for the ten MSD tasks.

The MPUnet reached state-of-the-art performance for DL methods on the three non-challenge datasets (MICCAI, HaRP and OAI) despite comparable methods being developed and tuned specifically to the cohorts and tasks. On MICCAI, with a mean F1 of 0.74 the MPUnet compares similar to the 0.74 obtained in [10] using a 2D multi-scale CNN on brain-extracted images and 0.75 obtained in [11] using a combination of a multi-scale 2D CNN, 3D patch-based CNN, a spatial information encoder network and a probabilistic atlas also on brain-extracted images. With a mean F1 of 0.85 on HarP, the MPUnet compares favorable to 0.78-0.83 (depending on subject disease state) reported in [12]. On OAI, with a mean F1 of 0.87, the MPUnet gets near the 0.88/0.89 (baseline/follow-up) obtained in [13] using a task-specific pipeline including 2D- and 3D U-nets along with multiple statistical shape model refinement steps. However, the comparison cannot be directly made as [13] worked on a smaller subset of the OAI data and predicted only 4 classes while we distinguished 7.

The MPUnet ranked 5th and 6th place in the first and second phases of the Medical Segmentation Decathlon respectively, in most cases comparing unfavorable only to significantly more compute intensive systems (see below).[4]

The question arises how the performance of a 2D U-net with multi-planar augmentation compares to a U-net with 3D convolutions. Such 3D models are computationally demanding and typically need – in our experience – large training datasets to achieve proper generalization. While we are not making the claim that the MPUnet is universally superior to 3D models, we did find the MPUnet to outperform a 3D U-net of comparable topology, learning and augmentation procedure across multiple tasks including one for which the 3D model had sufficient spatial extent to operate on the entire input volume at once. We refer to the supplementary Table S.5 for details. We also found the MPUnet superior to both single 2D U-Nets trained on individual planes as well as ensembles of separate 2D U-Nets trained on different planes, see Table S.2 & S.3 and Fig. S.1 in the supplementary material.

4 Discussion and Conclusions

The empirical evaluation over 13 segmentation tasks showed that multi-planar augmentation provides a simple mechanism for obtaining accurate segmentation models without hyperparameter tuning. With no task-specific modifications the MPUnet performs well across many non-pathological tissues imaged with various MR and CT protocols, in spite of the target compartments varying drastically in number, physical size, shape- and spatial distributions, as well as contrast to the surrounding tissues. Also the accuracies on the more difficult pathological targets are favorable compared to most other MSD contesters.

The MSD winning algorithm [14] relied on selecting a suitable model topology and/or cascade from an ensemble of candidates through cross-validation. In contrast to this and other top-ranking participants, we were interested to develop a task-agnostic segmentation system based on a single architecture and

[4] For comparison, the median F1 scores over all 10 tasks of the best five phase 1 submissions were 0.74, 0.67, 0.69, 0.66, and (our method) 0.69. Note that the official ranking was based on a more rigorous statistical analysis.

learning procedure that makes the system lightweight and easily transferable to clinical settings with limited compute resources.

That the MPUnet can be applied 'as is' across many tasks with high performance and its robustness against overfitting can be attributed to both the fully convolutional network approach, which is already known to generalize well, and our multi-planar augmentation framework. The latter allows us to apply a single 2D model with fixed hyperparameters, resulting in a fully autonomous segmentation system of low computational complexity. Multi-planar training improves the generalization performance in several ways: **1)** Sampling from multiple planes allows for a huge number of anatomically relevant images augmenting the training data; **2)** Exposing a 2D model to multiple planes takes the 3D nature of the input into account while maintaining the statistical and computational efficiency of 2D kernels; **3)** The systematic augmentation scheme allows test time augmentation to be performed, which increases the performance through variance reduction if errors across views are uncorrelated for a given subject (visualized in supplementary Fig. S.2). This makes the MPUnet an open source alternative to 3D fully convolutional neural networks.

Acknowledgements

We would like to thank both Microsoft and NVIDIA for providing computational resources on the Azure platform for this project.

