ADVANCING MEDICAL IMAGE REGISTRATION AND TUMOR SEGMENTATION WITH DEEP LEARNING: DESIGN, IMPLEMENTATION, AND TRANSFER INTO CLINICAL APPLICATION

by

Yaying Shi

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing & Information Systems

Charlotte

2024

Approved by:

Dr. Yonghong Yan

Dr. Min Shin

Dr. Razvan C. Bunescu

Dr. Srijan Das

Dr. Xiuxia Du

ABSTRACT

YAYING SHI. Advancing Medical Image Registration and Tumor Segmentation with Deep Learning: Design, Implementation, and Transfer into Clinical Application . (Under the direction of DR. YONGHONG YAN)

The advancement of medical imaging has significantly enhanced the ability to diagnose, monitor, and treat cancer. This dissertation focuses on the development of deep learning methodologies for the segmentation and registration of medical images, specifically Positron Emission Tomography(PET), Computed Tomography(CT), Magnetic Resonance Imageing(MRI), and pathology images, to improve the accuracy and efficiency of cancer diagnosis and treatment planning.

Segmentation, the process of delineating anatomical structures and pathological regions, is a crucial step in medical image analysis. This work introduces novel high-precision deep learning models for the automatic segmentation of tumors and organs at risk (OARs). These models utilize convolutional neural networks (CNNs) and transformer-based architectures to handle the complexities and variations inherent in PET, CT, and MRI. The segmentation models are trained on multi-modal imaging datasets, incorporating advanced techniques such as data augmentation, transfer learning, and ensemble learning to enhance robustness and generalization. Evaluation on various datasets demonstrates that these models achieve superior performance compared to traditional methods, with significant improvements in accuracy and reliability.

Registration, which aligns images from different modalities or time points, is another critical component in the analysis of medical images. This dissertation presents advanced deep learning approaches for the registration of CT, MRI, and pathology images, leveraging deep neural networks (DNNs) and unsupervised learning techniques. The proposed registration methods utilize two self-supervised vision transformer networks along with other novel architectures to learn feature representations

from publicly available data. These features are then processed through a feature matching network, enabling the accurate alignment of multi-modal images. We further extended this method with a Python script, employing it as an image fusion tool for MRI to enhance image resolution and quality through advanced up-scaling techniques. These approaches are designed to be computationally efficient and scalable, facilitating their integration into clinical workflows.

Our final goal is to streamline those deep learning methods to real clinical applications. This dissertation explores the practical applications of the developed models, including their deployment in microservices for common radiotherapy imaging tasks. The models are made accessible via Python scripts for clinical treatment planning software such as RayStation, allowing seamless integration into existing clinical systems. Evaluation using images and treatment planning data for prostate cancer underscores the potential of these models to enhance the quality of treatment planning and streamline the overall process of planning, response assessment, and adaptation. Additionally, this dissertation investigates the potential of federated learning for collaborative model training across multiple institutions without sharing sensitive patient data. This approach could enhance model robustness and generalizability by leveraging diverse datasets from various sources.

In conclusion, this dissertation explores the critical component of medical imaging for cancer diagnosis, monitoring, and treatment with advanced deep learning methods. We hope those innovative techniques developed in this research pave the way for more precise, efficient, and individualized patient care in oncology.

DEDICATION

Pursuing a Ph.D. is more than earning a degree; it is a way of life. When I first arrived at UNCC in 2019, I was overwhelmed by the unknown, unfamiliar with Charlotte, and uncertain about my future. Five years later, as I prepare to graduate, I feel a deep connection to this city, the university where I have lived, and, most importantly, the people I have come to love. My Ph.D. journey would not have been possible without the support of my advisor, my fiancée, my parents, my family, my friends, and many others. I dedicate this dissertation to them.

First and foremost, I extend my heartfelt gratitude to my fiancée, Jing, for her understanding, help, and companionship. Whenever I felt lost in my research or faced challenges in my daily life, she was always by my side, offering encouragement, inspiration, and a sense of home as a foreigner in the US.

I also wish to thank my parents for their endless, selfless, and unconditional support. Their unwavering love and sacrifices have been the foundation of my success. I am so proud to be the son of my parents. Additionally, I am grateful to my family and relatives for their understanding, help, and care for my parents during the pandemic.

Last but certainly not least, I would like to thank my advisor, Yonghong Yan. There is a saying in Chinese: "A teacher for a day is a father for a lifetime." Professor Yan's support extended far beyond this, guiding me not only in my research but also allowing me the freedom to explore various directions. His patience in teaching me how to write, conduct research, and propose ideas has been invaluable throughout my Ph.D. journey.

I am deeply thankful to all the people mentioned above and those not named here for their support and assistance.

ACKNOWLEDGEMENTS

Firstly, I would like to express my deepest gratitude to my advisor, Dr. Yonghong Yan, for his invaluable guidance and support throughout my research. I could not have completed this Ph.D. without his help. I greatly appreciate his guidance not only on technical problems but also on career paths and future research insights.

Secondly, I would like to thank all my committee members, Professor Min Shin, Srijan Das, Razvan C. Bunescu, Xiuxia Du, and Aidong Lu, for their efforts, help, and insightful suggestions along my journey. I am grateful for their dedication to serving as my committee.

Thirdly, I would like to thank Dr. Chunhua Liao for his guidance during my summer internship at LLNL. I learned a lot from him, especially about professional research and other valuable skills. I also thank Dr. Xiaodong Zhang for his mentorship during my summer internship at MD Anderson. His expertise in radiology oncology opened my mind to new research perspectives and helped me rethink the balance between academic and clinical research. Additionally, I would like to thank Dr. Don Chen and Dr. Wenwu Tang for their inspiration on how to translate my research into real engineering applications.

Fourthly, I would like to thank my collaborators Anjia Wang, Xinyao Yi, Kewei Yan, Gaurav Verma, Christian Micklisch, Erum Mushtaq, Salman Avestimehr, Hongji an Gao, and Anshuk Gottipati during my Ph.D. journey. I appreciate their collaboration and support.

Lastly, I would like to thank the College of Computing and Informatics, William States Lee College of Engineering, Lawrence Livermore National Laboratory, Texas Advanced Computing Center, and The University of Texas MD Anderson Cancer Center for their financial and resource support during my Ph.D. studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: Introduction

Cancer is one of the most challenging diseases in human history, characterized by the uncontrolled growth of the body's cells, surpassing the growth rate of normal tissue. While cancer itself does not directly cause death, it can be fatal by compromising essential bodily functions, such as the immune system and the digestive system. In 2020 alone, there were 9.7 million deaths attributed to cancer, highlighting its significant impact on global health [7].

Dealing with cancer involves a multifaceted approach that includes prevention, early detection, and various treatment modalities. Among the treatment options, radiation therapy is widely used for many types of cancer. This treatment involves the use of high-energy radiation to damage the DNA of cancer cells, which inhibits their ability to grow and divide. As a result, radiation therapy can significantly increase the survival rate of cancer patients by effectively targeting and reducing cancerous tumors.

As Fig. 1.1 shows, radiation therapy typically consists of several stages:

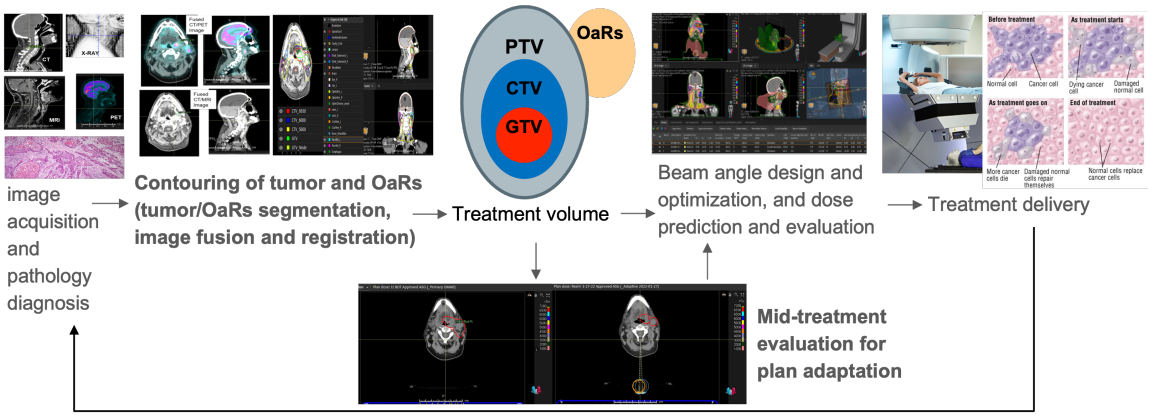


Figure 1.1: The process of radiation treatment planning and the use medical imaging

1. Image Acquisition and Pathology Diagnosis: The process begins with comprehensive imaging and pathology diagnosis to accurately identify the tumor and any affected organs. This includes the use of various imaging techniques such as CT, MRI, and PET scans.

2. Contouring of Tumor and Organs at Risk (OARs): After acquiring the images, the next step is contouring the tumor and OARs. This involves segmenting the tumor and surrounding healthy tissues, fusing images from different modalities, and registering them to create a precise treatment map.

3. Treatment Volume Definition: Defining the treatment volume involves delineating the Gross Tumor Volume (GTV), Clinical Target Volume (CTV), and Planning Target Volume (PTV). This step ensures that the radiation dose covers the entire tumor while sparing as much healthy tissue as possible.

4. Beam Angle Design and Optimization: The treatment plan is further refined by designing and optimizing the beam angles. This involves predicting and evaluating the radiation dose distribution to maximize tumor control and minimize exposure to healthy tissues.

5. Treatment Delivery: The patient undergoes the actual radiation treatment based on the optimized plan. This involves carefully targeting the defined treatment volumes with high precision.

6. Mid-Treatment Evaluation for Plan Adaptation: During the end of treatment, periodic evaluations are conducted to monitor the patient's response after initial planning, quickly adapt, and adjust the plan based on the feedback of patient during the mid-treatment. Adjustments to the treatment plan may be made to adapt to changes in the tumor size or position, ensuring continued effectiveness.

By understanding the nature of cancer, the damage it can cause, and the role of radiation therapy, we can better appreciate the importance of this treatment in improving the survival rates of cancer patients. This is especially true for the first

and second steps, which involve image registration and segmentation to provide better visualization to the physicians. With better visualization, physicians can make more precise beam angle settings and dose distributions, ultimately improving the survival rate of the patient.

My research will focus on providing better medical image segmentation and registration using advanced deep learning and machine learning technologies. By enhancing these critical steps in the radiation therapy process, we aim to improve the accuracy and effectiveness of treatment planning, ultimately contributing to better outcomes for cancer patients.

1.1 Problem Definition

1.1.1 Medical Image Registration

In radiation therapy, precise medical image segmentation and registration are critical for accurately identifying and targeting cancerous tissues while sparing healthy organs. Registration involves the process of aligning images from different modalities (such as CT, MRI, and PET scans) or different time points to create a coherent, comprehensive view of the patient's anatomy. This alignment is essential for ensuring that the various images overlay correctly, allowing for accurate diagnosis, treatment planning, and monitoring of tumor progression. Traditional registration techniques often rely on manual alignment or simple algorithms, which can be imprecise and time-consuming. Advances in image registration, particularly through the application of deep learning, have revolutionized this field. Deep learning algorithms can automatically detect and correct misalignments with high accuracy, improving the reliability and efficiency of the registration process.

A typical medical image registration can be shown as Fig. 1.2. Let I_f and I_m denote the fixed image and the moving image, respectively. The goal of image registration is to find a spatial transformation T that aligns I_m with I_f. The transformation T can be parameterized by a set of parameters θ.

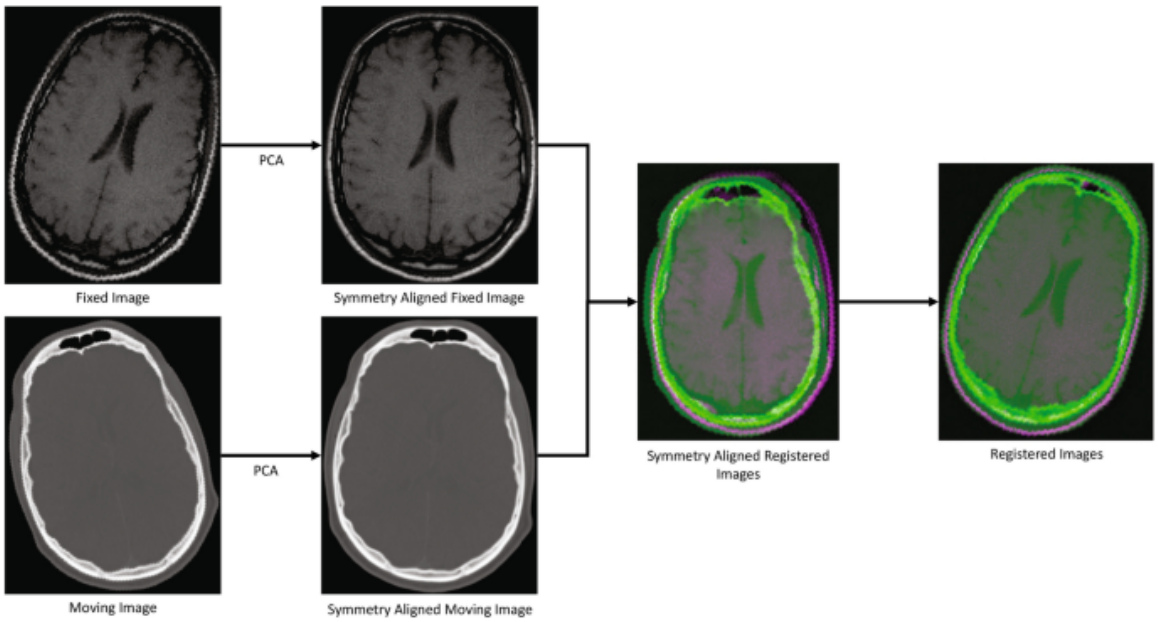


Figure 1.2: The process of Medical image registration for different modalities [2]

The transformed moving image is denoted by $I_m(T(x;\theta))$, where x represents the coordinates of a point in the image.

The objective is to find the optimal parameters θ^* that minimize the difference between the fixed image I_f and the transformed moving image $I_m(T(x;\theta))$. This can be formulated as an optimization problem:

$$\theta^* = \arg\min_{\theta} D(I_f(x), I_m(T(x;\theta)))$$

Here, D is a similarity measure or distance metric that quantifies the difference between the fixed image and the transformed moving image. Common evaluation metrics used in medical image registration include:

1. **Mean Squared Error (MSE)**:

$$D_{\text{MSE}}(I_f, I_m) = \frac{1}{N} \sum_{x} (I_f(x) - I_m(T(x;\theta)))^2$$

2. **Mutual Information (MI)**:

$$D_{\text{MI}}(I_f, I_m) = -\sum_{i,j} p_{I_f,I_m}(i,j) \log \frac{p_{I_f,I_m}(i,j)}{p_{I_f}(i)\, p_{I_m}(j)}$$

where $p_{I_f,I_m}(i,j)$ is the joint probability distribution of the intensities of I_f and I_m, and $p_{I_f}(i)$ and $p_{I_m}(j)$ are the marginal probability distributions of the intensities.

3. **Normalized Cross-Correlation (NCC)**:

$$D_{\text{NCC}}(I_f, I_m) = -\frac{\sum_x (I_f(x) - \bar{I}_f)(I_m(T(x;\theta)) - \bar{I}_m)}{\sqrt{\sum_x (I_f(x) - \bar{I}_f)^2 \sum_x (I_m(T(x;\theta)) - \bar{I}_m)^2}}$$

where $\bar{I}_f$ and $\bar{I}_m$ are the mean intensities of the fixed and moving images, respectively.

4. **Structural Similarity Index (SSIM)**:

$$D_{\text{SSIM}}(I_f, I_m) = \frac{(2\mu_{I_f}\mu_{I_m} + c_1)(2\sigma_{I_f I_m} + c_2)}{(\mu_{I_f}^2 + \mu_{I_m}^2 + c_1)(\sigma_{I_f}^2 + \sigma_{I_m}^2 + c_2)}$$

where: μ_{I_f} is mean of I_f, μ_{I_m} is mean of I_m, σ_{I_f} is standard deviation of I_f, σ_{I_m} is standard deviation of I_m, $\sigma_{I_f I_m}$ is covariance of I_m and I_f, c_1 is constant to stabilize the division with weak denominator, c_2 is constant to stabilize the division with weak denominator.

The optimization problem can be solved using various optimization techniques, such as gradient descent, Powell's method, or evolutionary algorithms, depending on the complexity of the transformation and the similarity measure used.

1.1.2 Medical Image Segmentation

Segmentation refers to the delineation of specific structures within medical images, such as tumors and organs at risk (OaRs). Accurate segmentation is crucial for defining the boundaries of the treatment area and for planning the radiation dose distribution. Conventional segmentation methods typically involve manual contouring by radiologists, which is both labor-intensive and subject to inter-observer variability.

To address these challenges, deep learning techniques, especially convolutional neural networks (CNNs), have been employed to automate and enhance the segmentation process. These networks can be trained on large datasets of annotated medical images to learn intricate patterns and features that distinguish different tissue types. As a result, deep learning-based segmentation provides consistent and precise delineations, facilitating better treatment planning and outcomes.

A typical medical image segmentation can be shown as Fig. 1.3. An input medical image set S with C segmentation classes can be represented as:

$$S = \{(x_i, y_i) \mid i = 1, 2, 3, \ldots, n\}$$

where $x_i = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m)$ is the input image vector from different modalities, and y_i is the segmentation mask in $C = \{c_1, c_2, \ldots, c_n\}$ classes. A medical image segmentation method can be considered as a unified function f, with the formula for an input x_i given by:

$$\hat{y}(x_i) = f(x_i)$$

$$\bigcup_{i=1}^{n} C_i = \hat{y}(x_i), \quad C_i \subset C, \quad C_i \cap C_j = \emptyset, \quad i, j \in [1, n]$$

The $\hat{y}(x_i)$ is the output of the image segmentation method. C_i is one of the segmentation classes. i and j are different numbers representing different classes in the image, and n is a positive number greater than 1. The goal of image segmentation is to find the segmentation function f that maps the input image S to the segmented image S_{seg}:

$$S = f(S_{seg})$$

The segmentation function f can be parameterized by a set of parameters θ. The objective is to find the optimal parameters θ^* that minimize the difference between the

segmented image S and the ground truth segmentation S_{gt}. This can be formulated as an optimization problem:

$$\theta^* = \arg\min_{\theta} L(S, S_{gt})$$

Here, L is a loss function that quantifies the difference between the segmented image and the ground truth segmentation. Common loss functions used in medical image segmentation include:

1. **Dice Coefficient (DICE)**:

$$L_{\text{DICE}}(S, S_{gt}) = 1 - \frac{2\sum_x S(x)S_{gt}(x)}{\sum_x S(x) + \sum_x S_{gt}(x)}$$

2. **Cross-Entropy Loss (CE)**:

$$L_{\text{CE}}(S, S_{gt}) = -\sum_x S_{gt}(x) \log S(x)$$

The optimization problem can be solved using various optimization techniques, such as gradient descent or more advanced methods, depending on the complexity of the segmentation function and the loss function used.

