

## ***Automated Segmentation and Registration of Brain Tumor***

Teammates: Yuliang Xiao (yxiao39), Yuanwu He (yhe87), Xiao Fu (xfu18)

### **INTRODUCTION & OBJECTIVE**

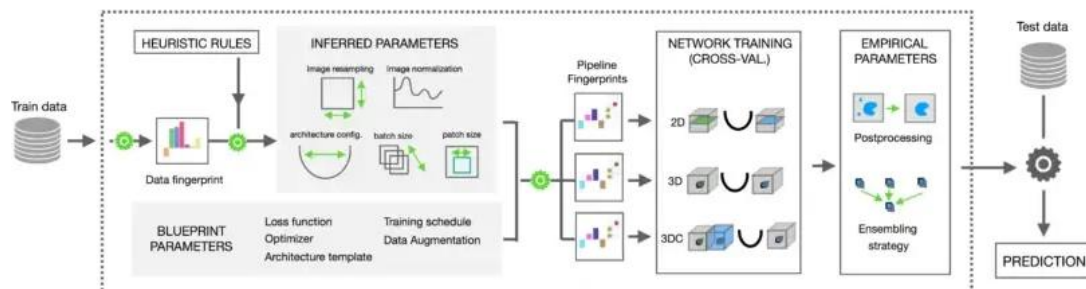
Before deep learning is applied to medical images, brain tumors are found by using CT/MRI scanners and manually detected by doctors, making the whole process extremely time-consuming. As a result, the accuracy of diagnosis could be lower regarding the doctor's workload. A computer-aided diagnosis will help eliminate those problems.

Our project developed a deep-learning platform for the automated segmentation of brain tumors. We tried two different algorithms to implement automated segmentation and registration, nnU-Net, and VoxelMorph. Finally, the feasibility of the method will be validated.

### **METHODS & PROCEDURES**

#### **1. nnU-Net (segmentation-based framework)**

nnU-Net is an open-source tool designed for medical image segmentation. It provides an end-to-end automated pipeline to deal with dataset diversity in the medical image segmentation domain. The full name of nnU-Net is no new U-net; it gets the name because nnU-Net is based on U-Net architecture. The output of nnU-Net is as same as U-Net, a probability map of where the label locates most likely located. The nnU-Net complete Workflow [1] is shown (Fig.1).



*Figure 1. nnU-Net Work Flow*

Compared to the traditional U-Net, nnU-Net has two signs of progress. First, nnU-Net does not need a change in network architecture. Second, nnU-Net can adjust hyper-parameters by itself. The rule of adjusting the data-dependent hyper-parameters is the heuristic rule [2], known as the “data fingerprint” (Fig.2).

Design choice	Required input	Heuristic Rule
Intensity normalization	Modality, intensity distribution	If CT: Global dataset percentile clipping + Z-Score with global foreground mean and std. Else: Z-Score with per image mean and std.
Image resampling strategy	Distribution of spacings	If anisotropic: In-plane with 3rd order spline, out-of-plane with nearest neighbour Else: 3rd 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 neighbour Else: linear interpolation
Image target spacing	Distribution of spacings	If anisotropic: Lowest resolution axis 10th percentile, other axes median. Else: 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 / img. 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.

Figure 2. Heuristic Rule of nnU-Net

To demonstrate how the heuristic works, we will introduce the fifth heuristic rule, network topology and patch size and batch size, in detail. When developing a network, we always prefer a large batch size and large patch size because large batch sizes allow for more accurate gradient estimation (in practice, any batch larger than one results in robust training), and larger patch sizes can increase the contextual information received by the network. However, these two parameters are limited by GPU memory. Therefore, what the nnU-Net has done is finding a balance between them. nnU-Net will initial a patch size and the median image size, and then, the nnU-Net will automatically configure the architecture according to the patch size. Then, based on that topology, a GPU memory estimation can be made. Finally, if the GPU estimation fits in the user's actual GPU memory, an architecture that maxes out the batch size is down; if not, the nnU-Net will reduce the patch size a little and repeat this process. In conclusion, based on a series of heuristic rules, nnU-Net can fit hyper-parameters.