References

1. Simpson, A.L., et al.: A large annotated medical image dataset for the development and evaluation of segmentation algorithms. CoRR **abs/1902.09063** (2019)
2. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer (2015) 234–241
3. Louring Koch, T., Perslev, M., Igel, C., Brand, S.S.: Accurate segmentation of dental panoramic radiographs with U-Nets. In: International Symposium on Biomedical Imaging (ISBI), IEEE (2019)
4. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning (ICML), PMLR (2015) 448–456
5. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill (2016)
6. Simard, P.Y., Steinkraus, D., Platt, J.: Best practices for convolutional neural networks applied to visual document analysis. In: International Conference on Document Analysis and Recognition (ICDAR), IEEE (2003)
7. Marcus, D.S., Wang, T.H., Parker, J., Csernansky, J.G., Morris, J.C., Buckner, R.L.: Open access series of imaging studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. Journal of Cognitive Neuroscience **19**(9) (2007) 1498–1507

8. Boccardi, M., et al.: Training labels for hippocampal segmentation based on the EADC-ADNI harmonized hippocampal protocol. Alzheimer's & Dementia **11**(2) (2015) 175–183
9. Dam, E., Lillholm, M., Marques, J., Nielsen, M.: Automatic segmentation of high- and low-field knee mris using knee image quantification with data from the osteoarthritis initiative. Journal of Medical Imaging **2**(2) (2015)
10. Moeskops, P., Viergever, M.A., Mendrik, A.M., de Vries, L.S., Benders, M.J.N.L., Isgum, I.: Automatic segmentation of MR brain images with a convolutional neural network. IEEE Transactions on Medical Imaging **35**(5) (2016) 1252–1261
11. Ganaye, P., Sdika, M., Benoit-Cattin, H.: Towards integrating spatial localization in convolutional neural networks for brain image segmentation. In: International Symposium on Biomedical Imaging (ISBI), IEEE (2018) 621–625
12. Roy, A.G., Conjeti, S., Navab, N., Wachinger, C.: Quicknat: Segmenting MRI neuroanatomy in 20 seconds. CoRR **abs/1801.04161** (2018)
13. Ambellan, F., Tack, A., Ehlke, M., Zachow, S.: Automated segmentation of knee bone and cartilage combining statistical shape knowledge and convolutional neural networks: Data from the osteoarthritis initiative (supplementary material) (2019) OAI-ZIB dataset.
14. Isensee, F., et al.: nnU-Net: Self-adapting framework for U-Net-Based medical image segmentation. CoRR **abs/1809.10486** (2018)

Supplementary Material

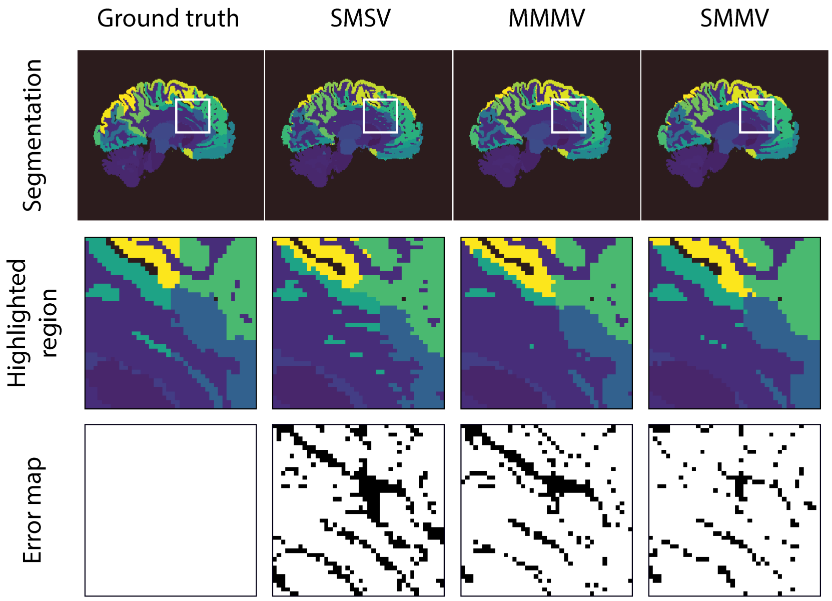

Fig. S.1. Visual comparison of the typical performance improvements obtained on a random subject of the MICCAI dataset when going from a single U-Net model fit to a single plane (single-model-single-view, SMSV, second column) to an ensemble of such models (multi-model-multi-view, MMMV, third column) to the MPUnet (single-model-multi-view, SMMV, fourth column). The first row shows the full segmentation on a single 2D slice. The second row presents a zoom of the highlighted region shown in each image of row 1. The third row shows a binary error-map for the highlighted region with black pixels representing errors compared to the ground truth and white pixels representing correctly classified pixels.

Table S.1. Fixed hyperparameter set for the optimization of the MPUnet core model on any segmentation task.

