Topic: Brain Tumor Segmentation in 3D Preoperative MRI Scans

Bryan Tan (Chen Zhengyu) | 1004318
Christy Lau Jin Yun | 1005330
Lee Pei Xuan | 1005513

Introduction

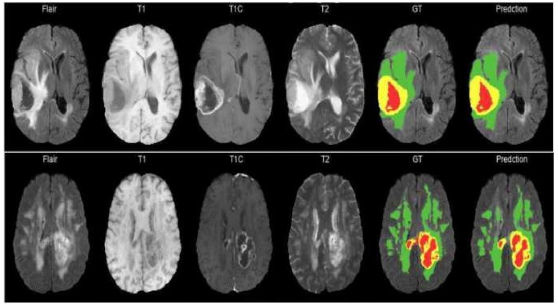
- Brain Tumour Segmentation is an important challenge in biomedical image processing.

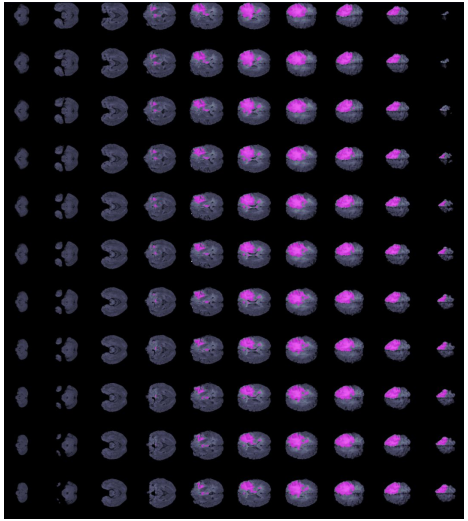
- Early detection of brain tumours is vital for the successful treatment of patients.

- One of the most prevalent types of primary brain tumours is Glioma, which comprises about 30 percent of brain tumours and 80 percent of all malignant brain tumours.

- Hence, there is a need for automated approaches to brain-tumour segmentation as it can **help reduce/augment hospital workloads** and **save lives**

Dataset: BraTS 2020
Problem Type: 3D Multi-class Image Segmentation





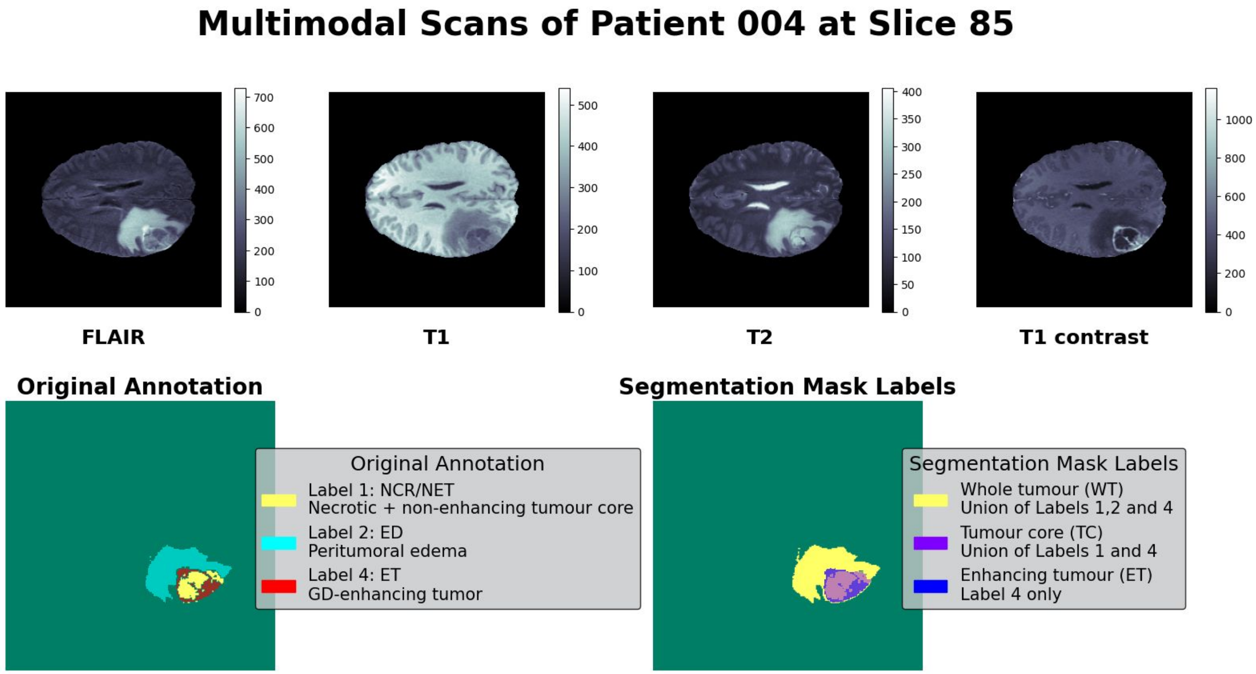
- **Description**: Multi-institutional (n=19) preoperative MRI Brain scans
- **File Extension**: Neuroimaging Informatics Technology Initiative (NIfTI) files (.nii.gz)
- **Image Size**: 240x240x155 Voxels
- **Number of Training Samples**: 369 Images total
- **Training Set**: 263
- **Validation and Testing Set**: 53 for both

Dataset: BraTS 2020
Problem Type: 3D Multi-class Image Segmentation

- **Image Modalities (Channels)**: 4
  - T1-Weighted (T1)
  - Post-contrast T1-Weighted (T1c)
  - T2-Weighted (T2)
  - T2 Fluid Attenuated Inversion Recovery (T2-FLAIR)

- **Original Labels**: 3
  - **NCR/NET — label 1**: Necrotic and Non-enhancing tumor core
  - **ED — label 2**: Peritumoral Edema
  - **ET — label 4**: GD-enhancing tumor

- **Ground Truth Segmentation Classes**: 3
  - **Whole tumor (WT)**: Union of NCR/NET, ED and ET (Label 1,2 and 4)
  - **Tumor core (TC)**: Union of NCR/NET and ET (Labels 1 and 4)
  - **Enhancing tumor (ET):** Label 4 only.



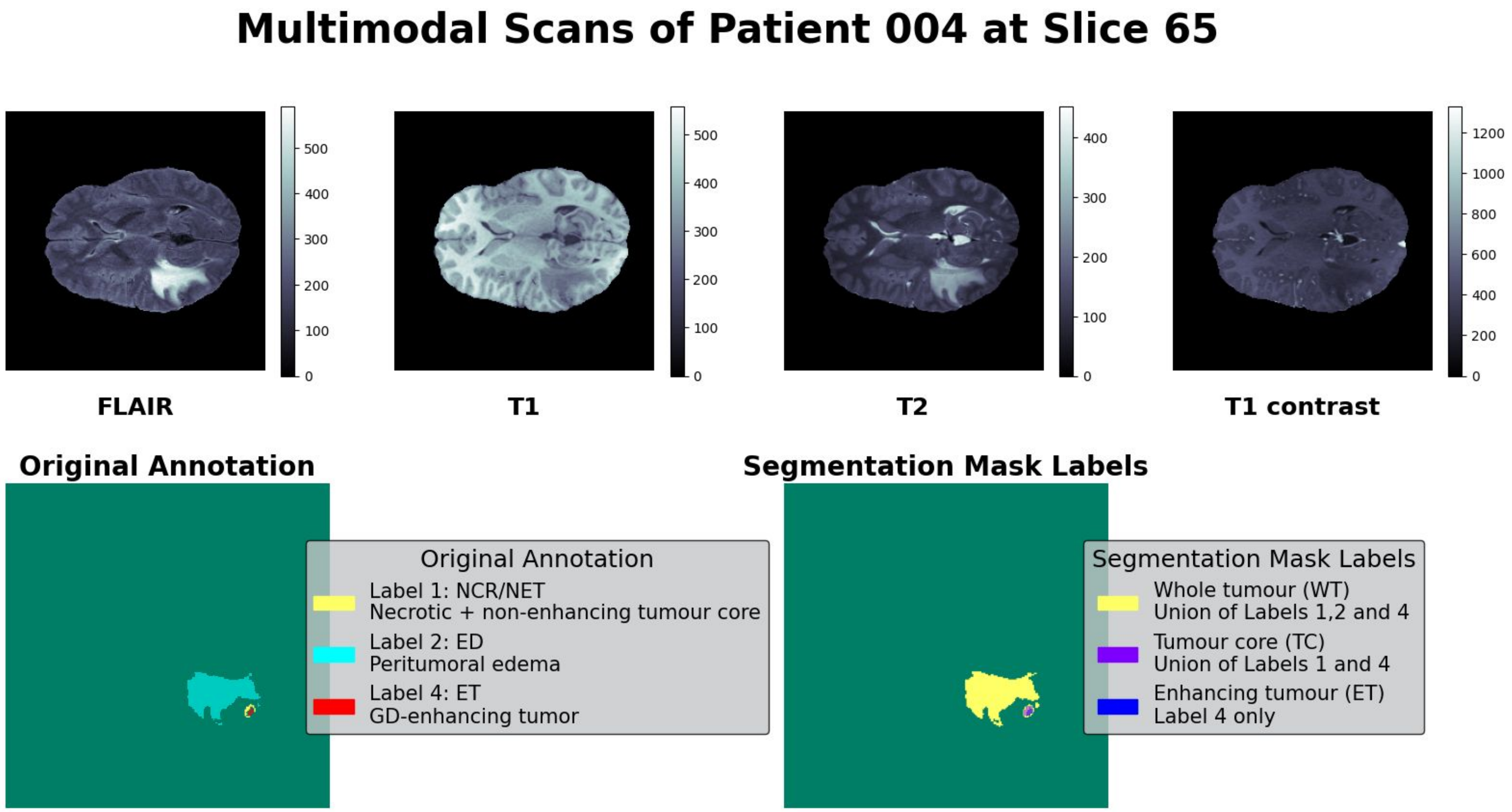
Pre-processing

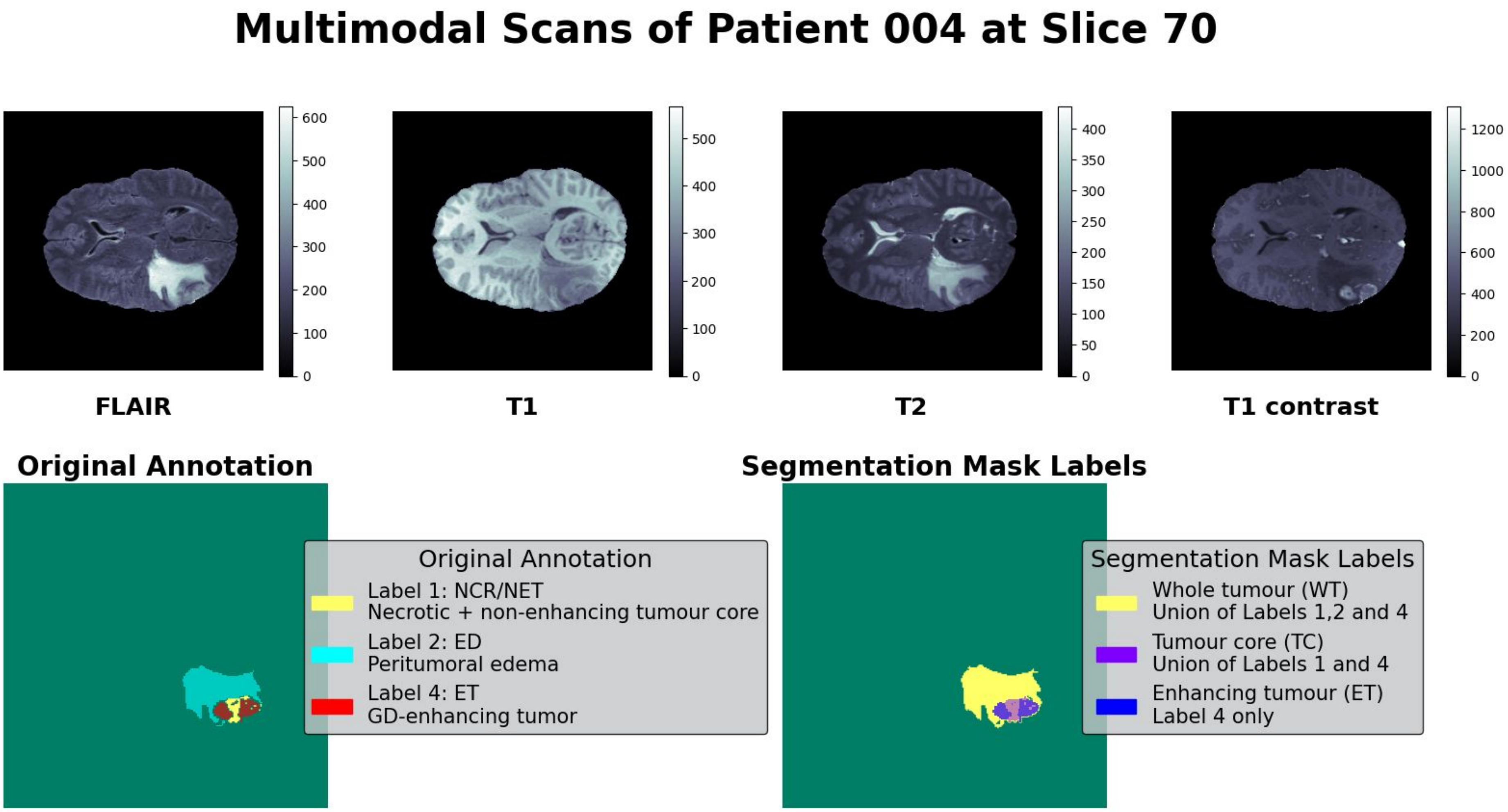
- **Min-Max Normalization:**
  - Image normalization is performed on input images to assist with convergence during training.

