nature research

https://doi.org/10.1038/s41592-020-01008-z

Supplementary information

nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation

In the format provided by the authors and unedited

Supplementary Notes

This document contains supplementary information for the manuscript 'Automated Design of Deep Learning Methods for Biomedical Image Segmentation'.

1 Dataset details

Table SN1.1 provides an overview of the datasets used in this manuscript including respective references for data access. The numeric values presented here are computed based on the training cases for each of these datasets. They are the basis of the dataset fingerprints presented in Figure 5.

ID	Dataset Name	Associated Challenges	Modalities	Median Shape (Spacing [mm])	N Class.	Rarest Class Ratio	N Training Cases	Segmentation Tasks
D1	Brain Tumour	[1, 2]	MRI (T1, T1c, T2, FLAIR)	138x169x138 (1, 1, 1)	3	$7.3 \cdot 10^{-3}$	484	edema, active tumor, necrosis
D2	Heart	[1]	MRI	115x320x232 (1.37, 1.25, 1.25)	1	$4.0 \cdot 10^{-3}$	20	left ventricle
D3	Liver	[1, 3]	CT	432x512x512 (1, 0.77, 0.77)	2	$2.6 \cdot 10^{-2}$	131	liver, liver tumors
D4	Hippocampus	[1]	MRI	36x50x35 (1, 1, 1)	2	$2.7 \cdot 10^{-2}$	260	anterior and posterior hippocampus
D5	Prostate	[1]	MRI (T2, ADC)	20x320x319 (3.6, 0.62, 0.62)	2	$5.4 \cdot 10^{-3}$	32	peripheral and transition zone
D6	Lung	[1]	CT	252x512x512 (1.24, 0.79, 0.79)	1	$3.9 \cdot 10^{-4}$	63	lung nodules
D7	Pancreas	[1]	CT	93x512x512 (2.5, 0.80, 0.80)	2	$2.0 \cdot 10^{-3}$	282	pancreas, pancreas cancer
D8	HepaticVessel	[1]	CT	49x512x512 (5, 0.80, 0.80)	2	$1.1 \cdot 10^{-3}$	303	hepatic vessels, tumors
D9	Spleen	[1]	CT	90x512x512 (5, 0.79, 0.79)	1	$4.7 \cdot 10^{-3}$	41	spleen
D10	Colon	[1]	CT	95x512x512 (5, 0.78, 0.78)	1	$5.6 \cdot 10^{-4}$	126	colon cancer
D11	AbdOrgSeg	[4]	CT	128x512x512 (3, 0.76, 0.76)	13	$4.4 \cdot 10^{-3}$	30	13 abdominal organs
D12	Promise	[5]	MRI	24x320x320 (3.6, 0.61, 0.61)	1	$2.0 \cdot 10^{-2}$	50	prostate
D13	ACDC	[6]	cine MRI	9x256x216 (10, 1.56, 1.56)	3	$1.2 \cdot 10^{-2}$	200 (100x2) *	left ventricle, right ventricle, myocardium
D14	LiTS **	[3]	CT	432x512x512 (1, 0.77, 0.77)	2	$2.6 \cdot 10^{-2}$	131	liver, liver tumors
D15	MSLesion	[7]	MRI (FLAIR, MPRAGE, PD, T2)	137x180x137 (1, 1, 1)	1	$1.7 \cdot 10^{-3}$	42 (21x2) *	multiple sclerosis lesions
D16	CHAOS	[8]	MRI	30x204x256 (9, 1.66, 1.66)	4	$3.3 \cdot 10^{-2}$	60 (20 + 20x2) *	liver, spleen, left and right kidney
D17	KiTS	[9]	CT	107x512x512 (3, 0.78, 0.78)	2	$7.5 \cdot 10^{-3}$	206	kidney, kidney tumor
D18	SegTHOR	[10]	CT	178x512x512 (2.5, 0.98, 0.98)	4	$4.6 \cdot 10^{-4}$	40	heart, aorta, esophagus, trachea
D19	CREMI	[11]	Electron Microscopy	125x1250x1250 $(4 \cdot 10^{-5}, 4 \cdot 10^{-6}, 4 \cdot 10^{-6})$	1	$5.2 \cdot 10^{-3}$	3	synaptic clefts
D20	Fluo-N2DH-SIM+	[12, 13]	Fluorescence Microscopy	773x739 $(1.25 \cdot 10^{-4}, 1.25 \cdot 10^{-4})$	1	$9.5 \cdot 10^{-2}$	215	HL60 nuclei
D21	Fluo-N3DH-SIM+	[12, 13]	Fluorescence Microscopy	59x349x639 $(2 \cdot 10^{-4}, 1.25 \cdot 10^{-4}, 1.25 \cdot 10^{-4})$	1	$5.9 \cdot 10^{-2}$	230	HL60 nuclei
D22	Fluo-C3DH-A549	[12, 13]	Fluorescence Microscopy	28.5x285x375 $(1 \cdot 10^{-3}, 1.26 \cdot 10^{-4}, 1.26 \cdot 10^{-4})$	1	$2.2 \cdot 10^{-2}$	30	A549 cell
D23	Fluo-C3DH-A549-SIM+	[12, 13]	Fluorescence Microscopy	33x300x375 $(1 \cdot 10^{-4}, 1.26 \cdot 10^{-4}, 1.26 \cdot 10^{-4})$	1	$1.5 \cdot 10^{-2}$	60	A549 cell

* multiple annotated examples per training case
** almost identical to Decathlon Liver; Decathlon changed the training cases and test set slightly

Table SN1.1: Overview over the challenge datasets used in this manuscript.

2 nnU-Net design principles

Here we present a brief overview of the design principles of nnU-Net on a conceptual level. Please refer to the online methods for a more detailed information on how these guidelines are implemented.

2.1 Fixed parameters

- Architecture Design decisions:
  - U-Net like architectures enable state of the art segmentation when the pipeline is well-configured. According to our experience, sophisticated architectural variations are not required to achieve state of the art performance.
  - Our architectures only use plain convolutions, instance normalization and Leaky non-linearities. The order of operations in each computational block is conv - instance norm - leaky ReLU.
  - We use two computational blocks per resolution stage in both encoder and decoder.
  - Downsampling is done with strided convolutions (the convolution of the first block of the new resolution has stride >1), upsampling is done with convolutions transposed. We should note that we did not observe substantial disparities in segmentation accuracy between this approach and alternatives (e.g. max pooling, bi/trilinear upsampling).
- Selecting the best U-Net configuration: It is difficult to estimate which U-Net configuration performs best on what dataset. To address this, nnU-Net designs three separate configurations and automatically chooses the best one based on cross-validation (see rule-based parameters). Predicting which configurations should be trained on which dataset is a future research direction.
  - 2D U-Net: Runs on full resolution data. Expected to work well on anisotropic data, such as D5 (Prostate MRI) and D13 (ACDC, cine MRI) (for dataset references see Table SN1.1).
  - 3D full resolution U-Net: Runs on full resolution data. Patch size is limited by availability of GPU memory. Is overall the best performing configuration (see results in 6). For large data, however, the patch size may be too small to aggregate sufficient contextual information.
  - 3D U-Net cascade: Specifically targeted towards large data. First, coarse segmentation maps are learned by a 3D U-Net that operates on low resolution data. These segmentation maps are then refined by a second 3D U-Net that operates on full resolution data.
- Training Scheme
  - All trainings run for a fixed length of 1000 epochs, where each epoch is defined as 250 training iterations (using the batch size configured by nnU-Net). Shorter trainings than this default empirically result in diminished segmentation performance.
  - As for the opimizer, stochastic gradient descent with a high initial learning rate (0.01) and a large nesterov momentum (0.99) empirically provided the best results. The learning rate is reduced during the training using the 'polyLR' schedule as described in [14], which is an almost linear decrease to 0.
  - Data augmentation is essential to achieve state of the art performance. It is important to run the augmentations on the fly and with associated probabilities to obtain a never ending stream of unique examples (see Section 4 for details).

  - Data in the biomedical domain suffers from class imbalance. Rare classes could end up being ignored because they are underrepresented during training. Oversampling foreground regions addresses this issue reliably. It should, however, not be overdone so that the network also sees all the data variability of the background.
  - The Dice loss function is well suited to address the class imbalance, but comes with its own drawbacks. Dice loss optimizes the evaluation metric directly, but due to the patch based training, in practice merely approximates it. Furthermore, oversampling of classes skews the class distribution seen during training. Empirically, combining the Dice loss with a cross-entropy loss improved training stability and segmentation accuracy. Therefore, the two loss terms are simply averaged.
- Inference
  - Validation sets of all folds in the cross-validation are predicted by the single model trained on the respective training data. The 5 models resulting from training on 5 individual folds are subsequently used as an ensemble for predicting test cases.
  - Inference is done patch based with the same patch size as used during training. Fully convolutional inference is not recommended because it causes issues with zero-padded convolutions and instance normalization.
  - To prevent stitching artifacts, adjacent predictions are done with a distance of patch_size / 2. Predictions towards the border are less accurate, which is why we use Gaussian importance weighting for softmax aggregation (the center voxels are weighted higher then the border voxels).

2.2 Rule-based parameters

These parameters are not fixed across datasets, but configured on-the-fly by nnU-Net according to the data fingerprint (low dimensional representation of dataset properties) of the task at hand.

- Dynamic Network adaptation:
  - The network architecture needs to be adapted to the size and spacing of the input patches seen during training. This is necessary to ensure that the receptive field of the network covers the entire input.
  - We perform downsampling until the feature maps are relatively small (minimum is $4 \times 4 (\times 4)$) to ensure sufficient context aggregation.
  - Due to having a fixed number of blocks per resolution step in both the encoder and decoder, the network depth is coupled to its input patch size. The number of convolutional layers in the network (excluding segmentation layers) is $(5 * k + 2)$ where k is the number of downsampling operations (5 per downsampling stems from 2 convs in the encoder, 2 in the decoder plus the convolution transpose).
  - Additional loss functions are applied to all but the two lowest resolutions of the decoder to inject gradients deep into the network.
  - For anisotropic data, pooling is first exclusively performed in-plane until the resolution matches between the axes. Initially, 3D convolutions use a kernel size of 1 (making them effectively 2D convolutions) in the out of plane axis to prevent aggregation of information across distant slices. Once an axes becomes too small, downsampling is stopped individually for this axis.
- Configuration of the input patch size:

  - Should be as large as possible while still allowing a batch size of 2 (under a given GPU memory constraint). This maximizes the context available for decision making in the network.
  - Aspect ratio of patch size follows the median shape (in voxels) of resampled training cases.

- Batch size:
  - Batch size is configured with a minimum of 2 to ensure robust optimization, since noise in gradients increases with fewer sample in the minibatch.
  - If GPU memory headroom is available after patch size configuration, the batch size is increased until GPU memory is maxed out.

- Target spacing and resampling:
  - For isotropic data, the median spacing of training cases (computed independently for each axis) is set as default. Resampling with third order spline (data) and linear interpolation (one hot encoded segmentation maps such as training annotations) give good results.
  - For anisotropic data, the target spacing in the out of plane axis should be smaller than the median, resulting in higher resolution in order to reduce resampling artifacts. To achieve this we set the target spacing as the 10th percentile of the spacings found for this axis in the training cases. Resampling across the out of plane axis is done with nearest neighbor for both data and one-hot encoded segmentation maps.

- Intensity normalization:
  - Z-score per image (mean substraction and division by standard deviation) is a good default.
  - We deviate from this default only for CT images, where a global normalization scheme is determined based on the intensities found in foreground voxels across all training cases.

2.3 Empirical parameters

Some parameters cannot be inferred by simply looking at the dataset fingerprint of the training cases. These are determined empirically by monitoring validation performance after training.

- Model selection: While the 3D full resolution U-Net shows overall best performance, selection of the best model for a specific task at hand can not be predicted with perfect accuracy. Therefore, nnU-Net generates three U-Net configurations and automatically picks the best performing method (or ensemble of methods) after cross-validation.

- Postprocessing: Often, particularly in medical data, the image contains only one instance of the target structure. This prior knowledge can often be exploited by running connected component analysis on the predicted segmentation maps and removing all but the largest component. Whether to apply this postprocessing is determined by monitoring validation performance after cross-validation. Specifically, postprocessing is triggered for individual classes where the Dice score is improved by removing all but the largest component.