Parameter	Value	Notes
Optimizer	Adam	The global learning rate is reduced by 10 % for every 2 consecutive epochs without validation performance improvements.
Learning rate -	$5 \cdot 10^{-5}$	
β_1 -	0.9	
β_2 -	0.999	
ϵ -	$1 \cdot 10^{-8}$	
Loss function	Cross entropy	
Regularization -	None	
Class balancing -	None	
Model Topology	2D U-Net	The input dimensions are inferred based on the sizes of the images of the training data cohort. The range of 128-512 is appropriate for typical compute systems, but may be expanded to work on larger images. Generalization properties outside of this suggested range have not been tested. Note that small images volumes may be oversampled.
Input dim -	128-512	
Depth -	4	
Up-sampling -	Nearest neighbour	
Activations -	ReLU	
Conv. kernel size -	3×3	
Max-pool kernel size -	2×2	
Padding -	True (’same’)	
Batch normalization -	True	
Parameters -	$6.2 \cdot 10^7$	
Image sampling	Multi-Planar	Plane unit vectors are sampled uniformly from the 3-sphere with at least 60 deg angle between them.
Image interp -	Tri-linear	
Label interp -	Nearest-neighbour	
Num. planes -	6	
Non-linear aug.	RED*	Strength and smoothness sampled on-the-fly to produce variable deformations. *Random Elastic Deformations.
Strength, α -	uniform(100, 500)	
Elasticity, σ -	uniform(20, 30)	
Apply prob. -	1/3	
Loss weight -	1/3	
Pre-processing	Robust scaling	Image- and channel-wise scaling to (non-background) intensity distribution of median 0 and IQR 1.
Post-processing	None	
Batch size	8-16	16 by default, reduced by 2 until batches fit in GPU memory. A fraction of 1 minus the mean validation recall of a batch must contain non-background images (≥ 1 pixel of class $\neq 0$).
Foreground fraction -	1 - recall	
Training epochs	∞	Training continues until 15 consecutive epochs of without validation performance improvements.
Train images/epoch -	2500	
Val. images/epoch -	3500	
Early stopping criteria	Validation F1	Mean per-class F1 scores (excluding background) computed over all images of a validation epoch.
Model selection criteria	Validation F1	

Table S.2. F1 improvement on the MICCAI and MSD Task 4 datasets for a MPUnet of 2-9 planes relative to the mean performance of 9 single-plane models each fit to 1 of the 9 planes of the 9-plane MPUnet model. While the absolute performance benefit of using higher numbers of planes vary between the two tasks, the gains are monotonically increasing with views across both. Note that these results are only guiding as the experiments were conducted just once for each MPUnet.

Num. planes, $i =$	9	8	7	6	5	4	3	2
MICCAI	0.041	0.037	0.037	0.035	0.029	0.024	0.015	0.012
MSD T4	0.017	0.017	0.016	0.015	0.015	0.014	0.013	0.012

Table S.3. Mean F1 performance on the MICCAI dataset for MPUnets of $i \in \{3, 6, 9\}$ planes compared to ensembles of individual single-plane model each trained on a unique plane. Each single-plane model is optimized under the same set of hyperparameter as the MPUnet. Note that the single-planar ensembles have i times the parameters of their MPUnet counterparts divided evenly across its i sub-models.

Num. planes, $i =$	9	6	3
Single-Planar Ensemble	0.717 ± 0.019	0.714 ± 0.021	0.710 ± 0.024
Multi-Planar U-Net	0.743 ± 0.028	0.737 ± 0.027	0.717 ± 0.030

Table S.4. Detailed report of the MPUnet mean and standard deviation F1 (dice) performance on individual target classes across the 10 tasks of the Medical Segmentation Decathlon.

Dataset	Description	Class	F1 Score
Task 1	Brain Tumours	Edema	0.70 ± 0.20
		Non-enhancing tumor	0.43 ± 0.31
		Enhancing tumour	0.67 ± 0.22
Task 2	Cardiac	Left atrium	0.89 ± 0.09
Task 3	Liver & Tumour	Liver	0.94 ± 0.03
		Cancer	0.57 ± 0.32
Task 4	Hippocampus ROI.	Anterior	0.90 ± 0.03
		Posterior	0.88 ± 0.04
Task 5	Prostate	Peripheral zone	0.69 ± 0.13
		Transition zone	0.86 ± 0.07
Task 6	Lung Tumours	Cancer	0.59 ± 0.23
Task 7	Pancreas & Tumour	Pancreas	0.71 ± 0.14
		Cancer	0.25 ± 0.27
Task 8	Hepatic Ves. & Tumour	Vessel	0.59
		Tumour	0.38
Task 9	Spleen	Spleen	0.95
Task 10	Colon Cancer	Cancer primaries	0.28