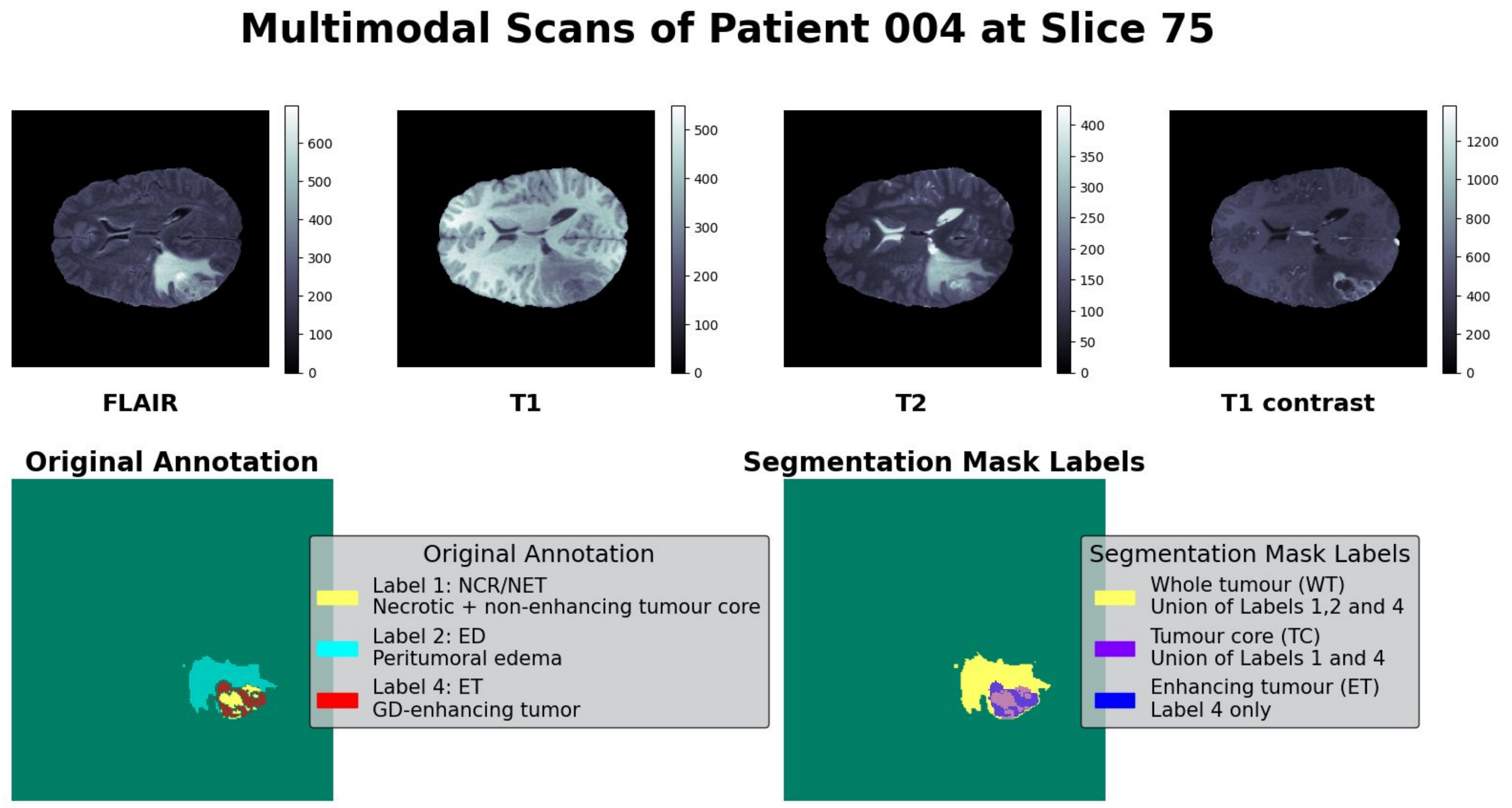
$$Normalized\ value = \frac{value - value_{min}}{value_{max} - value_{min}}$$

- **Cropping from size (240, 240, 155) to size (224, 224, 128).**
  - Reduces Image volume
  - Speeds up training without major performance losses due to margin of empty spaces in original images
  - Allows for input into certain models which only accepts 3D inputs with dimension that are in multiples of 32

- **Masking**:
  - Segmentation performance in the BraTS challenge is evaluated on three partially overlapping sub-regions of tumours: whole tumour (WT), tumour core (TC), and enhancing tumour (ET). To adhere to this, 3 sets of one-hot segmentation masks are created and stacked from unions of the original annotations.

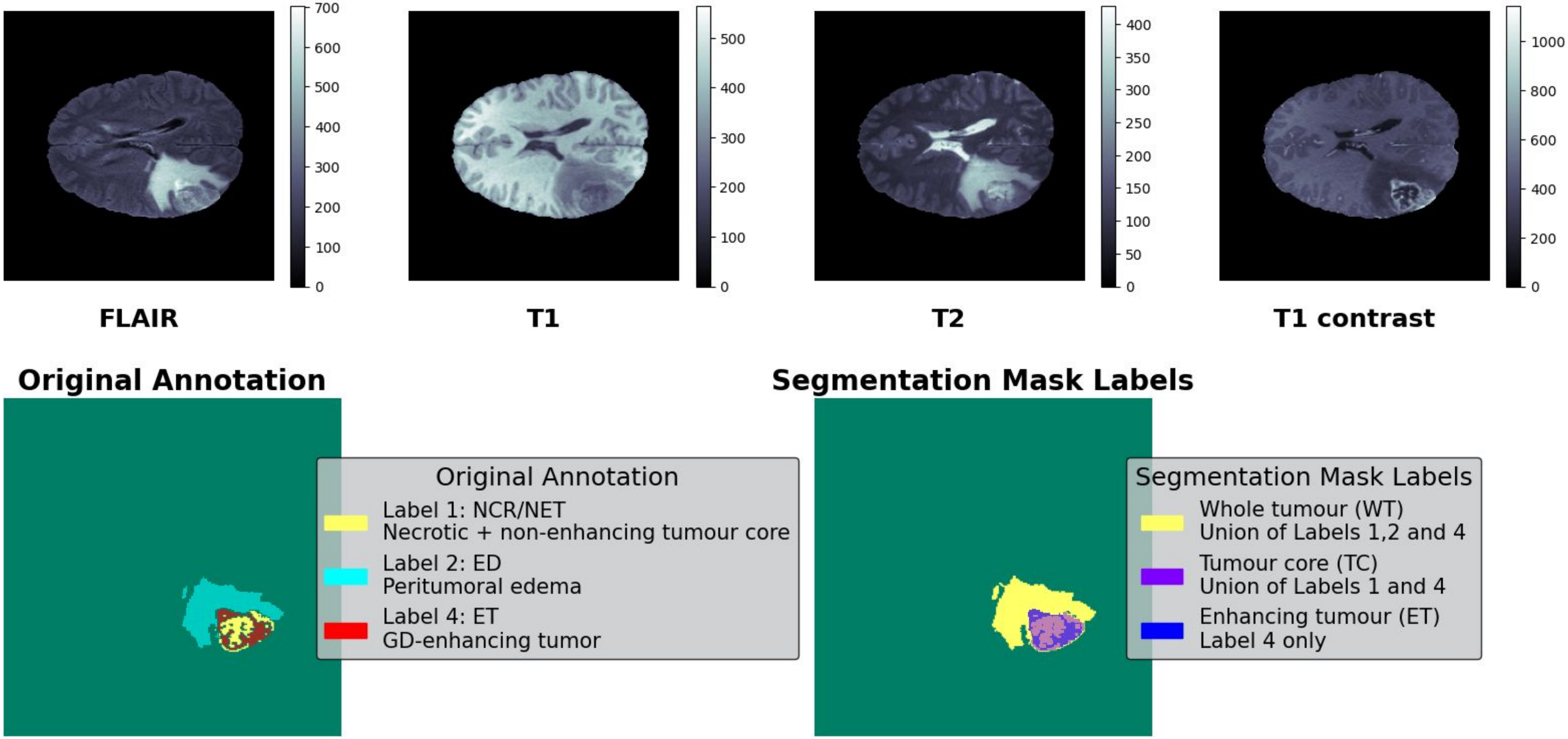

Multimodal Scans of Patient 004 at Slice 65
FLAIR
T1
T2
T1 contrast
Original Annotation
Segmentation Mask Labels
Original Annotation
Label 1: NCR/NET
Necrotic + non-enhancing tumour core
Label 2: ED
Peritumoral edema
Label 4: ET
GD-enhancing tumor
Segmentation Mask Labels
Whole tumour (WT)
Union of Labels 1,2 and 4
Tumour core (TC)
Union of Labels 1 and 4
Enhancing tumour (ET)
Label 4 only


Multimodal Scans of Patient 004 at Slice 70
FLAIR
T1
T2
T1 contrast
Original Annotation
Segmentation Mask Labels
Original Annotation
Label 1: NCR/NET
Necrotic + non-enhancing tumour core
Label 2: ED
Peritumoral edema
Label 4: ET
GD-enhancing tumor
Segmentation Mask Labels
Whole tumour (WT)
Union of Labels 1,2 and 4
Tumour core (TC)
Union of Labels 1 and 4
Enhancing tumour (ET)
Label 4 only
0
100
200
300
400
500
600
0
100
200
300
400
500
0
50
100
150
200
250
300
350
400
0
200
400
600
800
1000
1200