To evaluate the performance of an image segmentation method, common metrics include Intersection over Union (IoU), Dice Coefficient (DICE), and Hausdorff Distance (HD).

1. The IoU metric measures the overlap between the predicted segmentation $\hat{y}$ and the ground truth segmentation y. It is defined as:

$$\text{IoU} = \frac{|\hat{y} \cap y|}{|\hat{y} \cup y|}$$

2. The Dice Coefficient measures the similarity between the predicted segmentation $\hat{y}$ and the ground truth segmentation y. It is defined as:

$$\text{DICE} = \frac{2|\hat{y} \cap y|}{|\hat{y}| + |y|}$$

3. The Hausdorff Distance measures the maximum distance between the predicted segmentation boundary and the ground truth segmentation boundary. It is defined as:

$$\text{HD} = \max \left\{ \sup_{s \in S} \inf_{s_{gt} \in S_{gt}} d(s, s_{gt}), \sup_{s_{gt} \in S_{gt}} \inf_{s \in S} d(s_{gt}, s) \right\}$$

where $d(s, s_{gt})$ is the Euclidean distance between points s in S and s_{gt} in S_{gt}.

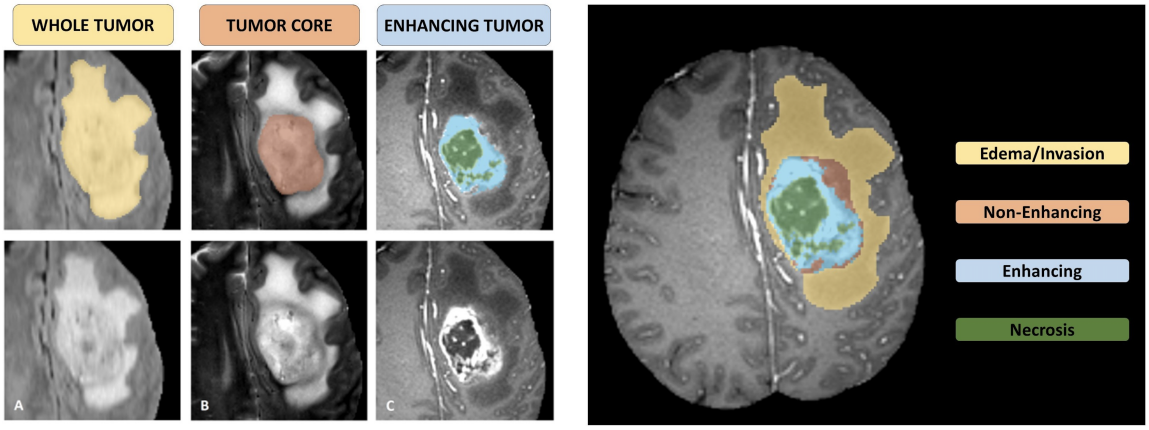


Figure 1.3: The process of Medical image segmentation for different modalities [3]

These metrics provide a quantitative measure of how well the segmentation method performs, with values ranging from 0 to 1 for IoU and DICE, where 1 indicates perfect agreement between the predicted and ground truth segmentations. The HD value, on the other hand, indicates the maximum distance between the boundaries of the segmented regions.

1.2 Overall Motivation and Challenge

Radiotherapy demands a high degree of personalization and precision to optimize patient treatment outcomes, known as personalized treatment and precision medicine. Despite significant advancements in technology and methodology, achieving this level

of precision in radiation treatment planning remains a major challenge. The current processes are time-consuming and exhibit considerable output variability, which poses significant hurdles in the radiation therapy workflow.

In my defense of proposal, I highlighted the critical need for accurate and efficient segmentation and registration of medical images to enable precision medicine and adaptive radiotherapy. These tasks are fundamental to ensuring that radiotherapy is both effective and precised.

1.2.1 Motivation

The primary motivation for this thesis is to address the limitations of current radiotherapy imaging techniques by enhancing the accuracy and efficiency of segmentation and registration processes. Traditional manual contouring methods are labor-intensive and prone to variability among radiation oncologists. The advent of deep learning (DL) models, such as convolutional neural networks (CNN) and transformer-based networks, has shown promise in automating these tasks. However, the improvement in accuracy over the past five years has been incremental rather than groundbreaking. For instance, in the BraTS brain tumor segmentation challenge, the DICE score has seen only modest improvements despite a fourfold increase in dataset size.

To overcome these limitations, we propose leveraging pathology images, which provide cell-level tissue structure, to enhance MRI/CT images using DL techniques. This approach aims to achieve or approach cell-level precision in medical imaging, thereby significantly improving the quality of tumor segmentation and radiotherapy planning. By integrating advances in next-generation sequencing and imaging technologies at the cellular and molecular levels, we aim to develop a novel framework that bridges the gap between pathology and radiology.

Furthermore, federated learning offers a promising solution for data sharing and privacy concerns. Federated learning allows multiple institutions to collaboratively

train a DL model without sharing patient data, thus preserving privacy and complying with data protection regulations. This method ensures that the model benefits from a diverse dataset while maintaining the confidentiality of individual patient information.

1.2.2 Challenges

Several challenges must be addressed to realize this vision:

Deep Learning Model Development: Developing DL models capable of handling the high complexity and variability in medical images is a major challenge. The models must be robust enough to generalize across different types of tumors and patient populations.

Federated Learning Implementation: Implementing federated learning poses its own set of challenges, including coordinating training across multiple institutions, ensuring model consistency, and managing communication overhead. Additionally, addressing security concerns to prevent data leakage during the training process is paramount.

High-Resolution Imaging: Achieving cell-level precision requires significantly enhancing the resolution of MRI/CT images. This involves reconstructing images with a resolution up to 39 times finer in each dimension, which is computationally intensive and technically demanding.

Validation and Clinical Implementation: Ensuring that the developed models are clinically valid and can be seamlessly integrated into existing radiotherapy workflows is crucial. This involves rigorous testing, validation, and collaboration with clinical practitioners.

1.3 The Stat-Of-Art Method

CNNs, UNet, and Variants. Convolutional neural networks have been the primary model architecture for computer vision tasks for a number of years. For image segmentation, the pioneering work of UNet [8] introduced the successful “U-shaped"

paradigm for medical image segmentation models. Subsequent variants, such as V-Net [9], Residual UNet [10], UNet++ [11], and nnUNet [12] have modified and improved CNN-based U-shaped networks and continue to demonstrate competitive performance on many image segmentation tasks.

Transformers complement CNNs. Multi-headed self-attention (MSA) employed by transformer models computes dynamic aggregation weights between pairs of tokens. This operation mimics the convolutional filters of CNNs but with the added benefit that weights are data-dependent rather than fixed. By stacking multiple attention heads, individual attention mechanisms can specialize to different aspects, akin to multiple convolutional filters per layer. MSA allows for the extraction of longer-range dependencies by computing attention scores over all pairs of tokens; though, this comes with significant memory overhead [13]. As such, researchers have used self-attention as a complement to CNNs by replacing certain layers with transformer blocks. In the area of medical image segmentation, such hybrid approaches have been developed for multi-modal tumor segmentation [14] and brain tumor segmentation from 3D image data [4] and CT images [15]. In contrast to these hybrid approaches, our proposed framework is constructed from a pure transformer basis.

Vision Transformer. Vision Transformer (ViT) [16], and subsequent variants [17, 18], adapt the transformer architecture from NLP for computer vision. ViT splits images into fixed-sized patches representing tokens, and combines the linearized patches with positional embeddings. The tokenized images are then passed through an encoder consisting of a series of transformer blocks. Swin-Transformer [19] introduced localized self-attention using shifted-windows to improve costs associated with MSA, and Swin-UNet combined the well-established U-shaped encoder-decoder architecture with transformers for medical image segmentation [20]. Our work extends the many of design choices of Swin-UNet to 3D medical images, while including SMoE to offset increased computation costs.

Mixture of Experts. Mixture of Experts (MoE) is an architecture for realizing conditional computation in neural networks. In MoEs, the model consists of a set of expert networks that are conditionally activated on the basis of a gating or routing network [21, 22]. Sparsely-gated MoEs employing a Top-K gating algorithm were introduced in the context of ensemble LSTM models for natural language processing [23], and GShard [24] and Switch Transformers [25] developed sparse MoE layers for transformer language models. Since only a subset of experts are activated for each input example, MoEs require significantly fewer computational resources during inference and can be flexibly adapted to new circumstances by adjusting the gating network. Such an architecture is designed to integrate various models and fits naturally for multi-modal data.

MoEs in Computer Vision. In computer vision, sparse MoE layers have been shown to have a broad range of potential benefits, including integrating large ensembles of experts, reducing consequences of data imbalance, and improving model efficiency—both in terms of model size and inference costs [26, 27, 28, 29, 30, 31]. Ahmed et al. [27] and later Condconv [30] use a shallow MoE with a single routing network for image classification. DeepMoE [29] incorporates a multi-headed sparse gating network to select convolution layers, and V-MoE [31] incorporates MoE layers with vision transformer models. In the area of medical image analysis, Rasti et al. [32] develop convolutional MoE models for retinal OCT image analysis, though this work and other existing frameworks that employ a mixture of experts still largely rely on dense implementations [33, 34, 35]. The use of sparsely-gated MoE for medical image segmentation remains an open area of research, and our work represents a novel application of sparse-gately MoEs with vision transformer architectures.

Multi-Task Learning Multi-Task Learning (MTL) is a widely used approach in various domains, including Natural Language Processing (NLP), Speech Recognition, computer vision, drug discovery, and more [36]. MTL enables a model to learn

multiple tasks simultaneously by optimizing more than one loss function. By leveraging the domain-specific information contained in the training signals of related tasks, MTL improves generalization. Human learning often follows a similar approach by applying the knowledge gained from learning related tasks to new ones. MTL can be achieved through two methods: 1) hard parameter sharing, where different tasks share the same parameters in some layers but have task-specific layers for different tasks under the same network structure [37, 38]. and 2) soft parameter sharing, where each task has its own model with its own parameters, and the model's similarity is regularized to get the distance between tasks. MTL trains the same network for the same input but produces different segmentation outputs for different tasks [39].

Ensemble Learning Ensemble learning involves combining multiple machine learning models to solve a single problem, with these individual models referred to as weak learners. The idea is that by combining the output of several weak learners, they can become a stronger, more accurate model. Each weak learner is trained on a training set and provides its own prediction. The final prediction is then made by combining the results of all the weak learners [40]. Traditional ensemble learning methods include bagging, boosting, stacking, and random forest. In the realm of deep learning, there have been a few attempts to combine ensemble learning with deep neural networks (DNNs), primarily by creating ensembles of DNNs.

Fundation and Univseral Model Foundation models and universal models have shown great success in the natural computer vision domain. Consequently, researchers have begun to shift their focus toward applying these models to the medical imaging domain. Recently, a paper published in Nature introduced a foundation model for medical imaging and a universal model for various types of medical images and cancers, based on Meta's Segment Anything Model (SAM), named MEDSAM [41]. This new method represents a significant step toward the development of universal models for the medical imaging domain. As I proposed in my defense, the development of

universal or unified models for medical imaging will be a key direction for the future.

Part I

Tumor Segmentation

Overview

This part delves into the development of various deep learning methods for tumor segmentation. It comprises three key chapters:

- **Chapter 2: An Ensemble Approach to Automatic Brain Tumor Segmentation [42]** Discusses the implementation of an ensemble of convolution-based networks to enhance brain tumor segmentation accuracy. We propose a novel ensemble method that combines different convolution-based networks as Level 1 weak learners and uses a convolution network as a meta-classifier to aggregate the Level 1 predictions from each subnetwork for a superior result.

- **Chapter 3: Stacking Feature Maps of Multi-Scaled Medical Images in U-Net for 3D Head and Neck Tumor Segmentation [43]** Introduces a novel method of stacking feature maps in a U-Net architecture for head and neck tumor segmentation. Due to the marginal performance improvements from previous subnetworks, we transitioned to using a single network and incorporated more spatial and original features from different scale input images to achieve better results.

- **Chapter 4: SMoE-MLP: 3D Medical Image Segmentation with Sparse MoEs-based Multiple Layer Perceptron of Vision Transformer** Presents a framework combining sparse mixture of experts (MoEs) with a vision transformer for superior 3D medical image segmentation. Building on the previous two approaches, we further utilize sparse MoEs and integrate them with a vision transformer to enhance segmentation efficiency and accuracy.

Each chapter in this part details the design, implementation, and evaluation of the proposed models, showcasing significant improvements in segmentation accuracy and robustness compared to state-of-the-art (STOA) methods.

CHAPTER 2: An Ensemble Approach to Automatic Brain Tumor Segmentation

2.1 Introduction

Brain tumors spawn from abnormal cells that replicate in the brain without control. There are several different types of brain tumors [44]. Some are noncancerous tumors, while others are cancerous or malignant. Noncancerous tumors do not extend or transform into surrounding normal brain tissue or other tissues in the human body, making them easily distinguishable from normal brain tissue. Cancerous tumors can originate in the brain or spread to the brain from other tissues in the body. These types of tumors are difficult to discern from normal brain tissue.

Magnetic resonance imaging (MRI) is an efficient tumor diagnostic imaging modality that generates detailed images of human body tissues using a magnetic field and computer-generated radio waves. A typical 3D brain MRI can be categorized by T1 and T2 relaxation times. T1 relaxation time is the time it takes the magnetic vector to return to the resting state. The T2 relaxation time is the time it takes the axial spin to return to the resting state. A 3D T1-weighted brain tumor image is one of the modalities in MRI, which can show the differences between normal brain tissue and brain tumors with the help of T1 relaxation time. Similarly, a T2-weighted brain tumor image is another modality based on T2 relaxation time, which is important for long-term tumor tracking. T1 with contrast agent (T1-ce) and T2 Fluid Attenuation Inversion Recovery (FLAIR) are two modalities that highlight the position and shape of tumors relative to normal brain tissue.

Although these four 3D brain MRIs (T1, T2, T1-CE, T2 FLAIR) aid physicians in locating, monitoring, tracking, and treating brain tumors, they are still time-consuming processes that challenge physicians when manually segmenting a tumor

from a 3D MRI. This is due to the complex structure of tumors. An automatic brain tumor segmentation method would help physicians save time by reducing the manual load of locating and segmenting a tumor, allowing them to focus more on the patient's diagnosis and treatment plan. Traditional segmentation methods, such as the Markov random field (MRF) model [45], atlas-based segmentation model [46], and edge-based method [47], are based on intensity images, image labels, and the clustering process [48]. With the success of ML/DL methods in computer vision tasks, convolutional neural networks (CNNs) show great potential and provide feasible solutions in automated brain tumor segmentation.

The RSNA-ASNR-MICCAI Brain Tumor Segmentation challenge (BraTS) is a segmentation competition to find the best state-of-the-art brain tumor segmentation algorithm [3]. The BraTS challenge provides plenty of algorithmic opportunities for medical image segmentation. Wang [49] proposed a triple cascaded framework for brain tumor segmentation, ranked second in the BraTS 2017 challenge. Three networks are proposed to hierarchically segment the whole tumor (WNet), tumor core (TNet), and enhancing tumor core (ENet) sequentially and fuse them in different views. EMMA, ranked first in the BraTS 2017 challenge, introduced a novel ensemble of multiple models and architectures to get better results from several models [50]. The EMMA ensemble is composed of DeepMedic [51], FCN [52], and U-Net [8] models, taking advantage of them to get better segmentation results. No New-Net, ranked second in the BraTS 2018 challenge, uses U-Net as a baseline and is trained with different patch sizes and loss functions to improve performance [53]. Andriy [54] proposed an auto-encoder network with auto-encoder regularization, ranking first in the BraTS 2018 challenge.

In this work, we develop four different CNN networks and ensemble their inference output with a classifier network to get better segmentation results. Our approach was inspired by EMMA, but we use a different ensemble method which trains a new

classifier model for the inference result from four base models. By using the ensemble method, we aim to achieve a more stable and robust segmentation result for brain tumor segmentation. We evaluate our approach through the BraTS 2021 challenge validation submission.

2.2 Convolution Based Ensemble Approach

In this section, we introduce the network architecture, each base network, training parameters, and training details that we used in the validation stage of the challenge. As mentioned in the previous paragraph, we employed an ensemble approach on four different models to achieve better segmentation results. We introduce each sub-network with its training details in the following subsections.

2.2.1 Ensemble Network

After training all four sub-models (3D Unet, Residual 3D Unet, 3D Vnet, TransBTS) separately, we begin to ensemble the inference results. Different from EMMA, we train a simple 3D classification model to integrate the results. The complete architecture is shown in Fig. 2.1. For the simple classifier, the input is the stacked inference results we obtained from already trained models. This input is processed with a 3D convolution layer. Then we flatten the output and pass it to the fully connected layer for pixel-level classification. Due to time constraints, we only trained it for 200 epochs.