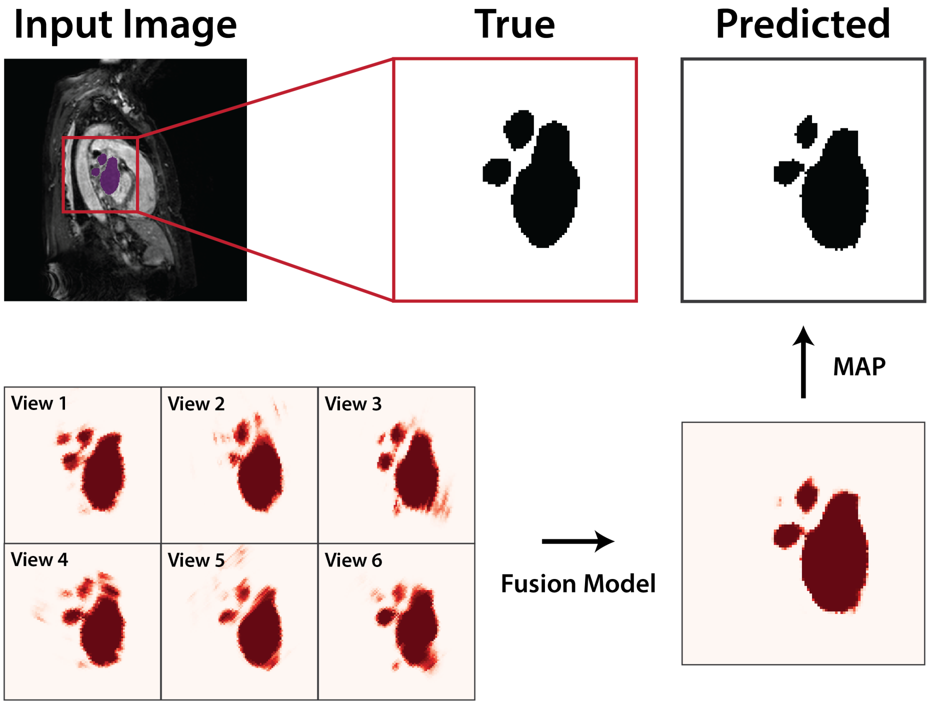

Fig. S.2. Visualization of the benefit of the MPUNet test-time augmentation approach. A 2D slice from an input image is shown in the upper left panel with a highlighted region of interest to the right giving the ground truth (binary) label map for the left atrium of an image in the Medical Segmentation Decathlon Task 4 dataset. A single MPUnet predicts on the entire image volume along 6 planes and maps the predictions to the input image space, producing a set of 6 segmentation volumes. For each of those, the corresponding slice to the input image is shown in the lower left panel. Darker red colors indicate higher confidence of the model in the foreground class at the given pixel as seen in a given view. Note that while each confidence map matches the ground truth to a large extend, the model has both false positive and false negative confidence in certain areas of individual views. After passing the 6 segmentation maps through the fusion model (lower right), a much cleaner output is produced, which after thresholding (upper right) coresponds well to the ground truth.

Table S.5. Comparison of the Multi-Planar UNet and a 3D UNet of identical topology (all 2D operations replaced by 3D operations) on the three non-challenge benchmark datasets MICCAI, HaRP and OAI as well as the Medical Segmentation Decathlon (MSD) Task 4 dataset (hippocampus in region-of-interest). The two models were trained under identical optimization parameters. The shown scores are mean per-class F1 scores pooled across three separate training and evaluation sessions. The MSD Task 4 dataset experiments were conducted on random splits of the challenge training data, as we do not have access to the test set. The 3D UNet was trained on isotropic ROIs of 64-cube voxels with random rotations and 3D random elastic deformations applied at batch-sampling time. This was done to emulate the benefit of the MPUNet's significant data augmentation. The sampled voxel-resolution was identical to that chosen for the MPUNet. The 3D model has a total of 90 million parameters against the 62 of the MPUnet. The MSD Task 4 dataset consists of small cut-out regions of interest spanning narrowly around the hippocampus to segment, and was include here to study the performance of the 3D model when the entire input image fits within the 64-cube input patch. **Note:** The OAI dataset used for those experiments was a smaller subset of the full dataset for which results are displayed in Table 1 (no follow-up scans included, specifically).