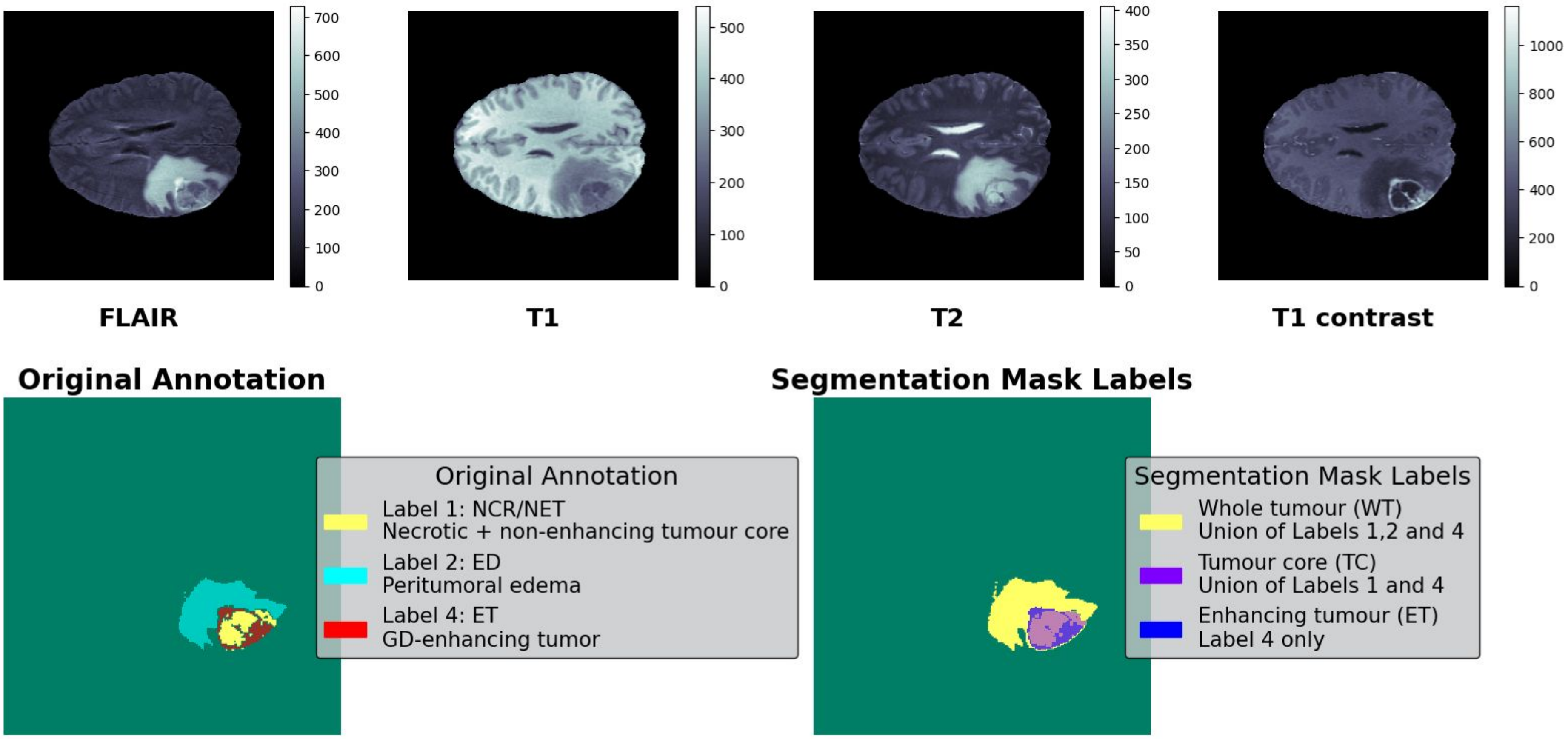


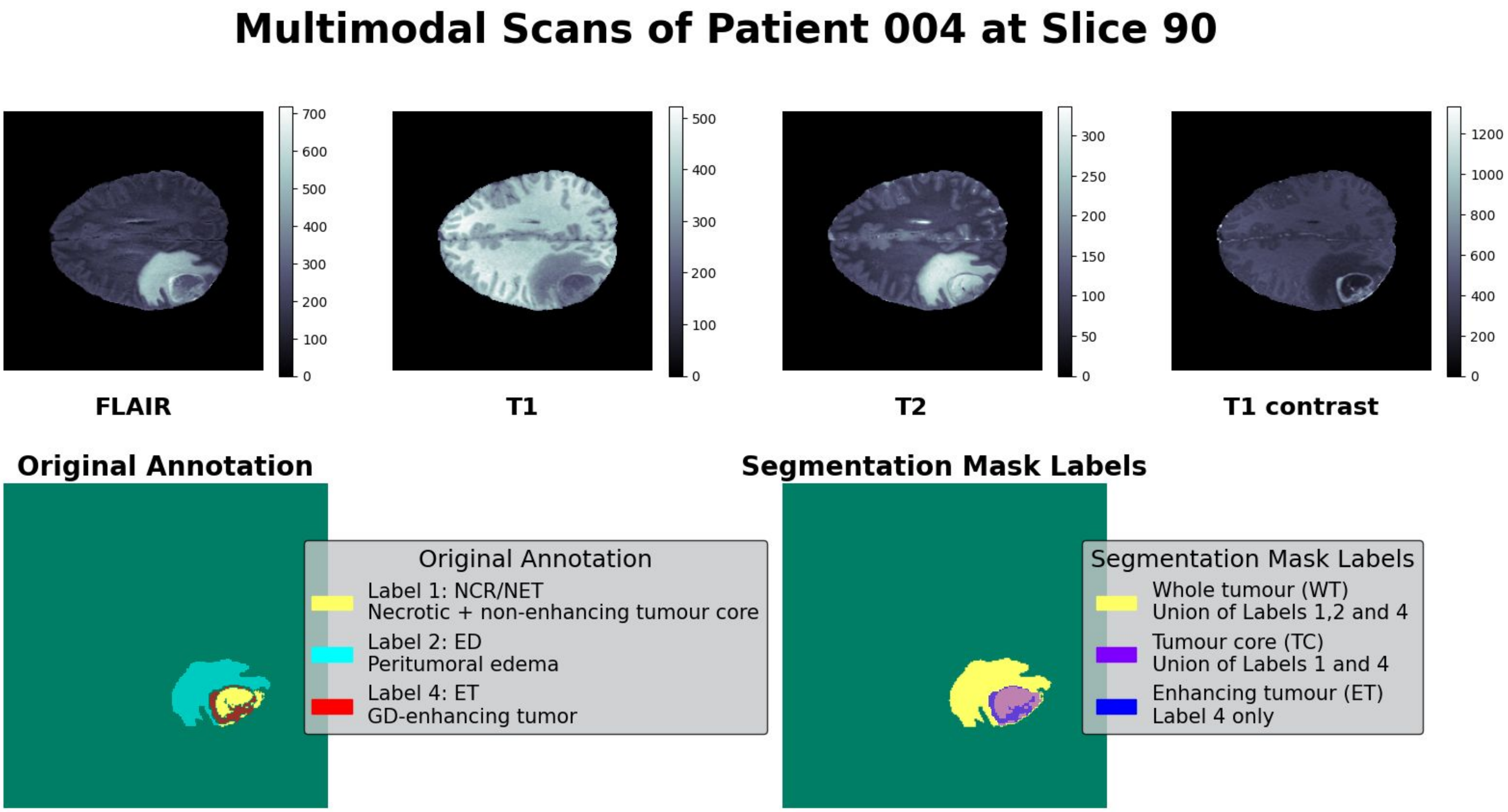
Multimodal Scans of Patient 004 at Slice 75
0
100
200
300
400
500
600
FLAIR
0
100
200
300
400
500
T1
0
50
100
150
200
250
300
350
400
T2
0
200
400
600
800
1000
1200
T1 contrast
Original Annotation
Original Annotation
Label 1: NCR/NET
Necrotic + non-enhancing tumour core
Label 2: ED
Peritumoral edema
Label 4: ET
GD-enhancing tumor
Segmentation Mask Labels
Segmentation Mask Labels
Whole tumour (WT)
Union of Labels 1,2 and 4
Tumour core (TC)
Union of Labels 1 and 4
Enhancing tumour (ET)
Label 4 only

Multimodal Scans of Patient 004 at Slice 80

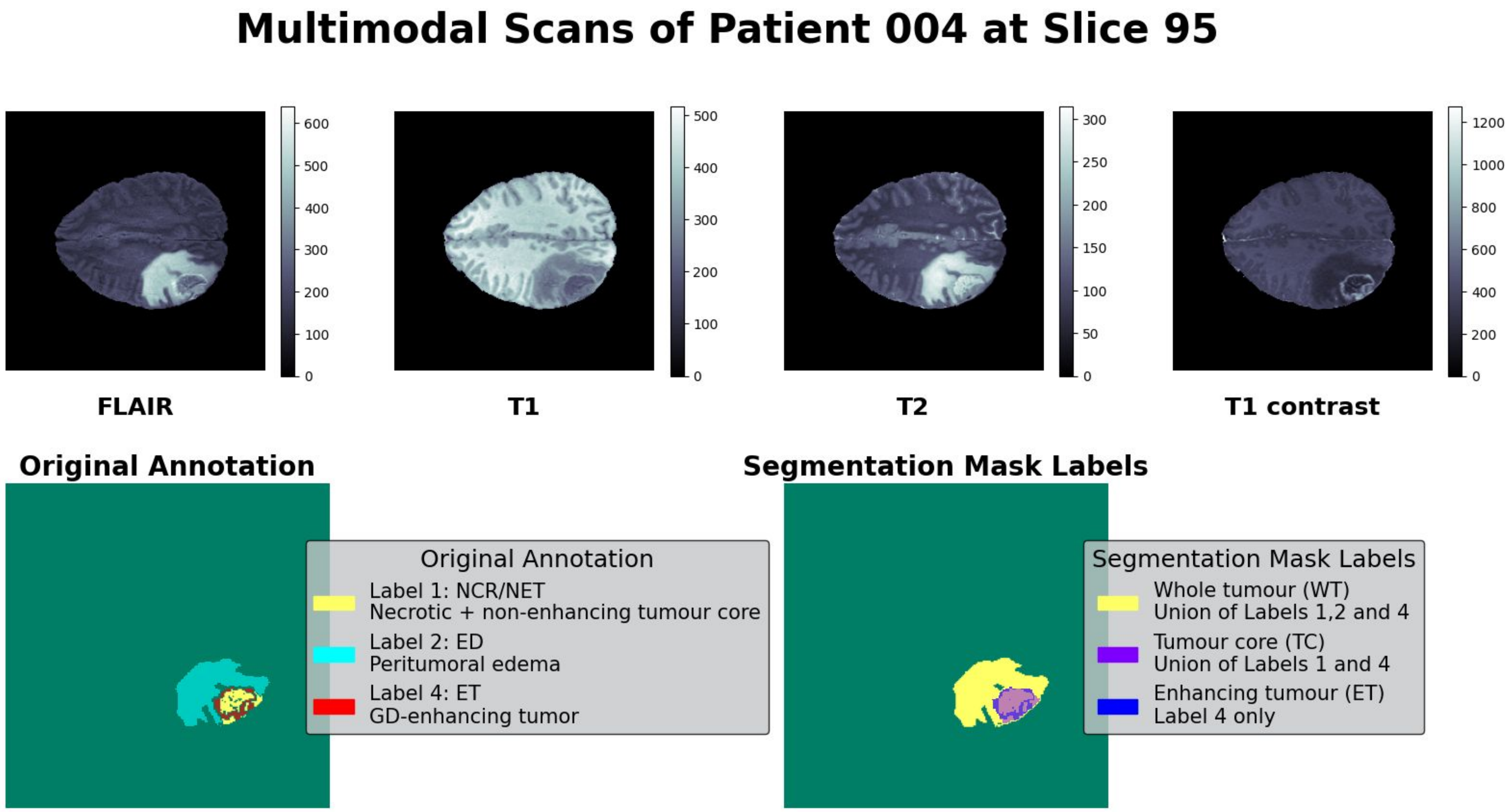


Multimodal Scans of Patient 004 at Slice 85



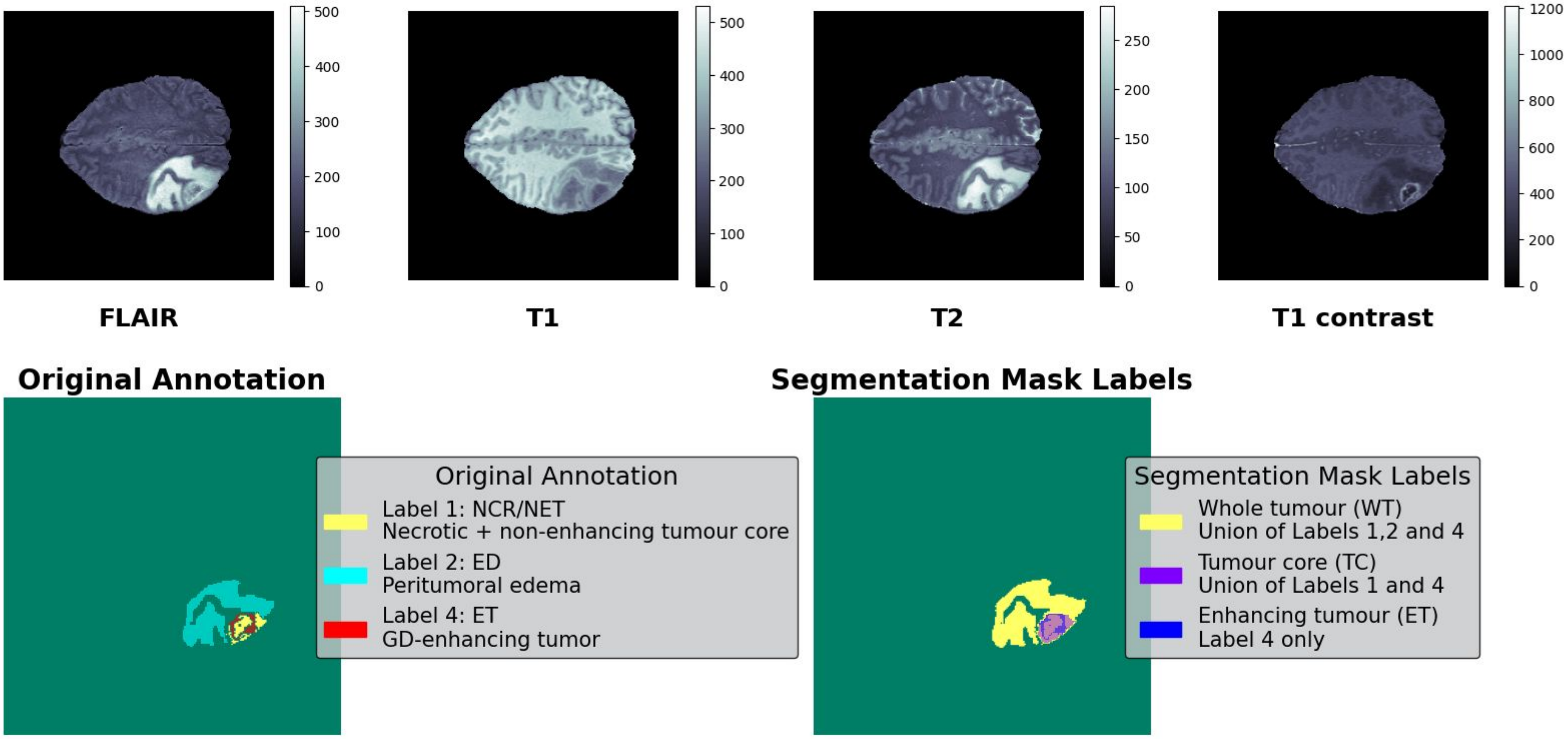


Multimodal Scans of Patient 004 at Slice 90
700
600
500
400
300
200
100
0
FLAIR
500
400
300
200
100
0
T1
300
250
200
150
100
50
0
T2
1200
1000
800
600
400
200
0
T1 contrast
Original Annotation
Original Annotation
Label 1: NCR/NET
Necrotic + non-enhancing tumour core
Label 2: ED
Peritumoral edema
Label 4: ET
GD-enhancing tumor
Segmentation Mask Labels
Segmentation Mask Labels
Whole tumour (WT)
Union of Labels 1,2 and 4
Tumour core (TC)
Union of Labels 1 and 4
Enhancing tumour (ET)
Label 4 only

