Medical Imaging Intelligent Reality Lab

ASAN Medical Center

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

Nature Methods, 2021

2023.05.27.월

Presenter: Hyunwook Kim

Table of Contents

- Introduction
- Materials and Methods
- Results
- Conclusions

Introduction

What is Image Segmentation?

- **Wikipedia**
  - In digital image processing and computer vision, image segmentation is **the process of partitioning a digital image into multiple image segments**, also known as image regions or image objects (sets of pixels).

Reference: https://en.wikipedia.org/wiki/Image_segmentation, https://www.superannotate.com/blog/image-segmentation-for-machine-learning

What is Image Segmentation?

- **Image Segmentation Using Deep Learning: A Survey**, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021
  - Image segmentation can be formulated as the problem of **classifying pixels** with semantic labels (semantic segmentation), or partitioning of individual objects (instance segmentation), or both (panoptic segmentation).

(a) Image (b) Semantic segmentation (c) Instance segmentation (d) Panoptic segmentation

Reference: https://ieeexplore.ieee.org/document/9356353, https://encord.com/blog/guide-to-semantic-segmentation/

Medical Image Segmentation

(a) Retinal Vessels

(b) Brain Tumors

(c) Multi-organ Abdomen

(d) Liver Tumors

(e) Nuclei

(f) Cells

(g) Skin Lesion

(h) Corneal Nerve

Reference: U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications, IEEE Access, 2021

Related Studies: U-Net

Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Fig. 4. Result on the ISBI cell tracking challenge. (**a**) part of an input image of the "PhC-U373" data set. (**b**) Segmentation result (cyan mask) with manual ground truth (yellow border) (**c**) input image of the "DIC-HeLa" data set. (**d**) Segmentation result (random colored masks) with manual ground truth (yellow border).

Table 2. Segmentation results (IOU) on the ISBI cell tracking challenge 2015.

Name	PhC-U373	DIC-HeLa
IMCB-SG (2014)	0.2669	0.2935
KTH-SE (2014)	0.7953	0.4607
HOUS-US (2014)	0.5323	-
second-best 2015	0.83	0.46
u-net (2015)	**0.9203**	**0.7756**

Reference: U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI, 2015

Related Studies: 3D U-Net

Input Shape
132 x 132 x 116 x 3

Output Shape
44 x 44 x 28

Fig. 2. The 3D u-net architecture. Blue boxes represent feature maps. The number of channels is denoted above each feature map.

Reference: 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation, MICCAI, 2016

Related Studies: U-Net and Its Variants

- **U-Net을 기반으로 한 다양한 모델이 있음**
  - 2D U-Net (Base)
  - 3D U-Net
  - Attention U-Net
  - Inception U-Net
  - Residual U-Net
  - Recurrent U-Net
  - Dense U-Net
  - UNet^{++}
  - …

Reference: U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications, IEEE Access, 2021

MI2RL Medical Imaging Intelligent Reality Lab

Limitations

- Despite the success of deep learning-based segmentation methods, **their applicability to specific image analysis problems** is often limited due to the need for high expertise and precision in their design and configuration.

- These limitations frequently force users into a **manual trial-and-error process** that relies heavily on personal experience, is poorly documented, and often results in suboptimal segmentation pipelines.

MI²RL Medical Imaging Intelligent Reality Lab

Main Contributions

- They developed nnU-Net, a deep learning-based segmentation method that **automatically configures itself, including preprocessing, network architecture, training and post-processing for any new task**.

- Without manual intervention, **nnU-Net surpasses most existing approaches**, including highly specialized solutions **on 23 public datasets** used in international biomedical segmentation competitions.

Materials and Methods

10 Datasets in Medical Segmentation Decathlon (D1 – D10)

- Their implementation of this recipe was developed and validated on the ten datasets provided by the Medical Segmentation Decathlon.
- The general applicability of nnU-Net's automated configuration is demonstrated in 13 additional datasets.