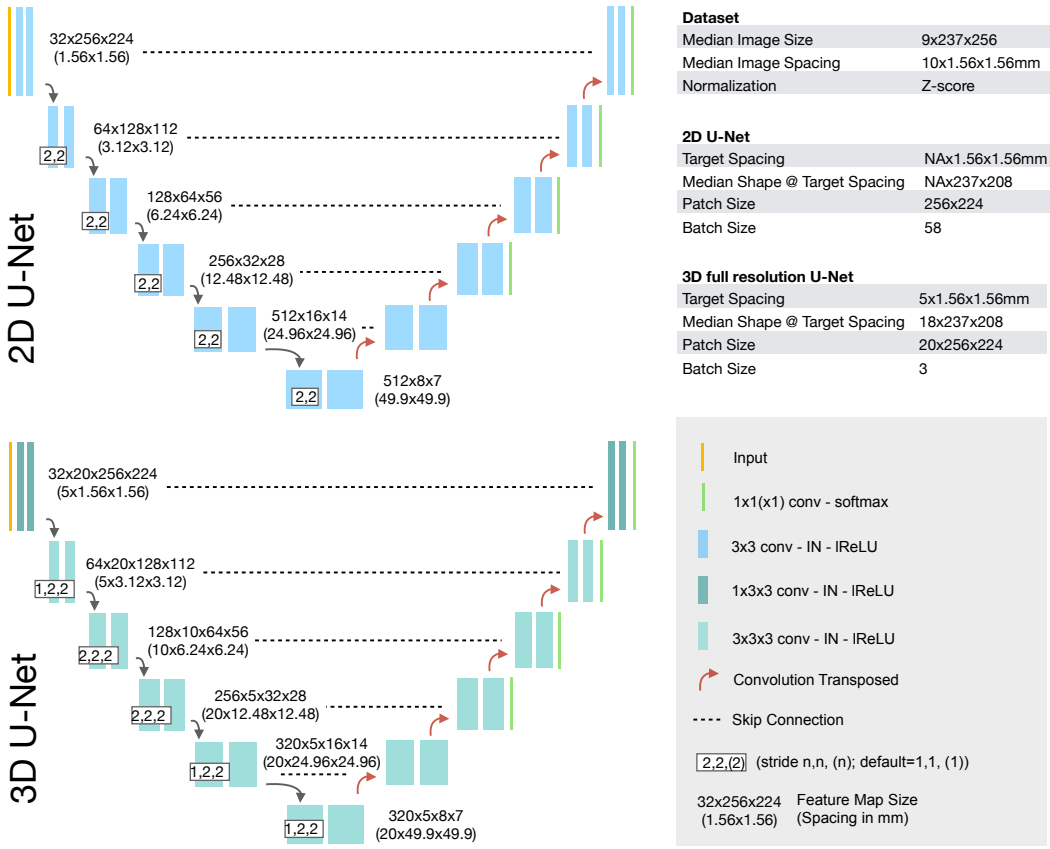

Figure SN3.1: Network architectures generated by nnU-Net for the ACDC dataset (D13)

3 Analysis of exemplary nnU-Net-generated pipelines

In this section we briefly introduce the pipelines generated by nnU-Net for D13 (ACDC) and D14 (LiTS) to create an intuitive understanding of nnU-Nets design principles and the motivation behind them.

3.1 ACDC

Figure SN3.1 provides a summary of the pipelines that were automatically generated by nnU-Net for this dataset.

Dataset description The Automated Cardiac Diagnosis Challenge (ACDC) [6] was hosted by MICCAI in 2017. Since then it is running as an open challenge with data and current leaderboard available at `https://acdc.creatis.insa-lyon.fr`. In the segmentation part of the challenge, participating teams were asked to generate algorithms for segmenting the right ventricle, the left myocardium and the left ventricular cavity from cine MRI. For each patient, reference segmentations for two time steps within the cardiac cycle were provided. With 100 training patients, this amounts to a total of 200 annotated images. One key property of cine MRI is that slice acquisition takes place across multiple cardiac cycles and breath holds. This results in a limited number of slices and thus a low out of plane resolution as well as the possibility for slice misalignments. Figure SN3.1 provides a summary of the pipelines that were automatically generated by nnU-Net for this dataset. The typical

image shape (here the median image size is computed for each axis independently) is $9 \times 237 \times 256$ voxels at a spacing of $10 \times 1.56 \times 1.56\,\mathrm{mm}$.

Intensity normalization With the images being MRI, nnU-Net normalizes all images individually by subtracting their mean and dividing by their standard deviation.

2D U-Net As target spacing for the in-plane resolution, $1.56 \times 1.56\,\mathrm{mm}$ is determined. This is identical for the 2D and the 3D full resolution U-Net. Due to the 2D U-Net operating on slices only, the out of plane resolution for this configuration is not altered and remains heterogeneous within the training set. The 2D U-Net is configured as described in the Online Methods 4 to have a patch size of 256×224 voxels, which fully covers the typical image shape after in-plane resampling (237×208). 3D U-Net The size and spacing anisotropy of this dataset causes the out-of-plane target spacing of the 3D full resolution U-Net to be selected as 5mm, corresponding to the 10th percentile of the spacings found in the training cases. In datasets such as ACDC, the segmentation contour can change substantially between slices due to the large slice to slice distance. Choosing the target spacing to be lower results in more images that are upsampled for U-Net training and then downsampled for the final segmentation export. Preferring this variant over the median causes more images to be downsampled for training and then upsampled for segmentation export and therefore reduces interpolation artifacts substantially. Also note that resampling the out of plane axis is done with nearest neighbor interpolation.The median image shape after resampling for the 3D full resolution U-Net is $18 \times 237 \times 208$ voxels. As described in the Online Methods 4 nnU-Net configures a patch size of $20 \times 256 \times 224$ for network training, which fits into the memory budget with a batch size of 3. Note how the convolutional kernel sizes in the 3D U-Net start with $(1 \times 3 \times 3)$ which is effectively a 2D convolution for the initial layers (see also Figure SN3.1). The reasoning behind this is that due to the large discrepancy in voxel spacing, too many changes are expected across slices and the aggregation of imaging information may therefore not be beneficial. Similarly, pooling is done in-plane only (conv kernel stride (1, 2, 2)) until the spacing between in-plane and out-of-plane axes are within a factor of 2. Only after the spacings approximately match the pooling and the convolutional kernel sizes become isotropic.

3D U-Net cascade Since the 3D U-Net already covers the whole median image shape, the U-Net cascade is not necessary and therefore omitted.

Training and Postprocessing During training, spatial augmentations for the 3D U-Net (such as scaling and rotation) are done in-plane only to prevent resampling of imaging information across slices which would cause interpolation artifacts. Each U-Net configuration is trained in a five-fold cross-validation on the training cases. Note that we interfere with the splits in order to ensure that patients are properly stratified (since there are two images per patient). Thanks to the cross-validation, nnU-Net can use the entire training set for validation and ensembling. To this end, the validation splits of each of the five fold are aggregated. nnU-Net evaluates the performance (ensemble of models or single configuration) by averaging the Dice scores over all foreground classes and cases, resulting in a single scalar value. Detailed results are omitted here for brevity (they are presented in Supplementary Information 6). Based on this evaluation scheme, the 2D U-Net obtains a score of 0.9165, the 3D full resolution a score of 0.9181 and the ensemble of the two a score of 0.9228. Therefore the ensemble is selected for predicting the test cases. Postprocessing is configured on the segmentation maps of the ensemble. Removing all but the largest connected component was found beneficial for the right ventricle and the left ventricular cavity.

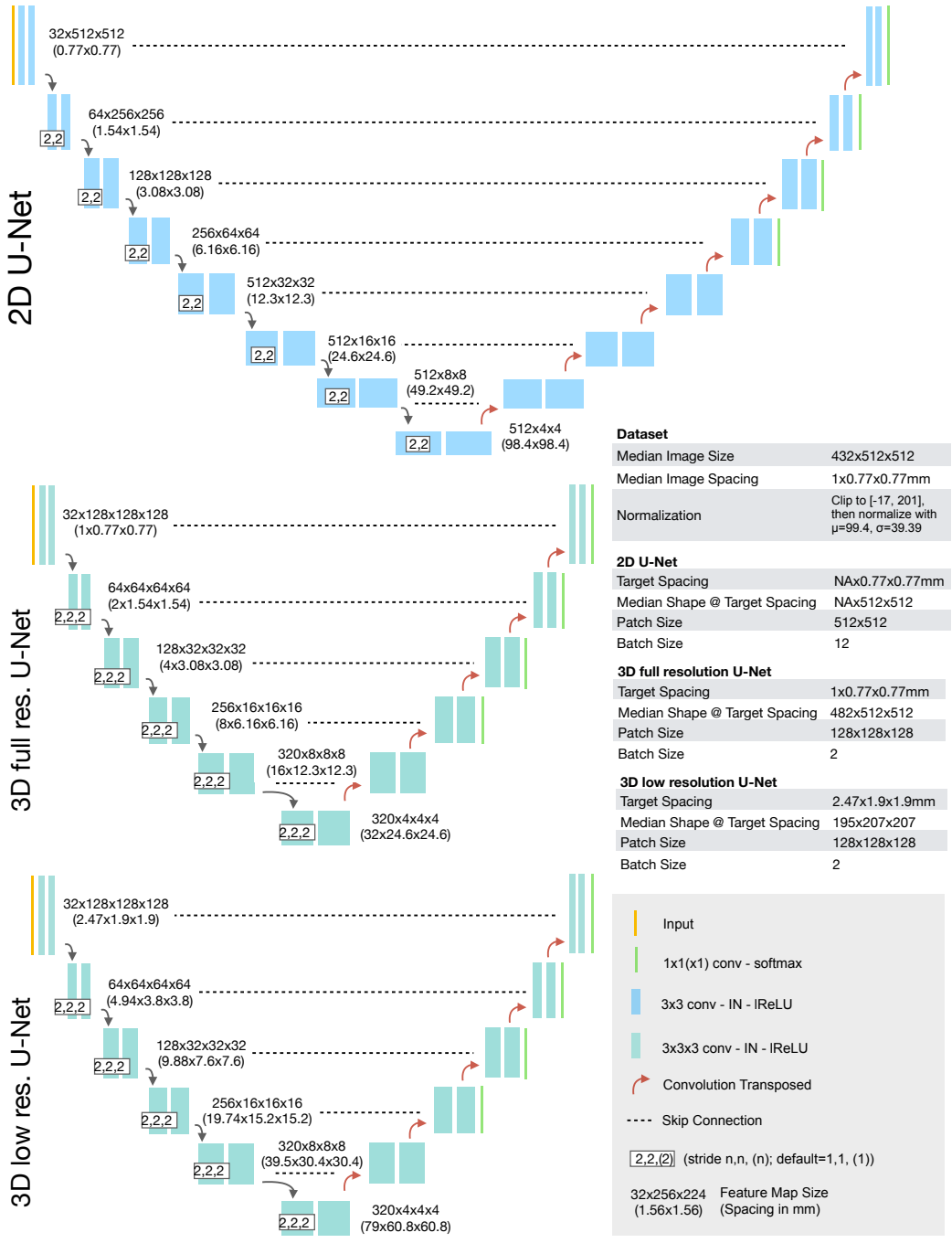

Figure SN3.2: Network architectures generated by nnU-Net for the LiTS dataset (D14)

3.2 LiTS

Figure SN3.2 provides a summary of the pipelines that were automatically generated by nnU-Net for this dataset.

Dataset description The Liver and Liver Tumor Segmentation challenge (LiTS) [3] was hosted by MICCAI in 2017. Due to the large, high quality dataset it provides, the challenge plays an important role in concurrent research. The challenge is hosted at https://competitions.codalab.org/competitions/17094. The segmentation task in LiTS is the segmentation of the liver and liver tumors in abdominal CT scans. The challenge provides 131 training cases with reference annotations. The test set has a size of 70 cases and the reference annotations are known only to the challenge organizers. The median image shape of the training cases is $432 \times 512 \times 512$ voxels with a corresponding voxel spacing of $1 \times 0.77 \times 0.77\,\mathrm{mm}$.

Intensity normalization Voxel intensities in CT scans are linked to quantitative physical properties of the tissue. The intensities are therefore expected to be consistent between scanners. nnU-Net leverages this consistency by applying a global intensity normalization scheme (as opposed to ACDC in Supplementary Information 3.1, where cases are normalized individually using their mean and standard deviation). To this end, nnU-Net extracts intensity information as part of the dataset fingerprint: the intensities of the voxels belonging to any of the foreground classes (liver and liver tumor) are collected across all training cases. Then, the mean and standard deviations of these values as well as their 0.5 and 99.5 percentiles are computed. Subsequently, all images are normalized by clipping them to the 0.5 and 99.5 percentiles, followed by subtraction of the global mean and division by the global standard deviation.

2D U-Net The target spacing for the 2D U-Net is determined to be $NA \times 0.77 \times 0.77\,\mathrm{mm}$, which corresponds to the median voxel spacing encountered in the training cases. Note that the 2D U-Net operates on slices only, so the out of plane axis is left untouched. Resampling the training cases results in a median image shape of $NA \times 512 \times 512$ voxels (we indicate by NA that this axis is not resampled). Since this is the median shape, cases in the training set can be smaller or larger than that. The 2D U-Net is configured to have an input patch size of 512×512 voxels and a batch size of 12.

3D U-Net The target spacing for the 3D U-Net is determined to be $1 \times 0.77 \times 0.77\,\mathrm{mm}$, which corresponds to the median voxel spacing. Because the median spacing is nearly isotropic, nnU-Net does not use the 10th percentile for the out of plane axis as was the case for ACDC (see Supplementary Information 3.1). The resampling strategy is decided on a per-image basis. Isotropic cases (maximum axis spacing / minimum axis spacing < 3) are resampled with third order spline interpolation for the image data and linear interpolation for the segmentations. Note that segmentation maps are always converted into a one hot representation prior to resampling which is converted back to a segmentation map after the interpolation. For anisotropic images, nnU-Net resamples the out-of-plane axis separately, as was done in ACDC.

After resampling, the median image shape is $482 \times 512 \times 512$. nnU-Net prioritizes a large patch size over a large batch size (note that these are coupled under a given GPU memory budget) to capture as much contextual information as possible. The 3D U-Net is thus configured to have a patch size of $128 \times 128 \times 128$ voxels and a batch size of 2, which is the minimum allowed according to nnU-Net heuristics. Since The input patches have nearly isotropic spacing, all convolutional kernel sizes and downsampling strides are isotropic ($3 \times 3 \times 3$ and $2 \times 2 \times 2$, respectively).

