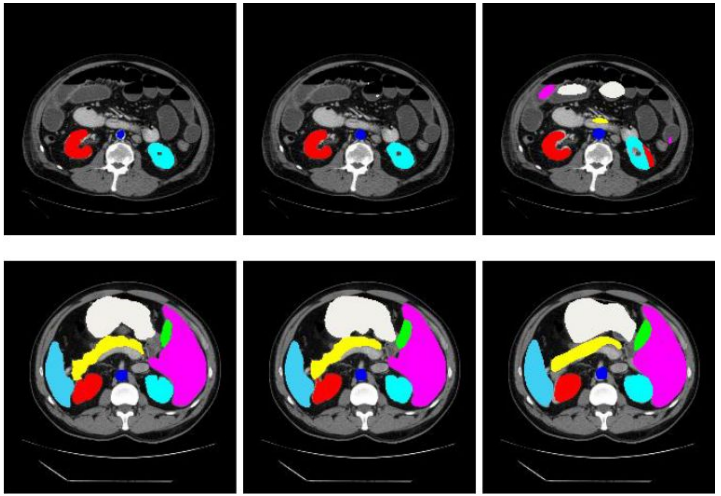
nnU-Net: A Self Configuring Method For deep-learning based biomedical image segmentation

Behnam Nikbakht
Majid Bahrehvar
Ali Salmani

Introduction

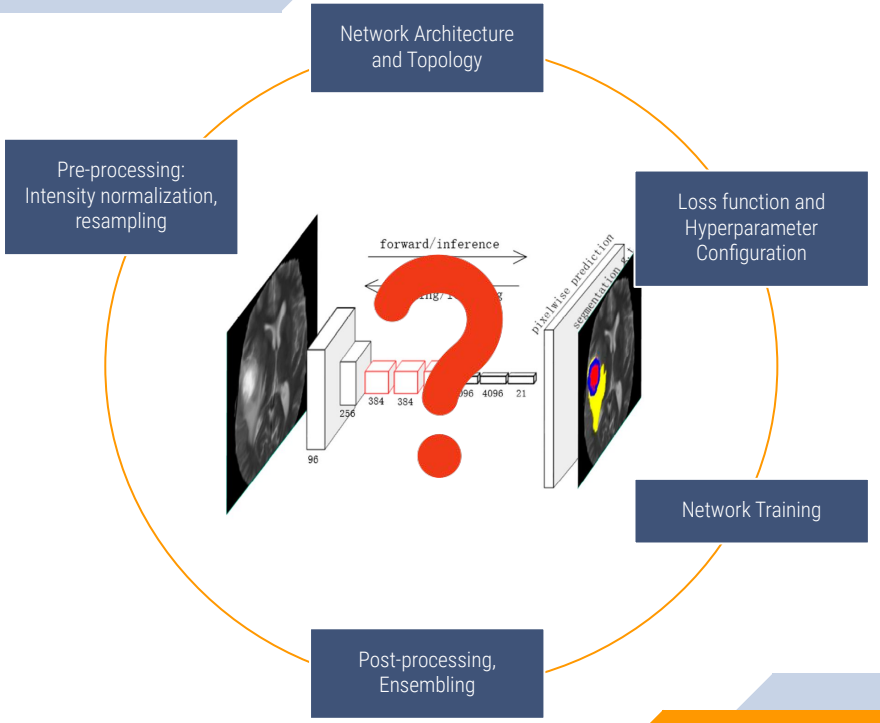
nnU-Net (“no new U-Net”) seeks to establish a standardized pipeline for the medical image segmentation process.