Reference: The Medical Segmentation Decathlon, Nature Communications, 2022

Datasets: D1 – D12 from International Challenges

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

Datasets: D13 – D23 from International Challenges

ID	Dataset Name	Associated Challenges	Modalities	Median Shape (Spacing [mm])	N Class.	Rarest Class Ratio	N Training Cases	Segmentation Tasks
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

Data Fingerprint: Brain Tumor (Edema, Necrosis, Enhancing Tumor)

- Median number of voxels
- Size anisotropy
- Spacing anisotropy
- Number of training cases
- Rarest class ratio
- Number of classes
- Modality

Data Fingerprints across Different Challenge Datasets

D1	MSD, brain tumor (edema, necrosis, enhancing tumor)	D9	MSD, spleen (spleen)	D17	KiTS (kidneys, kidney tumor)
D2	MSD, heart (left atrium)	D10	MSD, colon (colon cancer)	D18	SegTHOR (heart, aorta, esophagus, trachea)
D3	MSD, liver (liver, liver tumor)	D11	BCV-Abdomen (13 abdominal organs)	D19	CREMI (synaptic cleft)
D4	MSD, hippocampus (anterior and posterior)	D12	PROMISE12 (prostate)	D20	CTC - Fluo-N2DH-SIM+ (HL60 nuclei)
D5	MSD, prostate (peripheral zone, transition zone)	D13	ACDC (left and right ventricle, myocardium)	D21	CTC - Fluo-N3DH-SIM+ (HL60 nuclei)
D6	MSD, lung (lung nodules)	D14	LiTS (liver, liver tumor)	D22	CTC - Fluo-C3DH-A549 (A549 cell)
D7	MSD, pancreas (pancreas, pancreatic tumor)	D15	MSLes (MS lesions)	D23	CTC - Fluo-C3DH-A549-SIM (A549 cell)
D8	MSD, hepatic vessel (hepatic vessels, liver tumors)	D16	CHAOS (liver, spleen, left and right kidneys)		

MSD, Medical Segmentation Decathlon; CTC, Cell Tracking Challenge

Data Fingerprints across Different Challenge Datasets

- The **complexity of method configuration is amplified** by the fact that suitable pipeline settings either directly or indirectly depend on the data fingerprint under potentially complex relations.

- As a consequence, pipeline settings that are identified as **optimal for one dataset** (such as KiTS, see above) **may not generalize to others**, resulting in a need for reoptimization for each individual dataset.

Pipeline Fingerprint

Automated Method Configuration

Automated Method Configuration

Automated Method Configuration: Fixed Parameters

Automated Method Configuration: Fixed Parameters

- **Fixed parameters** that do not require adaptation between datasets and identify a robust common configuration

Data Augmentation	Rotations, Scaling, Gaussian noise, Gaussian blur, Brightness, Contrast, Simulation of low resolution (= Downsampling), Gamma correction, Mirroring (= Flipping)
Architecture Template	Encoder–decoder with skip-connection ('U-Net-like'), Instance normalization, Leaky ReLU, Deep Supervision (topology-adapted in inferred parameters)
Loss function	Dice, Cross-entropy
Optimizer	SGD with Nesterov momentum (μ = 0.99)
Learning Rate	Polynomial learning rate schedule (initial, 0.01)
Training Procedure	1,000 epochs × 250 mini-batches, Foreground oversampling
Inference Procedure	Sliding window with half-patch size overlap, Gaussian patch center weighting

Automated Method Configuration: Rule-based Parameters

MI2RL Medical Imaging Intelligent Reality Lab

Automated Method Configuration: Rule-based Parameters

- (Heuristic) **Rule-based parameters** that consider explicit dependencies between dataset properties (=data fingerprint) and design choices (= pipeline fingerprint)
  - Image/Annotation Resampling Strategies
  - Intensity Normalization
  - Image Target Spacing
  - Configuration of Full/Low Resolution 3D U-Net
    - Batch Size, Patch Size, Network Topology
  - Trigger of 3D U-Net Cascade