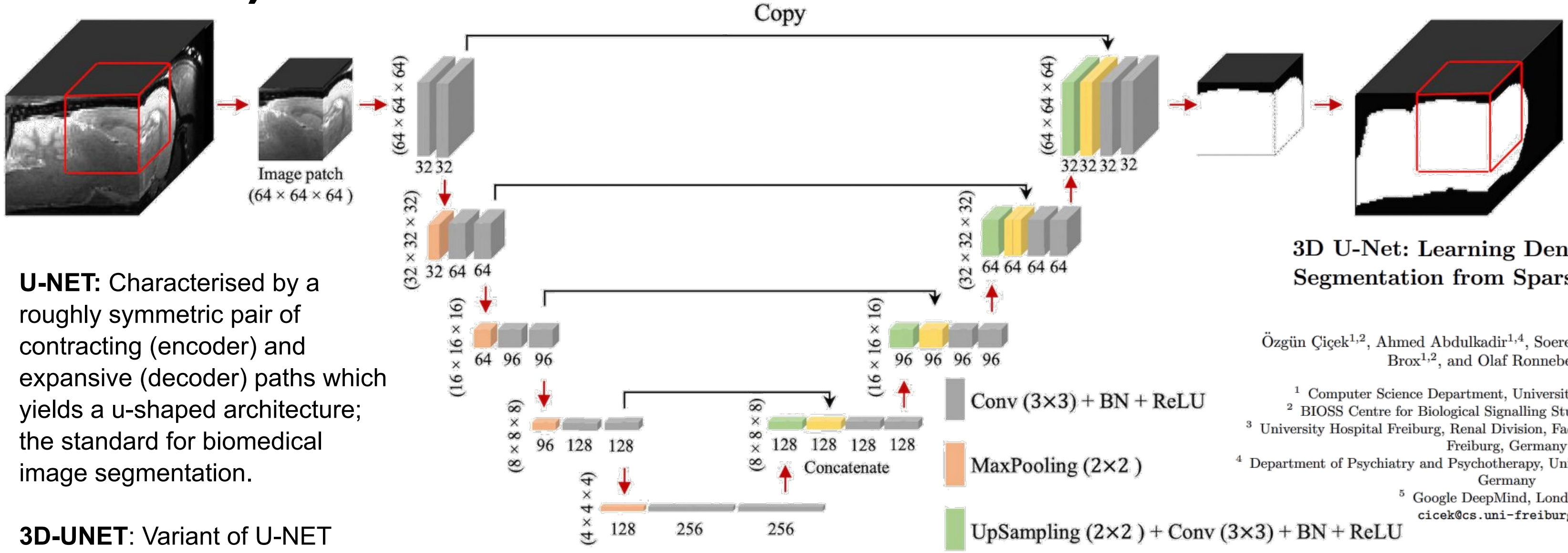


Multimodal Scans of Patient 004 at Slice 95
FLAIR
T1
T2
T1 contrast
Original Annotation
Segmentation Mask Labels
Original Annotation
Label 1: NCR/NET
Necrotic + non-enhancing tumour core
Label 2: ED
Peritumoral edema
Label 4: ET
GD-enhancing tumor
Segmentation Mask Labels
Whole tumour (WT)
Union of Labels 1,2 and 4
Tumour core (TC)
Union of Labels 1 and 4
Enhancing tumour (ET)
Label 4 only

Multimodal Scans of Patient 004 at Slice 100



Existing Works: 3D-UNet (Strong/Fast Baseline; many variants)



U-NET: Characterised by a roughly symmetric pair of contracting (encoder) and expansive (decoder) paths which yields a u-shaped architecture; the standard for biomedical image segmentation.

3D-UNET: Variant of U-NET which uses 3D convolutions instead of 2D ones.

24 Channels = 5,652,507 parameters
32 Channels = 10,045,731 parameters

3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation

Özgün Çiçek[1,2], Ahmed Abdulkadir[1,4], Soeren S. Lienkamp[2,3], Thomas Brox[1,2], and Olaf Ronneberger[1,2,5]

[1] Computer Science Department, University of Freiburg, Germany
[2] BIOSS Centre for Biological Signalling Studies, Freiburg, Germany
[3] University Hospital Freiburg, Renal Division, Faculty of Medicine, University of Freiburg, Germany
[4] Department of Psychiatry and Psychotherapy, University Medical Center Freiburg, Germany
[5] Google DeepMind, London, UK
cicek@cs.uni-freiburg.de

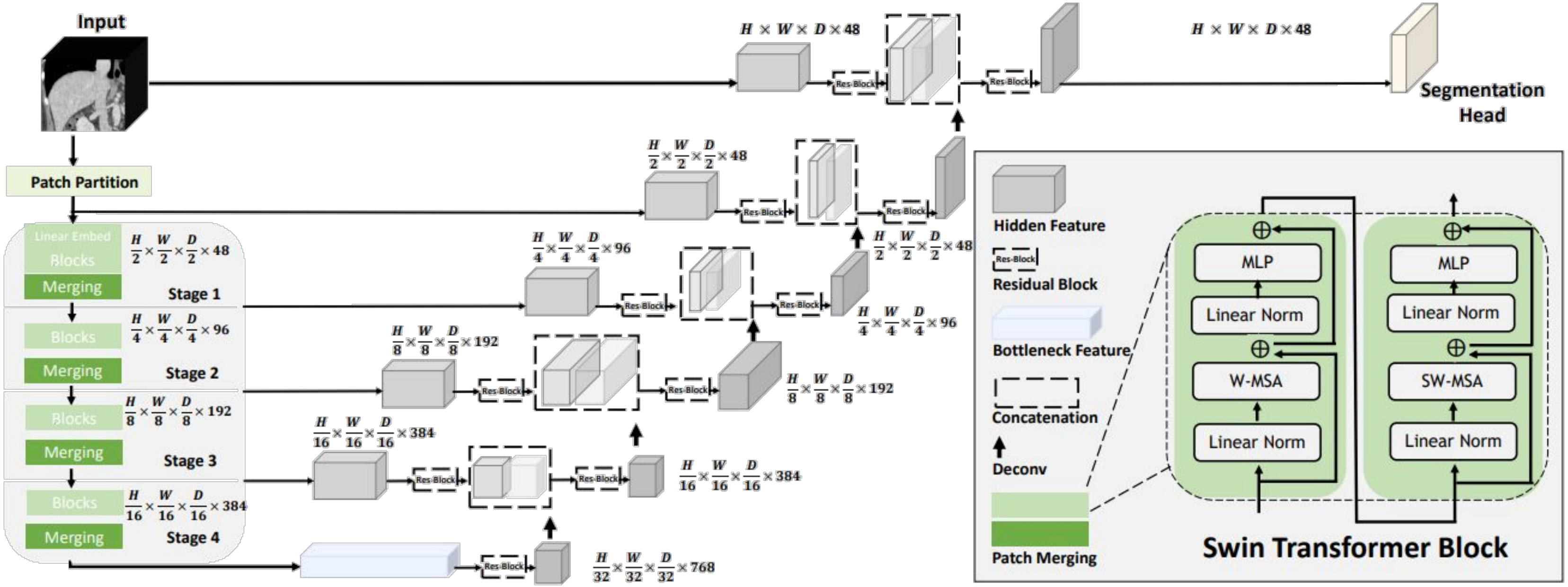
arXiv
https://arxiv.org › cs

3D U-Net: Learning Dense Volumetric Segmentation ... - arXiv

by Ö Çiçek · 2016 · Cited by 5023 — Abstract: This paper introduces a network for volumetric segmentation that learns from sparsely annotated volumetric images.

Cite as: arXiv:1606.06650

Existing Works: Swin UNETR (State-of-the-Art model)



Swin UNETR: Swin Transformers for Semantic Segmentation of Brain Tumors in MRI Images

Ali Hatamizadeh[1], Vishwesh Nath[1], Yucheng Tang[2], Dong Yang[1], Holger R. Roth[1], and Daguang Xu[1]

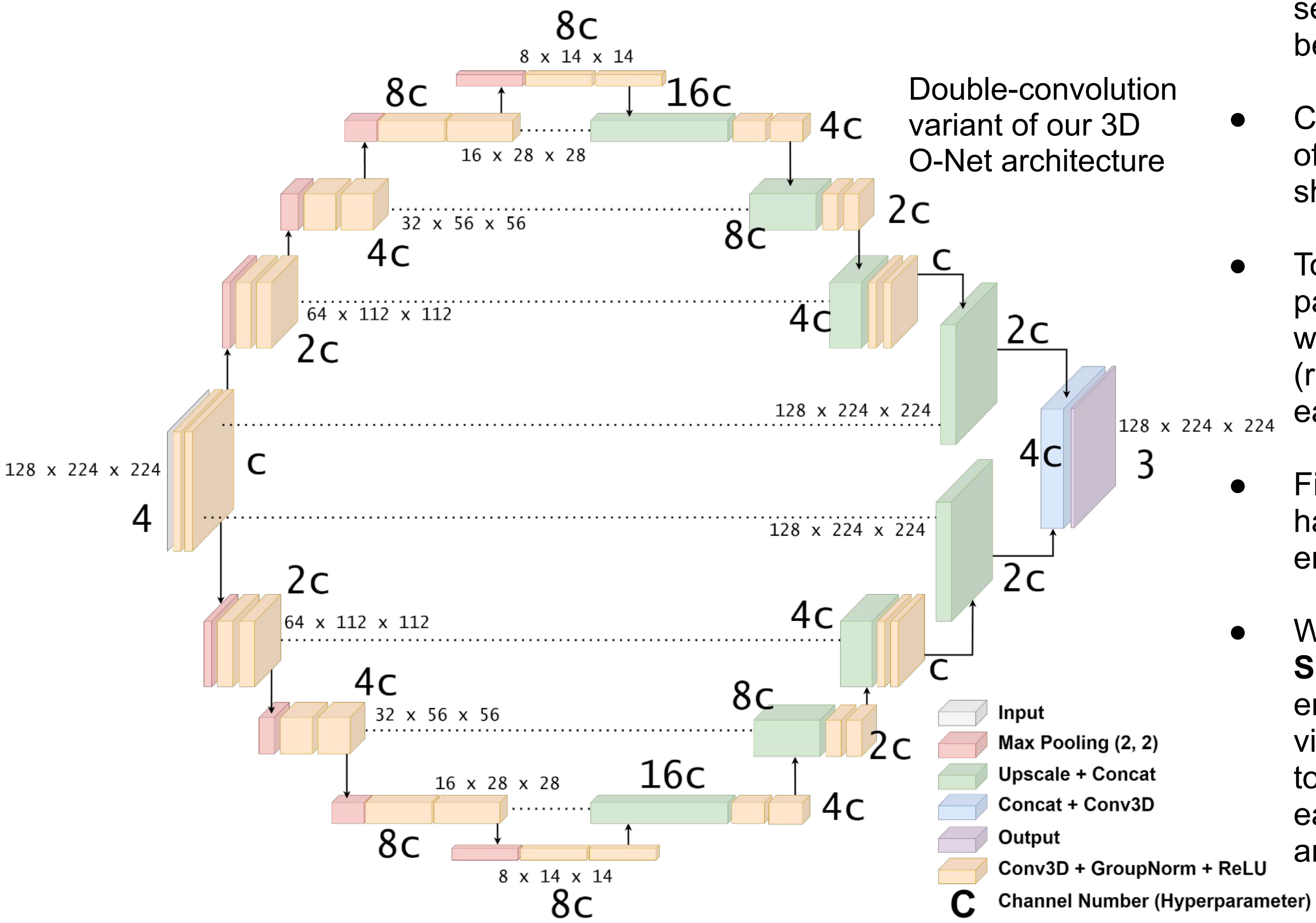
[1] NVIDIA
[2] Vanderbilt University
ahatamizadeh@nvidia.com

14,981,601 parameters!