Semantic Segmentation

Semantic segmentation is mostly used method in biomedical Imaging

Manual Segmentation Pipeline Design



Manual Method Configuration

- Time Consuming trial-and-error process
- Success depends on experience of the researcher
- Inaccessible to non-experts
- Needs to be repeated on every dataset
- Reduces method evaluation to one or few datasets in practice
- Might cause subpar baseline performance

nnU-Net

An out-of-the-box tool that automatically configures entire state-of-the-art segmentation pipelines for arbitrary biomedical datasets **without requiring expert knowledge** or extensive compute resources to run

AutoML Paradigm

nnU-Net Design Goals

- It should just work ‘out-of-the-box’, no expert knowledge required
- Requires standard deep-learning hardware
- Designed for biomedical datasets
- Holistic Configuration of the pipeline

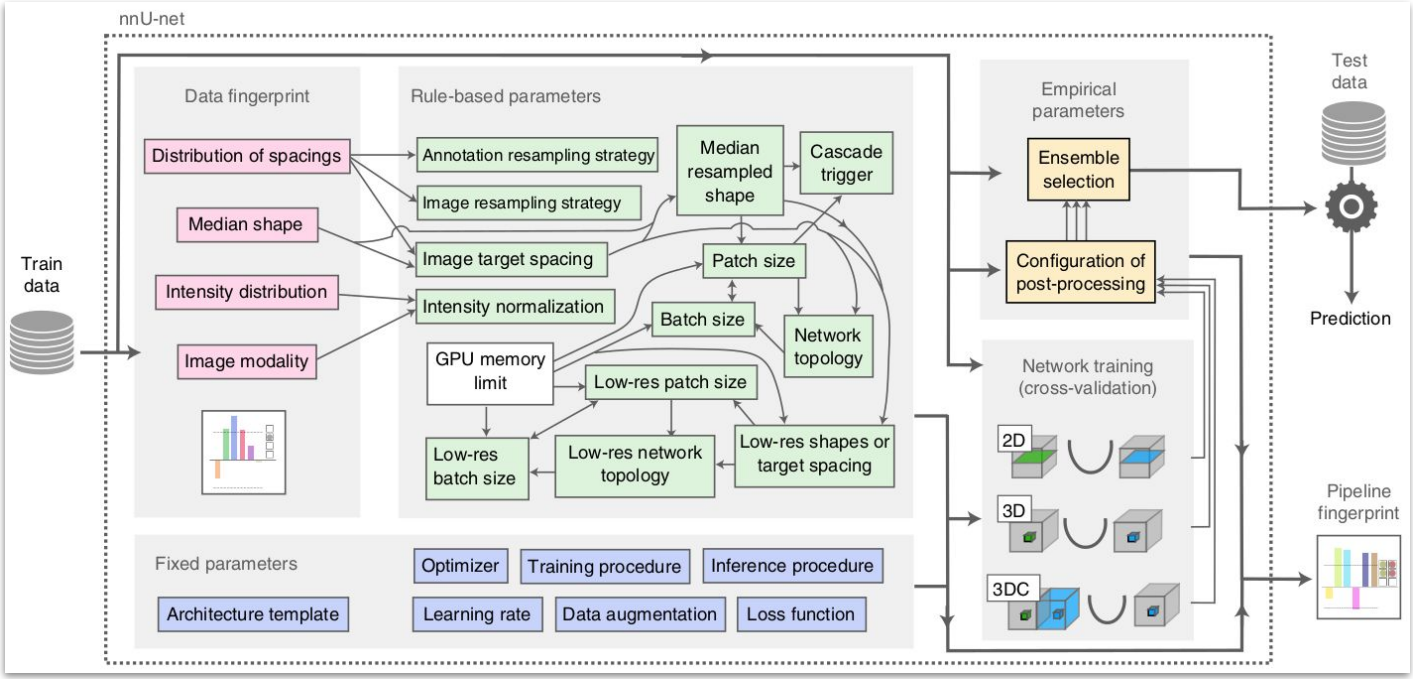
AutoML

- Outstanding performance in natural image processing, but not biomedical
- Reliance on empirical optimization:
  - Compute resources
  - Dataset size
- Search space requires expert knowledge
- Not holistic

Architecture Design decision

- U-Net like architectures enable state of the art segmentation when the pipeline is well-configured.
- Sophisticated architectural variations are not required to achieve state of the art performance.
- Architecture only use plain convolutions, instance normalization and Leaky non-linearities
- They use two computational blocks per resolution stage in both encoder and decoder.
- Downsampling is done with strided convolutions,upsampling is done with convolutions transposed.
- Predicting which configurations should be trained on which dataset is a future research direction.