Automated Method Configuration: Rule-based Parameters

Automated Method Configuration: Rule-based Parameters

- **Image Resampling Strategy**
  - Spacing Distribution이 Anisotropic인 경우 (= Spacing 크기가 각 Axis별로 같지 X),
    - In-plane → Third-order Spline Interpolation
    - Out-of-plane → Nearest Neighbor Interpolation
  - Spacing Distribution이 Anisotropic이 아닌 경우,
    - Third-order Spline Interpolation

Automated Method Configuration: Rule-based Parameters

- **Image Resampling Strategy**
  - Example

    Imagine you have an MRI scan where:

    - Voxel size in the x-direction: 1 mm
    - Voxel size in the y-direction: 1 mm
    - Voxel size in the z-direction: 5 mm

    This means that in the x-y plane, the voxel spacing is consistent (1 mm), but in the z-direction, the spacing is much larger (5 mm), leading to anisotropy.

Reference: https://chat.openai.com/

Automated Method Configuration: Rule-based Parameters

- **Image Resampling Strategy**
  - Example

    Applying the Strategy

    - **In-Plane (x-y plane)**:
      - Use third-order spline interpolation to resample the image. This maintains high quality and smooth transitions between voxels in the plane where the resolution is consistent.
    - **Out-of-Plane (z-direction)**:
      - Use nearest neighbor interpolation to resample the image. This avoids introducing artifacts and keeps the image true to its original data, despite the large voxel spacing.

Reference: https://chat.openai.com/

Automated Method Configuration: Rule-based Parameters

- **Annotation Resampling Strategy**
  - Annotation을 One-hot Encoding 형태로 변경
  - Spacing Distribution에 따라 Resampling 진행
    - Spacing Distribution이 Anisotropic인 경우,
      - In-plane → Linear Interpolation
      - Out-of-plane → Nearest Neighbor Interpolation
    - Spacing Distribution이 Anisotropic이 아닌 경우,
      - Linear Interpolation

Automated Method Configuration: Rule-based Parameters

- **Annotation Resampling Strategy**
  - Example

    1. **Original Annotation**:
       - Suppose you have an annotation map with three classes labeled as 0, 1, and 2.
    2. **One-Hot Encoding**:
       - Convert the annotation map to one-hot encoding:
         - Original: `[[0, 1, 2], [1, 0, 2]]`
         - One-Hot:
           - Class 0: `[[1, 0, 0], [0, 1, 0]]`
           - Class 1: `[[0, 1, 0], [1, 0, 0]]`
           - Class 2: `[[0, 0, 1], [0, 0, 1]]`

Reference: https://chat.openai.com/

Automated Method Configuration: Rule-based Parameters

- **Annotation Resampling Strategy**
  - Example

    3. **Resampling**:
       - Apply linear interpolation within the plane and nearest neighbor interpolation out of the plane if the data is anisotropic.
       - After resampling, convert the one-hot encoded annotations back to categorical labels by taking the argmax across the class dimension.

Reference: https://chat.openai.com/

Automated Method Configuration: Rule-based Parameters

Automated Method Configuration: Rule-based Parameters

- **Image Target Spacing**
  - Spacing Distribution이 Anisotropic인 경우,
    - Lowest Resolution Axis → Lowest Tenth Percentile Spacing
    - Other Axes → Median Spacing
  - Spacing Distribution이 Anisotropic이 아닌 경우,
    - All Axes → Median Spacing

Automated Method Configuration: Rule-based Parameters

- **Image Target Spacing**
  - Example

Consider a dataset with varying voxel spacings. Here's why you set a target spacing:

1. **Initial Spacings**:
   - Image 1: 1.0 x 1.0 x 2.5 mm
   - Image 2: 0.8 x 0.8 x 1.5 mm
   - Image 3: 1.2 x 1.2 x 2.0 mm
2. **Calculate Target Spacing**:
   - Median spacing for each axis.
   - Tenth percentile for the lowest resolution axis if anisotropic.