3D U-Net cascade Although nnU-Net prioritizes large input patches, the patch size of the 3D full resolution U-Net is too small to capture sufficient contextual information (it only covers 1/60 of the voxels of the median image shape after resampling). This can cause misclassifications of voxels

because the patches are too 'zoomed in', making for instance the distinction between the spleen and the liver particularly hard. The 3D U-Net cascade is designed to tackle this problem by first training a 3D U-Net on downsampled data and then refining the low-resolution segmentation output with a second U-Net that operates as full resolution. Using the process described in the Online Methods 4 as well as Figure SN5.1 b), the target spacing for the low resolution U-Net is determined to be $2.47 \times 1.9 \times 1.9\,\mathrm{mm}$, resulting in a median image shape of $195 \times 207 \times 207$ voxels. The 3D low resolution operates on $128 \times 128 \times 128$ patches with a batch size of 2. Note that while this setting is identical to the 3D U-Net configuration here, this is not necessarily the case for other datasets. If the 3D full resolution U-Net data was anisotropic, nnU-Net would prioritize to downsample the higher resolution axes first resulting in a deviating network architecture, patch size and batch size. After five-fold cross-validation of the 3D low resolution U-Net, the segmentation maps of the respective validation sets are upsampled to the target spacing of the 3D full resolution U-Net. The full resolution U-Net of the cascade (which has an identical configuration to the regular 3D full resolution U-Net) is then trained to refine the coarse segmentation maps and correct any errors it encounters. This is done by concatenating a one hot encoding of the upsampled segmentations to the input of the network.

Training and postprocessing All network configurations are trained as five fold cross-validation. nnU-Net again evaluates all configurations by computing the average Dice score across all foreground classes, resulting in a scalar metric per configuration. Based on this evaluation scheme, the scores are 0.7625 for the 2D U-Net, 0.8044 for the 3D full resolution U-Net, 0.7796 for the 3D low resolution U-Net and 0.8017 for the full resolution 3D U-Net of the cascade. The best combination of two models was identified as the one between the low and full resolution U-Nets with a score of 0.8111. Postprocessing is configured on the segmentation maps of this ensemble. Removing all but the largest connected component was found beneficial for the combined foreground region (union of liver and liver tumor label) as well as for the liver label alone, as both resulted in small performance gains when empirically testing it on the training data.

4 Details on nnU-Net's data augmentation

A variety of data augmentation techniques is applied during training. All augmentations are computed on the fly on the CPU using background workers. The data augmentation pipeline is implemented with the publicly available *batchgenerators* framework [4]. nnU-Net does not vary the parameters of the data augmentation pipeline between datasets.

Sampled patches are initially larger than the patch size used for training. This results in less out of boundary values (here 0) being introduced during data augmentation when rotation and scaling is applied. As a part of the rotation and scaling augmentation, patches are center-cropped to the final target patch size. To ensure that the borders of original images appear in the final patches, preliminary crops may initially extend outside the boundary of the image.

Spatial augmentations (rotation, scaling, low resolution simulation) are applied in 3D for the 3D U-Nets and applied in 2D when training the 2D U-Net or a 3D U-Net with anisotropic patch size. A patch size is considered anisotropic if the largest edge length of the patch size is at least three times larger than the smallest.

[4] https://github.com/MIC-DKFZ/batchgenerators

To increase the variability in generated patches, most augmentations are varied with parameters drawn randomly from predefined ranges. In this context, $x \sim U(a, b)$ indicates that x was drawn from a uniform distribution between a and b. Furthermore, all augmentations are applied stochastically according to a predefined probability.

The following augmentations are applied by nnU-Net (in the given order):

1. **Rotation and scaling.** Scaling and rotation are applied together for improved speed of computation. This approach reduces the amount of required data interpolations to one. Scaling and rotation are applied with a probability of 0.2 each (resulting in probabilities of 0.16 for only scaling, 0.16 for only rotation and 0.08 for both being triggered). If processing isotropic 3D patches, the angles of rotation (in degrees) α_x, α_y and α_z are each drawn from $U(-30, 30)$. If a patch is anisotropic or 2D, the angle of rotation is sampled from $U(-180, 180)$. If the 2D patch size is anisotropic, the angle is sampled from $U(-15, 15)$. Scaling is implemented via multiplying coordinates with a scaling factor in the voxel grid. Thus, scale factors smaller than one result in a "zoom out" effect while values larger one result in a "zoom in" effect. The scaling factor is sampled from $U(0.7, 1.4)$ for all patch types.
2. **Gaussian noise.** Zero centered additive Gaussian noise is added to each voxel in the sample independently. This augmentation is applied with a probability of 0.15. The variance of the noise is drawn from $U(0, 0.1)$ (note that the voxel intensities in all samples are close to zero mean and unit variance due to intensity normalization).
3. **Gaussian blur.** Blurring is applied with a probability of 0.2 per sample. If this augmentation is triggered in a sample, blurring is applied with a probability of 0.5 for each of the associated modalities (resulting in a combined probability of only 0.1 for samples with a single modality). The width (in voxels) of the Gaussian kernel σ is sampled from $U(0.5, 1.5)$ independently for each modality.
4. **Brightness.** Voxel intensities are multiplied by $x \sim U(0.7, 1.3)$ with a probability of 0.15.
5. **Contrast.** Voxel intensities are multiplied by $x \sim U(0.65, 1.5)$ with a probability of 0.15. Following multiplication, the values are clipped to their original value range.
6. **Simulation of low resolution.** This augmentation is applied with a probability of 0.25 per sample and 0.5 per associated modality. Triggered modalities are downsampled by a factor of $U(1, 2)$ using nearest neighbor interpolation and then sampled back up to their original size with cubic interpolation. For 2D patches or anisotropic 3D patches, this augmentation is applied only in 2D leaving the out of plane axis (if applicable) in its original state.
7. **Gamma augmentation.** This augmentation is applied with a probability of 0.15. The patch intensities are scaled to a factor of $[0, 1]$ of their respective value range. Then, a nonlinear intensity transformation is applied per voxel: $i_{new} = i_{old}^{\gamma}$ with $\gamma \sim U(0.7, 1.5)$. The voxel intensities are subsequently scaled back to their original value range. With a probability of 0.15, this augmentation is applied with the voxel intensities being inverted prior to transformation: $(1 - i_{new}) = (1 - i_{old})^{\gamma}$.
8. **Mirroring.** All patches are mirrored with a probability of 0.5 along all axes.

For the full resolution U-Net of the U-net cascade, nnU-Net additionally applies the following augmentations to the segmentation masks generated by the low resolution 3D U-net. Note that the segmentations are stored as one hot encoding.

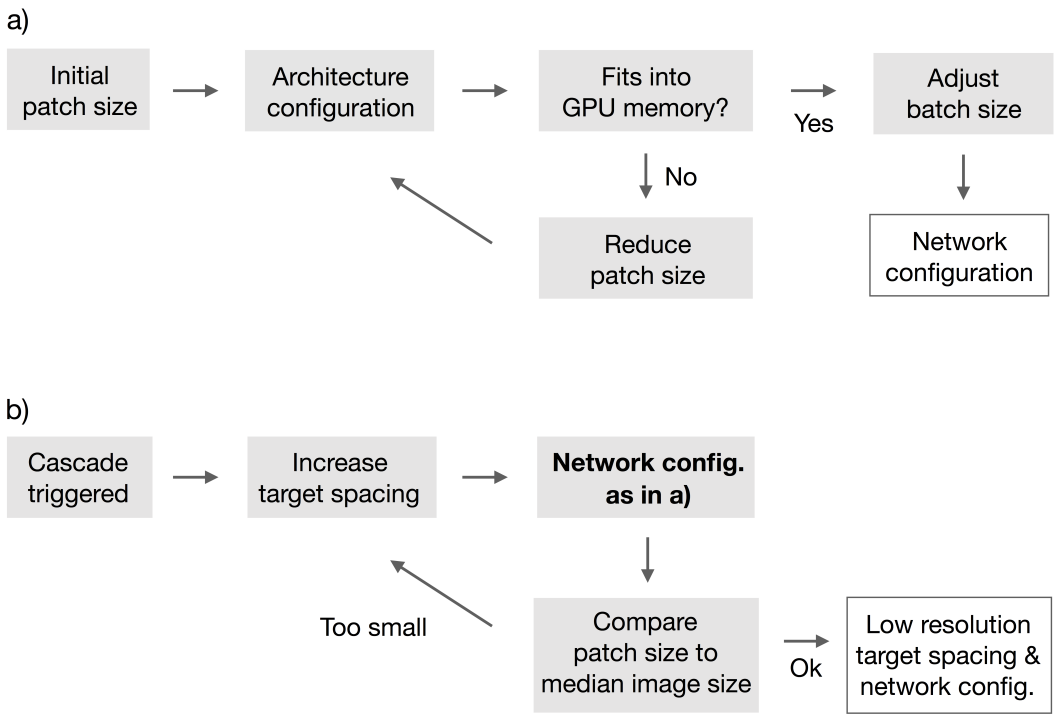

Figure SN5.1: Workflow for network architecture configuration. a) the configuration of a U-Net architecture given an input patch size and corresponding voxel spacing. Due to discontinuities in GPU memory consumption (due to changes in number of pooling operations and thus network depth), the architecture configuration cannot be solved analytically. b) Configuration of the 3D low resolution U-Net of the U-Net cascade. The input patch size of the 3D lowres U-Net must cover at least 1/4 of the median shape of the resampled trainig cases to ensure sufficient contextual information. Higher resolution axes are downsampled first, resulting in a potentially different aspect ratio of the data relative to the full resolution data. Due to the patch size following this aspect ratio, the network architecture of the low resolution U-Net may differ from the full resolution U-Net. This requires reconfiguration of the network architecture as depicted in a) for each iteration. All computations are based on memory consumption estimates resulting in fast computation times (sub 1s for configuring all network architectures).

1. **Binary operators.** With probability 0.4, a binary operator is applied to all labels in the predicted masks. This operator is randomly chosen from [dilation, erosion, opening, closing]. The structure element is a sphere with radius $r \sim U(1, 8)$. The operator is applied to the labels in random order. Hereby, the one hot encoding property is retained. Dilation of one label, for example, will result in removal of all other labels in the dilated area.

2. **Removal of connected components.** With probability 0.2, connected components that are smaller than 15% of the patch size are removed from the one hot encoding.

5 Network architecture configuration

Figure SN5.1 serves as a visual aid for the iterative process of architecture configuration described in the online methods.

6 Summary of nnU-Net challenge participations

In this section we provide details of all challenge participations.
In some participations, manual intervention regarding the format of input data or the cross-validation data splits was required for compatibility with nnU-Net. For each dataset, we disclose all manual interventions in this section. The most common cause for manual intervention was training cases that were related to each other (such as multiple time points of the same patient) and thus required to be separated for mutual exclusivity between data splits. A detailed description of how to perform this intervention is further provided along with the source code.

For each dataset, we run all applicable nnU-Net configurations (2D, 3D fullres, 3D lowres, 3D cascade) in 5-fold cross-validation. All models are trained from scratch without pretraining and trained only on the provided training data of the challenge without external training data. Note that other participants may be using external data in some competitions. For each dataset, nnU-Net subsequently identifies the ideal configuration(s) based on cross-validation and ensembling. Finally, The best configuration is used to predict the test cases.

The pipeline generated by nnU-Net is provided for each dataset in the compact representation described in Section 6.2. We furthermore provide a table containing detailed cross-validation as well as test set results.

All leaderboards were last accessed on December 12th, 2019.

6.1 Challenge inclusion criteria

When selecting challenges for participation, our goal was to apply nnU-Net to as many different datasets as possible to demonstrate its robustness and flexibility. We applied the following criteria to ensure a rigorous and sound testing environment:

1. The task of the challenge is semantic segmentation in any 3D imaging modality with images of any size.
2. Training cases are provided to the challenge participants.
3. Test cases are separate, with the ground truth not being available to the challenge participants.
4. Comparison to results from other participants is possible (e.g. through standardized evaluation with an online platform and a public leaderboard).

The competitions outlined below are the ones who qualified under these criteria and were thus selected for evaluation of nnU-Net. To our knowledge, CREMI [5] is the only competition from the biological domain that meets these criteria.

6.2 Compact architecture representation

In the following sections, network architectures generated by nnU-Net will be presented in a compact representation consisting of two lists: one for the convolutional kernel sizes and one for the downsampling strides. As we describe in this section, this representation can be used to fully reconstruct the entire network architecture. The condensed representation is chosen to prevent an

[5] https://cremi.org/leaderboard/

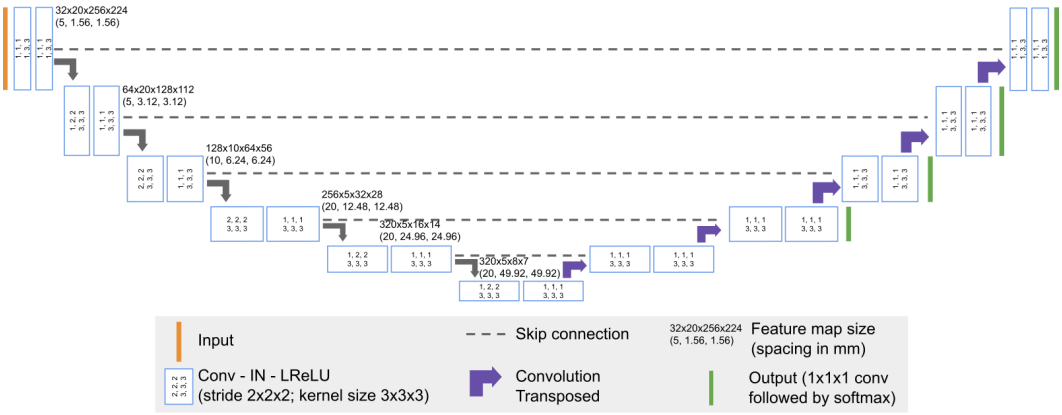

Figure SN6.1: Decoding the architecture. We provide all generated architectures in a compact representation from which they can be fully reconstructed if desired. The architecture displayed here can be represented by means of kernel sizes [[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]] and strides [[1, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2], [1, 2, 2]] (see description in the text).

excessive amount of figures.