Shifted-Window Transformer; a special transformer optimised for vision task; greater efficiency and performance

Swin UNETR is available in Project MONAI, which provides open-source libraries for biomedical AI and data analysis.

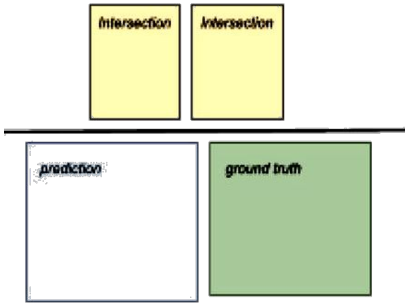
Our Proposed Architecture: 3D-ONet

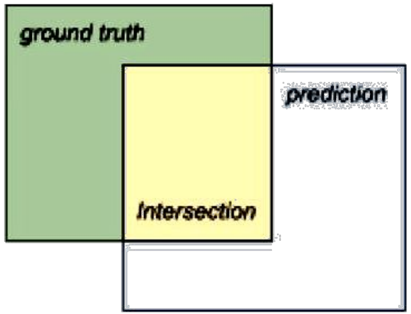


Double-convolution variant of our 3D O-Net architecture

- Two sets of encoder-decoder sections which are concatenated before the output convolution layer.
- Can be visualised as 2 parallel sets of U-Net in approximately an "O" shape.
- To maintain a similar parameter count as that of U-Net, we mainly experiment with single (rather than double) convolutions for each upscale and downscale step.
- Fixed kernel-size at 3 for the bottom half, varied kernel sizes at the top encoder-decoder half.
- We also experiment with “**3D SphereNet**”, with 4 sets of encoder-decoder pairs; can be visualised as being at the bottom, top, left and right in a 3D diagram; each with different kernel sizes 1,3,5 and 7. Large parameter count.

Performance Metrics:



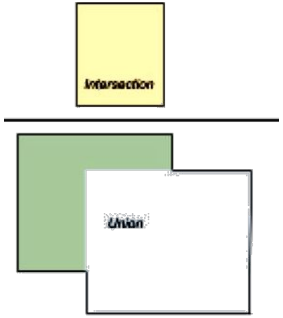


$$Dice\ Score = \frac{2 * |X \cap Y|}{|X| + |Y|}$$

Dice Score: 2x Intersection over Total Number of Pixels in A + B

Jaccard Score: Intersection over Union (IoU)

$$Jaccard\ Score = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}$$



<u>**Loss Function**</u>:
BCE-Dice Score = Binary Cross Entropy Loss + Dice Loss

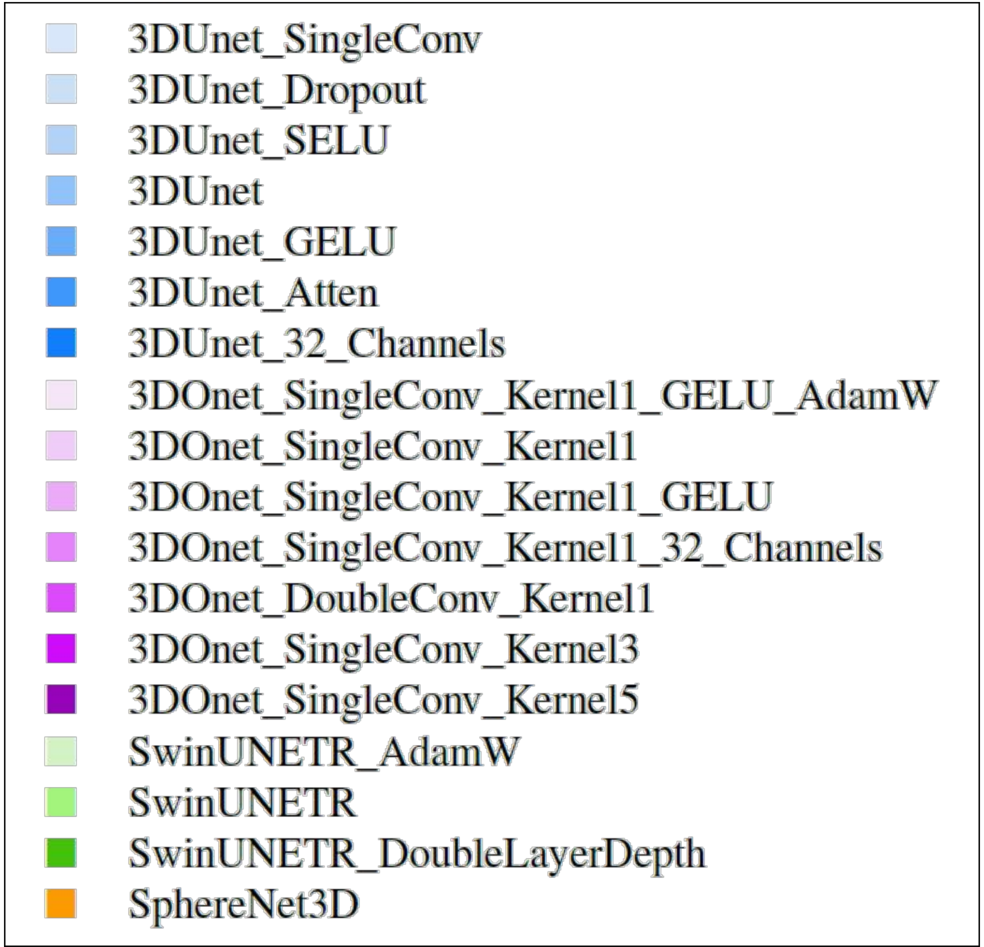
Rationale: Dice loss penalizes **final predictions**; BCE Loss penalizes **confident misclassifications / unconfident correct predictions**; deals with probabilities.

$$Dice\ Loss = 1 - Dice\ Score$$

$$BCE\ Loss = -Y \log(\sigma(Y')) - (1 - Y) \log(1 - \sigma(Y'))$$

$$BCEDice\ Loss = 0.5 * Dice\ Loss + 0.5 * BCE\ Loss$$

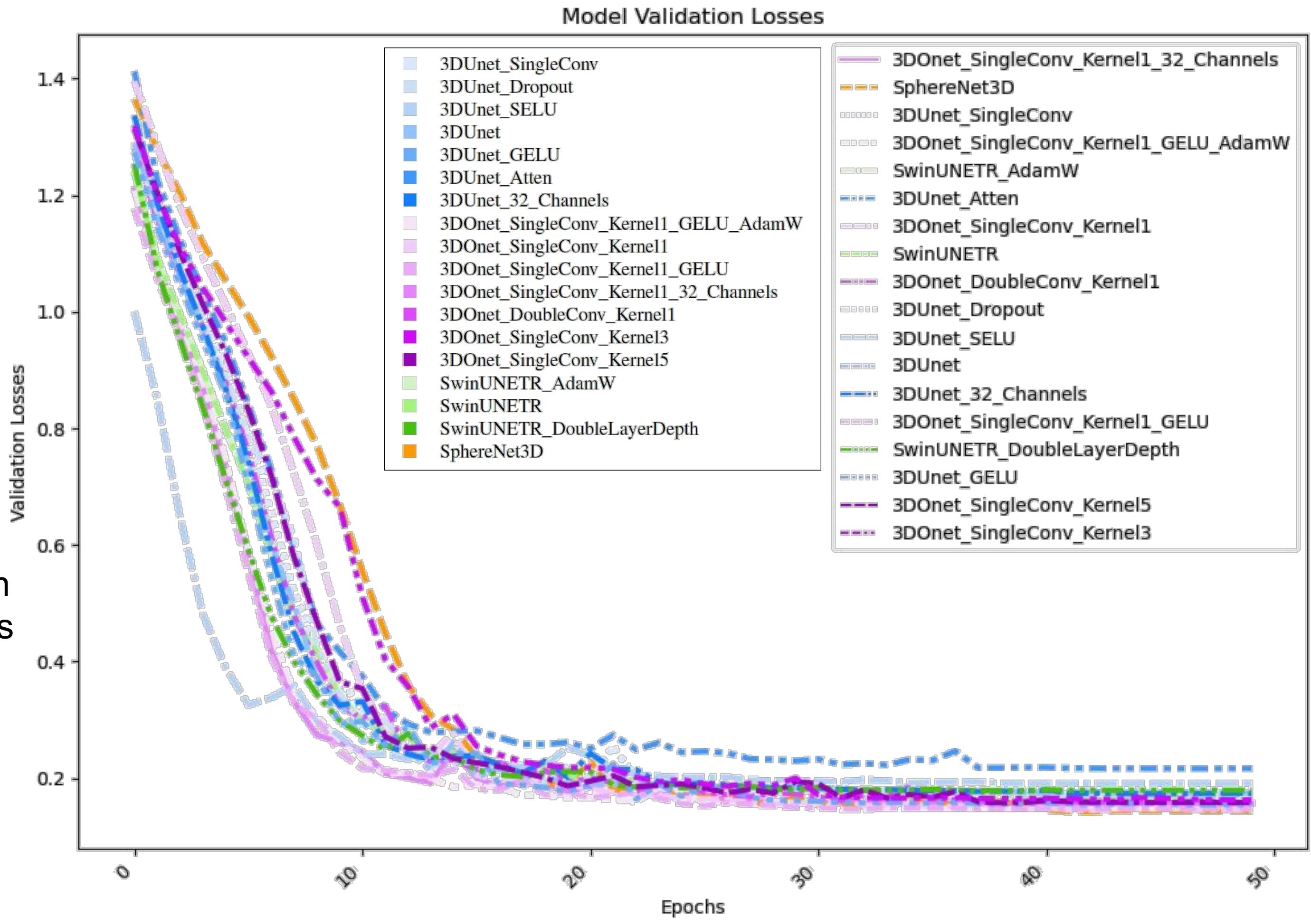
Models Evaluated



- **For ease of visualisation**
- **Darker shades** corresponds generally to a greater parameter count.
- 18 Models Evaluated.
- Blue corresponds 3D-UNet variants; 7 Models.
- Purple corresponds to 3D-ONet variant; 7 Models
- Green corresponds to Swin UNETR variants; 3 Models
- Orange corresponds to 3D-SphereNet; 1 Model

Training

- 50 epochs using the Adam optimiser with default parameters unless otherwise stated.
- Ubuntu-22.04 OS via WSL, with an Intel i5-13600k, 32GB RAM and an RTX 3090 with 24GB VRAM.
- Each model evaluated on the same 53 test samples by loading the best checkpoint.
- For reproducibility during training, we set seed=42 wherever possible.



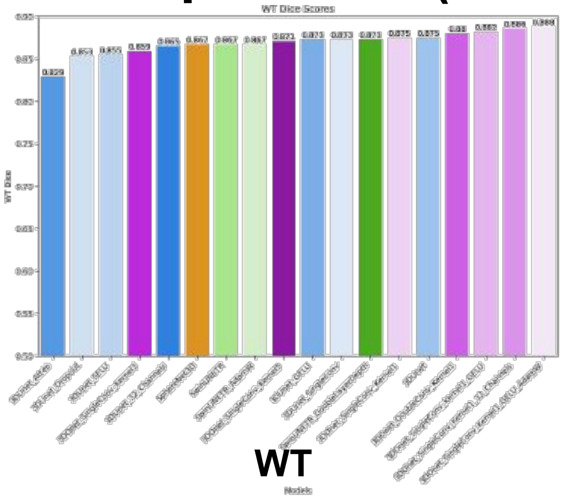
Results Comparisons (After 50 epochs on BraTS 2020)

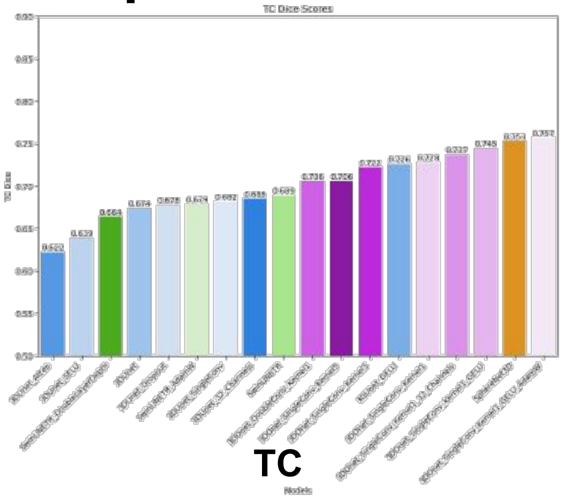
Bold represents best result within model family; <u>**underline**</u> represents best result across all models.