Reference: https://chat.openai.com/

Automated Method Configuration: Rule-based Parameters

- Image Target Spacing
  - Example

    3. **Resample to Target Spacing**:

       - All images are resampled to, for example, 1.0 x 1.0 x 1.5 mm.

Reference: https://chat.openai.com/

Automated Method Configuration: Rule-based Parameters

Automated Method Configuration: Rule-based Parameters

- **Intensity Normalization**
  - Modality에 따라 다른 방법 적용
    - Modality가 CT인 경우,
      - Global Dataset에 대해 0.5와 99.5 Percentile Clipping을 적용
      - Global Foreground의 Mean과 Standard Deviation을 이용하여 Z-score로 변환
    - Modality가 CT가 아닌 경우,
      - Global Foreground의 Mean과 Standard Deviation을 이용하여 Z-score로 변환

Automated Method Configuration: Rule-based Parameters

Automated Method Configuration: Rule-based Parameters

- **Batch Size, Patch Size, Network Topology**
  - Patch Size를 Median Resampled Image Shape으로 초기화
  - GPU 메모리가 허용되는 범위 내에서 Patch Size를 줄이는 작업과 Network Topology를 조정하는 작업을 반복함
    - 참고로 Batch Size는 최소한 2는 유지가 되어야 함

Automated Method Configuration: Rule-based Parameters

- **Batch Size, Patch Size, Network Topology**
  - Example

1. **Initial Patch Size**:
   - Median image shape: 256x256x128 voxels.
   - Start with a patch size of 256x256x128.
2. **GPU Memory Check**:
   - Suppose this patch size exceeds the GPU memory limit.
   - Reduce patch size to 128x128x64 voxels.

Reference: https://chat.openai.com/

Automated Method Configuration: Rule-based Parameters

- **Batch Size, Patch Size, Network Topology**
  - Example

3. **Adapt Network Topology**:
   - Adjust the number of downsampling layers:
     - Initial: 4 layers (down to 16x16x8).
     - After reduction: 3 layers (down to 16x16x8 but with different intermediate sizes).
   - Modify convolutional kernel sizes:
     - Initial: 3x3x3 kernels.
     - After reduction: 2x2x2 kernels.
   - Adjust pooling operations:
     - Initial: 2x2x2 pooling.
     - After reduction: 1x1x1 pooling (if necessary).

Reference: https://chat.openai.com/

Automated Method Configuration: Rule-based Parameters

Automated Method Configuration: Rule-based Parameters

- **Trigger of 3D U-Net Cascade**
  - 3D Full Resolution U-Net의 Patch Size가 Median Resampled Image Shape의 12.5%보다 작은 경우 3D U-Net Cascade를 사용함

Reference: nnU-Net: Self-adapting Framework for U-Net-Based Medical Image Segmentation, Bildverarbeitung für die Medizin, 2019

Automated Method Configuration: Rule-based Parameters

Automated Method Configuration: Rule-based Parameters

- **Configuration of Low Resolution 3D U-Net**
  - Image Target Spacing을 높이고 그에 따라 3D Low Resolution U-Net의 Patch Size, Network Topology, Batch Size를 조정함
  - 위 작업을 Patch Size가 Median Resampled Image Shape의 25%를 Cover 할 때까지 반복함

Automated Method Configuration: Rule-based Parameters

- Configuration of Low Resolution 3D U-Net
  - Example

    Let's consider an example to illustrate this process:

    - **Initial Configuration**:
      - Median image shape: 256 x 256 x 256 voxels
      - Initial target spacing: 1.0 x 1.0 x 1.0 mm
      - Initial patch size: 64 x 64 x 64 voxels

Reference: https://chat.openai.com/

Automated Method Configuration: Rule-based Parameters

- Configuration of Low Resolution 3D U-Net
  - Example
    - **Iterative Process**:
      1. **Increase Target Spacing**:
         - Increase to 1.5 x 1.5 x 1.5 mm
         - The image shape now effectively reduces (e.g., 170 x 170 x 170 voxels)
      2. **Adjust Patch Size**:
         - New patch size might be adjusted to 85 x 85 x 85 voxels
      3. **Adjust Network Topology**:
         - Modify the network layers to handle the new patch size. For example, reduce the number of downsampling layers or adjust the kernel sizes.