Architecture



Method

Design choice	Required input	Automated (fixed, rule-based or empirical) configuration derived by distilling expert knowledge (more details in online methods)
Learning rate	–	Poly learning rate schedule (initial, 0.01)
Loss function	–	Dice and cross-entropy
Architecture template	–	Encoder–decoder with skip-connection ('U-Net-like') and instance normalization, leaky ReLU, deep supervision (topology-adapted in inferred parameters)
Optimizer	–	SGD with Nesterov momentum ($\mu = 0.99$)
Data augmentation	–	Rotations, scaling, Gaussian noise, Gaussian blur, brightness, contrast, simulation of low resolution, gamma correction and mirroring
Training procedure	–	1,000 epochs × 250 minibatches, foreground oversampling
Inference procedure	–	Sliding window with half-patch size overlap, Gaussian patch center weighting
Intensity normalization	Modality, intensity distribution	If CT, global dataset percentile clipping & z score with global foreground mean and s.d. Otherwise, z score with per image mean and s.d.
Image resampling strategy	Distribution of spacings	If anisotropic, in-plane with third-order spline, out-of-plane with nearest neighbor Otherwise, third-order spline
Annotation resampling strategy	Distribution of spacings	Convert to one-hot encoding → If anisotropic, in-plane with linear interpolation, out-of-plane with nearest neighbor Otherwise, linear interpolation
Image target spacing	Distribution of spacings	If anisotropic, lowest resolution axis tenth percentile, other axes median. Otherwise, median spacing for each axis. (computed based on spacings found in training cases)
Network topology, patch size, batch size	Median resampled shape, target spacing, GPU memory limit	Initialize the patch size to median image shape and iteratively reduce it while adapting the network topology accordingly until the network can be trained with a batch size of at least 2 given GPU memory constraints. for details see online methods.
Trigger of 3D U-Net cascade	Median resampled image size, patch size	Yes, if patch size of the 3D full resolution U-Net covers less than 12.5% of the median resampled image shape
Configuration of low-resolution 3D U-Net	Low-res target spacing or image shapes, GPU memory limit	Iteratively increase target spacing while reconfiguring patch size, network topology and batch size (as described above) until the configured patch size covers 25% of the median image shape. For details, see online methods.
Configuration of post-processing	Full set of training data and annotations	Treating all foreground classes as one; does all-but-largest-component-suppression increase cross-validation performance? Yes, apply; reiterate for individual classes No, do not apply; reiterate for individual foreground classes
Ensemble selection	Full set of training data and annotations	From 2D U-Net, 3D U-Net or 3D cascade, choose the best model (or combination of two) according to cross-validation performance

nnU-Net divides hyper-parameters into 3 types:

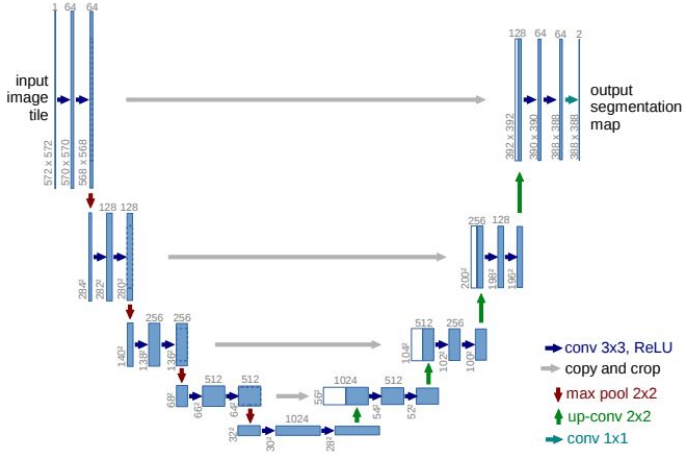
1. Fixed configurations
2. Rule-Based configurations
3. Empirical Configurations

In all cases, validation set performance does not affect training time, unlike in AutoML.