	MICCAI	HaRP	OAI	MSD T4
3D U-Net w. rotations	0.74 ± 0.04	0.84 ± 0.05	0.81 ± 0.07	0.87 ± 0.04
Multi-Planar U-Net	0.74 ± 0.03	0.85 ± 0.03	0.84 ± 0.07	0.88 ± 0.04

Table S.6. MPUnet base model topology (U-Net type) for images sampled with pixel dim $q = 256$. Note: Convolution strides of 1×1 where used in all layers.

Layer name	Output dim	Kernel dim	Filters	Activation	Pad
Input	$256 \times 256 \times C$	-	-	-	-
conv_1_1	$256 \times 256 \times 90$	3×3	90	ReLU	same
conv_1_2	$256 \times 256 \times 90$	3×3	90	ReLU	same
bn_1	$256 \times 256 \times 90$	-	-	-	-
pool_1	$128 \times 128 \times 90$	2×2	-	-	valid
conv_2_1	$128 \times 128 \times 181$	3×3	181	ReLU	same
conv_2_2	$128 \times 128 \times 181$	3×3	181	ReLU	same
bn_2	$128 \times 128 \times 181$	-	-	-	-
pool_2	$64 \times 64 \times 181$	2×2	-	-	valid
conv_3_1	$64 \times 64 \times 362$	3×3	362	ReLU	same
conv_3_2	$64 \times 64 \times 362$	3×3	362	ReLU	same
bn_3	$64 \times 64 \times 362$	-	-	-	-
pool_3	$32 \times 32 \times 362$	2×2	-	-	valid
conv_4_1	$32 \times 32 \times 724$	3×3	724	ReLU	same
conv_4_2	$32 \times 32 \times 724$	3×3	724	ReLU	same
bn_4	$32 \times 32 \times 724$	-	-	-	-
pool_4	$16 \times 16 \times 724$	2×2	-	-	valid
conv_5_1	$16 \times 16 \times 1448$	3×3	1448	ReLU	same
conv_5_2	$16 \times 16 \times 1448$	3×3	1448	ReLU	same
up_1	$32 \times 32 \times 1448$	2×2	-	-	-
conv_6_0	$32 \times 32 \times 724$	2×2	724	ReLU	same
bn_6	$32 \times 32 \times 724$	-	-	-	-
merge(bn4, bn6)	$32 \times 32 \times 1448$	-	-	-	-
conv_6_1	$32 \times 32 \times 724$	3×3	724	ReLU	same
conv_6_2	$32 \times 32 \times 724$	3×3	724	ReLU	same
bn_7	$32 \times 32 \times 724$	-	-	-	-
up_2	$64 \times 64 \times 724$	2×2	-	-	-
conv_7_0	$64 \times 64 \times 362$	2×2	362	ReLU	same
bn_8	$64 \times 64 \times 362$	-	-	-	-
merge(bn3, bn8)	$64 \times 64 \times 724$	-	-	-	-
conv_7_1	$64 \times 64 \times 362$	3×3	362	ReLU	same
conv_7_2	$64 \times 64 \times 362$	3×3	362	ReLU	same
bn_9	$64 \times 64 \times 362$	-	-	-	-
up_3	$128 \times 128 \times 362$	2×2	-	-	-
conv_8_0	$128 \times 128 \times 181$	2×2	181	ReLU	same
bn_10	$128 \times 128 \times 181$	-	-	-	-
merge(bn2, bn10)	$128 \times 128 \times 362$	-	-	-	-
conv_8_1	$128 \times 128 \times 181$	3×3	181	ReLU	same
conv_8_2	$128 \times 128 \times 181$	3×3	181	ReLU	same
bn_11	$128 \times 128 \times 181$	-	-	-	-
up_4	$256 \times 256 \times 181$	2×2	-	-	-
conv_9_0	$256 \times 256 \times 90$	2×2	90	ReLU	same
bn_12	$256 \times 256 \times 90$	-	-	-	-
merge(bn1, bn12)	$256 \times 256 \times 180$	-	-	-	-
conv_9_1	$256 \times 256 \times 90$	3×3	90	ReLU	same
conv_9_2	$256 \times 256 \times 90$	3×3	90	ReLU	same
bn_13	$256 \times 256 \times 90$	-	-	-	-
output	$256 \times 256 \times K$	1×1	K	softmax	-

Trainable parameters: $62,062,342$ **(for** $K = 135$**,** $C = 1$**)**