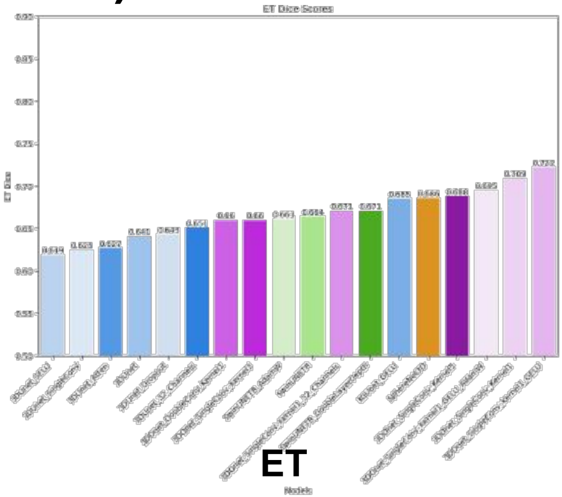
Models	WT dice	WT jaccard	TC dice	TC jaccard	ET dice	ET jaccard	Infer time (s)	Train time (h)	Params (1e6)
3DOnet_DoubleConv_Kernel1	0.8801	0.8022	0.7058	0.6017	0.6600	0.5678	27.41	**5.850**	5.90
3DOnet_SingleConv_Kernel1	0.8753	0.7956	0.7282	0.6363	0.7093	0.6172	22.71	10.533	**3.15**
3DOnet_SingleConv_Kernel1_32_Channels	0.8860	0.8085	0.7365	0.6300	0.6706	0.5761	29.72	12.167	5.60
3DOnet_SingleConv_Kernel1_GELU	0.8817	0.8050	0.7449	0.6435	<u>**0.7218**</u>	<u>**0.6266**</u>	23.18	6.650	**3.15**
3DOnet_SingleConv_Kernel1_GELU_AdamW	<u>**0.8879**</u>	<u>**0.8097**</u>	<u>**0.7572**</u>	**0.6565**	0.6948	0.6006	**22.53**	7.750	**3.15**
3DOnet_SingleConv_Kernel3	0.8593	0.7797	0.7217	0.6264	0.6597	0.5667	24.98	14.167	6.03
3DOnet_SingleConv_Kernel5	0.8705	0.7897	0.7064	0.5962	0.6877	0.5905	30.87	10.017	16.86
3DUnet	**0.8750**	0.7900	0.6736	0.5661	0.6414	0.5425	18.06	4.633	5.65
3DUnet_32_Channels	0.8651	0.7810	0.6849	0.5722	0.6508	0.5498	22.69	5.317	10.05
3DUnet_Atten	0.8290	0.7334	0.6216	0.5163	0.6267	0.5231	19.04	3.617	6.09
3DUnet_Dropout	0.8527	0.7719	0.6775	0.5690	0.6427	0.5393	17.58	3.883	5.65
3DUnet_GELU	0.8730	**0.7944**	**0.7262**	**0.6323**	**0.6849**	**0.5888**	18.50	5.833	5.65
3DUnet_SELU	0.8554	0.7690	0.6390	0.5303	0.6191	0.5144	17.88	<u>**3.567**</u>	5.65
3DUnet_SingleConv	0.8731	0.7901	0.6820	0.5785	0.6246	0.5210	<u>**15.20**</u>	5.567	<u>**3.01**</u>
SphereNet3D	**0.8667**	**0.7929**	**0.7526**	<u>**0.6582**</u>	**0.6860**	**0.5930**	**36.35**	**10.450**	**6.13**
SwinUNETR	0.8666	0.7902	**0.6888**	**0.5758**	0.6640	0.5749	20.39	**4.100**	**14.98**
SwinUNETR_AdamW	0.8674	0.7861	0.6786	0.5657	0.6611	0.5722	**20.24**	6.800	**14.98**
SwinUNETR_DoubleLayerDepth	**0.8733**	**0.7964**	0.6643	0.5569	**0.6706**	**0.5759**	24.15	15.233	15.64

Results Comparisons (After 50 epochs on BraTS 2020)

Dice



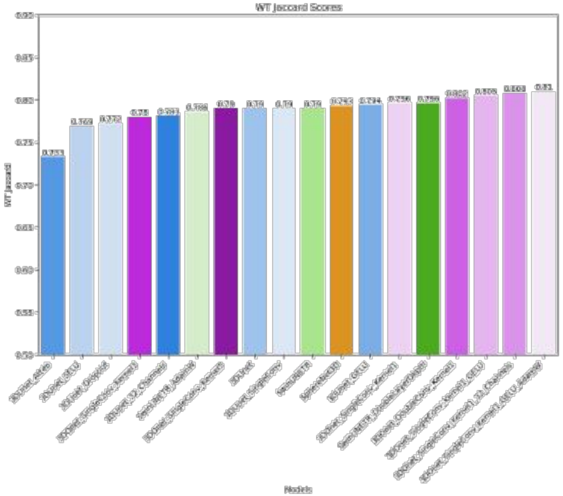


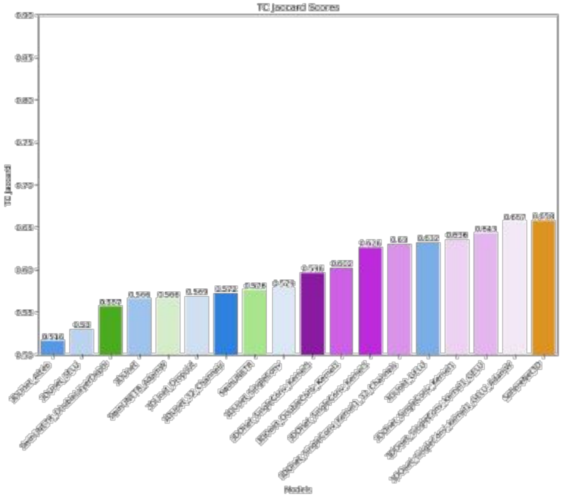


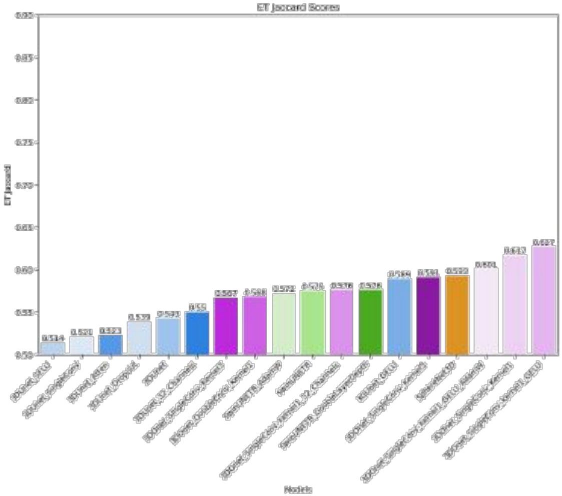
Y-axis ranges from 0.5-0.9

WT **TC** **ET**

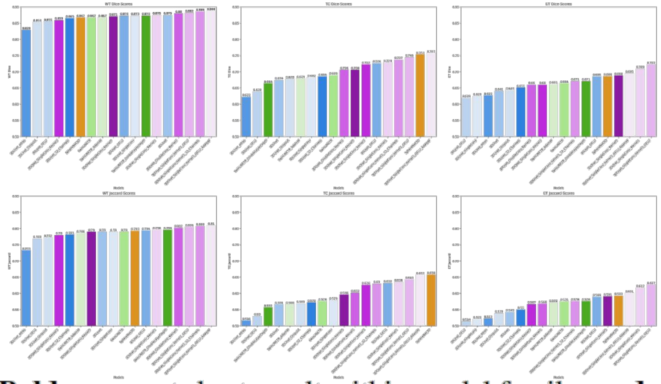
Jaccard







Discussion: Observation of Results



Observation 1: The smallest 3D O-Net model outperforms other models

Variants of our own 3D O-Net models with kernel size = 1 for the upper encoder-decoder section tend to outperform; these variants achieved the **highest score for 5 out of 6 of the tracked metrics** while having **among the lowest parameter.**

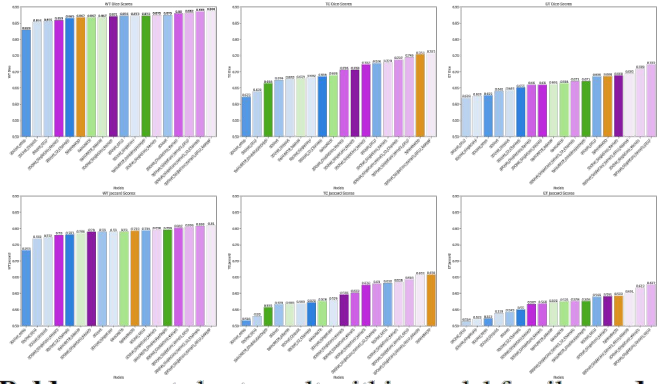
Bold represents best result within model family; **underline** represents best result across all models.

Models	WT dice	WT jaccard	TC dice	TC jaccard	ET dice	ET jaccard	Infer time (s)	Train time (h)	Params (1e6)
3DOnet_DoubleConv_Kernel1	0.8801	0.8022	0.7058	0.6017	0.6600	0.5678	27.41	**5.850**	5.90
3DOnet_SingleConv_Kernel1	0.8753	0.7956	0.7282	0.6363	0.7093	0.6172	22.71	10.533	**3.15**
3DOnet_SingleConv_Kernel1_32_Channels	0.8860	0.8085	0.7365	0.6300	0.6706	0.5761	29.72	12.167	5.60
3DOnet_SingleConv_Kernel1_GELU	0.8817	0.8050	0.7449	0.6435	**0.7218**	**0.6266**	23.18	6.650	**3.15**
3DOnet_SingleConv_Kernel1_GELU_AdamW	**0.8879**	**0.8097**	**0.7572**	**0.6565**	0.6948	0.6006	**22.53**	7.750	**3.15**
3DOnet_SingleConv_Kernel3	0.8593	0.7797	0.7217	0.6264	0.6597	0.5667	24.98	14.167	6.03
3DOnet_SingleConv_Kernel5	0.8705	0.7897	0.7064	0.5962	0.6877	0.5905	30.87	10.017	16.86
3DUnet	**0.8750**	0.7900	0.6736	0.5661	0.6414	0.5425	18.06	4.633	5.65
3DUnet_32_Channels	0.8651	0.7810	0.6849	0.5722	0.6508	0.5498	22.69	5.317	10.05
3DUnet_Atten	0.8290	0.7334	0.6216	0.5163	0.6267	0.5231	19.04	3.617	6.09
3DUnet_Dropout	0.8527	0.7719	0.6775	0.5690	0.6427	0.5393	17.58	3.883	5.65
3DUnet_GELU	0.8730	**0.7944**	**0.7262**	**0.6323**	**0.6849**	**0.5888**	18.50	5.833	5.65
3DUnet_SELU	0.8554	0.7690	0.6390	0.5303	0.6191	0.5144	17.88	**3.567**	5.65
3DUnet_SingleConv	0.8731	0.7901	0.6820	0.5785	0.6246	0.5210	**15.20**	5.567	**3.01**
SphereNet3D	**0.8667**	**0.7929**	**0.7526**	**0.6582**	**0.6860**	**0.5930**	**36.35**	**10.450**	**6.13**
SwinUNETR	0.8666	0.7902	**0.6888**	**0.5758**	0.6640	0.5749	20.39	**4.100**	**14.98**
SwinUNETR_AdamW	0.8674	0.7861	0.6786	0.5657	0.6611	0.5722	**20.24**	6.800	**14.98**
SwinUNETR_DoubleLayerDepth	**0.8733**	**0.7964**	0.6643	0.5569	**0.6706**	**0.5759**	24.15	15.233	15.64

Superior even to Nvidia Swin UNETR.

Possible reason is the difference in training environment; ViT models might need more data to work well; NVIDIA trained their model on DGX-1 cluster for 800 epochs.