Fixed Configurations

Parameters that don't need to change when we are moving from one dataset to another:

- **Model architecture** (U-Net), hence the name "no new U-Net".
  - The Encoder, Decoder with skip-connection don't have to be changed
- **Learning Rate:** 0.01



Fixed Configurations

- **Optimizer** (SGD with Nesterov momentum 0.9).
- **Training procedure** (1000 epocs * 250 minibatchs with 5-fold cross-validation and foreground over-sampling).
- **Inference procedure** (sliding window with half patch size overlap Gaussian patch importance weighting).

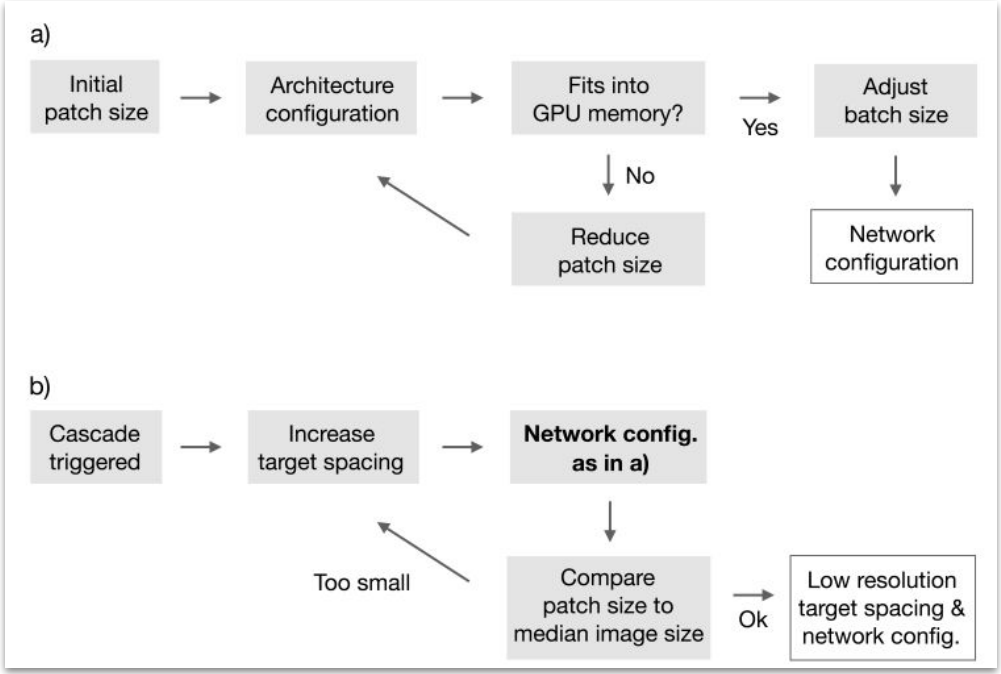
Rule-Based Configurations

- Image Intensity Normalization (Use HU if CT, else use z normalization).
- Image Resampling Strategy (If anisotropic, use cubic spline if ratio is within 3, else use nearest neighbor interpolation).
- Image Spacing (lowest 10th percentile if anisotropic, else median).
- Use 3D cascade (if image is too large).
- Model pooling depth (reduce anisotropic side until less than 3, pool until side length becomes 4).
- Mini-batch size (largest mini-batch that fits within 11 GB during training).