2.2.2 SubNetwork 1: 3D Unet

After training all four sub-models (3D Unet, Residual 3D Unet, 3D Vnet, TransBTS) separately, we begin to ensemble the inference results. Different from EMMA, we train a simple 3D classification model to integrate the results. The complete architecture is shown in Fig. 2.2. For the simple classifier, the input is the stacked inference results we obtained from already trained models. This input is processed with a 3D convolution layer. Then we flatten the output and pass it to the fully

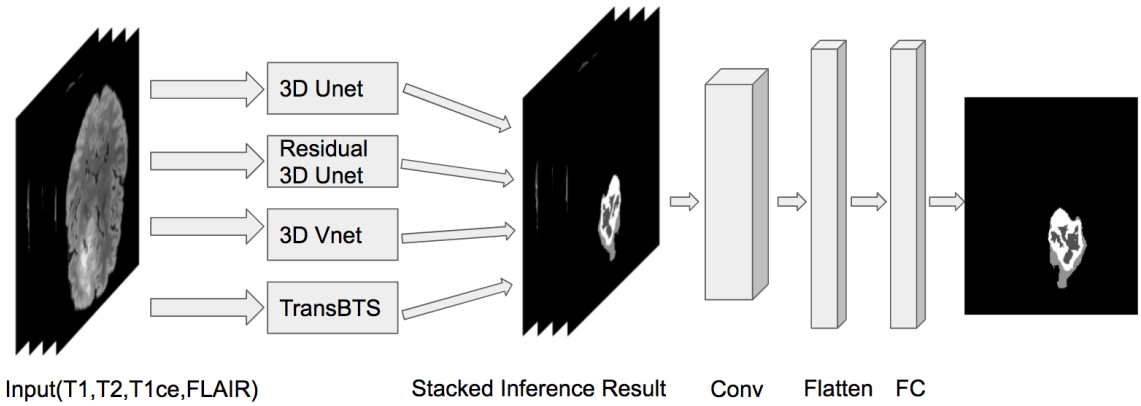


Figure 2.1: Schematic visualization of whole architecture

connected layer for pixel-level classification. Due to time constraints, we only trained it for 200 epochs.

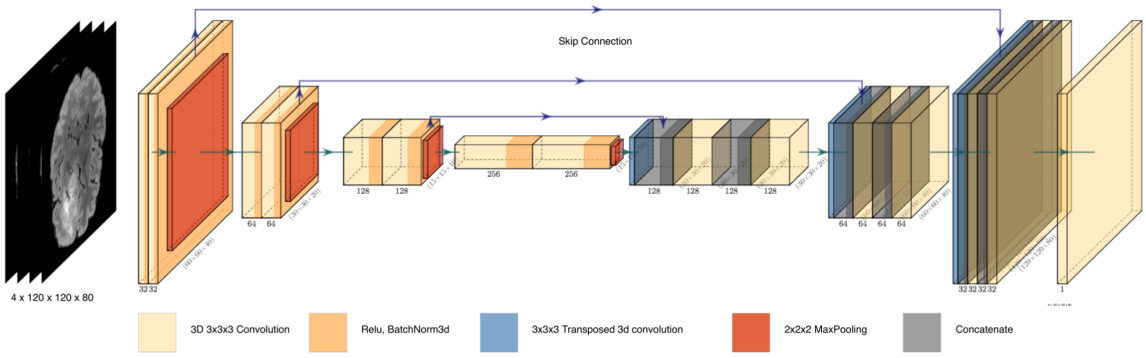


Figure 2.2: Schematic visualization of 3D Unet Network architecture

Our 3D Unet was developed based on the original Unet architecture and extended to 3 dimensions. Due to the limitation of CUDA memory on our GPU, the 3D Unet can only take a batch size of two. The detailed network structure is shown in Fig. 2.2. The input is a four-channel resized 3D MRI image. The input goes through a 3x3x3 3D convolution with 32 filters. After convolution, we apply the ReLU activation function, 3D batch normalization, and zero padding. The output has 4 channels, which is the same as the input. These 4 channels represent the background, whole tumor, tumor core, and enhanced tumor, respectively. After a sigmoid function, we obtain the segmentation result with three tumor categories.

For the training details, 3D Unet was trained with an input size of 120x120x80 and a batch size of 2. We used the Adam optimizer combined with the Dice loss function and a learning rate of 0.01. We also trained another 3D Unet model with the SGD optimizer and cross-entropy loss, maintaining the learning rate at 0.01. Due to time constraints, we trained the model for only 200 epochs.

2.2.3 SubNetwork 2: Residual 3D Unet

Residual 3D Unet has been used in the study of plant segmentation [55]. It was built upon the implementations of Çiçek et al.'s 3D Unet [56] and Lee et al.'s Residual UNet structure [57]. Each encoder is structured as a residual module whereby the output of the first convolution module is skipped over to the pooling module and combined with the output of the third convolution module, then passed to the ReLU. Due to GPU limitations, our Residual 3D Unet has been altered to accept the four 3D MRI images by decreasing the number of levels to 4, as opposed to 5. We set an input of 4 channels at 120x120x80 and 4 output channels for the segmentation classifications. Our Residual 3D Unet was trained with a batch size of 3. We used the Adam optimizer with the cross-entropy loss function. The learning rate was set to 0.01 and the model was trained for 100 epochs.

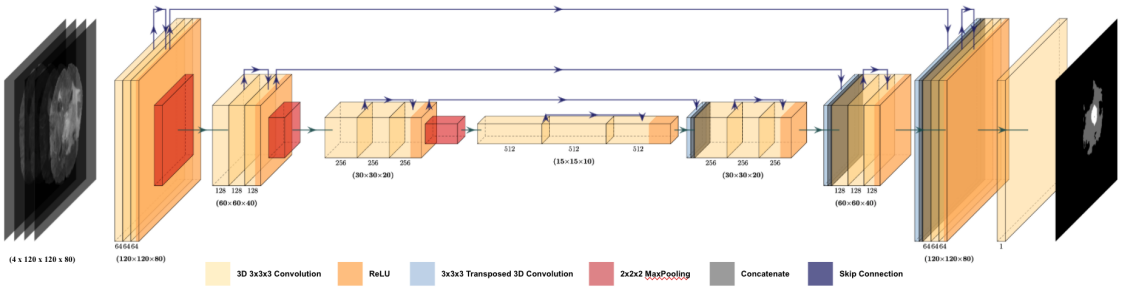


Figure 2.3: Schematic visualization of Residual 3D Unet Network architecture

2.2.4 SubNetwork 3: 3D Vnet

Vnet is a convolutional neural network that aims to segment MRI data by first compressing it to extract features and then decompressing it until the original size of the data is obtained [9]. The original Vnet work does not use all classes (4 channels/modality) data simultaneously. To adapt it to 3D, we apply a convolutional operation that takes 4-channel data as an input. Towards the end, instead of convolving 32 channels to reduce to 2 channels as done in the original work, we apply a filter to keep the output dimensions the same as the input dimensions (4x120x120x80). We also skip the softmax operation used in the original Vnet for this 3D adaptation. Another change compared to the original Vnet work is that we reduce the length of the compression path by one step and stop at 128 channels instead of 256 channels as it becomes infeasible to reduce the length and width of data 15x15x10 by 2. For training, we used a batch size of 1 with the Adam optimizer and Dice loss, with a learning rate of 0.01 for 200 epochs.

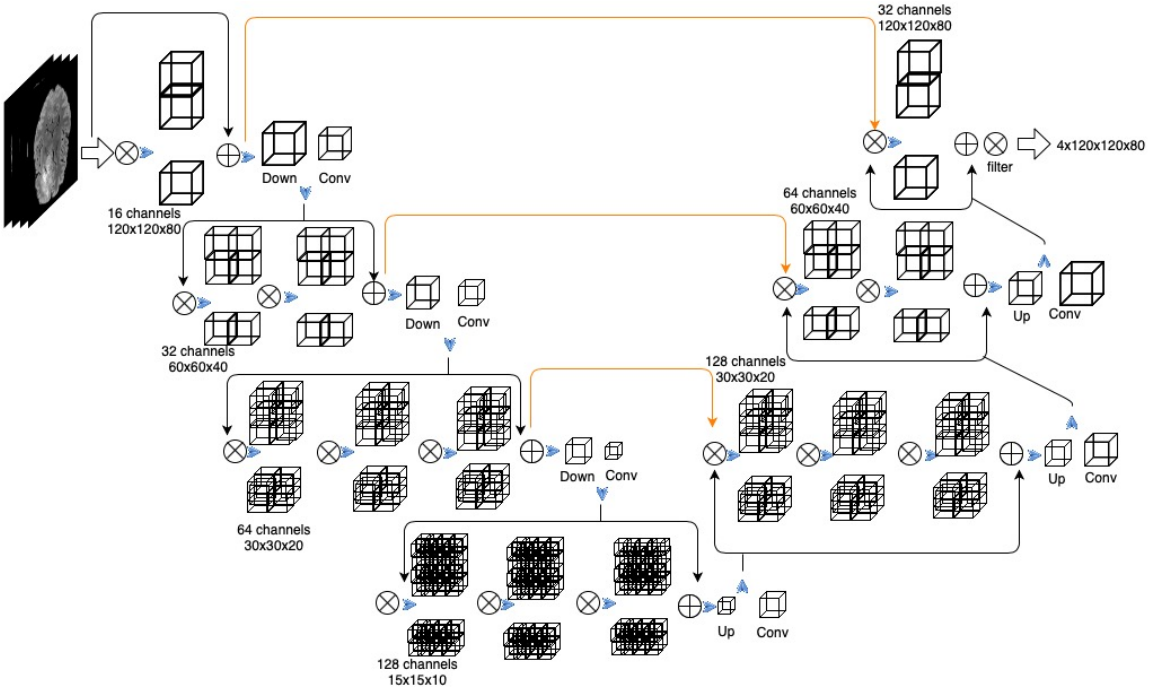


Figure 2.4: Schematic visualization of 3D Vnet Network architecture

2.2.5 SubNetwork 4: TransBTS

TransBTS is an encoder-decoder network that uses transformers in brain tumor segmentation [4]. The encoder part, which is similar to Unet, extracts semantic information with a 3D CNN structure and reduces the spatial features. By applying down-sampling, it captures local 3D context information. Inspired by the self-attention mechanism [58] and transformers [59] in natural language processing, TransBTS adds a transformer layer to the end of the encoder part. This transformer layer saves the local context information for global features. The decoder part uses the features from the transformer and performs up-sampling, combining them with high-resolution feature maps to segment the tumor.

Our TransBTS sub-model was developed based on the original TransBTS architecture. We made minor modifications to the TransBTS architecture. The original architecture is shown in Fig. 2.5. The input is still a four-channel random cropped 3D MRI with 3x3x3 3D convolution. After convolution, it applies the ReLU activation function, 3D batch normalization, and padding set to 1, which means padding is added to all four sides of the input. The decoder part uses pixel-level segmentation to restore the same dimension as the original MRI size.

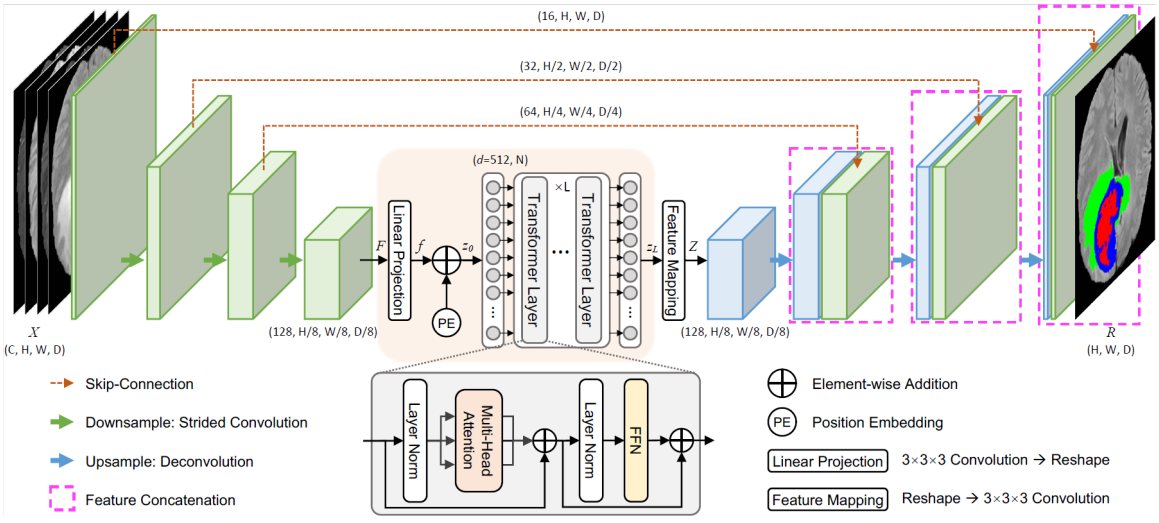


Figure 2.5: Schematic visualization of TransBTS Network architecture [4]

For the training details, TransBTS was trained with an input size of 128x128x128 and a batch size of 8. We used the Adam optimizer with Dice loss functions. The initial learning rate is 1e−4 and it is reduced using the following formula:

$$a = a * \left(1 - \frac{e}{es}\right)^{0.9} \tag{2.1}$$

where e is current epochs, es is total number of epochs. We use L2 norm regularization on the convolutional kernel parameters with a weight of 1e−5. The TransBTS was trained for a total of 1000 epochs.

2.3 Evaluation

2.3.1 Dataset Description

We used the BraTS 2021 challenge dataset for evaluation [3, 60, 61, 62, 63]. The BraTS 2021 dataset contains 1,251 cases in the training data. Each case has four modalities: 3D MRI, T1-weighted, T2-weighted, T1 contrast-enhanced, and T2 FLAIR. Our model was exclusively trained using the BraTS training dataset. In this work, we consider our four modalities as four input channels. Each volume has three dimensions: 240, 240, and 155, respectively. The segmented MRI has four labels: 0 represents the background, 4 represents the enhancing tumor, combined 1 and 4 represent the tumor core, combined 1, 2, and 4 represent the whole tumor, and 3 is not used. The validation dataset consists of 219 cases. The test data was not released by the deadline of this paper. We evaluated the performance of our model on the validation dataset using the dice score, sensitivity, specificity, and Hausdorff distances.

2.3.2 Preprocessing for Each Model

For the different sub-models, we applied various data augmentation techniques. For 3D Unet, Residual 3D Unet, and Vnet, we resized the images on different channels of the original 3D MRI. For TransBTS, we applied Z-score normalization to all modali-

ties of the 3D MRI using the mean and standard deviation. Following that, we applied linear normalization, random cropping, random clipping, and random intensity shifts.

2.3.3 Validation Phase Results

The performance of our model on the BraTS 2021 validation data is shown in Table 2.1. Our model achieved DICE scores of 0.81, 0.74, and 0.89 for ET (enhancing tumor), TC (tumor core), and WT (whole tumor), respectively. From the deviation, we find that the enhancing tumor and tumor core have high variation. The median and 25th quantile results indicate that our model did not perform well in certain test cases. Specifically, our model's performance was low on several test cases, particularly cases 213, 252, and 1721. The cause of these discrepancies still needs to be investigated.

Table 2.1: Dice score and Hausdorff distance on BraTS 2021 validation dataset. ET,TC,WT present enhancing tumor, tumor core, whole tumor respectively.

	Dice			Hausdorff95 (mm)		
validation dataset	ET	TC	WT	ET	TC	WT
mean	0.8194	0.7381	0.8915	16.6285	16.8743	5.7721
stdev	0.2439	0.2590	0.0987	69.6464	35.9145	7.5963
median	0.8915	0.8443	0.9220	1.4142	11.0000	4.0000
25quantile	0.8283	0.6793	0.8763	1.0000	6.8367	2.8284
75quantile	0.9465	0.9006	0.9475	2.9142	17.2612	6.0828

2.3.4 Test Phase Result

The performance of our model on the BraTS 2021 test data is shown in Table 2.2 and Table 2.3. Our model achieved DICE scores of 0.86, 0.73, and 0.58 for ET (enhancing tumor), WT (whole tumor), and TC (tumor core), respectively. Compared with the validation phase, the performance for WT decreased significantly. We will work on improving the performance for WT in future work. Additionally, we observed that the standard deviation for the whole tumor was higher than in the validation results. The 25th quantile results indicate that our model performed poorly in the tumor core. However, from the median and 75th quantile results, our model showed

good performance in most test cases. The overall performance was affected by some specific test cases. We will focus on those problematic test cases in future work.

Table 2.2: Dice score and Hausdorff distance on BraTS 2021 Test dataset.ET,TC,WT present enhancing tumor, tumor core, whole tumor respectively.

	Dice			Hausdorff95 (mm)		
validation dataset	ET	WT	TC	ET	WT	TC
mean	0.8553	0.7302	0.5836	16.3118	39.0873	64.1392
stdev	0.2113	0.3096	0.3552	70.1391	57.5957	82.7809
median	0.9279	0.8958	0.7630	1.0000	4.6904	24.9198
25quantile	0.8463	0.5978	0.1948	1.0000	2.4495	12.3915
75quantile	0.9644	0.9416	0.8866	2.2361	57.8802	90.7419

Table 2.3: Dice score and Hausdorff distance on BraTS 2021 Test dataset.ET,TC,WT present enhancing tumor, tumor core, whole tumor respectively.

Network	Dice			Hausdorff95 (mm)		
	ET	WT	TC	ET	WT	TC
3D Unet	0.7633	**0.7350**	0.8266	32.6349	17.4926	25.3428
Residual 3D Unet	0.2113	0.3096	0.3552	70.1391	57.5957	82.7809
3D Vnet	0.7723	0.6245	0.7899	22.6335	37.4873	24.7483
TransBTS	0.8098	0.7211	0.8615	16.6285	16.8743	5.7721
Our Approach	**0.8194**	0.7381	**0.8915**	2.2361	17.8802	6.07419

2.4 Related Work

Brain tumor segmentation is a growing field of interest for researchers as it aims to automate previously used tedious manual segmentation and yield higher specificity and sensitivity. In literature, U-Net [8], an encoder-decoder based model, has served as a great baseline architecture to attain low level details and provide good performance on brain segmentation tasks. Its variants U-Net++ [11] and Res-UNet [10] have also been successful to further improve the performance. However, all these convolutional neural network (CNNs) based models were employed to perform segmentation for 2D data. Given the 3D nature of MRI brain scans, it becomes time-consuming process to perform segmentation channel by channel. In light of this, U-Net has been adapted to cater the volumetric brain data to perform segmentation [64], [65]. In spite of its sophistication to extract low level details, it still suffered from capturing

long-range dependency.

On the other hand, attention based architecture, vision transformer have achieved state of the art performance in classification tasks and have shown potential in capturing long range dependency [16]. TransUnet [14] is a recent network in this direction that combines UNet model as a local features extractor with transformer model to gather global level information. However, it still processes images on slice by slice fashion and focuses on retaining spatial correlation between image patches via transformer. Swin-Unet [20] is another network that combines two models, Unet and swin transformer, to enhance the performance of the segmentation model. Unfortunately it only supports 2D MRI images. To overcome this challenge TransBTS [4] combines UNet with transformer but process all slices simultaneously, and thus captures global information in a better way.