Reference: https://chat.openai.com/

Automated Method Configuration: Rule-based Parameters

- Configuration of Low Resolution 3D U-Net
  - Example
    - **Iterative Process**:

      4. **Adjust Batch Size**:
         - Ensure the batch size fits within the GPU memory limits, potentially keeping it at 2.
      5. **Check Coverage**:
         - Calculate the percentage coverage:
           $\text{Coverage Percentage} = \left(\frac{85 \times 85 \times 85}{170 \times 170 \times 170}\right) \times 100 = 25\%$
         - Since the coverage is 25%, the condition is satisfied.

Reference: https://chat.openai.com/

Automated Method Configuration: Empirical Parameters

Automated Method Configuration: Empirical Parameters

- **Empirical parameters** decided empirically from the data
  - Configuration of Post-processing
    - 모든 Foreground Class를 하나로 설정 → All-but-largest-component-suppression

      → Cross-validation Performance가 오르는지 확인
      - Yes → 위 작업을 다시 반복함
      - No → Post-processing 중단
  - Ensemble Selection
    - 2D U-Net, 3D U-Net, 3D Cascade 중에서 Cross Validation 성능이 가장 좋은 모델 또는 조합을 최종적으로 선택함

Automated Method Configuration (Summary)

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

Materials and Methods

Network Architecture for ACDC Dataset

Dataset	
Median Image Size	9x237x256
Median Image Spacing	10x1.56x1.56mm
Normalization	Z-score

2D U-Net	
Target Spacing	NAx1.56x1.56mm
Median Shape @ Target Spacing	NAx237x208
Patch Size	256x224
Batch Size	58

3D full resolution U-Net	
Target Spacing	5x1.56x1.56mm
Median Shape @ Target Spacing	18x237x208
Patch Size	20x256x224
Batch Size	3

- Input
- 1x1(x1) conv - softmax
- 3x3 conv - IN - IReLU
- 1x3x3 conv - IN - IReLU
- 3x3x3 conv - IN - IReLU
- Convolution Transposed
- Skip Connection
- 2,2,(2) (stride n,n, (n); default=1,1, (1))
- 32x256x224 (1.56x1.56) Feature Map Size (Spacing in mm)

Network Architecture for LiTS Dataset

Dataset	
Median Image Size	432x512x512
Median Image Spacing	1x0.77x0.77mm
Normalization	Clip to [-17, 201], then normalize with μ=99.4, σ=39.39

2D U-Net	
Target Spacing	NAx0.77x0.77mm
Median Shape @ Target Spacing	NAx512x512
Patch Size	512x512
Batch Size	12

3D full resolution U-Net	
Target Spacing	1x0.77x0.77mm
Median Shape @ Target Spacing	482x512x512
Patch Size	128x128x128
Batch Size	2

3D low resolution U-Net	
Target Spacing	2.47x1.9x1.9mm
Median Shape @ Target Spacing	195x207x207
Patch Size	128x128x128
Batch Size	2

- Input
- 1x1(x1) conv - softmax
- 3x3 conv - IN - IReLU
- 3x3x3 conv - IN - IReLU
- Convolution Transposed
- Skip Connection
- 2,2,(2) (stride n,n, (n); default=1,1, (1))
- 32x256x224 (1.56x1.56) Feature Map Size (Spacing in mm)

Results

Results

Performances: D1 – D11

Results

Performances: D11 – D23

Pipeline Fingerprints from KiTS 2019 Leaderboard Entries

Pipeline Fingerprints from KiTS 2019 Leaderboard Entries

Evaluation of Design Decisions across Multiple Tasks

Conclusions

Conclusions

- nnU-Net achieves state-of-the-art semantic segmentation in biomedical imaging, automating method configuration without expert knowledge or special resources.