Figure 6.2 exemplary shows the 3D full resolution U-Net for the ACDC dataset (D13). The architecture has 6 resolution stages. Each resolution stage in both encoder and decoder consists of two computational blocks. Each block is a sequence of (conv - instance norm - leaky ReLU), as described in 4. In this figure, one such block is represented by an outlined blue box. Within each box, the stride of the convolution is indicated by the first three numbers (1,1,1 for the uppermost left box) and the kernel size of the convolution is indicated by the second set of numbers (1,3,3 for the uppermost left box). Using this information, along with the template with which our architectures are designed, we can fully describe the presented architecture with the following lists:

- **Convolutional kernel sizes**: The kernel sizes of this architecture are [[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]. Note that this list contains 6 elements, matching the 6 resolutions encountered in the encoder. Each element in this list gives the kernel size of the convolutional layers at this resolution (here this is three digits due to the convolutions being three dimensional). Within one resolution, both blocks use the same kernel size. The convolutions in the decoder mirror the encoder (dropping the last entry in the list due to the bottleneck).
- **Downsampling strides**: The strides for downsampling here are [[1, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2], [1, 2, 2]]. Each downsampling step in the encoder is represented by one entry. A stride of 2 results in a downsampling of factor 2 along that axis which a stride of 1 leaves the size unchanged. Note how the stride initially is [1, 2, 2] due to the spacing discrepancy. This changes the initial spacing of $5 \times 1.56 \times 1.56\,\mathrm{mm}$ to a spacing of $5 \times 3.12 \times 3.12\,\mathrm{mm}$ in the second resolution step. The downsampling strides only apply to the first convolution of each resolution stage in the encoder. The second convolution always has a stride of [1, 1, 1]. Again, the decoder mirrors the encoder, but the stride is used as output stride of the convolution transposed (resulting in appropriate upscaling of feature maps). Outputs of all convolutions transposed have the same shape as the skip connection originating from the encoder.

Segmentation outputs for auxiliary losses are added to all but the two lowest resolution steps.

6.3 Medical Segmentation Decathlon

Challenge summary The Medical Segmentation Decathlon[6] [1] is a competition that spans 10 different segmentation tasks. These tasks are selected to cover a large proportion of the dataset variability in the medical domain. The overarching goal of the competition was to encourage researchers to develop algorithms that can work with these datasets out of the box without manual intervention. Each of the tasks comes with respective training and test data. A detailed description of datasets can be found on the challenge homepage. Originally, the challenge was divided into two phases: In phase I, 7 datasets were provided to the participants for algorithm development. In phase II, the algorithms were applied to three additional and previously unseen datasets without further changes. Challenge evaluation was performed for the two phases individually and winners were determined based on their performance on the test cases.

Initial version of nnU-Net A preliminary version of nnU-Net was developed as part of our entry in this competition, where it achieved the first rank in both phases (see `http://medicaldecathlon.com/results.html`). We subsequently made the respective challenge report available on arXiv [15].

nnU-Net has since been refined using all ten tasks of the Medical Segmentation Decathlon. The current version of nnU-Net as presented in this publication was again submitted to the open leaderboard (`https://decathlon-10.grand-challenge.org/evaluation/results/`), and achieved the first rank outperforming the initial nnU-Net as well as other methods that held the state of the art since the original competition [16].

Application of nnU-Net to the Medical Segmentation Decathlon nnU-Net was applied to all ten tasks of the Medical Segmentation Decathlon without any manual intervention.

BrainTumour (D1)

Normalization: Each image is normalized independently by subtracting its mean and dividing by its standard deviation.

	2D U-Net	3D full resolution U-Net	3D low resolution U-Net
Target spacing (mm):	NA x 1 x 1	1 x 1 x 1	-
Median image shape at target spacing:	NA x 169 x 138	138 169 138	-
Patch size:	192 x 160	128 x 128 x 128	-
Batch size:	107	2	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table SN6.1: Network configurations generated by nnU-Net for the BrainTumour dataset from the Medical Segmentation Decathlon (D1). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

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

	edema	non-enhancing tumor	enhancing tumour	mean
2D	0.7957	0.5985	0.7825	0.7256
3D_fullres *	0.8101	0.6199	0.7934	0.7411
Best ensemble	0.8106	0.6179	0.7926	0.7404
Postprocessed	0.8101	0.6199	0.7934	0.7411
Test set	0.68	0.47	0.68	0.61

Table SN6.2: Decathlon BrainTumour (D1) results. Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set") Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 2D U-Net and the 3D full resolution U-Net.

Heart (D2)

Normalization: Each image is normalized independently by subtracting its mean and dividing by its standard deviation.

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 1.25 x 1.25	1.37 x 1.25 x 1.25	-
Median image shape at target spacing:	NA x 320 x 232	115 x 320 x 232	-
Patch size:	320 x 256	80 x 192 x 160	-
Batch size:	40	2	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 1]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3],]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table SN6.3: Network configurations generated by nnU-Net for the Heart dataset from the Medical Segmentation Decathlon (D2). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

	left atrium	mean
2D	0.9090	0.9090
3D_fullres *	0.9328	0.9328
Best ensemble	0.9268	0.9268
Postprocessed	0.9329	0.9329
Test set	0.93	0.93

Table SN6.4: Decathlon Heart (D2) results. Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 2D U-Net and the 3D full resolution U-Net.

Liver (D3)

Normalization: Clip to $[-17, 201]$, then subtract 99.40 and finally divide by 39.36.

	2D U-Net	3D full resolution U-Net	3D low resolution U-Net
Target spacing (mm):	NA x 0.7676 x 0.7676	1 x 0.7676 x 0.7676	2.47 x 1.90 x 1.90
Median image shape at target spacing:	NA x 512 x 512	482 x 512 x 512	195 x 207 x 207
Patch size:	512 x 512	128 x 128 x 128	128 x 128 x 128
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table SN6.5: Network configurations generated by nnU-Net for the Liver dataset from the Medical Segmentation Decathlon (D3). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

	liver	cancer	mean
2D	0.9547	0.5637	0.7592
3D_fullres	0.9571	0.6372	0.7971
3D_lowres	0.9563	0.6028	0.7796
3D cascade	0.9600	0.6386	0.7993
Best ensemble*	0.9613	0.6564	0.8088
Postprocessed	0.9621	0.6600	0.8111
Test set	0.96	0.76	0.86

Table SN6.6: Decathlon Liver (D3) results. Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 3D low resolution U-Net and the 3D full resolution U-Net.

Hippocampus (D4)

Normalization: Each image is normalized independently by subtracting its mean and dividing by its standard deviation.

	2D U-Net	3D full resolution U-Net	3D low resolution U-Net
Target spacing (mm):	NA x 1 x 1	1 x 1 x 1	-
Median image shape at target spacing:	NA x 50 x 35	36 x 50 x 35	-
Patch size:	56 x 40	40 x 56 x 40	-
Batch size:	366	9	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table SN6.7: Network configurations generated by nnU-Net for the Hippocampus dataset from the Medical Segmentation Decathlon (D4). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

	Anterior	Posterior	mean
2D	0.8787	0.8595	0.8691
3D_fullres *	0.8975	0.8807	0.8891
Best ensemble	0.8962	0.8790	0.8876
Postprocessed	0.8975	0.8807	0.8891
Test set	0.90	0.89	0.895

Table SN6.8: Decathlon Hippocampus (D4) results. Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 2D U-Net and the 3D full resolution U-Net.

Prostate (D5)

Normalization: Each image is normalized independently by subtracting its mean and dividing by its standard deviation.

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.62 x 0.62	3.6 x 0.62 x 0.62	-
Median image shape at target spacing:	NA x 320 x 319	20 x 320 x 319	-
Patch size:	320 x 320	20 x 320 x 256	-
Batch size:	32	2	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [1, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2], [1, 2, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table SN6.9: Network configurations generated by nnU-Net for the Prostate dataset from the Medical Segmentation Decathlon (D5). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

	PZ	TZ	mean
2D	0.6285	0.8380	0.7333
3D_fullres	0.6663	0.8410	0.7537
Best ensemble *	0.6611	0.8575	0.7593
Postprocessed	0.6611	0.8577	0.7594
Test set	0.77	0.90	0.835

Table SN6.10: Decathlon Prostate (D5) results. Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 2D U-Net and the 3D full resolution U-Net.

Lung (D6)

Normalization: Clip to $[-1024, 325]$, then subtract -158.58 and finally divide by 324.70.

	2D U-Net	3D full resolution U-Net	3D low resolution U-Net
Target spacing (mm):	NA x 0.79 x 0.79	1.24 x 0.79 x 0.79	2.35 x 1.48 x 1.48
Median image shape at target spacing:	NA x 512 x 512	252 x 512 x 512	133 x 271 x 271
Patch size:	512 x 512	80 x 192 x 160	80 x 192 x 160
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table SN6.11: Network configurations generated by nnU-Net for the Lung dataset from the Medical Segmentation Decathlon (D6). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

	cancer	mean
2D	0.4989	0.4989
3D_fullres	0.7211	0.7211
3D_lowres	0.7109	0.7109
3D cascade	0.6980	0.6980
Best ensemble*	0.7241	0.7241
Postprocessed	0.7241	0.7241
Test set	0.74	0.74

Table SN6.12: Decathlon Lung (D6) results. Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 3D low resolution U-Net and the 3D full resolution U-Net.

Pancreas (D7)

Normalization: Clip to $[-96.0, 215.0]$, then subtract 77.99 and finally divide by 75.40.

	2D U-Net	3D full resolution U-Net	3D low resolution U-Net
Target spacing (mm):	NA x 0.8 x 0.8	2.5 x 0.8 x 0.8	2.58 x 1.29 x 1.29
Median image shape at target spacing:	NA x 512 x 512	96 x 512 x 512	93 x 318 x 318
Patch size:	512 x 512	40 x 224 x 224	64 x 192 x 192
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table SN6.13: Network configurations generated by nnU-Net for the Pancreas dataset from the Medical Segmentation Decathlon (D7). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

	pancreas	cancer	mean
2D	0.7738	0.3501	0.5619
3D_fullres	0.8217	0.5274	0.6745
3D_lowres	0.8118	0.5286	0.6702
3D cascade	0.8101	0.5380	0.6741
Best ensemble *	0.8214	0.5428	0.6821
Postprocessed	0.8214	0.5428	0.6821
Test set	0.82	0.53	0.675

Table SN6.14: Decathlon Pancreas (D7) results. Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 3D full resolution U-Net and the 3D U-Net cascade.

Hepatic Vessel (D8)

Normalization: Clip to $[-3, 243]$, then subtract 104.37 and finally divide by 52.62.

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.8 x 0.8	1.5 x 0.8 x 0.8	2.42 x 1.29 x 1.29
Median image shape at target spacing:	NA x 512 x 512	150 x 512 x 512	93 x 318 x 318
Patch size:	512 x 512	64 x 192 x 192	64 x 192 x 192
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table SN6.15: Network configurations generated by nnU-Net for the HepaticVessel dataset from the Medical Segmentation Decathlon (D8). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

	Vessel	Tumour	mean
2D	0.6180	0.6359	0.6269
3D_fullres	0.6456	0.7217	0.6837
3D_lowres	0.6294	0.7079	0.6687
3D cascade	0.6424	0.7138	0.6781
Best ensemble *	0.6485	0.7250	0.6867
Postprocessed	0.6485	0.7250	0.6867
Test set	0.66	0.72	0.69

Table SN6.16: Decathlon HepaticVessel (D8) results. Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 3D full resolution U-Net and the 3D low resolution U-Net.

Spleen (D9)

Normalization: Clip to $[-41, 176]$, then subtract 99.29 and finally divide by 39.47.

	2D U-Net	3D full resolution U-Net	3D low resolution U-Net
Target spacing (mm):	NA x 0.79 x 0.79	1.6 x 0.79 x 0.79	2.77 x 1.38 x 1.38
Median image shape at target spacing:	NA x 512 x 512	187 x 512 x 512	108 x 293 x 293
Patch size:	512 x 512	64 x 192 x 160	64 x 192 x 192
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table SN6.17: Network configurations generated by nnU-Net for the Spleen dataset from the Medical Segmentation Decathlon (D9). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

	spleen	mean
2D	0.9492	0.9492
3D_fullres	0.9638	0.9638
3D_lowres	0.9683	0.9683
3D cascade	0.9714	0.9714
Best ensemble *	0.9723	0.9723
Postprocessed	0.9724	0.9724
Test set	0.97	0.97

Table SN6.18: Decathlon Spleen (D9) results. Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 3D U-Net cascade and the 3D full resolution U-Net.

Colon (D10)

Normalization: Clip to $[-30.0, 165.82]$, then subtract 62.18 and finally divide by 32.65.

	2D U-Net	3D full resolution U-Net	3D low resolution U-Net
Target spacing (mm):	NA x 0.78 x 0.78	3 x 0.78 x 0.78	3.09 x 1.55 x 1.55
Median image shape at target spacing:	NA x 512 x 512	150 x 512 x 512	146 x 258 x 258
Patch size:	512 x 512	56 x 192 x 160	96 x 160 x 160
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table SN6.19: Network configurations generated by nnU-Net for the Colon dataset from the Medical Segmentation Decathlon (D10). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

	colon cancer primaries	mean
2D	0.2852	0.2852
3D_fullres	0.4553	0.4553
3D_lowres	0.4538	0.4538
3D cascade *	0.4937	0.4937
Best ensemble	0.4853	0.4853
Postprocessed	0.4937	0.4937
Test set	0.58	0.58

Table SN6.20: Decathlon Colon (D10) results. Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform and only two significant digits are reported. Best ensemble on this dataset was the combination of the 3D U-Net cascade and the 3D full resolution U-Net.