Our work exploits a different direction than these works. Instead of merging two different models in one, we consider each model as an expert. We pick four good representative models of 3D brain segmentation tasks and ensemble them using the Ensembles of Multiple Models and Architectures technique. Although this technique has been explored before for brain segmentation in the work [51], our work exploits it with advanced 3D models which are completely different and more challenging than the original work.

2.5 Discussion

In this work, we introduced a new ensemble model that takes advantage of several sub-models to achieve more promising segmentation results on multi-modal 3D MRI. In the validation phase, we obtained good average results with our model. By combining the sub-models, we achieved better segmentation outputs than with each single model. We also introduced a new method to integrate the segmentation results from several models, yielding a more robust output.

However, there are several aspects we can improve upon following the BraTS 2021

challenge. In upcoming work, we can apply random cropping, flipping, and intensity adjustments to all our sub-models. In the preprocessing of data, the 3D Unet uses a resizing method, which results in the loss of a lot of pixel values during training. A poorly performing sub-model has limited contribution to the overall model. We will add more data augmentation techniques such as affine image transforms, random image rotations, and so on. We also aim to incorporate more data post-processing methods to generate a more stable and robust segmentation model. Additionally, we will apply data parallel distributed training methods or federated learning methods to speed up the training period. We will work on the test cases that exhibited significantly poor performance and investigate the causes of these performances. Furthermore, we will apply mixed-precision training to the code to accelerate the training phase. We will also attempt to train the entire network as one large network instead of training them separately.

To summarize, we achieved median DICE scores of 0.93, 0.90, and 0.76 on ET (enhancing tumor), WT (whole tumor), and TC (tumor core), respectively. We aim to further improve the aforementioned approaches for a better model in next year's challenge.

CHAPTER 3: Stacking Feature Maps of Multi-Scaled Medical Images in U-Net for 3D Head and Neck Tumor Segmentation

3.1 Introduction

Head and Neck (H&N) cancer is one of the most common cancers, affecting several areas of the throat, nose, and other head regions, excluding the brain and eyes. It is estimated that 277,597 people worldwide died from H&N cancer [66]. The 5-year survival rate for H&N cancer is around 90% if detected at stage 1 [67], which significantly reduces to 70% in stage 2. The survival rate further drops to 60% and 30% for stages 3 and 4, respectively [67]. Early diagnosis and treatment of H&N cancer can improve patient survival rates. Medical imaging techniques such as positron emission tomography (PET) and computed tomography (CT) have shown great value in localizing the primary tumor and assisting physicians with tumor contouring. However, manually contouring H&N tumors slide by slide is inefficient for physicians, and the distinctive sizes, types, and shapes of tumors make it challenging to define a uniform pattern [42].

An auto-segmentation method can be a feasible solution to the problems mentioned above. Traditional segmentation methods such as the threshold method [68], region-based method [69], and edge-based method [70] have their limitations and finding the best threshold for tumors is difficult. Recently, deep learning-based auto-segmentation methods have gained more attention for their great potential in computer vision tasks. In the medical domain, many studies have been conducted using deep learning (DL) methods. For example, Wang [49] proposed a triple cascaded framework that hierarchically segments three different types of brain tumors. EMMA [50] introduced a novel ensemble of DeepMedic [51], FCN [52], and U-Net [8] models,

achieving better segmentation results for brain tumors.

However, there are limited studies on H&N tumor segmentation until the rise of the Head and Neck Tumor Segmentation Challenge [71]. The Head and Neck Tumor Segmentation Challenge 2022 (HECKTOR) provides a well-labeled H&N tumor dataset for competitors to identify the best segmentation method for H&N tumors [72]. It offers a platform for H&N tumor segmentation and a high-quality tumor dataset. Additionally, it showcases various state-of-the-art H&N tumor segmentation algorithms and demonstrates the potential of deep learning in H&N tumor segmentation.

Recent post-challenge proceedings in the HECKTOR challenge show that most methods are implemented on U-Net [8]. Our network was inspired by U-Net with some modifications. In our work, we propose a 3D 'U'-shaped network architecture that takes multi-scaled PET/CT images as input for H&N tumor segmentation and concatenates their feature maps for deconvolution.

3.2 Design of the Stacked Feature Network

In this section, we will first introduce the pre-processing of raw HECKTOR challenge data and the augmentation techniques used. As mentioned in the previous section, we used stacked multiple resolution input images for training. We will also provide detailed splitting methods, training details, and the loss function used.

3.2.1 Data Preprocessing

Based on past challenge experiences, the dataset has no distribution shifts among different medical institutions. Since we did not perform any n-fold cross-validation metrics in this challenge, we did not split the training dataset. Thus, we trained with the whole dataset without any train-validation splits.

The PET/CT images from the HECKTOR challenge vary in size and are not registered. Therefore, we first applied a resampling script to resample all the input PET/CT images and well-labeled ground truth into a registered form. The script

we used was adapted from the official HECKTOR challenge GitHub website. The resample space is $2.0 \times 2.0 \times 2.0$. The directions and sizes of resampled images were determined by the pair of original PET/CT images, meaning that after registration, each training case has a different size in dimension. The interpolation for each PET/CT image is sitkBSpline from the SimpleITK library [73].

Unlike past challenges, HECKTOR 2022 did not provide any bounding boxes. Therefore, we applied a random crop on the resampled dataset and ensured all images were in the dimension of (144, 144, 144). If the dimension of the resampled image was smaller than (144, 144, 144), we applied zero-padding to make sure its dimension was (144, 144, 144). As mentioned in the previous section, we used multi-scale images as input, which are multi-resolution input images. We resampled these random cropped images into the dimensions of (72, 72, 72), (36, 36, 36), and (18, 18, 18).

3.2.2 Details of Stacked Multi-scale 'U' Shape Network

The overall network architecture is shown in Fig.3.1. It was implemented based on the U-Net. It has an encoder, a decoder part, a bottom layer, and feature maps from multi-scale input images. Since most medical images are 3-dimensional (3D), we first extended the original U-Net [8] into 3D. We did not add any blocks such as residual blocks, dense blocks, self-attention, and so on, to maintain the simplicity of our network. The encoder part consisted of three different resolution inputs. These inputs were registered in the same origin, spacing, and direction. For low-resolution PET/CT input images, two successive convolution layers without pooling were used to extract the feature maps from different scale input images. At the top, full-size input images are fed into three down-sampling blocks. Each down-sampling block consists of two convolution layers with batch normalization, ReLU activation function, zero padding, and a pooling layer. The decoder part contains regular deconvolution layers. At the bottom, we copied the low-resolution feature map from the encoder to the decoder. In the decoder part, we concatenated the deconvolution layer output,

encoder part feature maps, and low-resolution input feature maps. Thus, our decoder part contains richer feature information.

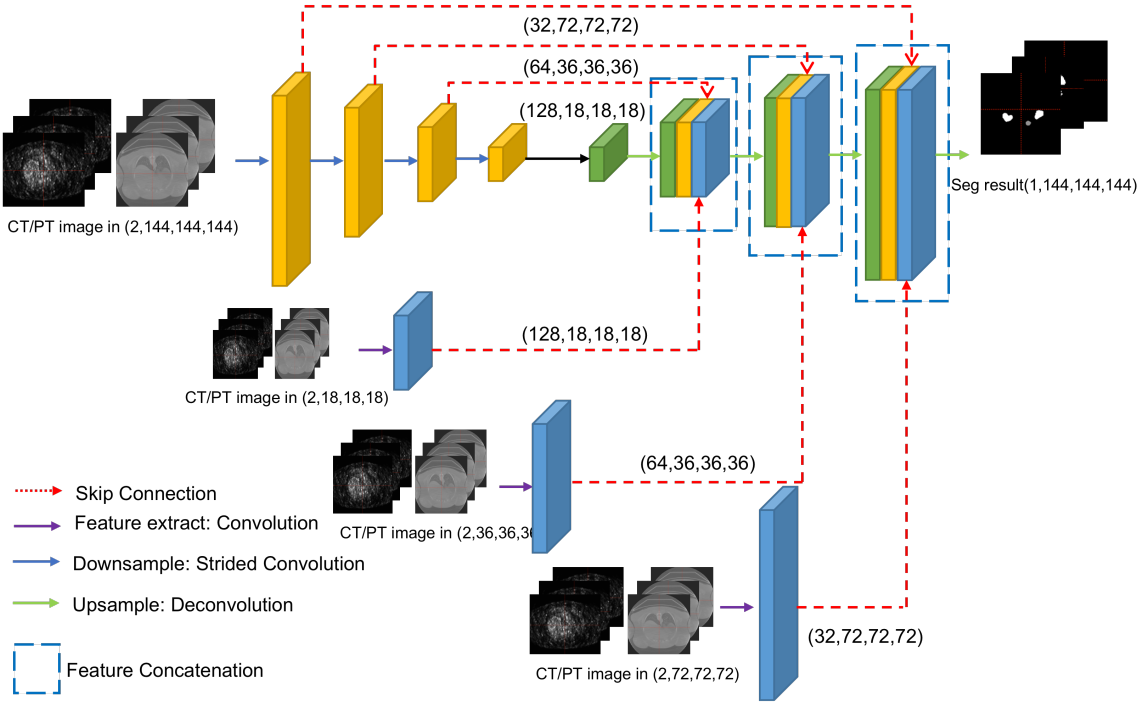


Figure 3.1: Stacked multi-scaled input 'U' Network Architecture. We used an input patch size of 144×144×144, 72×72×72, 36×36×36, and 18×18×18 with PET/CT as two modalities for the network. The network structure is essentially U-shaped architecture implemented based on U-net. The down-sampling is implemented with three down blocks, each with a strided 3D convolution operation with a 3×3×3 filter for each modality. The up-sampling is done with deconvolution. The size of the feature map is displayed in the figure. We directly copied the feature maps at the bottom layer. We concatenated different resolution input image feature maps with the deconvolution output. We also used skip connections to directly concatenate feature maps from the encoder part.

From the left side of Fig.3.1, the input consists of 2-channel 3D images which are CT and PET images. The first level of input image size is 144×144×144. In the encoder part, it is fed into a down-sampling block for feature extraction. The initial filter size for the convolution layer was set to 32. At the bottleneck, we mirrored the feature maps from the encoder to the decoder part. For low-resolution input at the second level, such as (72, 72, 72), it went through two convolution layers with an initial filter size of 32. When we have all three low-resolution images' feature maps, we

concatenate them with the same resolution output in the decoder part. The learning rate of our model was set to 2e−4 and it was reduced using the following formula:

$$Lr = Lr * \left(1 - \frac{e}{te}\right)^{0.9} \tag{3.1}$$

where e is the current epoch, and te is the total number of epochs. We used L2 norm regularization on the convolutional kernel parameters with a weight of 1e−5. We ran the training for a total of 2000 epochs.

3.2.3 Optimization and Data Augmentation

nnUnet[74] provides some helpful guidelines for medical image segmentation. In this paper, we applied some data augmentations and optimizations according to the suggestions of nnUnet.

Since we did not use the n-fold cross-validation technique, we did not have any validation set. Thus, we did not employ an early stopping mechanism or ensembling strategies.

We increased the batch size from 2 (as in the original Unet paper) to 16. According to the description of nnUnet, a lower batch size generates unnecessarily noisier gradients. These noisier gradients can reduce overfitting but decrease overall performance. Therefore, we used a higher batch size for better performance and less data copying between the host and the device.

We also applied several data augmentation techniques to obtain a more robust model from the training. In the pre-processing section, we had already applied random cropping and zero-padding for the input dataset. We also used Z-score normalization on the input PET/CT images. Additionally, we applied a random mirror flipping method. The flipping was performed in three dimensions across the axial, sagittal, and coronal planes. The random flipping rate was set to 0.5. We applied a random rotation to the training dataset at a ratio of 0.5. The rotation angle ranged from -10

to 10 degrees. We applied a random intensity shift at a ratio of 0.5. The intensity of the training data was shifted between -0.1 and 0.1. The scale of the training data was set from 0.9 to 1.1.

We chose the Dice loss function as the evaluation metric during training. For the optimizer, we chose the Adam optimizer. We also used batch normalization instead of instance normalization. Batch normalization can reduce the performance difference between the training dataset and the testing dataset. By applying the previous data augmentation techniques, we made our best effort to reduce the domain gap between the training and testing data.

3.3 Evaluation

3.3.1 HECKTOR 2022 Datasets

The HECKTOR 2022 dataset contains 359 testing cases and 524 training cases. In the HECKTOR training dataset, each training case consists of a set of PET/CT images and a well-labeled ground truth image. All images are in NIfTI format. There are two types of tumors: H&N primary tumors (GTVp) and H&N nodal Gross Tumor Volumes (GTVn), which are labeled as 1 and 2, respectively. Our model was trained exclusively on the HECKTOR training dataset, without using any other public or private datasets.

In this work, we used four different resolutions of PET/CT images, as detailed in the previous section. All of these different-resolution images have two input channels. The ground truth labels have three classes: label 0 for the background, label 1 for GTVp tumor, and label 2 for GTVn. The test dataset consists of 319 cases with no ground truth provided. Since we did not use cross-validation, we can only evaluate the results on the test dataset. For the evaluation metrics, we used the Dice score provided by the HECKTOR 2022 challenge.

3.3.2 Implementation Details

Our model was run using PyTorch 1.9.0 with CUDA 11.1. The Python version used was 3.8.5, and the model was trained from scratch on a server with 4 NVIDIA A100 GPUs (40GB VRAM each). The total number of training epochs was 2000, and the batch size was set to 16. As mentioned in the previous section, we applied pre-processing techniques and data augmentations to the original training data.

For inference on the testing data, we used the sliding windows inference method with a window size set to (144, 144, 144). In the first step, we used the same pre-processing procedure as the training stage. Then, we cropped the input data into small pieces with dimensions of (144, 144, 144), and applied the same resampling method to those small pieces into low resolution. If the resampled image dimension could not be cropped into an integer number of pieces, we extended the last piece to the dimension of (144, 144, 144). For example, if an input image had dimensions of (160, 160, 160), we would crop it into small pieces of numpy arrays $(:144, :144, :144), (:144, :144, 16:160), (:144, 16:160, :144), (16:160, :144, :144), (:144, 16:160, 16:160), (16:160, :144, 16:160), (16:160, 16:160, 16:160)$. We then concatenated the results together.

For the challenge, we also applied test time augmentation, which involved applying augmentations to different batches of test data and merging predictions during the inference stage.

3.3.3 HECKTOR 2022 Test Result

In terms of the challenge, we have this section specifically dedicated to the competition results. Since we did not use a validation dataset, there are no validation results to report. The test performance was obtained from the official HECKTOR challenge website. As there is no additional information provided, we are unable to analyze our results in detail. The performance of our two submissions on the HECKTOR 2022

testing data is reported in Table 3.1.

Table 3.1: Dice score and on HECKTOR 2022 validation dataset. GTVp,GTVn present tumors H&N primary tumors and H&N nodal Gross Tumor Volumes respectively.

	Dice		
Test Dataset	GTVp	GTVn	Mean
First Submission	0.69786	0.66730	0.68258
Second Submission	0.68610	0.66482	0.67546

3.3.4 Qualitative Results

We randomly selected 100 training cases as a validation dataset. The validation cases were chosen based on their Dice scores, specifically selecting the best, worst, mean, median, and the 75th and 25th percentiles. The results are shown in Fig. 3.2. The best case is CHUS-094 with a mean Dice score of 0.960. The 75th percentile case is CHUS-040 with a mean Dice score of 0.880. The mean case is CHUM-015 with a mean Dice score of 0.678. The median case is MDA-185 with a mean Dice score of 0.778. The 25th percentile case is MDA-180 with a mean Dice score of 0.53. The worst case is CHUS-028 with a mean Dice score of 0.223.

Table 3.2: Dice scores on the HECKTOR 2022 validation dataset. GTVp and GTVn represent H&N primary tumors and H&N nodal Gross Tumor Volumes, respectively.

	Dice
Validation	Mean
Best	0.960
75th quantile	0.880
mean	0.678
median	0.778
25th quantile	0.53
worst	0.223

3.4 Discussion

In this paper, we introduced a new Stacked Multi-Scale 3D PET/CT input image model for a 'U' Shape Network to achieve more promising segmentation results on

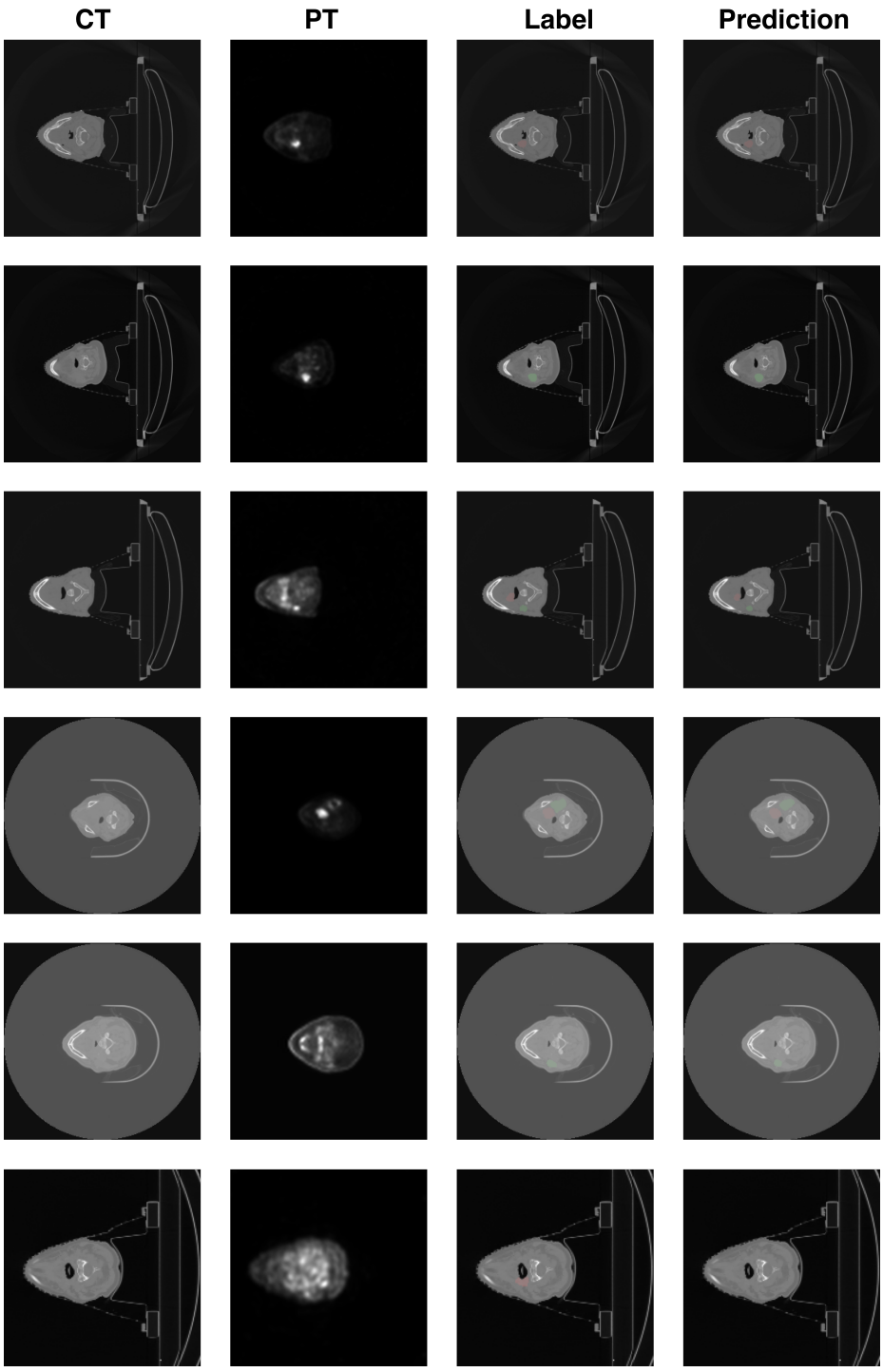


Figure 3.2: Visualization of Qualitative Results. For each row, the PET image is shown in the first left column. The second left column displays the CT image. The label is next to the CT image. The predicted outcome is in the last right column. GTVp is shown in green, and GTVn in red. From the first row to the last row, we displayed the best, 75th percentile, mean, median, 25th percentile, and worst validation case, respectively.