Rule-Based Model Configurations

- Network Topology, Patch Size, and Batch Size are configured at the start of training.
- Networks are expected to give approximate GPU memory usage expectations.
- Patch size is given high priority as large patch size is necessary for segmentation.
- Network topology is designed to pool until side length is 4 and anisotropic degree is within 3.
- Batch size is set to be lower than 5% of total data.

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.

b) The input patch size of the 3D lowres U-Net must cover at least 1/4 of the median shape of the resampled training cases to ensure sufficient contextual information.

Empirical Parameters

- **Model Selection**
  - nnU-Net generate multiple models
    - 2D U-Net
    - 3D U-Net
    - 3D Cascade
      - only for datasets with large images
  - 5-fold cross-validation to assess accuracy/performance
  - Pick the best performing model, or a combination of them
- **Post-Processing**
  - The medical image contains only one instance of the target
  - Connected component analysis is useful here

Evaluating the performance on the validation set

IoU

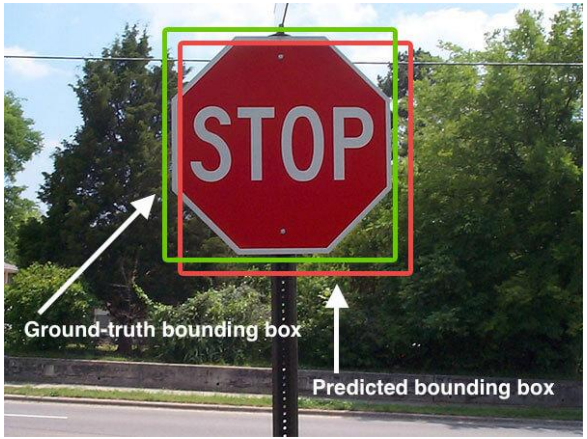
area of overlap / area of union

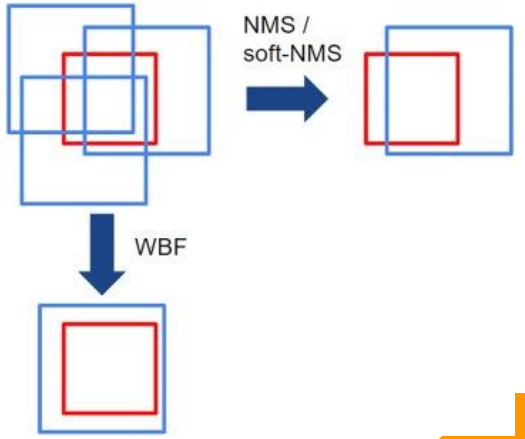
NMS

uses the objectiveness score (confidence) and IoU

WBC (WBF)

combines all of the segments





Post-Processing

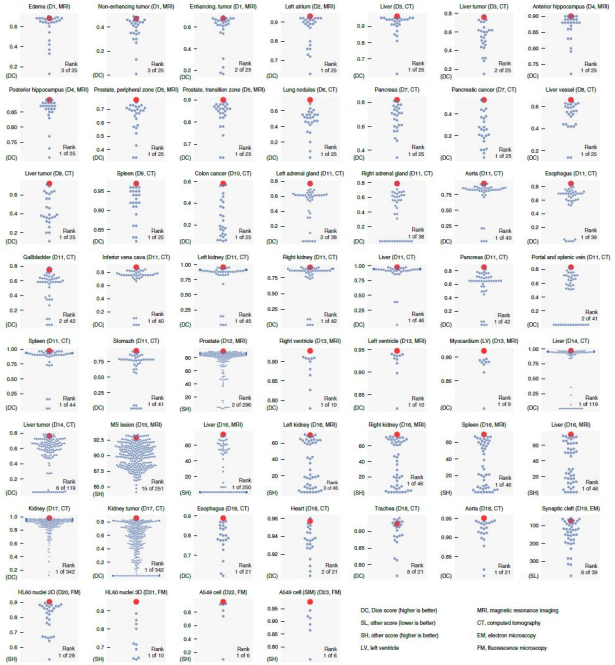
- Connected-component based
- Removing all but the largest connected component
- Benchmarks the effect of suppressing smaller components on the cross-validation results:
    - Treat all foreground classes as one component.
    - Check for the suppression of all but the largest region
    - Apply the same procedure for individual classes.