6.4 Multi Atlas Labeling Beyond the Cranial Vault: Abdomen (D11)

Challenge summary The Multi Atlas Labeling Beyond the Cranial Vault - Abdomen Challenge[7] [4] (denoted BCV for brevity) comprises 30 CT images for training and 20 for testing. The segmentation target are thirteen different organs in the abdomen.

Application of nnU-Net to BCV nnU-Net was applied to the BCV challenge without any manual intervention.

Normalization: Clip to $[-958, 327]$, then subtract 82.92 and finally divide by 136.97.

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.76 x 0.76	3 x 0.76 x 0.76	3.18 x 1.60 x 1.60
Median image shape at target spacing:	NA x 512 x 512	148 x 512 x 512	140 x 243 x 243
Patch size:	512 x 512	48 x 192 x 192	80 x 160 x 160
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table SN6.21: Network configurations generated by nnU-Net for the BCV challenge (D131. For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

[7] https://www.synapse.org/Synapse:syn3193805/wiki/217752

	1	2	3	4	5	6	7	8
2D	0.8860	0.8131	0.8357	0.6406	0.7724	0.9453	0.8405	0.9128
3D_fullres	0.9083	0.8939	0.8675	0.6632	0.7840	0.9557	0.8816	0.9229
3D_lowres	0.9132	0.9045	0.9132	0.6525	0.7810	0.9554	0.8903	0.9209
3D cascade	0.9166	0.9069	0.9137	0.7036	0.7885	0.9587	0.9037	0.9215
Best ensemble *	0.9135	0.9065	0.8971	0.6955	0.7897	0.9589	0.9026	0.9248
Postprocessed	0.9135	0.9065	0.8971	0.6959	0.7897	0.9590	0.9026	0.9248
Test set	0.9721	0.9182	0.9578	0.7528	0.8411	0.9769	0.9220	0.9290

	9	10	11	12	13	mean
2D	0.8140	0.7046	0.7367	0.6269	0.5909	0.7784
3D_fullres	0.8638	0.7659	0.8176	0.7148	0.7238	0.8279
3D_lowres	0.8571	0.7469	0.8003	0.6688	0.6851	0.8223
3D cascade	0.8621	0.7722	0.8210	0.7205	0.7214	0.8393
Best ensemble *	0.8673	0.7746	0.8299	0.7218	0.7287	0.8393
Postprocessed	0.8673	0.7746	0.8299	0.7262	0.7290	0.8397
Test set	0.8809	0.8317	0.8515	0.7887	0.7674	0.8762

Table SN6.22: Multi Atlas Labeling Beyond the Cranial Vault Abdomen (D11) results. Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. Postprocessing was applied to the model marked with *. This model (incl postprocessing) was used for test set predictions. Note that the Dice scores for the test set are computed with the online platform. Best ensemble on this dataset was the combination of the 3D U-Net cascade and the 3D full resolution U-Net.

6.5 PROMISE12 (D12)

Challenge summary The segmentation target of the PROMISE12 challenge [5] is the prostate in T2 MRI images. 50 training cases with prostate annotations are provided for training. There are 30 test cases which need to be segmented by the challenge participants and are subsequently evaluated on an online platform[8].

Application of nnU-Net to PROMISE12 nnU-Net was applied to the PROMISE12 challenge without any manual intervention.

Normalization: Each image is normalized independently by subtracting its mean and dividing by its standard deviation.

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.61 x 0.61	2.2 x 0.61 x 0.61	-
Median image shape at target spacing:	NA x 327 x 327	39 x 327 x 327	-
Patch size:	384 x 384	28 x 256 x 256	-
Batch size:	22	2	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2], [1, 2, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table SN6.23: Network configurations generated by nnU-Net for the PROMISE12 challenge (D12). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

[8] https://promise12.grand-challenge.org/

	prostate	mean
2D	0.8932	0.8932
3D_fullres	0.8891	0.8891
Best ensemble *	0.9029	0.9029
Postprocessed	0.9030	0.9030
Test set	0.9194	0.9194

Table SN6.24: PROMISE12 (D12) results. Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the scores for the test set are computed with the online platform. The evaluation score of our test set submission is 89.6507. The test set Dice score reported in the table was computed from the detailed submission results (Detailed results available here `https://promise12.grand-challenge.org/evaluation/results/89044a85-6c13-49f4-9742-dea65013e971/`). Best ensemble on this dataset was the combination of the 2D U-Net and the 3D full resolution U-Net.

6.6 The Automatic Cardiac Diagnosis Challenge (ACDC) (D13)

Challenge summary The Automatic Cardiac Diagnosis Challenge [6] (ACDC) comprises 100 training patients and 50 test patients. The target structures are the cavity of the right ventricle, the myocardium of the left ventricle and the cavity of the left ventricle. All images are cine MRI sequences of which the enddiastolic (ED) and endsystolic (ES) time points of the cardiac cycle were to be segmented. With two time instances per patient, the effective number of training/test images is 200/100.

Application of nnU-Net to ACDC Since two time instances of the same patient were provided, we manually interfered with the split for the 5-fold cross-validation of our models to ensure mutual exclusivity of patients between folds. A part from that, nnU-Net was applied without manual intervention.

Normalization: Each image is normalized independently by subtracting its mean and dividing by its standard deviation.

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 1.56 x 1.56	5 x 1.56 x 1.56	-
Median image shape at target spacing:	NA x 237 x 208	18 x 237 x 208	-
Patch size:	256 x 224	20 x 256 x 224	-
Batch size:	58	3	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2], [1, 2, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table SN6.25: Network configurations generated by nnU-Net for the ACDC challenge (D13). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

	RV	MLV	LVC	mean
2D	0.9053	0.8991	0.9433	0.9159
3D_fullres	0.9059	0.9022	0.9458	0.9179
Best ensemble *	0.9145	0.9059	0.9479	0.9227
Postprocessed	0.9145	0.9059	0.9479	0.9228
Test set	0.9275	0.9135	0.9475	0.9295

Table SN6.26: ACDC results (D13). Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform. The online platform reports the Dice scores for enddiastolic and endsystolic time points separately. We averaged these values for a more condensed presentation. Best ensemble on this dataset was the combination of the 2D U-Net and the 3D full resolution U-Net.

6.7 Liver and Liver Tumor Segmentation Challenge (LiTS) (D14)

Challenge summary The Liver and Liver Tumor Segmentation challenge [17] provides 131 training CT images with ground truth annotations for the liver and liver tumors. 70 test images are provided without annotations. The predicted segmentation masks of the test cases are evaluated using the LiTS online platform[9].

Application of nnU-Net to LiTS nnU-Net was applied to the LiTS challenge without any manual intervention.

Normalization: Clip to $[-17, 201]$, then subtract 99.40 and finally divide by 39.39.

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.77 x 0.77	1 x 0.77 x 0.77	2.47 x 1.90 x 1.90
Median image shape at target spacing:	NA x 512 x 512	482 x 512 x 512	195 x 207 x 207
Patch size:	512 x 512	128 x 128 x 128	128 x 128 x 128
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table SN6.27: Network configurations generated by nnU-Net for the LiTS challenge (D14). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

[9] https://competitions.codalab.org/competitions/17094

	liver	cancer	mean
2D	0.9547	0.5603	0.7575
3D_fullres	0.9576	0.6253	0.7914
3D_lowres	0.9585	0.6161	0.7873
3D cascade	0.9609	0.6294	0.7951
Best ensemble*	0.9618	0.6539	0.8078
Postprocessed	0.9631	0.6543	0.8087
Test set	0.9670	0.7630	0.8650

Table SN6.28: LiTS results (D14). Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform. Best ensemble on this dataset was the combination of the 3D low resolution U-Net and the 3D full resolution U-Net.

6.8 Longitudinal Multiple Sclerosis Lesion Segmentation Challenge (MSLesion) (D15)

Challenge summary The longitudinal multiple sclerosis lesion segmentation challenge [7] provides 5 training patients. For each patient, 4 to 5 images acquired at different time points are provided (4 patients with 4 time points each and one patient with 5 time points for a total of 21 images). Each time point is annotated by two different experts, resulting in 42 training annotations (on 21 images). The test set contains 14 patients, again with several time points each, for a total of 61 MRI acquisitions. Test set predictions are evaluated using the online platform[10]. Each train and test image consists of four MRI modalities: MPRAGE, FLAIR, Proton Density, T2.

Application of nnU-Net to MSLesion We manually interfere with the splits in the cross-validation to ensure mutual exclusivity of patients between folds. Each image was annotated by two different experts. We treat these annotations as separate training images (of the same patient), resulting in a training set size of $2 \times 21 = 42$. We do not use the longitudinal nature of the scans and treat each image individually during training and inference.

Normalization: Each image is normalized independently by subtracting its mean and dividing by its standard deviation.

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 1 x 1	1 x 1 x 1	-
Median image shape at target spacing:	NA x 180 x 137	137 x 180 x 137	-
Patch size:	192 x 160	112 x 128 x 96	-
Batch size:	107	2	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 1], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table SN6.29: Network configurations generated by nnU-Net for the MSLesion challenge (D15). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

[10] https://smart-stats-tools.org/lesion-challenge

	lesion	mean
2D	0.7339	0.7339
3D_fullres *	0.7531	0.7531
Best ensemble	0.7494	0.7494
Postprocessed	0.7531	0.7531
Test set	0.6785	0.6785

Table SN6.30: MSLesion results (D15). Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. * marks the best performing model selected for subsequent postprocessing (see "Postprocessed") and test set submission (see "Test set"). Note that the Dice scores for the test set are computed with the online platform based on the detailed results (which are available here `https://smart-stats-tools.org/sites/lesion_challenge/temp/top25/nnUNetV2_12032019_0903.csv`). The ranking is based on a score, which includes other metrics as well (see [7] for details). The score of our submission is 92.874. Best ensemble on this dataset was the combination of the 2D U-Net and the 3D full resolution U-Net.

6.9 Combined Healthy Abdominal Organ Segmentation (CHAOS) (D16)

Challenge summary The CHAOS challenge [8] is divided into five tasks. Here we focused on Tasks 3 (MRI Liver segmentation) and Task 5 (MRI multiorgan segmentation). Tasks 1, 2 and 4 also included the use of CT images, a modality for which plenty of public data is available (see e.g. BCV and LiTS challenge). To isolate the algorithmic performance of nnU-Net relative to other participants we decided to only use the tasks for which a contamination with external data was unlikely. The target structures of Task 5 are the liver, the spleen and the left and right kidneys. The CHAOS challenge provides 20 training cases. For each training case, there is a T2 images with a corresponding ground truth annotation as well as a T1 acquisition with its own, separate ground truth annotation. The T1 acquisition has two modalities which are co-registered: T1 in-phase and T1 out-phase. Task 3 is a subset of Task 5 with only the liver being the segmentation target. The 20 test cases are evaluated using the online platform[11].

Application of nnU-Net to CHAOS nnU-Net only supports images with a constant number of input modalities. The training cases in CHAOS have either one (T2) or two (T1 in & out phase) modalities. To ensure compatibility with nnU-Net we could have either duplicated the T2 image and trained with two input modalities or use only one input modality and treat T1 in phase and out phase as separate training examples. We opted for the latter because this variant results in more (albeit highly correlated) training images. With 20 training patients being provided, this approach resulted in 60 training images. For the cross-validation we ensure that the split is being done on patient level. During inference, nnU-Net will generate two separate predictions for T1 in and out phase which need to be consolidated for test set evaluation. We achieve this by simply averaging the softmax probabilities between the two to generate the final segmentation. We train nnU-Net only for Task 5. Because task 3 represents a subset of Task 5, we extract the liver from our Task 5 predictions and submit it to Task 3.

Normalization: Each image is normalized independently by subtracting its mean and dividing by its standard deviation.

[11] `https://chaos.grand-challenge.org/`

	2D U-Net	3D full resolution U-Net	3D low resolution U-Net
Target spacing (mm):	NA x 1.66 x 1.66	5.95 x 1.66 x 1.66	-
Median image shape at target spacing:	NA x 195 x 262	45 x 195 x 262	-
Patch size:	224 x 320	40 x 192 x 256	-
Batch size:	45	2	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [1, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2], [1, 1, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table SN6.31: Network configurations generated by nnU-Net for the CHAOS challenge (D16). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2.

	liver	right kidney	left kidney	spleen	mean
2D	0.9132	0.8991	0.8897	0.8720	0.8935
3D_fullres	0.9202	0.9274	0.9209	0.8938	0.9156
Best ensemble *	0.9184	0.9283	0.9255	0.8911	0.9158
Postprocessed	0.9345	0.9289	0.9212	0.894	0.9197
Test set	-	-	-	-	-

Table SN6.32: CHAOS results (D16). Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. Postprocessing was applied to the model marked with *. This model (incl postprocessing) was used for test set predictions. Note that the evaluation of the test set was performed with the online platform of the challenge which does not report Dice scores for the individual organs. The score of our submission was 72.44 for Task 5 and 75.10 for Task3 (see [8] for details). Best ensemble on this dataset was the combination of the 2D U-Net and the 3D full resolution U-Net.

6.10 Kidney and Kidney Tumor Segmentation (KiTS) (D17)

Challenge summary The Kidney and Kidney Tumor Segmentation challenge [18] was the largest competition (in terms of number of participants) at MICCAI 2019. The target structures are the kidneys and kidney tumors. 210 training and 90 test cases are provided by the challenge organizers. The organizers provide the data both in their original geometry (with voxel spacing varying between cases) as well as interpolated to a common voxel spacing. Evaluation of the test set predictions is done on the online platform[12].