H&N tumors. In the testing phase, we achieved overall good results. However, there are still many areas where we can improve in the future. For instance, we can use batch dice rather than an instant mini dice. In the current method, we evaluate the dice loss in every mini-batch, which is considered an instant mini dice. In the future, we will use dice loss for the whole dataset, referred to as batch dice. By performing batch dice, we can consider the entire dataset as a large sample trained in one batch. This presents a trade-off between bias and variance.

We can also improve the results by using cross-fold validation techniques. By doing so, we can keep and record the best weights and evaluate the model during training. After training, we can also ensemble those models into a better-performing model.

Overall, we achieved mean dice scores of 0.69786 and 0.66730 for GTVp and GTVn H&N tumors, respectively. We will address the proposed improvements in the next challenge.

CHAPTER 4: SMoE-MLP: 3D Medical Image Segmentation with Sparse MoEs-based Multiple Layer Perceptron of Vision Transformer

4.1 Introduction

Deep learning has established itself as the most effective technique for a broad variety of tasks across multiple types of data, especially in natural language processing and computer vision. Historically it has been demonstrated that increasing network complexity and dataset quantity will generally improve performance [75, 76], and large models pre-trained on large datasets currently hold state-of-the-art performance in both computer vision and in natural language processing [31]. Despite their high performance, training and deploying such large-scale models present a number of challenges because the computational costs scale poorly due to the dense nature of these networks, e.g., the largest models could top 100B parameters [77, 31]. Therefore, there is a need for techniques to reduce computational costs while maintaining performance, especially for resource-constrained environments. Additionally, not all application areas have large, well-annotated datasets readily available for training such models. One area where the emphasis on large models pre-trained on large datasets presents particular difficulties is medical image analysis, where training data are limited and expensive to annotate and deployment of deep learning models in clinical settings requires efficient training and inference using often limited computational resources.

For medical image segmentation, most state-of-the-art methods are convolutional neural network (CNN) models utilizing the “U-shaped" encoder-decoder architecture pioneered by UNet [8] and subsequent variants [12]. Despite the powerful representational ability of such U-shaped CNNs, the reliance on local receptive fields leads to a

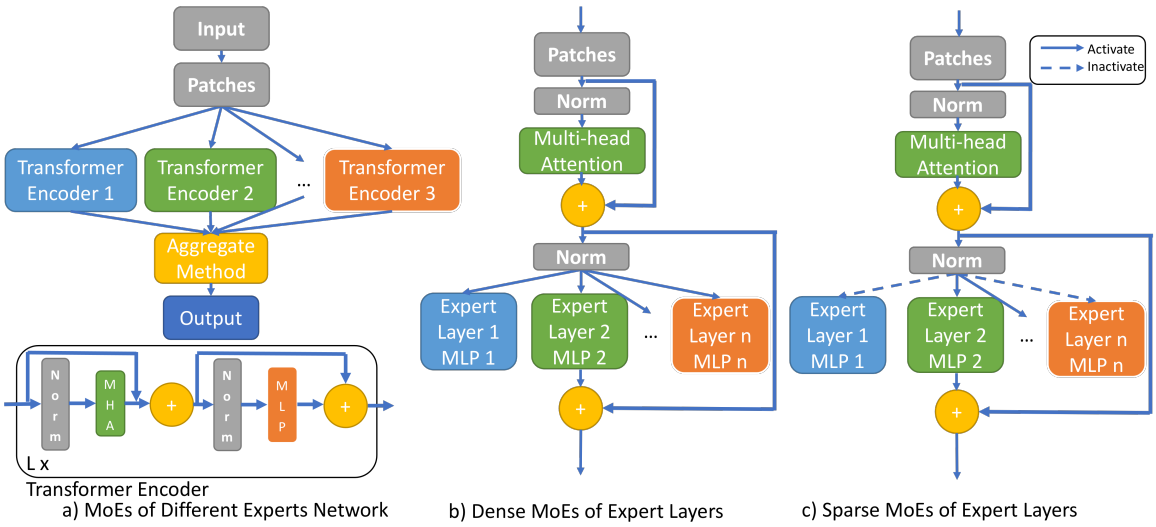


Figure 4.1: Different Types of Mixture of Experts: a) The traditional type combines the output from each expert, which comprises different network structures, to produce a final result; b) In the dense type, each expert forms one layer in the same network structure, performing layer-level aggregation facilitated by a gating network; c) The sparse type involves activating specific expert layers instead of all of them.

deficiency in capturing global semantic information and long-range dependencies in images [78, 79]. Transformer models for computer vision attempt to alleviate such shortcomings by utilizing multi-headed self-attention (MSA) in place of convolutional kernels. However, vanilla transformers compute attention scores between all pairs of tokens—small image patches—leading to an $\mathcal{O}(n^2)$ complexity, posing computational challenges for training models for 3D medical images.

To counteract the growing resource demand of ever-expanding deep neural networks, conditional computation [21] aims to reduce computational costs associated with large, highly-deep models while preserving model representational capacity. Conditional computation applies only a subset of parameters to each example during training and inference, maintaining a relatively constant computational cost. One of the methods is Sparsely-gated Mixture-of-Expert networks (SMoEs) [23], which have been explored for large transformer-based language models [24, 25] and 2D image classification models based on Vision Transformer (ViT) [16, 18, 17]. Fig. 4.1, showcases three diverse types of Mixture of Experts (MoEs): a traditional MoEs network,

a dense MoEs layer, and a sparse MoEs layer. In SMoEs, dense feedforward layers are replaced by a layer of sparse experts, and each input is routed to a particular subset of experts. Such routing procedures come with their own suite of challenges though due to the non-differentiable nature of the process. Riquelme et al. [31] have addressed many of these difficulties in the context of 2D image classification. Still, limited research has been done on adapting SMoEs for image segmentation tasks, particularly in challenging domains like medical image analysis. The routing function and design of the gating network remain open questions for models designed for 3D medical image segmentation [31].

In this work, we designed and implemented a novel architecture named Sparse MoEs-based Multiple Layer Perceptron of Vision Transformer(SMoE-MLP) which extended the U-shaped ViT architecture for 3D medical image segmentation and integrated it with SMoEs. To the best of our knowledge, this represents the first effort to combine SMoEs with ViT models for 3D medical image segmentation. In the SMoE-MLP network, we improved the routing and gating method based on Top-K routing [31] and adapted the SMoE layer to image segmentation. Our experiments demonstrated that the models trained with the SMoE-MLP network achieved strong performance on the tasks of brain tumor segmentation (BraTS) and head & neck tumor segmentation (Hecktor). Specifically, it performs on par with or better than other models such as TransBTS, nnUnet, Uneter, etc., in the segmentation tasks of whole tumor, tumor core, and gross tumor volume of the primary(GTVp). We implemented conditional computation in the SMoE-MLP network to reduce the original ViT-MoE model computation cost by sparsely activating parts of experts while maintaining the same level of performance. SMoE-MLP outperformed the original ViT-MoE model with extra fine-tuning work. Compared to densely connected ViT-MoE, our proposed SMoE-MLP approach accelerates the training process by 1.37 and 1.83 times on BraTS 2021 [3] and Hecktor 2022 [72], respectively. In addition,

SMoE-MLP speeds up the inference process by 2 and 1.8 times on BraTS 2021 and Hecktor 2022, respectively.

4.2 Related Work

CNNs, UNet, and their variants. Convolutional neural networks have been the primary model architecture for computer vision tasks for a number of years. For image segmentation, the pioneering work of UNet [8] introduced the successful "U-shaped" paradigm for medical image segmentation models. Subsequent variants, such as V-Net [9], Residual UNet [10], UNet++ [11], and nnUNet [12] have modified and improved CNN-based U-shaped networks and continue to demonstrate competitive performance on many image segmentation tasks.

Transformers to complement CNNs. Transformer models employ the multi-headed self-attention (MSA) methods to computes dynamic aggregation weights between pairs of tokens. This operation mimics the convolutional filters of CNNs but with the added benefit that weights are data-dependent rather than fixed. By stacking multiple attention heads, individual attention mechanisms can specialize to different aspects, akin to multiple convolutional filters per layer. MSA allows for extraction of longer-range dependencies by computing attention scores over all pairs of tokens; though, this comes with significant memory overhead [13]. As such, researchers have used self-attention as a complement to CNNs by replacing certain layers with transformer blocks. In the area of medical image segmentation, such hybrid approaches have been developed for multi-modal tumor segmentation [14] and brain tumor segmentation from 3D image data [4] and CT images [15]. In contrast to these hybrid approaches, our proposed framework is constructed from a pure transformer basis.

Vision Transformer Vision Transformer (ViT) [16], and subsequent variants [17, 18], adapt the transformer architecture from NLP for computer vision. ViT splits images into fixed-sized patches representing tokens and combines the linearized patches with positional embeddings. The tokenized images are then passed through an

encoder consisting of a series of transformer blocks. Swin-Transformer [19] introduced localized self-attention using shifted-windows to improve costs associated with MSA. Swin-UNet combined the well-established U-shaped encoder-decoder architecture with transformers for medical image segmentation [20]. Swin-Unetr extended the UNetr with swin block [80].

Mixture of Experts Mixture of Experts (MoE) is an architecture for realizing conditional computation in neural networks. In MoEs, the model consists of a set of expert networks that are conditionally activated on the basis of a gating or routing network [21, 22]. SMoEs employing a Top-K gating algorithm were introduced in the context of ensemble LSTM models for natural language processing [23], GShard [24], and Switch Transformers [25] developed sparse MoE layers for transformer language models.

MoEs in Computer Vision In computer vision, sparse MoE layers have been shown to have a broad range of potential benefits, including integrating large ensembles of experts, reducing consequences of data imbalance, and improving model efficiency—in terms of model size and inference costs [30, 27]. Ahmed et al. [27] and Condconv [30] used a shallow MoE with a single routing network for image classification. In the area of medical image segmentation, Yanglan et al. [81] utilized MoE as a decoder for stroke lesion segmentation, though this work and other existing frameworks that employ a MoE still largely rely on dense implementations [33, 34, 35]. The use of SMoE for 3D medical image segmentation remains an open area of research, and our work represents a novel application of SMoEs with vision transformer architectures.

4.3 Design of the SMoE-MLP Framework

Our SMoE-MLP framework was inspired by previous research works, including 'unter'[15], swin-unter[1], and VMoE [31]. Compared with previous approaches, our SMoE-MLP combines the power of a transformer 'U'-shaped network and a sparse mixture of experts. Unlike existing methods, we have also enhanced the gating

method for both the training and inference stages to reduce computational costs and enhance overall efficiency in training and inference. As illustrated in Fig. 4.1, the sparse MoEs do not reduce the size of our model but rather decrease the number of active parameters during training. Our SMoE-MLP further enhances this by sparsely activating some experts during training and updating their weights and gating during backward propagation. The details of our network will be introduced in the following subsection.

4.3.1 U-shape Architecture

An overview of our proposed SMoE-MLP network is presented in Fig. 4.2. The SMoE-MLP network is composed of three main components: an encoder, a bottleneck, and a decoder. The encoder and decoder are composed of a series of ViT-MoE blocks. For a given input MRI image $X \in R^{M \times H \times W \times D}$, where M represents the channels or modalities of the image, and H, W, and D represent the height, width, and depth respectively. The image is split into non-overlapping patches of size $2 \times 2 \times 2$ as individual 3D image tokens. For an image with C channels, each token has an initial feature dimension of $2 \times 2 \times 2 \times 4$. A linear embedding layer is applied on this raw-valued feature to project it to an arbitrary dimension (denoted as C). Each image token is then passed through the encoder consisting of ViT-MoE blocks and transformed to a latent feature representation. After each layer, patch merging is used to downsample the image and learn hierarchical feature representations. At the bottom of the encoder, a bottleneck layer consolidates the feature representations before passing them to the decoder, which consists of a mirrored set of ViT-MoE blocks coupled with patch-expanding layers to transform the latent feature representation to the original image resolution. A patch-expanding layer is used to linear expand the resolution of the input features to match the same dimensions as the corresponding encoder layer. Parallel skip connections between encoder and decoder layers are employed to preserve information learned at different resolutions and avoid

gradient decay. A final linear projection layer transforms the feature representation to a voxel-wise segmentation mask.

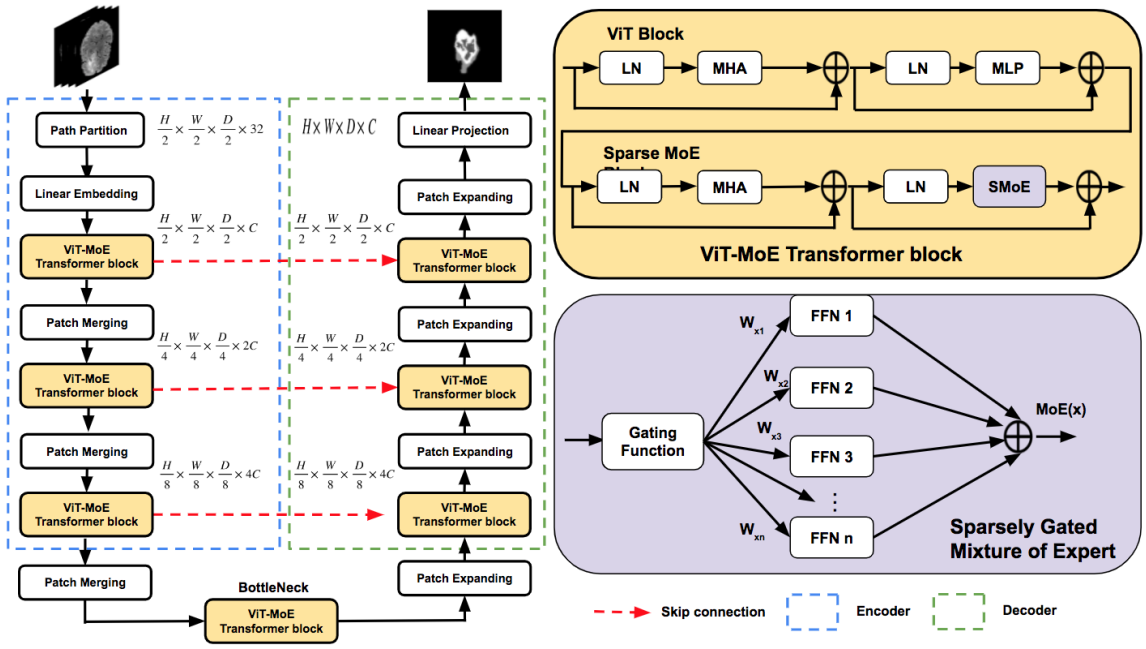


Figure 4.2: Overall Architecture of the Proposed SMoE-MLP: the right top of figure is a modified transformer block, we replaced the second MLP with experts. The experts has same structure as MLP. For each training round, we dispatch the input image and embedding it, then send it to the modified transformer block. The first part of the block remains the same, but the expert is controlled by the gating network with proposed algorithm.

4.3.2 ViT-MoE Transformer Block

The ViT-MoE block consists of two components: a ViT block and an SMoE block. The ViT block is a regular transformer block with Layer Normalization, multi-headed self-attention, residual connection, and multi-layer perceptron. The Sparse MoE block has a structure similar to that of the ViT block, except that its dense MLP is substituted with a sparsely activated set of MLPs where each MLP represents an expert in a mixture of experts model. The ViT block is a regular transformer block and it is computed as:

$$
\begin{aligned}
h'_l(x) &= MHA\left(LayerNorm\left(h_{l-1}(x)\right)\right) + h_{l-1}(x) \\
h_l(x) &= MLP\left(LayerNorm\left(h'_l(x)\right)\right) + h'_l(x) \\
h'_{l+1}(x) &= MHA\left(LayerNorm\left(h_l(x)\right)\right) + h_l(x) \\
h_{l+1}(x) &= MoE\left(LayerNorm\left(h'_l(x)\right)\right) + h'_l(x)
\end{aligned}
\tag{4.1}
$$

Where $h_l(x)$, $h_{l+1}(x)$ represent the output feature for the ViT block and MoE block on l layer and $l+1$ layer respectively. $h'_l(x)$, $h'_{l+1}(x)$ represent the intermediate results. MHA is a multiple-head attention mechanism to jointly learn from different representation subspaces at different positions.