Discussion: Observation of Results



Observation 2: Replacing ReLU with GELU improves performance

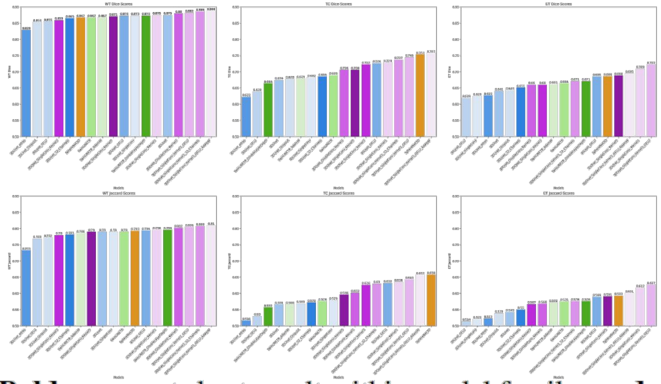
3D U-Net and 3D O-Net show **noticeable improvements in performance** when the ReLU activation function in their default variants is replaced by GELU. There is a significant increase in performance for 4 out of the 6 metrics between 3DUnet and its GELU variant 3DUnet_GELU.

Bold represents best result within model family; **underline** represents best result across all models.

Models	WT dice	WT jaccard	TC dice	TC jaccard	ET dice	ET jaccard	Infer time (s)	Train time (h)	Params (1e6)
3DOnet_DoubleConv_Kernel1	0.8801	0.8022	0.7058	0.6017	0.6600	0.5678	27.41	**5.850**	5.90
3DOnet_SingleConv_Kernel1	0.8753	0.7956	0.7282	0.6363	0.7093	0.6172	22.71	10.533	**3.15**
3DOnet_SingleConv_Kernel1_32_Channels	0.8860	0.8085	0.7365	0.6300	0.6706	0.5761	29.72	12.167	5.60
3DOnet_SingleConv_Kernel1_GELU	0.8817	0.8050	0.7449	0.6435	**0.7218**	**0.6266**	23.18	6.650	**3.15**
3DOnet_SingleConv_Kernel1_GELU_AdamW	**0.8879**	**0.8097**	**0.7572**	**0.6565**	0.6948	0.6006	**22.53**	7.750	**3.15**
3DOnet_SingleConv_Kernel3	0.8593	0.7797	0.7217	0.6264	0.6597	0.5667	24.98	14.167	6.03
3DOnet_SingleConv_Kernel5	0.8705	0.7897	0.7064	0.5962	0.6877	0.5905	30.87	10.017	16.86
3DUnet	**0.8750**	0.7900	0.6736	0.5661	0.6414	0.5425	18.06	4.633	5.65
3DUnet_32_Channels	0.8651	0.7810	0.6849	0.5722	0.6508	0.5498	22.69	5.317	10.05
3DUnet_Atten	0.8290	0.7334	0.6216	0.5163	0.6267	0.5231	19.04	3.617	6.09
3DUnet_Dropout	0.8527	0.7719	0.6775	0.5690	0.6427	0.5393	17.58	3.883	5.65
3DUnet_GELU	0.8730	**0.7944**	**0.7262**	**0.6323**	**0.6849**	**0.5888**	18.50	5.833	5.65
3DUnet_SELU	0.8554	0.7690	0.6390	0.5303	0.6191	0.5144	17.88	**3.567**	5.65
3DUnet_SingleConv	0.8731	0.7901	0.6820	0.5785	0.6246	0.5210	**15.20**	5.567	**3.01**
SphereNet3D	**0.8667**	**0.7929**	**0.7526**	**0.6582**	**0.6860**	**0.5930**	**36.35**	**10.450**	**6.13**
SwinUNETR	0.8666	0.7902	**0.6888**	**0.5758**	0.6640	0.5749	20.39	**4.100**	**14.98**
SwinUNETR_AdamW	0.8674	0.7861	0.6786	0.5657	0.6611	0.5722	**20.24**	6.800	**14.98**
SwinUNETR_DoubleLayerDepth	**0.8733**	**0.7964**	0.6643	0.5569	**0.6706**	**0.5759**	24.15	15.233	15.64

Simple change of using GELU for 3DUnet made it outperform Swin UNETR and its variants for nearly every performance metric. GELU activation function may be much more suitable for this particular segmentation task.

Discussion: Observation of Results



Observation 3: Simpler models tend to outperform

Increasing or decreasing the number of convolution layers per upscale/downscale steps appears to (un-intuitively) have the opposite effect on performance;

Might be due to overfitting on small dataset size or a result of other complex interplay between model architecture and hyperparameters.

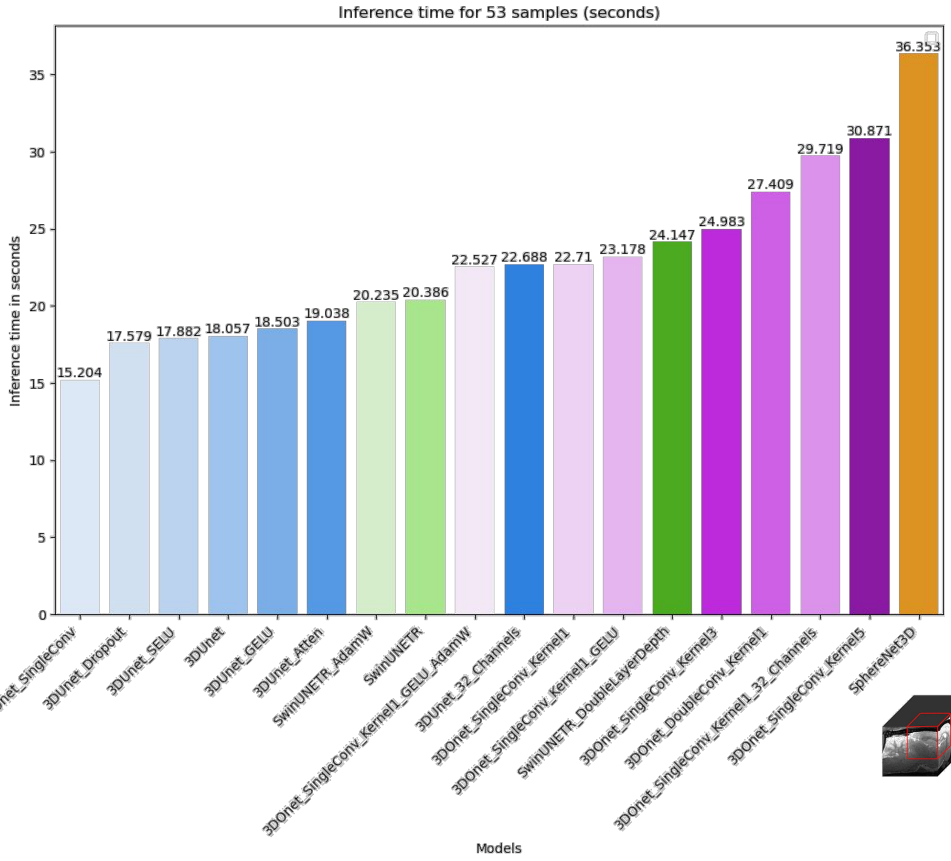
Bold represents best result within model family; **underline** represents best result across all models.

Models	WT dice	WT jaccard	TC dice	TC jaccard	ET dice	ET jaccard	Infer time (s)	Train time (h)	Params (1e6)
3DOnet_DoubleConv_Kernel1	0.8801	0.8022	0.7058	0.6017	0.6600	0.5678	27.41	**5.850**	5.90
3DOnet_SingleConv_Kernel1	0.8753	0.7956	0.7282	0.6363	0.7093	0.6172	22.71	10.533	**3.15**
3DOnet_SingleConv_Kernel1_32_Channels	0.8860	0.8085	0.7365	0.6300	0.6706	0.5761	29.72	12.167	5.60
3DOnet_SingleConv_Kernel1_GELU	0.8817	0.8050	0.7449	0.6435	**0.7218**	**0.6266**	23.18	6.650	**3.15**
3DOnet_SingleConv_Kernel1_GELU_AdamW	**0.8879**	**0.8097**	**0.7572**	**0.6565**	0.6948	0.6006	**22.53**	7.750	**3.15**
3DOnet_SingleConv_Kernel3	0.8593	0.7797	0.7217	0.6264	0.6597	0.5667	24.98	14.167	6.03
3DOnet_SingleConv_Kernel5	0.8705	0.7897	0.7064	0.5962	0.6877	0.5905	30.87	10.017	16.86
3DUnet	**0.8750**	0.7900	0.6736	0.5661	0.6414	0.5425	18.06	4.633	5.65
3DUnet_32_Channels	0.8651	0.7810	0.6849	0.5722	0.6508	0.5498	22.69	5.317	10.05
3DUnet_Atten	0.8290	0.7334	0.6216	0.5163	0.6267	0.5231	19.04	3.617	6.09
3DUnet_Dropout	0.8527	0.7719	0.6775	0.5690	0.6427	0.5393	17.58	3.883	5.65
3DUnet_GELU	0.8730	**0.7944**	**0.7262**	**0.6323**	**0.6849**	**0.5888**	18.50	5.833	5.65
3DUnet_SELU	0.8554	0.7690	0.6390	0.5303	0.6191	0.5144	17.88	**3.567**	5.65
3DUnet_SingleConv	0.8731	0.7901	0.6820	0.5785	0.6246	0.5210	**15.20**	5.567	**3.01**
SphereNet3D	**0.8667**	**0.7929**	**0.7526**	**0.6582**	**0.6860**	**0.5930**	**36.35**	**10.450**	**6.13**
SwinUNETR	0.8666	0.7902	**0.6888**	**0.5758**	0.6640	0.5749	20.39	**4.100**	**14.98**
SwinUNETR_AdamW	0.8674	0.7861	0.6786	0.5657	0.6611	0.5722	**20.24**	6.800	**14.98**
SwinUNETR_DoubleLayerDepth	**0.8733**	**0.7964**	0.6643	0.5569	**0.6706**	**0.5759**	24.15	15.233	15.64

Performance for 3DOne decreased for 4 out of the 6 metrics compared to its simpler single convolution counterpar.

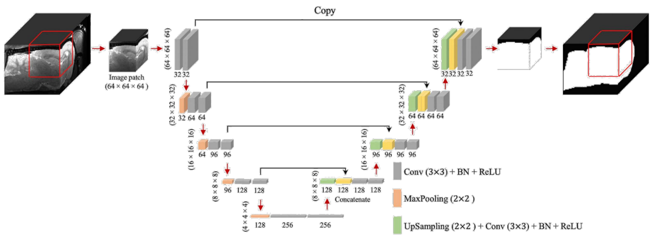
Performance for 3DUnet increased for 3 out of the 6 metrics compared to its more complex double convolution counterpart.

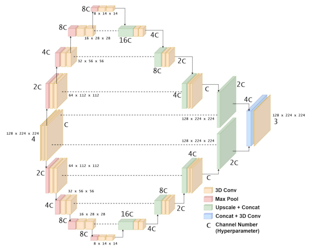
Discussion: Observation of Results



Observation 4: Longer inference time is required for 3D O-Net and 3D Sphere-Net

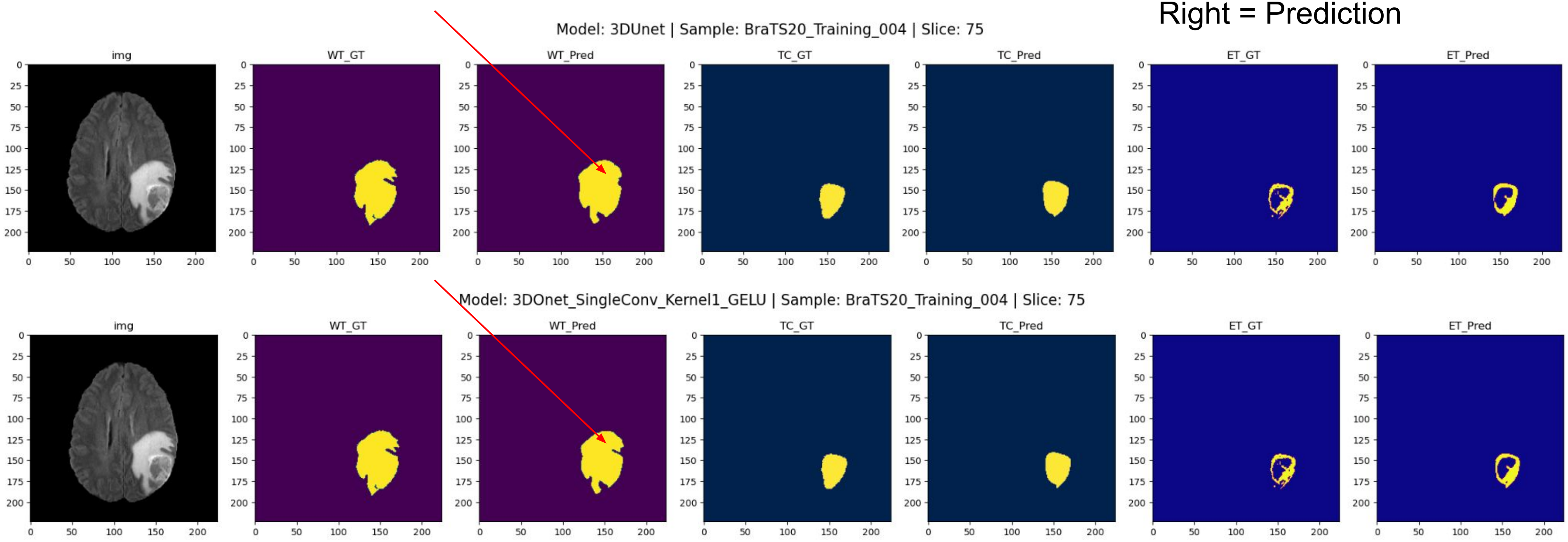
- 3D O-Net and 3D Sphere-Net models have greater inference time compare to both 3D U-Net and Swin UNETR.
- This is despite the fact that many 3D O-Net models have a much lower parameter count compared to 3D U-Net and Swin UNETR.
- One explanation is that backpropagation has to be performed across more layers due to its sequential nature, despite the presence of seemingly parallel paths.