We participated in the original KiTS 2019 MICCAI challenge with a manually designed residual 3D U-Net. This algorithm, described in [19] obtained the first rank in the challenge. For this submission, we did slight modifications to the original training data: Cases 15 and 37 were confirmed to be faulty by the challenge organizers (`https://github.com/neheller/kits19/issues/21`) which is why we replaced their respective segmentation masks with predictions of one of our networks. We furthermore excluded cases 23, 68, 125 and 133 because we suspected labeling errors in these cases as well. At the time of conducting the experiments for this publication, no revised segmentation masks were provided by the challenge organizers, which is why we re-used the modified training dataset for training nnU-Net.

After the challenge event at MICCAI 2019, an open leaderboard was created. The original

[12] `https://kits19.grand-challenge.org/`

challenge leaderboard is retained at http://results.kits-challenge.org/miccai2019/. All submissions of the original KiTS challenge were mirrored to the open leaderboard. The submission of nnU-Net as performed in the context of this manuscript is done on the open leaderboard, where many more competitors have entered since the challenge. As presented in Figure 3, nnU-Net sets a new state of the art on the open leaderboard, thus also outperforming our initial, manually optimized solution.

Application of nnU-Net to KiTS Since nnU-Net is designed to automatically deal with varying voxel spacings within a dataset, we chose the original, non-interpolated image data as provided by the organizers and let nnU-Net deal with the homogenization of voxel spacing. nnU-Net was applied to the KiTS challenge without any manual intervention.

Normalization: Clip to $[-79, 304]$, then subtract 100.93 and finally divide by 76.90.

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.78 x 0.78	0.78 x 0.78 x 0.78	1.99 x 1.99 x 1.99
Median image shape at target spacing:	NA x 512 x 512	525 x 512 x 512	206 x 201 x 201
Patch size:	512 x 512	128 x 128 x 128	128 x 128 x 128
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table SN6.33: Network configurations generated by nnU-Net for the KiTS challenge (D17). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

	Kidney	Tumor	mean
2D	0.9613	0.7563	0.8588
3D_fullres	0.9702	0.8367	0.9035
3D_lowres	0.9629	0.8420	0.9025
3D cascade	0.9702	0.8546	0.9124
Best ensemble*	0.9707	0.8620	0.9163
Postprocessed	0.9707	0.8620	0.9163
Test set	-	0.8542	-

Table SN6.34: KiTS results (D17). Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. Postprocessing was applied to the model marked with *. This model (incl postprocessing) was used for test set predictions. Note that the Dice scores for the test set are computed with the online platform which computes the kidney Dice score based of the union of the kidney and tumor labels whereas nnU-Net always evaluates labels independently, resulting in a missing value for kindey in the table. The reported kindey Dice by the platform (which is not comparable with the value computed by nnU-Net) is 0.9793. Best ensemble on this dataset was the combination of the 3D U-Net cascade and the 3D full resolution U-Net.

6.11 Segmentation of THoracic Organs at Risk in CT images (SegTHOR) (D18)

Challenge summary In the Segmentation of THoracic Organs at Risk in CT images [10] challenge, four abdominal organs (the heart, the aorta, the trachea and the esopahgus) are to be segmented in CT

images. 40 training images are provided for training and another 20 images are provided for testing. Evaluation of the test images is done using the online platform[13].

Application of nnU-Net to SegTHOR nnU-Net was applied to the SegTHOR challenge without any manual intervention.

Normalization: Clip to $[-986, 271]$, then subtract 20.78 and finally divide by 180.50.

	2D U-Net	**3D full resolution U-Net**	**3D low resolution U-Net**
Target spacing (mm):	NA x 0.89 x 0.89	2.50 x 0.89 x 0.89	3.51 x 1.76 x 1.76
Median image shape at target spacing:	NA x 512 x 512	171 x 512 x 512	122 x 285 x 285
Patch size:	512 x 512	64 x 192 x 160	80 x 192 x 160
Batch size:	12	2	2
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]

Table SN6.35: Network configurations generated by nnU-Net for the SegTHOR challenge (D18). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

	esophagus	heart	trachea	aorta	mean
2D	0.8181	0.9407	0.9077	0.9277	0.8986
3D_fullres	0.8495	0.9527	0.9055	0.9426	0.9126
3D_lowres	0.8110	0.9464	0.8930	0.9284	0.8947
3D cascade	0.8553	0.9520	0.9045	0.9403	0.9130
Best ensemble*	0.8545	0.9532	0.9066	0.9427	0.9143
Postprocessed	0.8545	0.9532	0.9083	0.9438	0.9150
Test set	0.8890	0.9570	0.9228	0.9510	0.9300

Table SN6.36: SegTHOR results (D18). Note that all reported Dice scores (except the test set) were computed using five fold cross-validation on the training cases. Postprocessing was applied to the model marked with *. This model (incl postprocessing) was used for test set predictions. Note that the Dice scores for the test set are computed with the online platform. Best ensemble on this dataset was the combination of the 3D U-Net cascade and the 3D full resolution U-Net.

6.12 Challenge on Circuit Reconstruction from Electron Microscopy Images (CREMI) (D19)

Challenge summary The Challenge on Circuit Reconstruction from Electron Microscopy Images is subdivided into three tasks. The synaptic cleft segmentation task can be formulated as semantic segmentation (as opposed to e.g. instance segmentation) and is thus compatible with nnU-Net. In this task, the segmentation target is the cell membrane in locations where the cells are forming a synapse. The dataset consists of serial section Transmission Electron Microscopy scans of the *Drosophila melanogaster* brain. Three volumes are provided for training and another three are provided for testing. Test set evaluation is done using the online platform[14].

Application of nnU-Net to CREMI Since to the number of training images is lower than the number of splits, we cannot run a 5-fold cross-validation. Thus, we trained 5 model instances,

[13] https://competitions.codalab.org/competitions/21145
[14] https://cremi.org/

each of them on all three training volumes and subsequently ensembled these models for test set prediction. Because this training scheme leaves no validation data, selection of the best of three model configurations as performed by nnU-Net after cross-validation was not possible. Hence, we intervened by only configuring and training the 3D full resolution configuration.

Normalization: Each image is normalized independently by subtracting its mean and dividing by its standard deviation.

	2D U-Net	3D full resolution U-Net	3D low resolution U-Net
Target spacing (mm):	-	$4 \cdot 10^{-5}$ x $4 \cdot 10^{-6}$ x $4 \cdot 10^{-6}$	-
Median image shape at target spacing:	-	125 x 1250 x 1250	-
Patch size:	-	24 x 256 x256	-
Batch size:	-	2	-
Downsampling strides:	-	[[1, 2, 2], [1, 2, 2], [1, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2]]	-
Convolution kernel sizes:	-	[[1, 3, 3], [1, 3, 3], [1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table SN6.37: Network configurations generated by nnU-Net for the CREMI challenge (D19). For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2

Results Because our training scheme for this challenge left no validation data, a performance estimate as given for the other datastes is not available for CREMI. The CREMI test set is evaluated by the online platform. The evaluation metric is the so called CREMI score, a description of which is available here `https://cremi.org/metrics/`. Dice scores for the test set are not reported. The CREMI score of our test set submission was 74.96 (lower is better).

6.13 Cell Tracking Challenge

Challenge summary The cell tracking challenge [12, 13] (`http://celltrackingchallenge.net/3d-datasets/`) is subdivided into two benchmarks: the cell segmentation and the cell tracking benchmark. It comprises a total of 9 2D+t and 10 3D+t datasets, each showing a different type of cell or nucleus acquired with microscopic imaging techniques. Since each dataset is evaluated independently, teams usually only participate in a fraction of all available datasets.

The challenge evaluates submissions to the cell segmentation benchmark using two different scores, DET and SEG. A detailed description of the scores is provided here: `https://public.celltrackingchallenge.net/documents/SEG.pdf`. The ranking of the cell segmentation benchmark is based on OP_{CSB} which is the average value of DET and SEG (see also here `http://celltrackingchallenge.net/evaluation-methodology/`).

In the following we present our participation in the cell tracking challenge. To understand our selection of datasets from within this challenge, we should note that nnU-Net is not designed to be trained on data where the vast majority of slices (or objects) has not been annotated. While this can be addressed in future work (for example as in [20]), we did not want to modify nnU-Net so as to keep our algorithm consistent throughout the manuscript.

The reference annotation instructions of the cell tracking challenge state the following: “For each 3D frame, we also randomly selected at least one of its 2D z-slices that contained some objects” (`https://public.celltrackingchallenge.net/documents/Annotation%20procedure.pdf`). As

a result, the overwhelming majority of 3D datasets in the competition come with (possibly incomplete) reference annotations on only a few 2D slices of the 3D volumes and therefore lack dense reference labels which could be used to train nnU-Net with. There are, however, two simulated 3D datasets (Fluo-C3DH-A549-SIM and Fluo-N3DH-SIM) as well as one real 3D Dataset (Fluo-C3DH-A549, single time steps are annotated densely in this dataset as opposed to sparse 2D slices) for which suitable reference annotations are available.

Even though nnU-Net was primarily developed for solving 3D segmentations and the difficulties in method configuration that go along with it, we also demonstrate that it can be applied successfully to 2D data by including the Fluo-N2DH-SIM dataset in our experiments.

Instance segmentation with nnU-Net Fluo-N2DH-SIM+ and Fluo-N3DH-SIM+ require instance segmentations of cell nuclei. nnU-Net, however, was developed for semantic segmentation. To close this gap we convert the provided instance-level segmentations to a semantic segmentation map prior applying nnU-Net. This is done by converting each nucleus instance into two semantic labels: the center and the border. As a result, the converted reference segmentations in these datasets contain the semantic classes 'background', 'nucleus center' and 'nucleus border'.

After applying nnU-Net, the predicted semantic segmentations of the test cases are converted back to instance segmentations by connected component analysis of the center class, followed by iterative outgrowing into the border region until all border pixels have been assigned a cell instance. To properly assign instance labels to cells entering or exiting the frame, we also identify isolated 'nucleus border' components and assign a unique instance ID to them.

Fluo-N2DH-SIM+ (D20)

Dataset summary The Fluo-N2DH-SIM+ dataset [21] simulates the behavior of dividing HL60 cells. In the simulation, the nuclei were stained with Hoescht and acquired with fluorescence microscopy. The training cases are subdivided into two time series of images with 65 and 150 training images, respectively. The segmentation task is instance segmentation of the cell nuclei.

Application of nnU-Net to Fluo-N2DH-SIM+ To enable the application of nnU-Net to this dataset, we convert the provided nuclei instance segmentation into a semantic segmentation problem (as described above) prior to applying nnU-Net. We set the border thickness to 0.7 micrometers.

We enable nnU-Net to use the time series information by concatenating the four time steps preceding the frame of interest to its input. This is implemented by using them as additional input modalities and thus does not require any changes to nnU-Net.

Due to the lack of sufficiently diverse training data (there are only two separate time series), we do not run cross-validation and instead train a single model on all training cases.

Normalization Each image is normalized independently by subtracting its mean and dividing by its standard deviation.

	2D U-Net	3D full resolution U-Net	3D low resolution U-Net
Target spacing (mm):	$1.25 \cdot 10^{-4}$ x $1.25 \cdot 10^{-4}$	-	-
Median image shape at target spacing:	773 x 739	-	-
Patch size:	896 x 768	-	-
Batch size:	4	-	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	-	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	-	-

Table SN6.38: Network configurations generated by nnU-Net for the Fluo-N2DH-SIM+ (D20) of the cell tracking challenge. Note that this is a 2D dataset. For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2.

Results Due to the lack of sufficiently diverse data to run a cross-validation, a single U-Net from nnU-Nets 2D configuration was trained on all training data. Therefore, no postprocessing could be configured and no performance estimate on the training cases is available. We use this single model to predict the test cases. Our predictions, as well as the code used to generate them were uploaded to and evaluated by the challenge organizers. The leaderboard is available here `http://celltrackingchallenge.net/latest-csb-results/`. We achieved a DET score of 0.978 (rank 4), a SEG score of 0.832 (rank 1) and a combined OP_{CBS} score of 0.905 (rank 1). Detailed results as well as the software used to generate them (nnU-Net, model parameters and glue code) can be accessed here `http://celltrackingchallenge.net/participants/DKFZ-GE/#`.

Fluo-N3DH-SIM+ (D21)

Dataset summary The Fluo-N3DH-SIM+ dataset [21] simulates the behavior of dividing HL60 cells. In the simulation, the nuclei were stained with Hoescht and acquired with fluorescence microscopy. The training cases are subdivided into two time series of images with 150 and 80 training images, respectively. The segmentation task is instance segmentation of the cell nuclei.

Application of nnU-Net to Fluo-N2DH-SIM+ To enable the application of nnU-Net to this dataset, we convert the provided nuclei instance segmentation into a semantic segmentation problem (as described above) prior to applying nnU-Net. We set the border thickness to 0.5 micrometers.

We do not make use of the time information and process each frame independently.

Due to the lack of sufficiently diverse training data (there are only two separate time series), we do not run cross-validation and instead train a single model on all training cases. The 3D full resolution U-Net was manually selected for this dataset (with 2D being the only other option).

Normalization Each image is normalized independently by subtracting its mean and dividing by its standard deviation.

	2D U-Net	3D full resolution U-Net	3D low resolution U-Net
Target spacing (mm):	NA x $1.25 \cdot 10^{-4}$ x $1.25 \cdot 10^{-4}$	$2 \cdot 10^{-4}$ x $1.25 \cdot 10^{-4}$ x $1.25 \cdot 10^{-4}$	-
Median image shape at target spacing:	NA x 349 x 639	59 x 349 x 639	-
Patch size:	384 x 640	32 x 192 x 384	-
Batch size:	13	2	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [1, 2]]	[[2, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2], [1, 2, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table SN6.39: Network configurations generated by nnU-Net for the Fluo-N3DH-SIM+ dataset (D21) of the cell tracking challenge. For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2.