Sparsely Gated MoE For a given transformer MoE layer with n experts, the feature representation is computed as:

$$
MoE(x) = \sum_{i=1}^{k} g\left(x\right)_i e_i\left(x\right) \tag{4.2}
$$

where x represents the input token representation, $e_i(x)$ is the feedforward neural network output generated by expert i, and $g\left(x\right)_i$ is the gating function that credits each expert i. $g\left(x\right)$ are learnable parameters. k denotes number of activated experts.

Gating Method To achieve a sparsely-gated MoE, we define a gating network method as following equation:

$$
g\left(x\right)_i = \begin{cases} Random_{k_{train}}\left(SoftMax\left(\omega_i * x + \omega_{Noise}\right)\right) \\ \qquad\qquad\qquad\qquad if\ training \\ Top_{k_{infer}}\left(SoftMax\left(\omega_i * x\right)\right) \\ \qquad\qquad\qquad\qquad if\ inference \end{cases} \tag{4.3}
$$

where ω_g is an expert-specific weight and ω_{Noise} is a noise weight sampled from a normal distribution $\omega_{Noise} \sim (0, \frac{1}{E^2})$. The noise weight is added during the training

process to increase the stability of the network.

During the training, k_{train} experts are randomly selected for each token. The value of k_{train} should be smaller than the total experts number n, but larger than k_{infer}. By applying the random expert selection method during training state, the bias of selected experts is reduced. For the training stage, $g(x)_{k_{train}}$, which is a subset of the total experts pool $g(x)_n$, was selected to achieve sparse training that only trains a partial network. In contrast, a regular top-K method is applied to ensure that the network is sparsely activated.

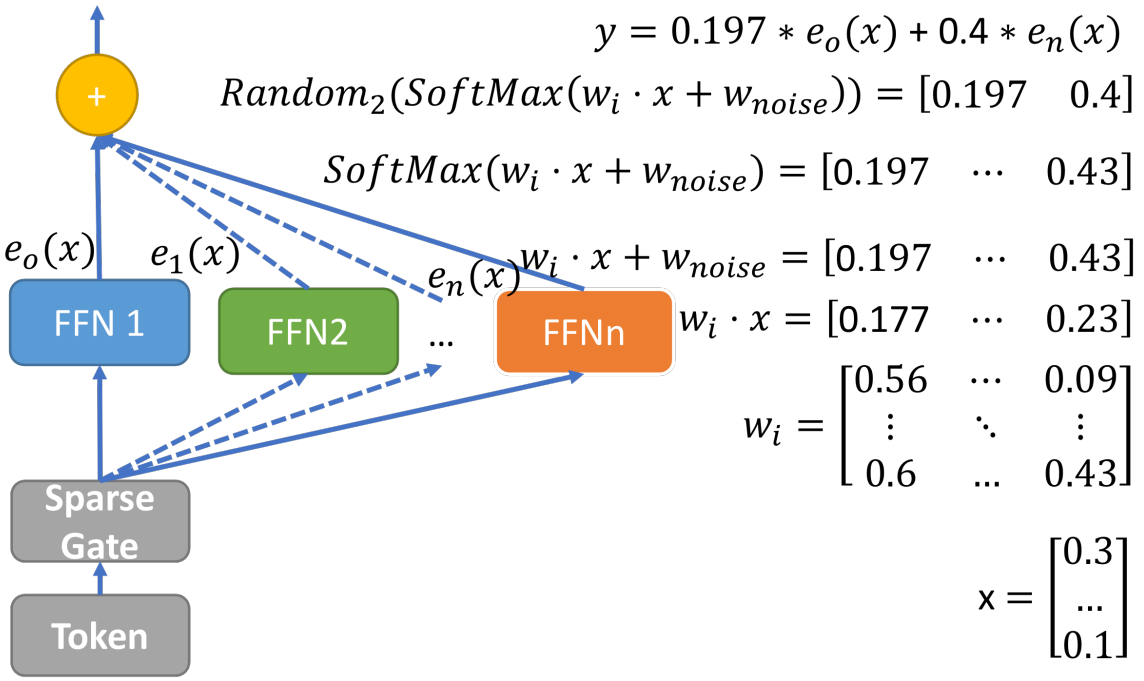


Figure 4.3: Workflow of Sparsely Gated Mixture of Experts (MoEs) Block: the input, x, is taken from the previous layer. It passes through a feed-forward network (FFN) with different experts, denoted as e_i. The computation involves a position-wise feed-forward network. Subsequently, it undergoes a gating weight matrix multiplication. During the training stage, we introduce noise and apply softmax to ensure the sum of the results equals one. After randomly selecting K expert's results, y is generated by the linearly weighted combination of each expert's output on the token, guided by the gate's output.

An example workflow of an expert network is demonstrated in Fig. 4.3. In the sparsely gated Mixture of Experts block, there are n different experts $e_0, e_1, ..., e_n$

with the same network structures but holding their own weights. These experts increase the capacity of our network.

The input, x, is the output from the previous layer. Each expert takes x as a token and computes its own output $e_i(x)$. On the right side of Fig. 4.3, the expert network takes the input token x:[0.3,...,0.1] and produces the output of each individual expert on x, such as $e_0(x), e_1(x), ..., e_n(x)$.

The sparse gating network takes the input of each expert and computes it with a dot product using a gating matrix ω_g, which considers the credits or possibilities of each expert. ω_g is an m $\times$ n matrix, where m is the size of the input x feature dimension, and n is the number of experts. We perform a dot product on x and ω_g to calculate the similarity between the input token and the experts. The result, [0.177, ..., 0.23], indicates that the input prefers $e_n > ... > e_0$.

During the training phase, we add noise, as mentioned before, to increase the robustness of our model. After adding the noise, ω_g changes to [0.197, ..., 0.43]. A softmax function is applied to ensure the sum result is set to 1. Then, we randomly pick k (set by the user) experts to perform the final calculation. In this workflow, we randomly pick 2 experts, 0 and n. Thus, the output of the MoE block is $y = 0.197 * e_0(x) + 0.4 * e_n(x)$.

4.4 Evaluation

Data and Evaluation Metric. We evaluate our SMoE-MLP model on the Brain Tumor Segmentation (BraTS) 2021 [3] and Head & Neck Tumor Segmentation (Hecktor) 2022 challenge data [72]. Since there are no BraTS 2022 challenges, we continue to use BraTS 2021 for evaluation.

The BraTS 2021 dataset comprises 1,251 annotated cases as training data and 219 unlabeled cases as the validation set. Each case consists of a set of four mpMRI images using four different volumetric representations, including T1-weighted, T2-weighted, T1 contrast-enhanced, and T2 Fluid Attenuated Inversion Recovery (FLAIR). Each

3D MRI image has dimensions of $240 \times 240 \times 155$. The ground truth labels consist of one benign class and three tumor classes, representing the necrotic tumor core (NCR), peritumoral edematous/invaded tissue (ED), and GD-enhancing (enhanced) tumor (ET). The combined NCR and ET represent the tumor core (TC), while all three tumor classes combined represent the whole tumor (WT) [42].

The Hecktor 2022 dataset contains 359 testing cases and 524 training cases. In the Hecktor training dataset, each case consists of a set of PET/CT images and a well-labeled ground truth image. There are two types of tumors: H&N primary tumors (GTVp) and H&N nodal Gross Tumor Volumes (GTVn), labeled as 1 and 2, respectively [43].

For the BraTS 2021 dataset, we use Dice score and Hausdorff distance metrics to evaluate image segmentation performance. For the Hecktor 2022 dataset, we only use the Dice score since the official Hecktor dataset doesn't provide Hausdorff distance evaluation. The Dice score measures the overlap of two images, with higher values indicating better performance. Conversely, Hausdorff distance evaluates the contour distance between two images, with lower values being preferable.

Table 4.1: Dice score and Hausdorff distance on BraTS 2021 validation dataset and Hecktor 2022 test dataset. ET, TC, WT represent enhancing tumor, tumor core, and whole tumor respectively. GTVp, GTVn present tumors H&N primary tumors and H&N nodal Gross Tumor Volumes respectively. The results of Unetr, Swin-Unet are from the paper [1] without Hausdorff distance. The result of Swin-Unetr was reported in its own paper. We trained, adopted, and evaluated the rest of the network if they did not present the result in the original paper.

Network	Computation Parameter(M)	BraTS 2021						Hecktor 2022	
		Dice			Hausdorff95 (mm)			Dice	
		TC	ET	WT	TC	ET	WT	GTVp	GTVn
3D-Unet [56]	5	0.7633	0.8266	0.7350	32.6349	17.4926	25.3428	0.69786	0.66730
TransBTS [4]	32	0.8194	0.7381	0.8915	16.6285	16.8743	5.7721	0.68610	0.66482
nnUnet [12]	25	0.8731	0.8443	0.9220	1.7823	11.0000	4.0000	0.77782	**0.77960**
TransUnet [14]	116	0.8176	0.8397	0.9149	4.91	4.20	2.93	0.65498	0.66292
Unetr [15]	102	0.8420	0.8530	0.9050	N/A	N/A	N/A	0.66271	0.65743
Swin-Unet [20]	33.7	0.8660	0.8340	0.9050	N/A	N/A	N/A	0.69054	0.67321
Swin-Unetr [15]	N/A	0.8850	**0.8580**	0.9260	5.831	6.016	3.770	0.71213	0.63644
ViT-MoE	27	0.8644	0.8363	0.9026	3.3166	**3.7417**	2.4495	0.76505	0.76032
SMoE-MLP	12	**0.8899**	0.8503	**0.9276**	**1.4142**	11.7983	**3.7417**	**0.77882**	0.77600

Implementation Details. The implementation of our network uses Python 3.8.5 and Pytorch 1.9.0. For each training case, we apply several data augmentations including linear normalization, random crop, random clip, and random intensity shift. Each image is cropped to a dimension of $128 \times 128 \times 128$ before being input to the segmentation model. The network is trained from scratch on four NVIDIA A100 GPUs (40GB VRAM) for 2000 epochs with a batch size of 16. That means we do not make any pre-train on other datasets. We use the Adam optimizer with Dice loss as the loss function with L2 regularization with $\lambda = 10^{-5}$. The initial learning rate is set to 0.0001 with a polynomial learning rate schedule with initial rate decay set at a power of 0.9.

Baselines To compare the performance of our proposed method, we made serval comparisons with the current representative methods such as 3D-Unet [56], TransBTS [4], nnUnet [12], TransUnet [14], Unetr [15], Swin-unet [1] and Swin-unetr [80]. We tried to get the result from the original paper if they used the same dataset for evaluation. If not we adopted their method to those two datasets without changing parameters or hyper-parameters in the original paper. We also have our implementation of ViT-MoE which replaces the MLP layers by 4 experts.

4.4.1 Experimental Results on BraTS 2021

Model Setting The model architecture we used in the BraTS 2021 consists of 3 ViT-MoE layers for the encoder block, 3 ViT-MoE layers for the decoder block, a hidden dimension set as 512, and the number of heads set as 8. For the SMoE block, we replaced the MLP with MLP experts (FFN with different parameters). The number of MLP experts was set by the user. In this experiment, we set it as 4. For the training stage, we randomly selected 2 experts to perform inference training for each iteration. During the inference stage, the top-k experts were set as 3 which is a top-3 method for inference.

Model Performance The image segmentation results on the BraTS 2021 valida-

tion set are reported in Table 4.1. Our SMoE-MLP model achieved a Dice score of 88.99%, 85.03%, 92.46%, and a Hausdorff Distance of 1.41mm, 11.79mm, 3.74mm for the ET, tumor core, and whole tumor, respectively. We compared our framework against 3D-UNet, TransBTS, nnUNet, and other transformer-based networks on the BraTS 2021 validation. Our SMoEs-MLP model outperformed 3D-UNet, TransBTS and nnUNet on all three segmentation tasks, demonstrating the strong potential of pure transformer-based architectures compared with CNN or hybrid approaches. Additionally, our model outperformed the transformer-based model such as ViT-MoE on TC, WT segmentation tasks and matches Swin-Unetr's performance on ET segmentation. Slightly reduced performance on ET segmentation, given the results on ET segmentation, suggests that our model could improve on identifying the necrotic core (NCR). For the Hausdorff distance, our model achieved best on the TC. Even though our model has a good dice score on ET but ViT-MoE model has the lowest ET Hausdorff distance. For the WT, our model outperformed all other models in distance. Overall, these results clearly demonstrate the potential of SMoE-MLP models for medical image segmentation. In addition, as a sparse model, our SMoE-MLP model achieved these results with a significantly smaller size of computation parameter.

4.4.2 Experimental Results on Hecktor 2022

Model Setting The model architecture we used in the hecktor 2022 is similar to the BraTS 2021. SMoE-MLP has 3 ViT-MoE for encoder-decoder layers, a hidden dimension set as 512 as well. The number of heads also set as 8. For the SMoE block, we replaced every other FFN with MLP experts to construct. The number of MLP experts was set as 4. For the training stage, we randomly selected 2 experts to perform training for each iteration. During the inference stage, the top-k experts were set as 3 which is a top-3 method for inference.

Model Performance The image segmentation results on the Hecktor 2022 test set

are reported in Table 4.1. Our SMoE-MLP model achieved a Dice score of 77.88%, 77.60% for GTVp, GTVn respectively. We compared our framework against the same network for BraTS 2021 task. Our SMoE-MLP model outperformed all based line models on GTVp, and we also outperformed the performance of all baseline models except nnUnet on GTVn. Consider two evaluations, that demonstrated the strong potential of pure transformer-based architectures compared with CNN or hybrid approaches. Besides, we still get a great performance with fewer computational parameters.

4.4.3 Ablation Study

4.4.3.1 Qualitative results

A qualitative visualization can be observed in Fig. 4.4. Since we do not have ground truth for the test dataset, we selected the cases from the training dataset for the visualization. The cases chosen are one from the Brats 2021 training set and one from the Hector 2022 training set. As shown in Fig. 4.4, SMoEs-MLP can capture enriched image features due to the cooperated work of each expert.

4.4.3.2 The Efficacy of the Number of Experts

As a mixture of experts model, the number of experts employed can be flexibly tuned to balance computational costs with performance benefits. In the results presented above, we utilized three experts per token for inference. We conducted tests with varying maximum numbers of experts and different numbers of experts during the inference for our SMOE-MLP model.

More Experts: In the previous section, we described setting the maximum number of experts on each MoE-MLP block as 4. In our subsequent experiments, we increased the maximum number of experts for the SMoE-MLP network, as illustrated in Fig. 4.5. The number of inference experts is set as one less than the maximum number of experts. As we increased the maximum number of experts, the Dice score showed

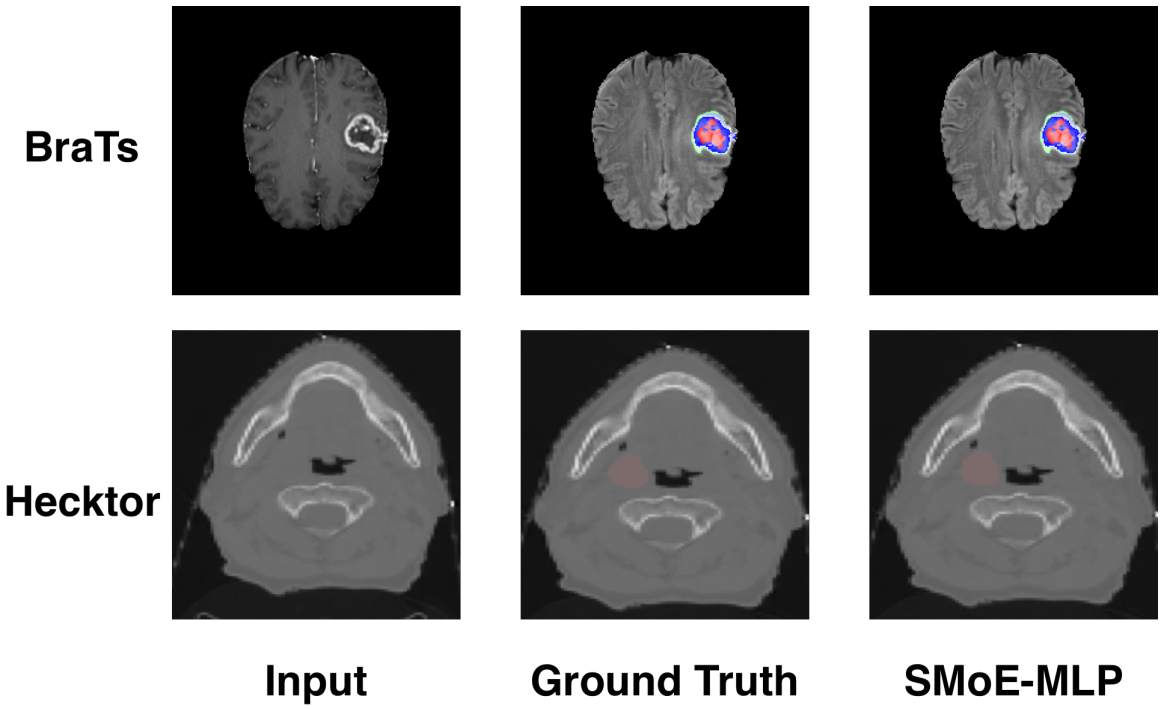
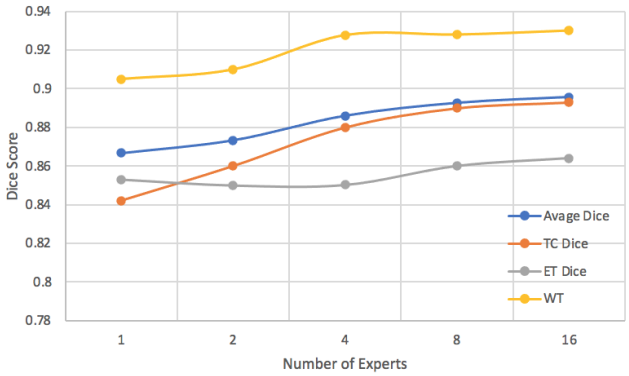
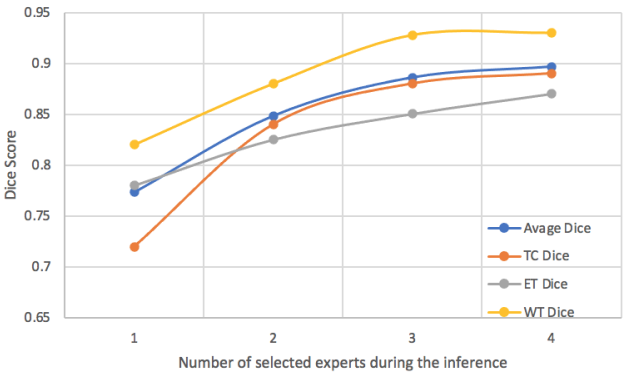


Figure 4.4: Visualization of Qualitative Results

a slight improvement. However, the number of parameters in our network increased dramatically, resulting in longer inference and training times. The trade-off between accuracy and efficiency appears to be too low. Therefore, we recommend using 4 as the maximum number of experts and 3 experts for inference.