3DUnet vs 3DONet Prediction Comparisons

Left = Ground Truth
Right = Prediction



Our best 3D-ONet model was able to discern the crevice in the tumour for the WT (Whole Tumor) class, unlike the 3D-UNet baseline.

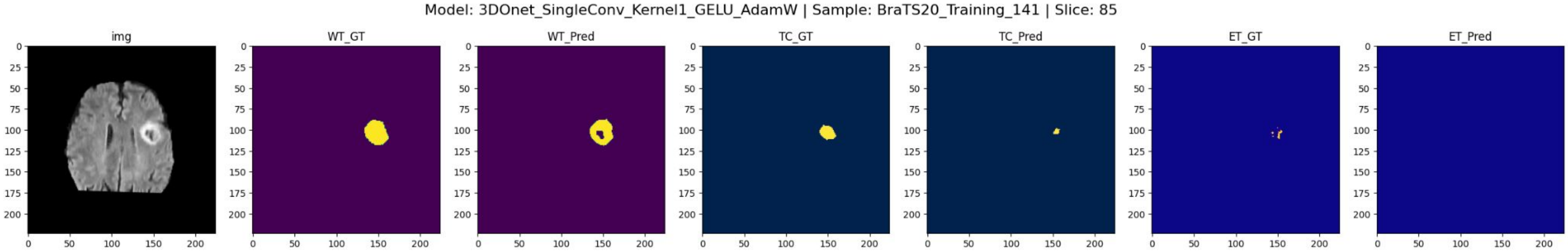
Challenges Faced

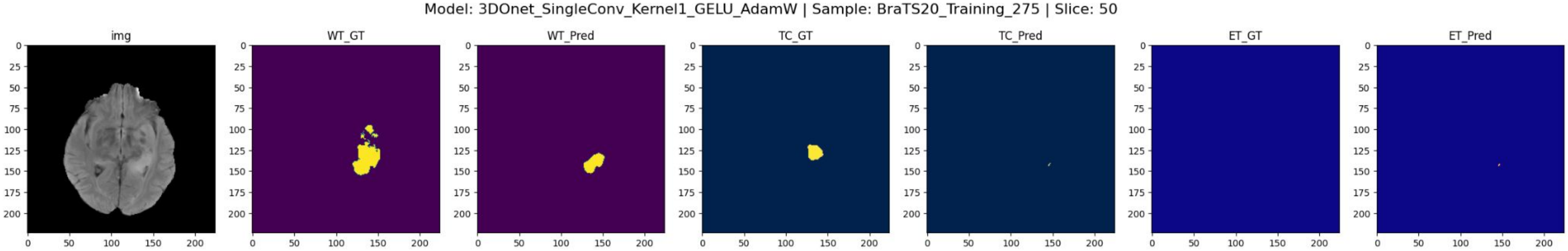
- **Incompatible input sizes**
  - One of the initial challenges faced was the fact that the default input size of (240, 240, 155) was incompatible with one of the evaluated models (specifically MONAI's implementation of Swin UNETR).
  - We hence implement the function crop_3d_array to crop the input to size (224, 224, 128) with dimensions that are multiples of 32.

- **Long training time**
  - Due to the complexity of the task and model, training is prone to interruptions as it can take up to several hours to complete 50 epochs. To address this, the trainer always saves the model checkpoints your_best_model_{YYYYMMDD_HHmmss}_{epoch}.pth in the model's ./Logs folder for the epoch with the best validation accuracy in the format.
  - This allows the checkpoint to be loaded in the future using the Trainer's load_pretrain_model(path) method.

Limitations

- **Unusually poor results for certain samples**
  - Certain samples show unusually poor results for classes TC and ET, with dice/jaccard scores of less than 0.5 and 0.001 for TC and ET.
  - One explanation is that these samples are **anomalously labelled** or have input images that are misrepresented by their labels.
  - Another explanation is that the area covered by the ground truth TC and ET **masks is unusually small** for these samples.
  - **Sample 141 (Top) and 275 (Bottom) are examples of such cases**

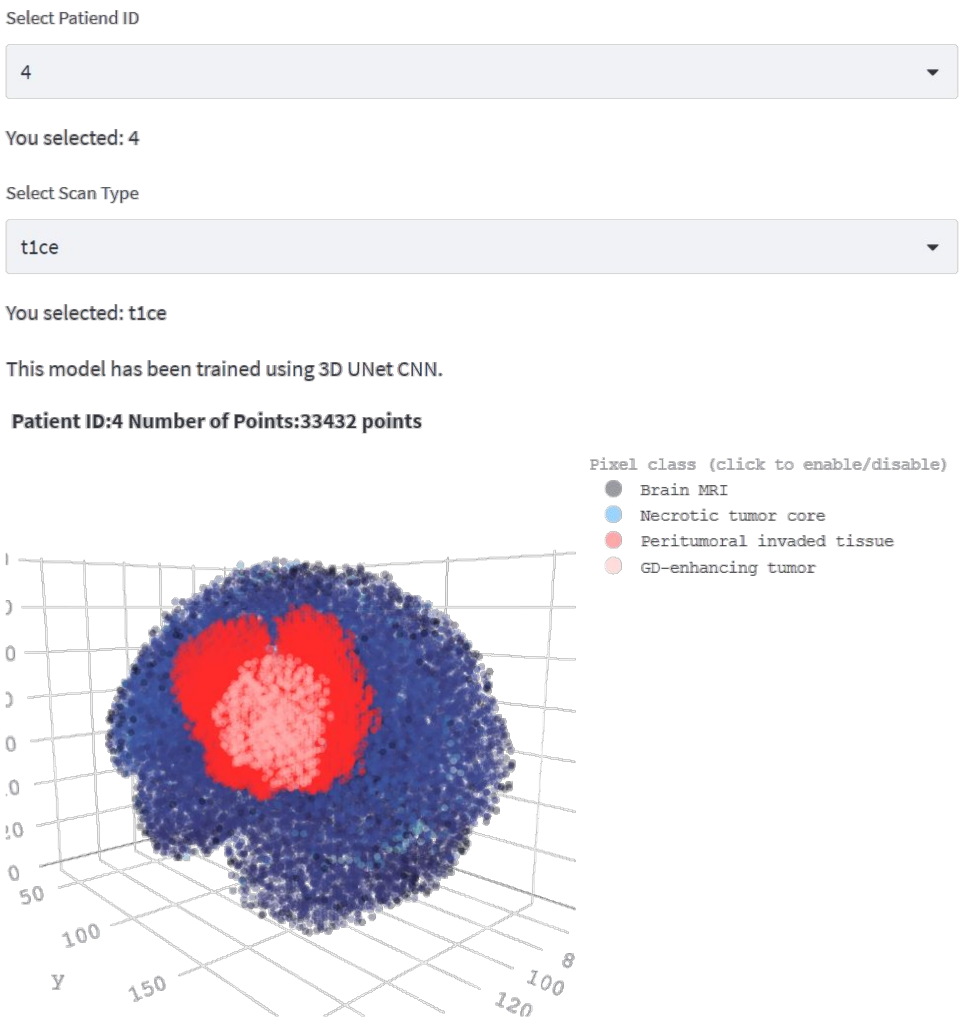




Limitations

- **Limited Compute Power**
    - The task complexity restricted the number of train, validation and test samples; Trained models are **potentially overfitted** to the 53 test samples.
    - The size and complexity of the dataset **prevents** us from performing **extensive hyperparameter tuning** and exploring further techniques such as increasing the number and sizes of layers
    - We **cannot replicate** the conditions used to train NVIDIA's state-of-the-art approach (Swin UNETR).

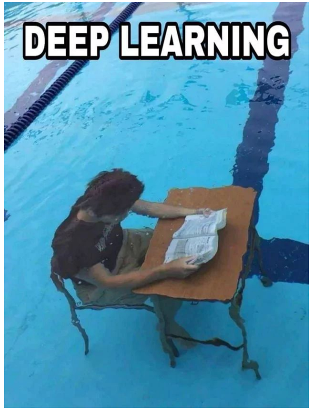
- **Lack of Data Augmentation**
    - We did not perform augmentation due to the **greater compute power required (more data)** and the **danger of generating anatomically incorrect samples.**
    - There could be danger in incorporating anatomically unrealistic examples into training sets; impact of such inclusions has yet to be studied extensively.



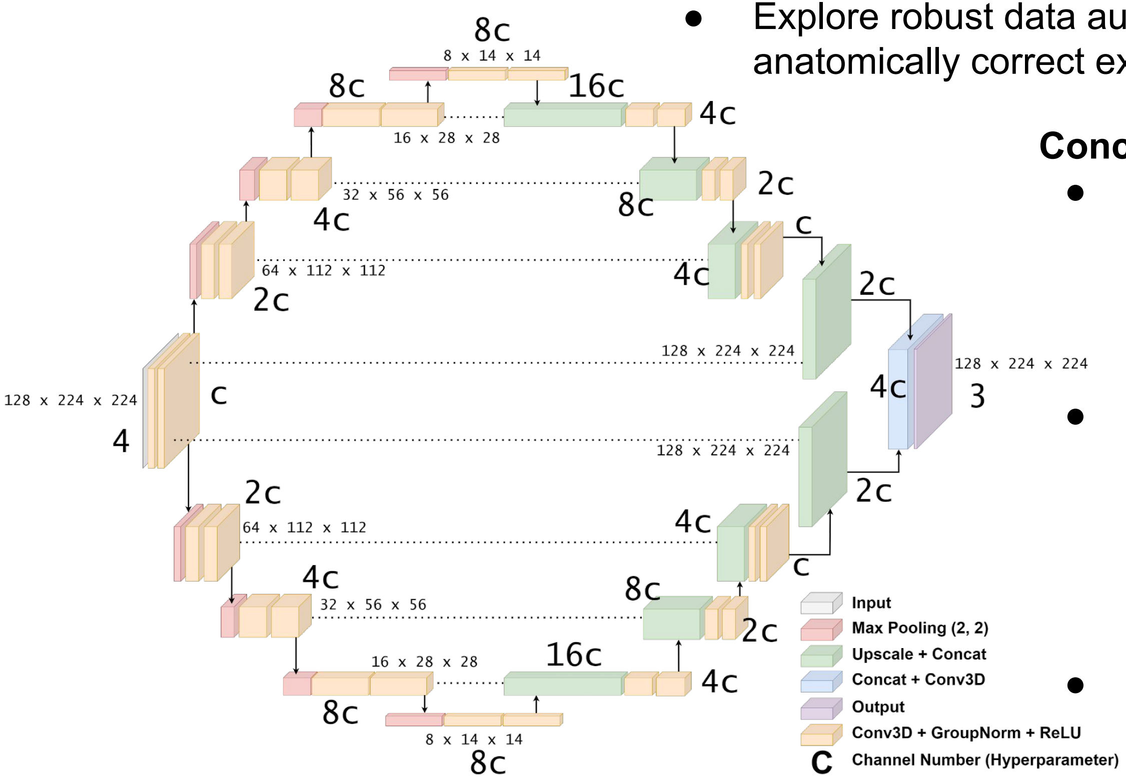
Demonstration

(If time permits)

- [GUI (Visualisation)]
- **Notebook (Single Eval)**
- **Notebook (Initiate Training Process)**



Future Improvements and Conclusion



Future Improvements

- Combine best-performing models into an ensemble model.
- Explore robust data augmentation methods that can generate anatomically correct examples to increase size of dataset.

Conclusion

- We introduced the novel 3D-ONet architecture which we found to be **performant** yet **parameter-efficient** for 3D brain tumour segmentation.
- **Increasing the number of encoder-decoder paths in UNet-style architectures is hence a promising approach that ought to be explored**.We demonstrated its potential through test results against current SOTA.
- However, more work is required to further verify the performance of the model compared to state-of-the-art approaches,especially under data-abundant conditions.