Comments

- Not depend on performance on validation metrics.
- No need to have any prior requirement or specific knowledge
- out-of-the-box

Results

- First place in 33 of 53 challenges with no modifications.
- Is close to the winning submissions in the remaining 20 tasks
- Many of the competitors use a same architecture with varying results



Results

- baseline for medical segmentation
- Results for COVID19 segmentation challenge in 2020.
- nnUNet features in the top-tier of nearly all medical segmentation challenges.
- minor modifications specializing it for the task

Table 1 | Top-10 finalists after statistical ranking. "Value" represents the average rank the algorithm achieved across all tasks. We also show if methods were automated, used external data for training, the input data dimensions used in the algorithms, and the network architecture.

Rank	Value	ID #	Fully Automated	Extra Data	Pretrained	Ensemble	Data Dimension	Network Architecture	Authors	Country
1	2.6	53	✓	✓	✗	✗	3D	nnU-Net	S. Hu et al.	China
2	6.0	38	✓	✗	✗	✓	3D	nnU-Net	F. Isensee et al.	Germany
3	7.7	65	✓	✗	✗	✓	2D/3D	nnU-Net	C. Tang	USA
4	8.4	58	✓	✗	✗	✓	3D	nnU-Net	Q. Yu et al.	China
5	8.5	31	✓	✗	✗	✓	3D	nnU-Net	J. Sölter et al.	Luxembourg
6	9.2	50	✓	✗	✗	✓	2D/3D	nnU-Net	T. Zheng & L. Zhang	Japan
6	9.2	68	✓	✗	✓	✗	2D/3D	VGG16 Hybrid, MONAI	V. Liauchuk et al.	Belarus
8	9.4	95	✓	✗	✗	✓	3D	nnU-Net	Z. Zhou et al.	China
9	10.6	29	✓	✗	✗	✗	3D	nnU-Net	J. Moltz et al.	Germany
10	11.3	15	✓	✗	✗	✗	3D	U-Net	B. Oliveira et al.	Portugal

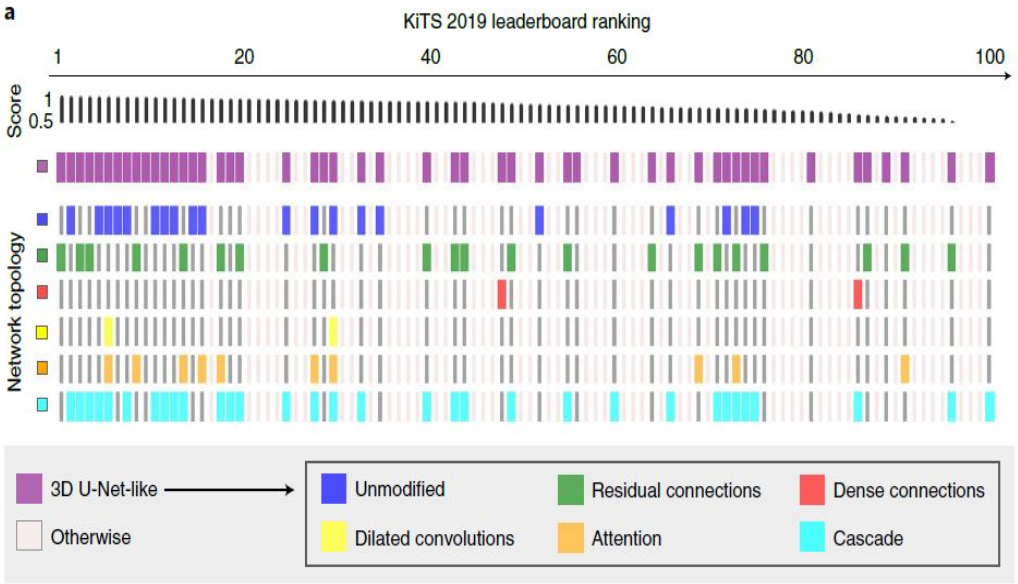
THANKS!