(a) Dice score for varying numbers of maximum experts on the BraTS dataset

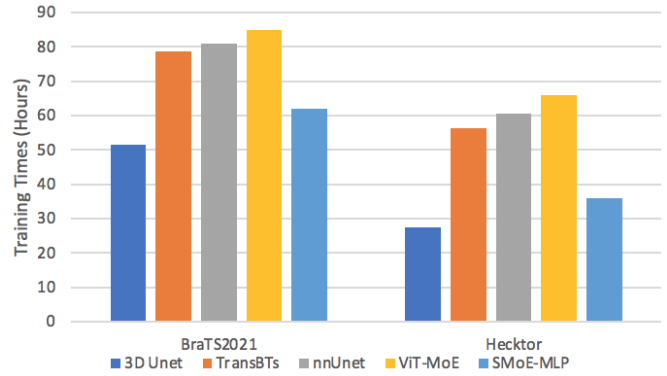


(b) Dice score for varying numbers of activated experts, with a maximum of 4 experts, on the BraTS dataset

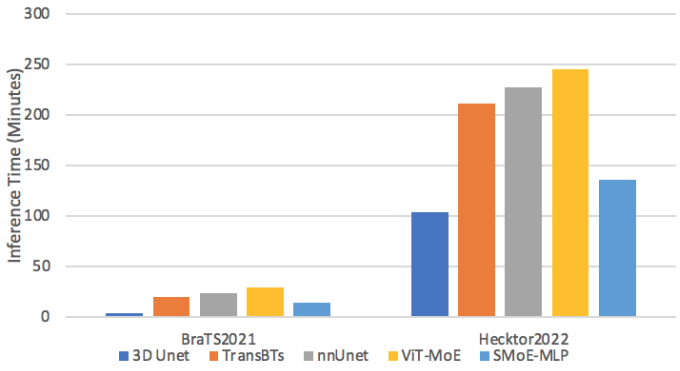
Figure 4.5: Efficacy of the Number of Experts

More Numbers of Activated Experts during Inference: In the previous section, we described setting the number of activated experts on each MoE-MLP block as 3. In subsequent experiments, we explored different numbers of experts for the SMoE-MLP network, as illustrated in Fig. 4.5. The maximum number of experts is still set at 4, with the number of inference experts set as one less than the maximum. As we increased the activated number of experts, the Dice score showed a slight improvement, similar to the results in the 'More Experts' section. However, similar challenges persist. The trade-off between accuracy and efficiency remains a major concern for now. Empirically, we recommend using 3 experts for inference when the

maximum number of experts is set at 4.



(a)



(b)

Figure 4.6: Training Time and Inference Time of Various Network Architectures.

4.4.3.3 Model Analysis

In order to evaluate the training and inference time of various machine learning networks, we implemented each network on the same hardware and input data. The training and inference time were then measured and recorded for each network. By comparing the training and inference times, we were able to identify which network had the fastest speed and which had the slowest.

The results of this evaluation can be used to inform the selection of a network for a specific application or to identify areas for optimization in the current networks. The training time of different networks on the BraTS 2021 validation and Hecktor 2022

test datasets is included in Fig. 4.6.

Our model used the setting of 4 maximum experts, with 2 randomly selected for the training phase and the top 3 for inference. Our analysis indicates that SMoE-MLP achieves 1.37 and 1.83 times faster training speeds than a ViT-MoE model on the BraTS and Hecktor datasets, respectively, through the use of sparsely active partial experts. In comparison with CNN-based methods such as TransBTs and nnUnet, SMoE-MLP shows up to 1.25 times faster training speeds. The longer training time for the Hecktor dataset can be attributed to the smaller size of the training data.

Moreover, we evaluated the inference time of the different networks, as depicted in Fig. 4.6. SMoE-MLP exhibits faster inference times than the ViT-MoE model, achieving a speedup of 2x and 1.8x on the BraTS 2021 and Hecktor 2022 datasets, respectively. However, it is worth noting that the pre-processing of the Hecktor test data takes longer than the actual model computation time, which may affect the overall performance evaluation.

4.5 Discussion, Limitation and Future Work

In this paper, we propose a novel Sparse Mixture of Experts-based Multiple Layer Perceptron of Vision Transformer (SMoE-MLP) for 3D medical image segmentation. Our framework efficiently combines vision transformers with sparse MoE to balance the performance of large-scale models with their computational costs. We evaluate our model on the BraTS 2021 [3] brain tumor segmentation task and the Hecktor 2022 head and neck tumor segmentation task. Our results clearly demonstrate the performance and faster training/inference of SMoE-MLP.

However, there are several limitations to our work. First, we only evaluated our method on two public datasets. Second, we did not test the scalability of our network to a larger network with more SMoE-MLP layers. Third, this network can be fully pipelined for distributed training on multiple servers. In this paper, we did not develop such a parallel training framework. Fourth, the Dice score of our framework

only improves upon the current state-of-the-art methods on some subtasks of the two datasets.

Our future work will focus on addressing the above limitations. In the near future, we will examine alternative gating strategies, localized self-attention, and hierarchical attention mechanisms to further improve the computational efficiency and accuracy of transformer-based models for medical image segmentation. We will include more datasets for evaluation, and a parallel distribution training framework will be implemented to improve inference and training speed.

Part II

Medical Image Registration

Overview

This part focuses on advanced deep learning approaches for the registration of medical images, which is critical for accurate analysis and treatment planning. Based on our previous segmentation methods, we observed that performance improvements were limited, regardless of the enhancements made. Therefore, we considered using more precise images, such as pathology images, for tumor segmentation. To achieve this, we first needed to register radiology images and pathology images together. However, several challenges arose due to the resolution gap between these two types of images and the lack of well-labeled paired data. In this part, we propose our own approach to registration. It includes:

- **Chapter 5: Path-CT Image Registration with Self-Supervised Vision Transformer for Lung Cancer [82]** This chapter explores a self-supervised learning approach using a vision transformer to align pathology and CT images in lung cancer. We address the data problem by using two separately trained feature extractors for each modality, which also improves segmentation results for downstream tasks.

- **Chapter 6: Upscaling Prostate Cancer MRI Images to Cell-level Resolution with Pathology WSI Using Self-Supervised Learning [83]** This chapter fully extends the previous work and addresses the challenge of fusing MRI and pathology images to achieve cell-level resolution for prostate cancer. We introduce a new script to fuse MRI and pathology images, resulting in improved downstream segmentation outcomes.

Each chapter in this part details the design, implementation, and evaluation of the

proposed models, showcasing significant improvements in segmentation accuracy and robustness compared to original methods.

CHAPTER 5: Path-CT Image Registration with Self-Supervised Vision Transformer for Lung Cancer

5.1 Introduction

Automatic medical image registration employs computational methods to identify an optimal spatial transformation that effectively aligns underlying anatomical structures. Traditional image registration follows an iterative procedure involving the collection of necessary features, determination of a similarity measure, selection of a transformation model, and finally, a search mechanism. Conventional approaches, like PI-RADS [2], strive to determine the transformation field for source and target images through intensive computation, which may be less efficient. In recent times, deep learning has emerged as the state-of-the-art method, significantly improving the performance of intensity-based registration techniques [84, 85, 86, 29, 87, 88, 89].

However, existing methods, imaging tasks and application focus on registering tissue images of the same precision-level such as MRI, CT and PET images. The research on integrating tissue-resolution images with cellular-resolution pathology images has been mainly used for identifying definitive prognostic biomarker [90], and very few aims to achieve a more accurate registration [86, 91]. In addition, there is a known challenge of limited well-labeled paired data to train supervised learning model for high accurate registration.

In this study, we assert that registering the cellular features in pathology images into tissue images, such as CT images, can produce images with more structural information of tumor and tissues. Leveraging the power of self-supervised learning, particularly in scenarios with limited labeled data, our research focuses on bridging the high-resolution gap between CT images and pathology modalities. Our method

includes a transformer-based network for the extraction of discriminative information inherent in both CT and pathology images, and a feature-matching network for aligning and mapping the distinctive features extracted from both image types.

We evaluate our framework using CT and pathology images and have improved the the Dice score from 65% to 72% on the lung cancer dataset. Additionally, we provide a fusing script for post-processing, offer radiologists and pathologists an integrated and cohesive view of registered CT and pathology images. Beyond improving the precision of existing CT and pathology image registration techniques, our approach exemplifies a new direction in medical image registration.

5.2 Design of the SSL Vision Transformer Framework

The designed DL network, as illustrated in Fig. 5.1, is a self-supervised learning network for registration images that have high-resolution gaps, such as the tissue-level CT images and the cellular-level pathology images studied in this work. The network includes 1) a self-supervised feature extractor, for both CT images and for whole-slide pathology images, 2) a feature-matching sub-network that aligns and maps the distinctive features extracted from both image types, 3) post-processing for fusing pathology and CT patches based on the correlation maps produced from the previous step.

5.2.1 Data and Pre-processing

In this study, CT and pathology data were sourced from TCIA [92] and TCGA [93] using a data retriever. TCIA provided a dataset of paired CT/pathology data, thoroughly documented in the original work [91], including 6 patient cases. Each case contains a series of CT images in 3D, 4 to 6 pathology slides, and corresponding annotations of lesions, blood vessels, and so on. Traditional registration methods, namely affine and deformable registration using the Elastix tool in Matlab, were utilized as detailed in the paper [91]. Although the data used in the original paper is sufficient for

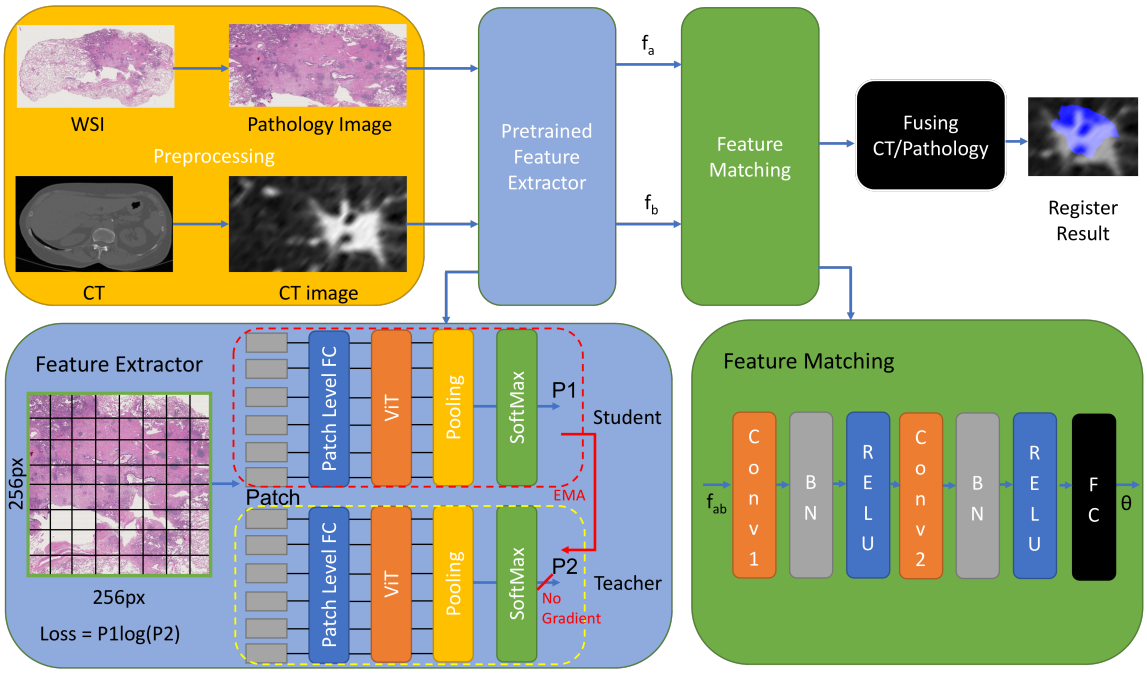


Figure 5.1: Overall Core Architecture Design of Path-CT Registration

traditional methods, it is limited to machine learning-based methods. Given the lack of well-labeled CT/pathology data, we leveraged extra unlabeled CT/pathology images from TCIA [94] and TCGA [93] respectively for pre-training in a self-supervised manner. Each pre-training dataset contains 500 cases.

Data preparation involved dividing the acquired whole slide images (WSI) from TCGA and CT scans from TCIA into smaller patches (256 × 256 pixels). As we know, the mounting of tissue sections on glass slides introduced inherent artifacts such as shrinkage, rotation, and flipping [86]. To mitigate these effects we extended the pathology images to a 3D representation and matched the reconstructed 3D feature with the original 3D CT [91]. Subsequently, corresponding CT slides and sections were located, as illustrated in *PathologyImage* in Fig. 5.1. Similar processing steps were applied to those annotations, including blood vessels, in both pathology slices and CT scans, as illustrated in the *CTimage* section of Fig. 5.1. Following normalization of CT slice intensities (ranging from 0 to 255), images were resampled to 256 × 256

before integration into the feature extractor pipeline for the feature matching.

5.2.2 Self-supervised Feature Extractor Based on DINO

In this framework, we implemented a self-supervised feature extractor inspired by DINO [95]. DINO, operating on the principles of self-distillation without labels, offers a mechanism for extracting features imbued with explicit information relevant to the semantic segmentation of images. DINO has a two-step process: a self-supervised pre-training stage followed by a fine-tuning stage. During the pre-training stage, the model learns to associate similar features of images while separating unrelated ones, creating meaningful representations. This is achieved through a self-distillation mechanism, where the model acts as both a student and a teacher, guiding itself to learn useful representations.

In our paper, we adapted the DINO framework to extract the features for both CT and pathology images as a feature extractor. Since these CT and pathology images were downloaded from different databases for pre-training, the feature extractors were trained independently for CT and pathology images in this study. The overall architecture of our self-supervised feature extractor is illustrated in Fig. 5.1. To enhance performance, pre-trained weights on ImageNet were utilized for both the student and teacher networks [95]. In Fig. 5.1, note the larger size of the pathology image compared to the natural image. To effectively manage this, we divided each image into smaller pathology images, each measuring 256 $\times$ 256 pixels. These small patch images were then fed into both the student and teacher networks, which shared identical network structures. The networks utilized standard vision transformer blocks, as illustrated in the feature extractor block at the bottom of Fig. 5.1, with characteristics such as Layer Normalization, multi-headed self-attention, residual connection, and a multi-layer perceptron. We adopted the vit-small as the backbone for our feature extractor, with parameter settings unchanged [95].

Our training strategy involved keeping the teacher network frozen, with weight

updates exclusively driven by the student network. Regular training updates were applied to the student network. Regarding training loss, we adhered to the loss function shown in the bottom left of Fig. 5.1. The teacher network underwent updates using Exponential Moving Average (EMA), consistent with prior work.

5.2.3 Feature Matching Network Based on CNN

Our feature-matching network, inspired by the methodology outlined in the work of Shao et al. [86], consists of two parts: 1. correlation map calculation and 2. feature regression. In the previous step, we obtained features for both CT scans and pathology images. Each feature map, *f*, represents an image of dimensions *(w, h, d)*, where d represents the number of features, w represents the width, and h represents the height. Subsequently, the feature maps f_A and f_B were input into a correlation layer, which is a dot production of input features to describe the similarity between the two images. This correlation layer combines f_A and f_B to create a correlation map C_{ab} of the same size. The computation of the correlation map is expressed as:

$$C_{ab}\big(i,j,k\big) = f_B\big(i,j\big)^T * f_A\big(i_k, j_k\big) \tag{5.1}$$

where $k = h(j_k - 1) + i_k$. The resulting correlation map C_{ab} indicates the similarity of f_b at the position (i, j) and all features of f_a. To address potential ambiguous matches, normalization is applied to obtain the resulting tentative correspondence map f_{ab}. After that, f_{ab} needs to pass through a regression network to estimate the parameters of the geometric transformation related to the input CT and pathology image. Following the same architecture in paper [86], the regression network consists of two layers, with each layer beginning with a convolutional unit, followed by batch normalization and ReLU. A final fully connected (FC) layer conducts the regression of parameters for the geometric transform and outputs it as the θ, as illustrated in Fig. 5.1. The θ is considered as the affine matrix for the registration. In this paper,

we use the same modified affine transformation and loss function to improve stability as the detailed description in paper [86].

5.2.4 Fusing Pathology and CT Image

After the affine image registration, the pathology images, CT scans, annotated lesions, blood vessels, and other labels, such as invasive, were aligned with the corresponding CT slices using the estimated composite affine transformation θ. It is essential to note that pathology images typically have a larger size than sliced CT images, which are usually 256×256. Consequently, the deformed pathology images maintain the same size as the original high-resolution images because the affine transformation was only applied to the original image. To visualize our results effectively, we utilized OpenCV package to deploy a post-processing Python script to resize the CT scans to a larger dimension, stack the pathology images, and fuse all the images together. An example of this process is presented in the result section to showcase the functionality of our post-processing script.

5.3 Evaluation

In this section, we demonstrate the effectiveness of our proposed registration methods on lung cancer datasets [96] and provide ablation study to offer insight into the importance of its different components. Our evaluation consists of two parts. In the first part, we evaluate the registration of CT and pathology images on the lung cancer dataset. In the second part, we assess the effectiveness of the pre-trained self-supervised learning on the CT/pathology image feature extractor.

5.3.1 Result of Registration

We evaluated the registration results both qualitatively and quantitatively on the lung cancer dataset [96] and compared the results with those presented in the original paper [91].