## 2. VoxelMorph Algorithm (registration-based segmentation framework)

VoxelMorph is a fast unsupervised-learning-based framework for deformable, pairwise medical image registration [3].

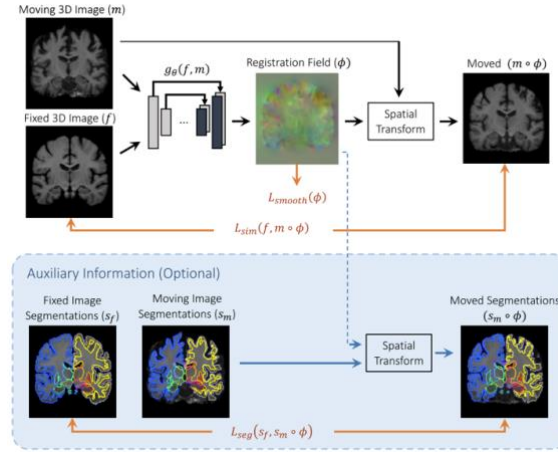


Figure 3. Overview of VoxelMorph Method

It treats registration as a function to map paired input images to a deformation field that makes them aligned. Registration is formulated as an objective function and used in the convolution neural network to build the model that can optimize this function.

Before training the model, we first preprocessed the data. In order to make the data compatible with the VoxelMorph frameworks, the raw images must be cropped to a specific shape. Besides, it also requires the images to be scaled between 0 and 1 (z-score normalization). We used ANTsPy, a Python library that includes blazing-fast IO, registration, segmentation, statistical learning, and visualization functionalities, among others [4].

After training the model, we input two images, the first image was segmented, and the second was not. The network found the registration field between these two images and then applied it on the label of the first image. Finally, we got the label of the second image. This method was especially suitable for situations where the ground truth is insufficient.

## WORKFLOW

Here we attached a flowchart to illustrate a clearer workflow on how we proceed the two algorithms, as shown in Figure 4.

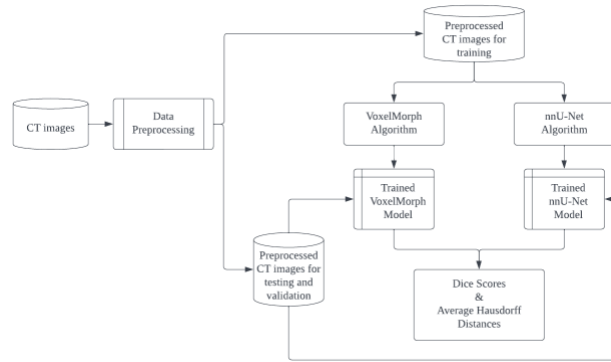


Figure 4. Flowchart of the process

First, 59 groups of CT images for training and 15 for testing were collected. Among the 59 groups of CT images, there are 47 for model training and 12 for validation since we are using 5-fold cross-validation. These images are unsuitable for directly sending the model to train due to the GPU and data format limitations. Therefore, the images are preprocessed before the training process. We first cropped the region of interest (ROI) of the images from  $512 * 512 * 256$  into the dimension of  $240 * 240 * 240$  to fit our GPU limitation. Later, we extract the image data and convert the file format of these images from nifti to npz for model training. Finally, the preprocessed CT images for training are sent to the VoxelMorph and nnU-Net algorithms for model training, and the validation CT images are used for model evaluation for each hyperparameter setting. Once the models were determined, we applied our preprocessed testing data to the model to evaluate the performances by comparing the results with the ground truth based on two methods, Dice Coefficient Scores and Average Hausdorff Distances (AHD).

## RESULTS

The accuracy of our models for the two algorithms are listed in Table 1.