Results Due to the lack of sufficiently diverse data to run a cross-validation, a single U-Net from nnU-Net's 3D full resolution configuration was trained on all training data. Therefore, no postprocessing could be configured and no performance estimate on the training cases is available. We use this single model to predict the test cases. Our predictions, as well as the code used to generate them, were uploaded to and evaluated by the challenge organizers. The leaderboard is available here `http://celltrackingchallenge.net/latest-csb-results/`. We achieved a DET score of 0.992 (rank 1), a SEG score of 0.906 (rank 1) and a combined OP_{CBS} score of 0.949 (rank 1). Detailed results as well as the software used to generate them (nnU-Net, model parameters and glue code) can be accessed here `http://celltrackingchallenge.net/participants/DKFZ-GE/#`.

Fluo-C3DH-A549 and Fluo-C3DH-A549-SIM+ (D22 & D23)

Dataset summary Fluo-C3DH-A549 [22] shows GFP-actin-stained A549 lung cancer cells in a matrixgel matrix. Fluo-C3DH-A549-SIM+ [23] is a simulated dataset showing the same cell type. Each of these datasets comes with two time series. The sequences of Fluo-C3DH-A549 contain 15 and 15 manually annotated images. The sequences of Fluo-C3DH-A549-SIM+ are fully annotated (30+30) due to its simulated nature. The segmentation task in these datasets is semantic segmentation of the single cell that is present in its images. Due to the strong similarities of the datasets, we merge them together and obtain one large dataset which we refer to as Fluo-C3DH-A549(-SIM+). The joint dataset has four time sequences and a total of 90 annotated training images.

Application of nnU-Net to Fluo-C3DH-A549(-SIM+) We do not make use of the time information and process each frame independently.

We manually select nnU-Net's 3D full resolution configuration and train four U-Net models. Each model was trained on three of the four available time series. No further modifications to nnU-Net were made.

Normalization Each image is normalized independently by subtracting its mean and dividing by its standard deviation.

	2D U-Net	3D full resolution U-Net	3D low resolution U-Net
Target spacing (mm):	NA x $1.26 \cdot 10^{-4}$ x $1.26 \cdot 10^{-4}$	$1 \cdot 10^{-3}$ x $1.26 \cdot 10^{-4}$ x $1.26 \cdot 10^{-4}$	-
Median image shape at target spacing:	NA x 300 x 375	29 x 300 x 375	-
Patch size:	320 x 384	24 x 256 x 256	-
Batch size:	27	2	-
Downsampling strides:	[[2, 2], [2, 2], [2, 2], [2, 2], [2, 2], [2, 2]]	[[1, 2, 2], [1, 2, 2], [2, 2, 2], [2, 2, 2], [1, 2, 2], [1, 2, 2]]	-
Convolution kernel sizes:	[[3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3], [3, 3]]	[[1, 3, 3], [1, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]	-

Table SN6.40: Network configurations generated by nnU-Net for the Fluo-C3DH-A549(-SIM+) dataset (D22 & D23) of the cell tracking challenge. For more information on how to decode downsampling strides and kernel sizes into an architecture, see 6.2.

	A549 cell	mean
2D	0.9247	0.9247
3D_fullres*	0.9394	0.9394
Best ensemble	0.9356	0.9356
Postprocessed	0.9394	0.9394
Test set	-	-

Table SN6.41: Fluo-C3DH-A549(-SIM+) results (D22 & D23). Note that all reported Dice scores were computed using stratified four fold cross-validation on the training cases. Postprocessing was applied to the model marked with *. This model (incl postprocessing) was used for test set predictions. Note that we merged the Fluo-C3DH-A549 and Fluo-C3DH-A549-SIM+ datasets for our training (thus enabling the four fold cross-validation) but that these datasets are evaluated independently. For Fluo-C3DH-A549 we achieved a DET score of 1, a SEG score of 0.908 (rank 1) and a combined OP_{CBS} score of 0.954 (rank 1). For Fluo-C3DH-A549(-SIM) we achieved a DET score of 1, a SEG score of 0.955 (rank 1) and a combined OP_{CBS} score of 0.977 (rank 1). Note that the DET score is uninformative on these datasets because only a single cell is present in the images. Detailed results as well as the software used to generate them (nnU-Net, model parameters and glue code) can be accessed here `http://celltrackingchallenge.net/participants/DKFZ-GE/`

7 Description of all leaderboard submissions

When developing machine learning methods, the use of independent test sets is fundamental to ensure objective and convincing reporting on their performance. It is for this reason that we decided to evaluate nnU-Net solely in the context of international segmentation competitions. In this process, it is quintessential to separate the test sets (and in the case of nnU-Net also the test datasets) from the development set in order to prevent overfitting. As we have already stated in the manuscript, the development of nnU-Net was conducted solely by running five-fold cross-validations (or single train-val splits) on the ten datasets provided by the Medical Segmentation Decathlon [1]. Ideally, this process should result in a single test set submission. When inspecting some of the leaderboards, however, one can find multiple submissions that can be associated with our research group. Since this may raise concerns about the independent nature of nnU-Net's testing, we would like to clarify the purpose and nature of each of these entries.

nnU-Net has undergone a long development process with three major milestones:

1. September 2018: The first beta version of nnU-Net was developed as part of our participation in the Medical Segmentation Decathlon (http://medicaldecathlon.com/) challenge. Our challenge participation is described in this preprint: [15]. Source code was not released.
2. April 2019: First open source release of nnU-Net together with an updated preprint ([24]). This version is still available in our code repository and can be selected by choosing the nnUNetTrainer trainer class (over the default which is nnUNetTrainerV2).
3. April 2020: Journal submission of the latest version described in this manuscript, along with an update to the preprint ([25]).

We would like to emphasize again that during all times the development of nnU-Net, i.e. finding appropriate fixed parameters as well as the heuristic rules, was solely done using cross-validation (or single train-val splits) on the training sets of the ten datasets provided by the Medical Segmentation Decathlon. All improvements made between versions of nnU-Net were found and validated on these datasets only. Only when releasing a new version, nnU-Net was applied to other challenges. This was a necessary step to support the claims made in the corresponding preprints.

As a research group with multiple years of experience in designing segmentation methods, there are naturally also other segmentation projects that necessitated test set evaluations on several isolated competitions, for example [26, 19, 27].

In the following we present an exhaustive list of all submissions to the challenges used in this manuscript.

Medical Segmentation Decathlon (D1-D10) [1]

Note that the leaderboard of the Medical Segmentation Decathlon is split into the original challenge leaderboard [15] and the open leaderboard [16]. The results of the original leaderboard were not ported to the open leaderboard resulting in the need for consulting both leaderboards in order to get a complete overview of the results.

1. **2018.** Our participation in the Medical Segmentation Decathlon challenge with the initial beta version of nnU-Net (first milestone). This entry is visible on the original challenge leaderboard only.
2. **December 2019.** Evaluation of the final version of nnU-Net for this manuscript (third milestone). This entry is visible in the open leaderboard only.

Multi Atlas Labeling Beyond the Cranial Vault: Abdomen (D11) [4]

The leaderboard can be accessed at https://www.synapse.org/#!Synapse:syn3193805/wiki/217785.

1. **March 2019**. Submission title: 'submitted_ensemble_3d_fullres_cascade_and_3d_fullres.zip'. Evaluation of nnU-Net for its open source release (second milestone)
2. **March 2019**. Submission title: 'submit_proper_names'. Same as above (as evident by identical score). User error on our end.
3. **March 2019**. Submission title: 'submit_proper_names'. Same as above (as evident by identical score). User error on our end.

[15] http://medicaldecathlon.com/results.html
[16] https://decathlon-10.grand-challenge.org/evaluation/leaderboard/

4. **December 2019**. Submission title: 'ready_for_submission'. Intended as evaluation for this manuscript, but had a bug in the inference code which needed to be fixed (see below).
5. **December 2019**. Submission title: 'submission_bugfixed'. Final evaluation for this manuscript (with bug fixed). nnU-Net was not changed between the two submission apart from this bugfix.

PROMISE12 (D12) [5]

The leaderboard can be accessed at `https://promise12.grand-challenge.org/evaluation/leaderboard/`.

1. **August 2018**. Submission title: ' ' (empty). Application of the initial nnU-Net (first milestone).
2. **March 2019**. Submission title: 'nnUNet final, trained only on Promise dataset'. Evaluation of nnU-Net for the open source release (second milestone).
3. **December 2019.** Submission title: 'nnU-Net v2'. Evaluation of the final version of nnU-Net for this manuscript (third milestone).
4. **December 2019.** Submission title: 'nnU-Net v2'. Same as above (as evident by identical score), User error on our end.

Automatic Cardiac Diagnosis Challenge (ACDC) (D13) [6]

The leaderboard can be accessed at `https://acdc.creatis.insa-lyon.fr/#phase/59db86a96a3c7706f64dbfed`. Note that the leaderboard only displays the last entry and does not provide information about the multiple entries described below.

1. **2017**. Our entry in the competition as described here [26]. Not related to nnU-Net.
2. **March 2019**. Evaluation of nnU-Net for the open source release (second milestone).
3. **December 2019**. Evaluation of the final version of nnU-Net for this manuscript (third milestone).

Liver and Liver Tumor Segmentation Challenge (LiTS) (D14) [17]

The leaderboard can be accessed at `https://competitions.codalab.org/competitions/17094#results`. Note that the leaderboard only displays one entry for us ('FabianIsensee'), which is incorrect:

1. **March 2019.** Evaluation of nnU-Net for the open source release (second milestone).
2. **December 2019.** Evaluation of the final version of nnU-Net for this manuscript (third milestone).

Longitudinal multiple sclerosis lesion segmentation challenge (MSLesion) (D15) [7]

The leaderboard can be accessed at `https://smart-stats-tools.org/lesion-challenge`.

1. **February 2019.** Related to a different publication of our research group ([27]).
2. **March 2019**. Evaluation of nnU-Net for the open source release (second milestone).
3. **December 2019**. Evaluation of the final version of nnU-Net for this manuscript (third milestone).

Combined Healthy Abdominal Organ Segmentation (CHAOS) (D16) [8]

The leaderboard can be accessed at https://chaos.grand-challenge.org/evaluation/leaderboard/. Note that we only participated in Task3 and Task5. The scores for these tasks can be accessed by clicking the 'show all metrics' button.

1. **Sept 2019**. We were contacted by the challenge organizer, Emre Kavur, who encouraged us to participate in the competition (and be listed in the corresponding challenge summary [8]). This submission is mostly based on the final version of nnU-Net, but contains some manual modifications to the pipeline configuration (target spacing) and can therefore not be re-used to evaluate nnU-Net's true out-of-the-box performance.
2. **December 2019**. Evaluation of the final version of nnU-Net for this manuscript (third milestone).

Kidney and Kidney Tumor Segmentation (KiTS) (D17) [18, 26]

The results of the MICCAI competition can be accessed at http://results.kits-challenge.org/miccai2019/. The open leaderboard is available at https://kits19.grand-challenge.org/evaluation/leaderboard/. All results of the original challenge were copied to the open leaderboard. The open leaderboard thus gives a complete picture of all challenge entries.

1. **July 2019**. Our entry in the KiTS competition which achieved the first place in the MICCAI leaderboard is described in [19]. Some parts of this submission were implemented based on a previous version of nnU-Net, but the residual U-Net architecture we submitted shares little similarity with the configuration generated as part of this manuscript.
2. **December 2019**. Evaluation of the final version of nnU-Net for this manuscript (third milestone).

Segmentation of THoracic Organs at Risk in CT images (SegTHOR) (D18) [10]

The leaderboard can be accessed at https://competitions.codalab.org/competitions/21145#results.

1. **December 2019**. Evaluation of the final version of nnU-Net for this manuscript (third milestone). Note that the evaluation platform claims there are three entries. We would like to emphasize that the platform also counts failed submissions (incorrect file names, no feedback about the performance was given) giving the wrong impression that multiple successful submissions were made.

Challenge on Circuit Reconstruction from Electron Microscopy Images (CREMI)(D19)

The leaderboard can be accessed at https://cremi.org/leaderboard/.

1. **December 2019**. Evaluation of the final version of nnU-Net for this manuscript (third milestone).