Any questions?

Architecture Design decision

- U-Net like architectures enable state of the art segmentation when the pipeline is well-configured.
- It should just work ‘out of the box’, no expert knowledge required
- Required standard deep learning hardware
- Designed for biomedical dataset.
- Holistic configuration for entire pipeline.

Problem Statement

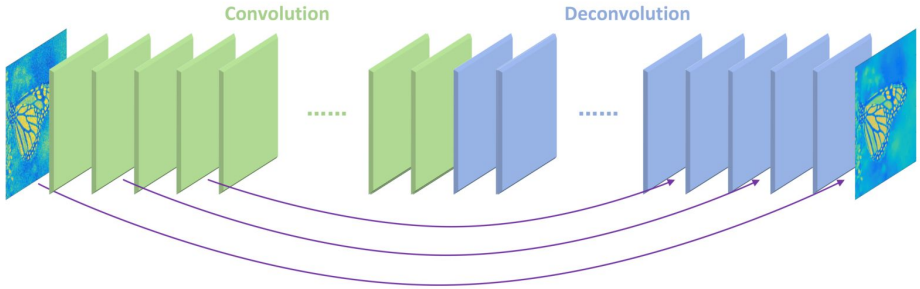


Analysis of challenge leaderboard entries shows that using superficially similar methods and model architectures can lead to vastly different results due to implementation details.

Moreover, no single method stands out as being necessary for high scores (except deep learning).

Skip-Connection (OR Shortcut Connections)

Skips some of the layers in the neural network and feeds the output of one layer as the input to the next layers



Data Augmentation

- 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.
- Gaussian Noise. Zero centered additive Gaussian noise is added to each voxel in the sample independently
- Gaussian Blur. Blurring is applied with a probability of 0.2 per sample.
- Brightness. Voxel intensities are multiplied by x ~ U (0.7, 1.3) with a probability of 0.15
- Contrast. Voxel intensities are multiplied by x ~ U (0.65, 1.5) with a probability of 0.15.
- Simulation of low resolution. This augmentation is applied with a probability of 0.25 persample and 0.5 per associated modality.
- Gamma augmentation. This augmentation is applied with a probability of 0.15.Mirroring. All patches are mirrored with a probability of 0.5 along all axes.
- Mirroring. All patches are mirrored with a probability of 0.5 along all axes.

Inferred Parameters

- Dynamic Network adaptation
  - Needs to be adapted to the size and spacing of the input patches seen during training
  - perform downsampling until the feature maps are relatively small (minimum is 4 × 4(×4))
  - 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 inthe 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
- Configuration of the input patch size
- Batch size
- Target spacing and resampling
- Intensity normalization

Training Scheme

- All trainings run for a fixed length of 1000 epochs, where each epoch is defined as 250 training iterations
- As for the optimizer, stochastic gradient descent with a high initial learning rate (0.01) and a large nesterov momentum (0.99) empirically provided the best results
- 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
- 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.
- combining
- the Dice loss with a cross-entropy loss improved training stability and segmentation accuracy. Therefore, the two loss terms are simply averaged.

K-fold cross validation

- Shuffle randomly.
- Split into k groups
- For each group:
  - Take the group as test data set
  - Take the remaining groups as training data set
  - Fit a model
  - Retain the evaluation score
  - discard the model and continue
- Summarize the model

Dice Coefficient

- Class Imbalance
  - background is much larger than foreground
  - Used for connected-component based analysis
- IoU (left) vs Dice Coefficient (Right) - the result is in black color