Table 1: The accuracy of the models for the two algorithms based on Dice Score and AHD

Name	nnU-Net		VoxelMorph	
	Dice Score	AHD	Dice Score	AHD
CVHN_308	0.73386351	3.102099068	0.550708322	3.422939684
CVHN_367	0.803976272	1.907423993	0.625511323	3.77215801
CVHN_412	0.163808638	16.75043043	0.557578369	3.69789234
CVHN_869	0.708315434	2.395088666	0.424232461	5.859322669
CVHN_838	0.450067758	8.710050878	0.277045361	13.18029103
CVHN_488	0.627526769	3.896385943	0.438617091	6.019322202
CVHN_609	0.240037951	3.869196222	0.082198863	10.02374945

CVHN_088	0.794220902	1.628269231	0.423731139	4.634777631
CVHN_067	0.713886717	2.063536097	0.608115355	3.287171587
CVHN_560	0.843824965	1.316909633	0.699757371	2.791373435
CVHN_519	0.565362565	7.192020796	0.36735845	9.496100228
CVHN_356	0.817110694	1.844885556	0.082198863	10.02374945
CVHN_192	0.780641673	1.744061265	0.398588453	5.141719313
CVHN_813	0.881820123	1.012853212	0.524958264	5.249528823
	<b>average = 0.65</b>		<b>average = 0.43</b>	

The best matched image set for nnU-net algorithm model is CVHN\_813, and the best matched image set for VoxelMorph algorithm model is CVHN\_560. Some hyperparameters of our models are listed here:

- Alpha: 0.25
- Loss: Dice + Focal loss
- Gamma: 2
- Training epoch: 150
- Learning rate: 0.01
- Patch\_size: [96 160 128]
- Gradient Descent Method: Stochastic Gradient Descent (SGD)

The link to the source code of our implementation was listed in the Appendix section.

The closer the dice score is close to 1, the better the result is. We had an average dice score of 0.65 for the nnU-net model and 0.43 for the VoxelMorph model. In general, this is an acceptable result but not the great one, and there are still some modifications we could make to improve our models. A detailed analysis is shown in the Future Improvement section.

## **FUTURE IMPROVEMENT**

Deep learning requires a large dataset to improve the performance of the models. For our project, we only used 59 groups of data to train our model, which might lead to a poor result. We should collect more labeled data to improve performance. Furthermore, we could improve the architecture of our model by adding residual units and modifying the encoder and decoder. The hyperparameters are also significant to the performance of our trained models. We could modify some of the hyperparameters, such as the optimizer, learning rate, loss function, and dropout rate,

to increase the accuracy of our auto-detection model. Our hardware is another limitation to our models since it might take too long to train a perfect model under specific hyperparameters.

### **THINGS LEARNED DURING THE PROJECT**

First, we learned how to search for a proper dataset for training. In the beginning, we wanted to use some MRI images for training, but there needed to be more ground truth to compare the results. Hence, we switched our data from MRI to CT images containing ground truth. Secondly, we learned how to construct a deep learning model and adjust the hyperparameters to achieve a better result. It is fascinating to accomplish the out-of-textbook task using the knowledge we learned during the lectures. Thirdly, and most importantly, we learned how to cooperate with teammates during the busy semester to complete a project from scratch. The dedication of every teammate is the critical factor that we could finish the project on time.

### **ADVICE TO NEW CV STUDENTS**

Students need to know how to implement the specific algorithm, but it is more beneficial for them if they learn the principle of different algorithms. Be patient with the assignment and be active in either peer communication or office hours since, from our point of view, this is the critical factor for a student to learn new knowledge efficiently and effectively.

### **REFERENCE**

- [1] Prateek Gupta, Kumar T. Rajamani, Mattias P. Heinrich. nnU-Net: The no-new-UNet for automatic segmentation
- [2] Isensee F, Jaeger PF, Kohl SAA, Petersen J, Maier-Hein KH. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nat Methods*. 2021 Feb;18(2):203-211. doi: 10.1038/s41592-020-01008-z. Epub 2020 Dec 7. PMID: 33288961.
- [3] Guha, B., Amy, Z., Mert, S.R., John, G., & Adrain, D.V (2019). VoxelMorph: A Learning Framework for Deformable Medical Image Registration.
- [4] Avants, B. B., Tustison, N., & Song, G. (2009). Advanced normalization tools (ANTs). *Insight j*, 2(365), 1-35.

### **APPENDIX**

Link to the source code: <https://github.com/MIC-DKFZ/nnUNet>