Cell Tracking Challenge (D20-D23) [12, 13]

The leaderboard can be accessed at http://celltrackingchallenge.net/latest-csb-results/. Note that the identifier for our submission was assigned by the organizers is therefore different from the other competitions: 'DKFZ-GE'. An overview over all

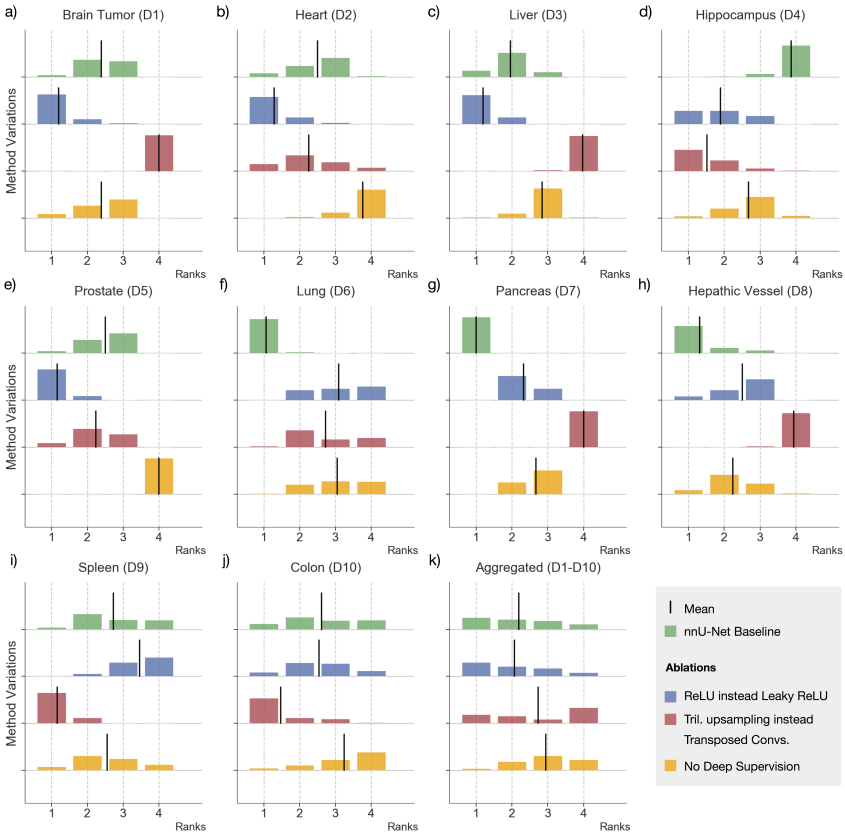

Figure SN8.1: Ablation studies of nnU-Net's architecture template. We configure and train nnU-Nets 3D full resolution U-Net configuration on the ten Decathlon datasets with different architecture template variations. The presentation follows Figure 6 in the main manuscript: for each dataset, we run five-fold cross-validation. 1000 virtual validation datasets are generated via bootstrapping. Algorithms are ranked on each of the virtual validation sets resulting in a distribution over rankings. The results indicate that replacing the default leaky ReLU nonlinearity with regular ReLUs did not adversely affect performance. Replacing convolution transposed with trilinear upsampling and removing the auxiliary losses ('deep supervision') resulted in a slight decrease in performance.

submissions made by each team can be accessed by clicking the teams identifier in the 'Participants & Algorithms' tab (`http://celltrackingchallenge.net/participants/`).

1. **July 2020**. Evaluation of the final version of nnU-Net for this manuscript (third milestone).

8 Ablation studies of nnU-Net's architecture template

Some design choices in the network template could be considered non-standard. We evaluate these design choices in the same way we would compare different methods with the nnU-Net framework. This allows us to integrate the changes only once and apply them to an arbitrary number of datasets. Following the methodology also used to generate Figure 6, we present the results of running five-fold cross-validations on the 10 Decathlon datasets. As can be seen in the Figure, there is no conceivable difference in performance between ReLU and leaky ReLU nonlinearities. The performance drops, however, when replacing convolution transpose in the decoder with trilinear upsampling and when removing the auxiliary loss layers.

9 Using nnU-Net with limited compute resources

Reduction of computational complexity was one of the key motivations driving the design of nnU-Net. The effort of running all the configurations generated by nnU-Net should be manageable for most users and researchers. There are, however, some shortcuts that can be be taken in case computational resources are extremely scarce.

9.1 Reducing the number of network trainings

Depending on whether the 3D U-Net cascade is configured for a given dataset, nnU-Net requires 10 (2D and 3D U-Net with 5 models each) or 20 (2d, 3D, 3D cascade (low resolution and high resolution U-Net) with 5 models each) U-Net trainings to run, each of which takes a couple of days on a single GPU. While this approach guarantees the best possible performance, training all models may exceed reasonable computation time if only a single GPU is available. Therefore, we present two strategies to reduce the number of total network trainings when running nnU-Net.

Manual selection of U-Net configurations
Overall, the 3D full resolution U-Net shows the best segmentation results. Thus, this configuration is a good starting point and could simply be selected as default choice. Users can decide whether to train this configuration using all training cases (to train a single model) or run a five-fold cross-validation and ensemble the 5 resulting models for test case predictions.

In some scenarios, other configurations than the 3D full resolution U-Net can yield best performance. Identifying such scenarios and selecting the respective most promising configuration, however, requires domain knowledge for the dataset at hand. Datasets with highly anisotropic images (such as D12 PROMISE12), for instance, could be best suited for running a 2D U-Net. There is, however, no guarantee for this relation (see D13 ACDC). On datasets with very large images, the 3D U-Net cascade seems to marginally outperform the 3D full resolution U-Net (for example D11, D14, D17, D18, ...) because it improves the capture of contextual information. Note that this is only true if the target structure requires a large receptive field for optimal recognition. On CREMI (D19) for example, despite large image sizes, only a limited field of view is required, because the target structure are relatively small synapses that can be identified using only local information, which is why we selected the 3D full resolution U-Net for this dataset (see Section 6.12).

Not running all configurations as 5-fold cross-validation
Another computation shortcut is to not run all models as 5-fold cross-validation. For instance, only one split for each configuration can be run (note, however, that the 3D low resolution U-Net of the cascade is required to be run as a 5-fold cross-validation in order to generate low resolution segmentation maps of all training cases for the second full resolution U-net of the cascade). Even when running multiple configurations to rely on empirical selection of configurations by nnU-Net, this reduces the total number of models to be trained to 2 if no cascade is configured or 8 if the cascade is configured (the cascade requires 6 model trainings: 5 3D low resolution U-Nets and 1 full resolution 3D U-Net training). nnU-Net subsequently bases selection of the best configuration on this single train-val split. Note that this strategy provides less reliable performance estimates and may result in sub optimal configuration choices. Finally, users can decide whether they wish to re-train the selected configuration on the entire training data or run a five-fold cross-validation for this selected configuration. The latter is expected to result in better test set performance because the 5 models can be used as an ensemble.

9.2 Reduction of GPU memory

nnU-Net is configured to utilize 11GB of GPU memory. This requirement is, based on our experience, a realistic requirement for a modern deep-learning capable GPU (such as a Nvidia GTX 1080 ti (11GB), Nvidia RTX 2080 ti (11GB), Nvidia TitanX(p) (12GB), Nvidia P100 (12/16 GB), Nvidia Titan RTX (24GB), Nvidia V100 (16/32 GB), ...). We strongly recommend using nnU-Net with this default configuration, because it has been tested extensively and, as we show in this manuscript, provides excellent segmentation accuracy. Should users still desire to run nnU-Net on a smaller GPU, the amount of GPU memory used for network configuration can be adapted easily. Corresponding instructions are provided along with the source code.

References

[1] Amber L Simpson, Michela Antonelli, Spyridon Bakas, Michel Bilello, Keyvan Farahani, Bram Van Ginneken, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, et al. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *arXiv preprint arXiv:1902.09063*, 2019.

[2] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014.

[3] Patrick Bilic, Patrick Ferdinand Christ, Eugene Vorontsov, Grzegorz Chlebus, Hao Chen, Qi Dou, Chi-Wing Fu, Xiao Han, Pheng-Ann Heng, Jürgen Hesser, et al. The liver tumor segmentation benchmark (lits). *arXiv preprint arXiv:1901.04056ada*, 2019.

[4] B Landman, Z Xu, J Eugenio Igelsias, M Styner, TR Langerak, and A Klein. Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge, 2015.

[5] Geert Litjens, Robert Toth, Wendy van de Ven, Caroline Hoeks, Sjoerd Kerkstra, Bram van Ginneken, Graham Vincent, Gwenael Guillard, Neil Birbeck, Jindang Zhang, et al. Evaluation of prostate segmentation algorithms for mri: the promise12 challenge. *Med Image Analysis*, 18(2):359–373, 2014.

[6] Olivier Bernard, Alain Lalande, Clement Zotti, Frederick Cervenansky, Xin Yang, Pheng-Ann Heng, Irem Cetin, Karim Lekadir, Oscar Camara, Miguel Angel Gonzalez Ballester, et al. Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: Is the problem solved? *IEEE TMI*, 37(11):2514–2525, 2018.

[7] Aaron Carass, Snehashis Roy, Amod Jog, Jennifer L Cuzzocreo, Elizabeth Magrath, Adrian Gherman, Julia Button, James Nguyen, Ferran Prados, Carole H Sudre, et al. Longitudinal multiple sclerosis lesion segmentation: resource and challenge. *NeuroImage*, 148:77–102, 2017.

[8] A Emre Kavur, N Sinem Gezer, Mustafa Barış, Pierre-Henri Conze, Vladimir Groza, Duc Duy Pham, Soumick Chatterjee, Philipp Ernst, Savaş Özkan, Bora Baydar, et al. Chaos challenge–combined (ct-mr) healthy abdominal organ segmentation. *arXiv preprint arXiv:2001.06535*, 2020.

[9] Nicholas Heller, Fabian Isensee, Klaus H Maier-Hein, Xiaoshuai Hou, Chunmei Xie, Fengyi Li, Yang Nan, Guangrui Mu, Zhiyong Lin, Miofei Han, et al. The state of the art in kidney and kidney tumor segmentation in contrast-enhanced ct imaging: Results of the kits19 challenge. *arXiv preprint arXiv:1912.01054*, 2019.

[10] Zoé Lambert, Caroline Petitjean, Bernard Dubray, and Su Ruan. Segthor: Segmentation of thoracic organs at risk in ct images. *arXiv preprint arXiv:1912.05950*, 2019.

[11] Larissa Heinrich, Jan Funke, Constantin Pape, Juan Nunez-Iglesias, and Stephan Saalfeld. Synaptic cleft segmentation in non-isotropic volume electron microscopy of the complete drosophila brain. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 317–325. Springer, 2018.

[12] Martin Maška, Vladimír Ulman, David Svoboda, Pavel Matula, Petr Matula, Cristina Ederra, Ainhoa Urbiola, Tomás España, Subramanian Venkatesan, Deepak MW Balak, et al. A benchmark for comparison of cell tracking algorithms. *Bioinformatics*, 30(11):1609–1617, 2014.

[13] Vladimír Ulman, Martin Maška, Klas EG Magnusson, Olaf Ronneberger, Carsten Haubold, Nathalie Harder, Pavel Matula, Petr Matula, David Svoboda, Miroslav Radojevic, et al. An objective comparison of cell-tracking algorithms. *Nature methods*, 14(12):1141–1152, 2017.

[14] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[15] Fabian Isensee, Jens Petersen, Andre Klein, David Zimmerer, Paul F Jaeger, Simon Kohl, Jakob Wasserthal, Gregor Koehler, Tobias Norajitra, Sebastian Wirkert, et al. nnu-net: Self-adapting framework for u-net-based medical image segmentation. *arXiv preprint arXiv:1809.10486*, 2018.

[16] Qihang Yu, Dong Yang, Holger Roth, Yutong Bai, Yixiao Zhang, Alan L Yuille, and Daguang Xu. C2fnas: Coarse-to-fine neural architecture search for 3d medical image segmentation. *arXiv preprint arXiv:1912.09628*, 2019.

[17] Patrick Bilic, Patrick Ferdinand Christ, Eugene Vorontsov, Grzegorz Chlebus, Hao Chen, Qi Dou, Chi-Wing Fu, Xiao Han, Pheng-Ann Heng, Jürgen Hesser, et al. The liver tumor segmentation benchmark (lits). *arXiv preprint arXiv:1901.04056*, 2019.

[18] Nicholas Heller, Niranjan Sathianathen, Arveen Kalapara, Edward Walczak, Keenan Moore, Heather Kaluzniak, Joel Rosenberg, Paul Blake, Zachary Rengel, Makinna Oestreich, et al. The kits19 challenge data: 300 kidney tumor cases with clinical context, ct semantic segmentations, and surgical outcomes. *arXiv preprint arXiv:1904.00445*, 2019.

[19] Fabian Isensee and Klaus H Maier-Hein. An attempt at beating the 3d u-net. *arXiv preprint arXiv:1908.02182*, 2019.

[20] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.

[21] David Svoboda and Vladimír Ulman. Mitogen: A framework for generating 3d synthetic time-lapse sequences of cell populations in fluorescence microscopy. *IEEE transactions on medical imaging*, 36(1):310–321, 2016.

[22] Carlos Castilla, Martin Maška, Dmitry V Sorokin, Erik Meijering, and Carlos Ortiz-de Solórzano. 3-d quantification of filopodia in motile cancer cells. *IEEE transactions on medical imaging*, 38(3):862–872, 2018.

[23] Dmitry V Sorokin, Igor Peterlik, Vladimír Ulman, David Svoboda, Tereza Nečasová, Katsiarina Morgaenko, Lívia Eiselleová, Lenka Tesařová, and Martin Maška. Filogen: a model-based generator of synthetic 3-d time-lapse sequences of single motile cells with growing and branching filopodia. *IEEE transactions on medical imaging*, 37(12):2630–2641, 2018.

[24] Fabian Isensee, Jens Petersen, Simon AA Kohl, Paul F Jäger, and Klaus H Maier-Hein. nnu-net: Breaking the spell on successful medical image segmentation. *arXiv preprint arXiv:1904.08128*, 2019.

[25] Fabian Isensee, Paul F Jäger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. Automated design of deep learning methods for biomedical image segmentation. *arXiv preprint arXiv:1904.08128*, 2019.

[26] Fabian Isensee, Paul F Jaeger, Peter M Full, Ivo Wolf, Sandy Engelhardt, and Klaus H Maier-Hein. Automatic cardiac disease assessment on cine-mri via time-series segmentation and domain specific features. In *STACOM*, pages 120–129. Springer, 2017.

[27] Gianluca Brugnara, Fabian Isensee, Ulf Neuberger, David Bonekamp, Jens Petersen, Ricarda Diem, Brigitte Wildemann, Sabine Heiland, Wolfgang Wick, Martin Bendszus, et al. Automated volumetric assessment with artificial neural networks might enable a more accurate assessment of disease burden in patients with multiple sclerosis. *European Radiology*, pages 1–9, 2020.