- It serves as an out-of-the-box tool for segmentation, a standardized baseline for comparison, and a framework for evaluating new ideas without manual effort.

Medical Imaging Intelligent Reality Lab

References

- https://youtu.be/3po8qVzz5Tc?si=2_Jx-BdcELazmNDo
- https://youtu.be/K3qRY9Q-BFo?si=_vnyXQXLHGf5R5HE

ASAN FOUNDATION
아산재단

ASAN
Medical Center

Thank you

MI2RL

Medical
Imaging
Intelligent
Reality
Lab

Appendix

Datasets

Anisotropy

- Anisotropy is a structural property of non-uniformity in different directions, as opposed to isotropy and an anisotropic object or pattern has properties that differ according to direction of measurement.

- For example, many materials exhibit very different physical or mechanical properties when measured along different axes, e.g. absorbance, refractive index, conductivity, and tensile strength.

WMAP image of the (extremely tiny) anisotropies in the cosmic microwave background radiation

Reference: https://en.wikipedia.org/wiki/Anisotropy

How to Calculate Size and Spacing Anisotropies?

- Let's consider an example where the dimensions of an image are 200×300×400 voxels and the voxel spacings are 1.0mm×1.5mm×2.0mm.

Size Anisotropy Calculation:

1. Dimensions: $L_x = 200$, $L_y = 300$, $L_z = 400$
2. Ratios:

$$R_{xy} = \frac{200}{300} = 0.67, \quad R_{xz} = \frac{200}{400} = 0.50, \quad R_{yz} = \frac{300}{400} = 0.75$$

3. Size Anisotropy:

$$\text{Size Anisotropy} = \max(0.67, 0.50, 0.75) = 0.75$$

Reference: https://chat.openai.com/

How to Calculate Size and Spacing Anisotropies?

- Let's consider an example where the dimensions of an image are 200×300×400 voxels and the voxel spacings are 1.0mm×1.5mm×2.0mm.

Spacing Anisotropy Calculation:

1. Voxel Spacings: $S_x = 1.0\,\text{mm}$, $S_y = 1.5\,\text{mm}$, $S_z = 2.0\,\text{mm}$
2. Ratios:

$$R_{sx} = \frac{1.0}{1.5} = 0.67, \quad R_{sz} = \frac{1.0}{2.0} = 0.50, \quad R_{sy} = \frac{1.5}{2.0} = 0.75$$

3. Spacing Anisotropy:

$$\text{Spacing Anisotropy} = \max(0.67, 0.50, 0.75) = 0.75$$

Reference: https://chat.openai.com/

Gamma Correction

- Gamma correction or gamma is a nonlinear operation used to encode and decode luminance or tristimulus values in video or still image systems.

Reference: https://en.wikipedia.org/wiki/Gamma_correction

Polynomial Learning Rate Schedule

```python
import numpy as np
import matplotlib.pyplot as plt


def polynomial_decay_schedule(initial_lr: float, power: float, max_epochs: int):
    """
    Generate a polynomial decay learning rate schedule.

    Args:
        initial_lr: The initial learning rate.
        power: The power of the polynomial.
        max_epochs: The maximum number of epochs.

    Returns:
        An array of learning rates for each epoch.
    """
    epochs = np.arange(max_epochs)
    lr = initial_lr * ((1 - (epochs / max_epochs)) ** power)
    return lr
```

Reference: https://medium.com/@theom/a-very-short-visual-introduction-to-learning-rate-schedulers-with-code-189eddffdb00

Gaussian Patch Center Weighting

- 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).