5.3.1.1 Quantitative results

We evaluated the quantitative results using two metrics: the Euclidean distance and the Dice score for CT and registered pathology images in six cases. The Dice score is better when higher, indicating increased similarity. Conversely, the Euclidean distance is better when lower, indicating decreased separation. The evaluation of multimodal registration includes the assessment of blood vessels, manually annotated on both CT and pathology images, treated as landmarks. These blood vessels serve as landmarks solely for evaluating the registration result, as they are not utilized during the registration process. The same transformation applied to the original blood vessels on the pathology images ensures consistency. The Euclidean distance measures the distance between landmarks on CT and registered pathology images, while the Dice score indicates the overlap of blood vessels between the two images. To mitigate bias, we compute the average Euclidean distance and Dice score for all pathology slices for each patient. The initial findings are summarized in Table 5.1. The overall Dice score averages $72.6\% \pm 3.8\%$ across the six cases, with an average Euclidean distance of $1.73\,\text{mm} \pm 0.29\,\text{mm}$. Compared with the original work [91], we improved the Dice score from 65.9% to 72.6% on average and reduced the distance from 1.9 mm to 1.7 mm.

Table 5.1: Quantitative result of registering pathology images with CT image on all the six cases. The Euclidean Distance and the Dice score metrics are calculated between the ground truth of blood vessels on CT and registered pathology image.

Case ID	Euclidean Distance (mm)↓		Dice ↑	
	Original	Our	Original	Our
LungFCP-01-0001	1.75	1.782	0.731	0.692
LungFCP-01-0002	2.15	1.764	0.624	0.739
LungFCP-01-0003	2.02	1.811	0.595	0.687
LungFCP-01-0004	1.81	1.447	0.689	0.764
LungFCP-01-0005	1.42	1.671	0.692	0.745
LungFCP-01-0006	2.67	1.915	0.624	0.728

5.3.1.2 Qualitative results

We present the visualization of our registered results in Fig. 5.2. The blood vessel and lesion labels were annotated by an expert pathologist on the pathology image. We applied the same transformation to the labels and registered them with the CT image, as shown in sub-figures d and e. The lesions and blood vessels were utilized solely for evaluation purposes and were not part of the training procedure. The quantitative evaluation in the last section calculated the registered vessels' Euclidean distance compared to the labeled vessels for the visualized patient with the case ID LungFCP-01-0001.

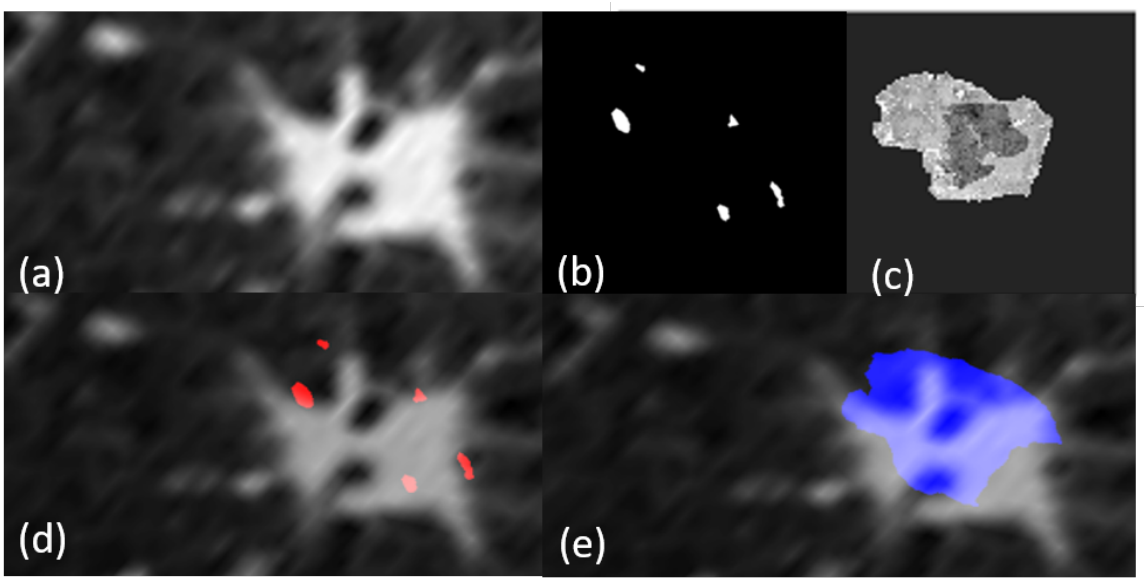


Figure 5.2: Visualization of Qualitative Result. Red color presents blood vessel. Blue color presents lesion. (a) is the visualization of one slides CT image which aligned with the sagittal view and resized to (13, 600, 1050). (b) is the visualization of lesion part of pathology image in sagittal view. (c) is stacked pathology image. (d) is the visualization of registered blood vessels segmentation. (e) is the visualization of registered lesion with CT image.

5.3.2 Ablation Study

In this paper, we employed the DINO framework for the pre-trained feature extractor [95]. As described in the methodology section, we conducted pre-training using

DINO self-supervised learning (DINO-SSL) on various datasets, and the results are compared in Table 5.2. The token size was set to 16, and the dimension was set to 384. We utilized the pre-trained model weights on ImageNet released by the official DINO paper. Due to the limited availability of paired CT and pathology images, we separately trained the feature extraction networks for CT and pathology images. Since the data from TCIA [94] and TCGA [93] are unlabeled, we employed KNN [95] classification results as ground truth for evaluating the accuracy of our pre-trained feature extractor. If the feature extractor output matches the same class as the KNN classifier, we consider it correct; otherwise, we deem it incorrect. We categorized the classes into three: background, lesion, and blood vessel, labeled as 0, 1, and 2, respectively. The accuracy of the feature extractor is presented in Table 5.2 for different datasets. The accuracy on the TCIA-Prostate dataset is lower than that of the TCGA-PRAD and TCGA-various datasets, possibly due to the smaller size of the TCIA dataset. Despite increasing the training epochs from 100 to 800 for TCIA-Prostate data, the accuracy improvement was marginal. Conversely, for the other two datasets, we achieved an accuracy of 95% with only 100 epochs.

Table 5.2: Accuracy of pre-trained feature extractor on different dataset

Dataset	Epochs	Accuracy
TCIA-PROSTATE	100	0.88
TCIA-PROSTATE	800	0.89
TCGA-PRAD	100	0.95
TCGA-various	100	0.96

5.4 Conclusion

In this paper, we have introduced a self-supervised learning framework for the registration of CT and pathology images in the lung cancer dataset. Leveraging the pre-trained self-supervised learning feature extractor, our framework captures improved feature representations of both CT images and pathology, easing the registration process. The performance of our framework is notable, achieving a 72.9% Dice score and

a 1.73 mm Euclidean distance, surpassing the registration results that use conventional method. This underscores the promising potential of self-supervised learning frameworks for future advancements in medical image registration. Additionally, our framework offers a novel approach to fuse high-resolution images to low-resolution counterparts.

CHAPTER 6: Upscaling Prostate Cancer MRI Images to Cell-level Resolution with Pathology WSI Using Self-supervised Learning

6.1 Introduction

Radiological imaging constitutes a cornerstone in the study of cancer, spanning critical stages from foundational research to diagnostic elucidation, therapeutic strategizing, and ongoing surveillance. Modalities such as computed tomography (CT), magnetic resonance imaging (MRI), and positron emission tomography (PET) furnish intricate depictions of internal anatomical structures, affording clinicians invaluable insights into tumor localization, metastatic dissemination, and anomalous tissue proliferation. These imaging modalities, renowned for their capacity to discern subtle nuances in size, shape, and morphological attributes, empower medical practitioners to delineate various cancer phenotypes, and ascertain tumor staging and grading, etc.

Yet, the interpretation of radiographic imagery has its challenge, as discerning malignant from benign tissue can often be nuanced and subjective, even among seasoned experts. Manual demarcation of cancerous lesions on radiological scans, while essential, is often fraught with potential inaccuracies, leading to the potential underestimation of tumor dimensions or overlooking of less conspicuous lesions that are not clearly visible on radiology images due to the low resolution. In contrast, pathology whole slide images (WSI), often boasting gigapixel-level resolution of cells, afford pathologists unprecedented insight into tissue microstructures gleaned from biopsies or surgical specimens. This microscopic granularity enables the discernment of cellular morphology, tissue architecture, and aberrant cellular features with exceptional precision. Leveraging the registration of histopathology images with their corresponding radiological slices, clinicians can overlay cancerous regions identified in

histopathological analyses onto radiological scans. This registered images would enable the precise segmentation of tumors, encompassing lesions imperceptible on MRI scans, and facilitates cancer evaluation.

However, fusing images from different modalities present technical hurdles due to inherent disparities in resolution, particularly for integrating images of large resolution gaps such as pathology and radiology images. Currently, image registration techniques are primarily developed for medical images with similar resolutions, such as PET, CT, and MRI scans [85, 29, 89]. These methods typically fall into two categories: traditional approaches, which often suffer from low computational efficiency, and machine learning-based methods. However, the machine learning-based methods are limited in their applicability as they tend to work only on specific datasets and struggle to extend to larger resolution gaps, such as between MRI and pathology WSI. Furthermore, these machine learning approaches face challenges due to the lack of paired and well-labeled data available for training purposes.

To address the limitations of existing machine learning-based registration methods and enhance the generation of fusion images with improved diagnostic capabilities, we present a novel self-supervised learning framework specifically designed for the registration of radiological and pathological images in this study. Our primary aim is to reduce the resolution gap between radiology and pathology images to facilitate precise registration. This is achieved through the utilization of a self-supervised transformer-based feature extraction network and a feature-matching network. Our ultimate objective is to enhance the resolution of MRI images to the cell level using pathology data, with the registration process serving as a crucial tool in achieving this goal. By leveraging cutting-edge self-supervised transformer-based architectures for feature extraction and matching, our framework overcomes the constraints of current registration methods which need multiple paired and well-labeled data. It accomplishes this by extracting relevant features with a self-supervised feature extrator

from high-resolution pathology specimens and low-resolution radiological scans, all without requiring labeled training data. Subsequently, the feature-matching network aligns, maps, and registers distinct features from both image modalities in a self-supervised manner. This approach ensures more accurate and detailed registration, facilitating the creation of fusion images that offer enhanced diagnostic insights. Our contributions can be summarized as follows:

1. We have introduced a novel fused image framework aimed at upscaling prostate MRI images to cell-level resolution, leveraging registration as a key tool. This innovative framework enables the creation of fused images that seamlessly integrate high-resolution pathology data with MRI scans. These fused images hold great promise for downstream tasks such as cancer segmentation, offering enhanced details and accuracy crucial for improved diagnostic capabilities.

2. We evaluated our framework using datasets related to prostate cancer, comparing it with the original paper that presented those datasets. Our framework demonstrated an enhancement in accuracy from 56.3% to 64.6% for prostate cancer.

3. We tackled the issue of insufficient well-labeled paired pathology and radiology images by introducing a novel approach to self-supervised learning, which separates the learning process on unlabeled datasets. The efficiency of pre-trained self-supervised learning can be seen in the ablation study part.

4. We addressed the resolution gap between radiology images and pathology modalities through our innovative registration concept to achieve a 39 times resolution difference between the original MRI and the new fusion image.

5. We evaluated the similarity of our new fused image and up-scaled MRI image by Mutal Information and Structural Similarity Index. We achieved a mutual information score on average of 4 and a Structural Similarity Index at least of 0.933 which suggests our new fusion image has high similarity to the MRI image.

6.2 Design of the Extended SSL Vision Transformer Framework

For medical images, which are multimodal in nature, fusion and the registration of images of different modality have been well studied topics. Our method advanced the state-of-art by using self-supervised learning, which alleviates the dependency on labeled data by leveraging the inherent structure and redundancy within the data itself.

In this work, we present a novel self-supervised learning framework inspired by DINO [95] and Prosregnet [86]. In contrast to traditional DINO models and Prosregnet, our framework is an image fusion network combining the features of images that have high-resolution gaps, such as the tissue-level MRI and the cellular-level pathology images, as illustrated in Fig. 6.1. The key component of the network includes 1) Two self-supervised feature extractors, each devoted to MRI and whole-slide pathology images respectively. 2) A correlation mapping block responsible for generating correlation maps of features extracted by the previous feature extractors. 3) A feature-matching sub-network designed to align and map distinctive features extracted from both image types. 4) Post-processing techniques for fusing pathology and MRI patches based on the correlation maps obtained in the previous step.

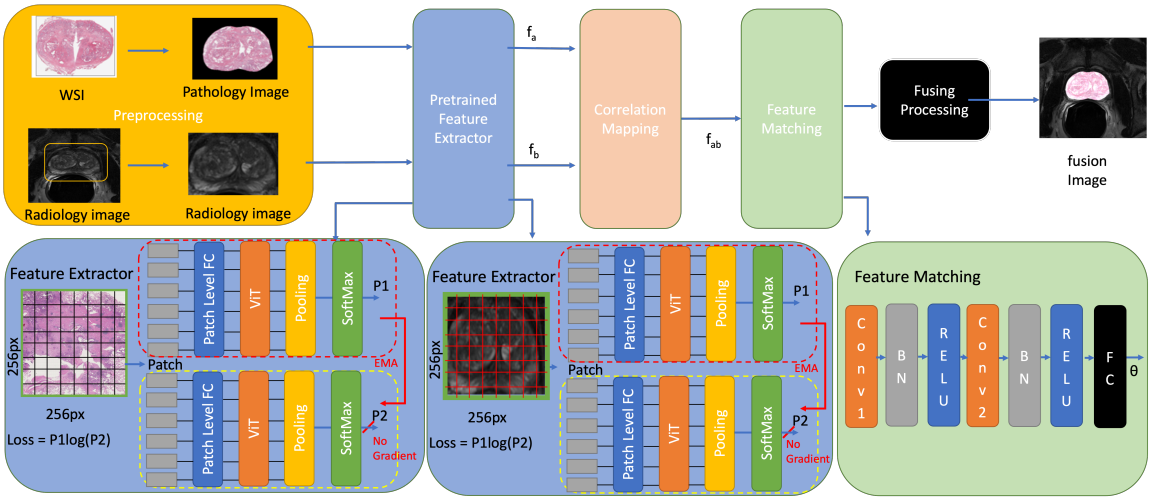


Figure 6.1: Overall Core Architecture Design of Cell Level Precision Registration

6.2.1 Data and Pre-processing

In this study, all MRI and pathology data were obtained from publicly available repositories, specifically TCIA [92] and TCGA [93], utilizing a data retriever tool. TCIA provided two datasets of paired MRI/pathology data, extensively described in the original work [86]. The first dataset, PROSTATE-MRI [97], consists of 26 cases, with each case containing multiple pathology slides. In total, we have 82 paired MRI and Pathology WSI from this dataset. The second dataset [98], referred to as fused MRI-Prostate, comprises 28 cases, each containing 3 Tesla T1-weighted, T2-weighted, Diffusion weighted, and Dynamic Contrast-Enhanced prostate MRI scans, accompanied by corresponding digitized histopathology (H&E stained) images of radical prostatectomy specimens. For training purposes, we utilized all images from the first dataset and 26 cases from the second dataset. The remaining two cases, along with 6 slides annotated with lesions, were reserved as the test dataset. The first dataset was not used for the test cases due to the absence of lesion annotations necessary for evaluating downstream segmentation tasks. Given the limited size of the datasets, comprising only 82 paired data points, we relied on machine learning-based methods. To mitigate the data limitations, we extended the training data by incorporating additional unlabeled MRI and pathology images from TCIA [94] and TCGA [93], respectively. Each pre-training dataset consisted of 500 cases, enabling us to pre-train a feature extractor in a self-supervised manner.

In the initial stage, we trained the feature extractor using unlabeled MRI and pathology image data. We subdivided the acquired WSI from TCGA and MRI scans from TCIA into smaller patches measuring 256 * 256 pixels. For the subsequent step, we trained our feature matching subnet utilizing two paired MRI/pathology datasets. We standardized the dimensions of these datasets to match those of our pre-trained feature extraction data. Given the paired nature of the data from these two datasets, corresponding MRI slides and pathology images were aligned, as depicted in the

PathologyImage and *RadiologyImage* sections of Fig. 6.1. Furthermore, when performing downstream tasks on both pathology slices and MRI scans, such as lesion annotation, we applied the same methodology. This ensured alignment and correspondence between the annotations, as illustrated in the *RadiologyImage* section of Fig. 6.1. To prepare the MRI images for integration into the feature extractor pipeline during training of the feature matching sub-network and for inference, we normalized the MRI intensities to a scale ranging from 0 to 255.

6.2.2 Self-supervised Feature Extractor Based on DINO

The first component of our framework comprises a self-supervised feature extractor inspired by DINO [95]. DINO is a self-supervised learning method for visual representation learning. It achieves cutting-edge performance by aligning representations of the same image across different layers of the neural network through self-distillation. The DINO framework operates in two stages: a self-supervised pre-training stage and a fine-tuning stage. During the pre-training stage, the model learns similar features across the images while separating unrelated ones, thereby constructing meaningful representations. This is facilitated by a self-distillation mechanism, where the model serves both as a student and a teacher, guiding itself to learn valuable representations. Subsequently, in the fine-tuning stage, the pre-trained model can be further optimized for specific downstream tasks, such as image classification or object detection, ensuring its adaptability to diverse application scenarios.

In this paper, the initial phase involves pre-training a pathology and MRI feature extractors by leveraging the DINO framework. Given the disparity in resolution and format between MRI and pathology images, we undertake separate pre-training processes for each modality. Consequently, we develop distinct feature extractors tailored to MRI and pathology images, as illustrated by the two feature extractors depicted in Fig. 6.1.

To optimize performance, we leverage pre-trained weights obtained from the TCIA

dataset for both the student and teacher networks, as outlined in [95]. Given the larger size of pathology images compared to natural images, we decompose each image into smaller patches, each measuring 256×256 pixels. These patch images are fed into both the student and teacher networks, which share identical network structures. The networks employed standard vision transformer blocks, depicted in the feature extractor block at the bottom of Fig. 6.1, featuring components such as Layer Normalization, multi-headed self-attention, residual connections, and a multi-layer perceptron. We adopted the ViT-small as the backbone for our feature extractor, maintaining the parameter settings as per [95].

During the pre-training phase, the teacher network is frozen with weight updates solely in the student network. A distillation loss across the teacher-student predictions is imposed to train the self-supervised framework (see Fig. 6.1). The parameters in the teacher network is updated using Exponential Moving Average (EMA).

6.2.3 Feature Matching Sub-Network Based on CNN

Our feature-matching network, inspired by the methodology outlined in the work of Shao et al. [86], consists of two main components: correlation mapping and feature matching. Initially, we obtained features for both MRI and pathology images from two separate feature extractors. Each feature map, denoted as *f*, represents an image with dimensions *(w, h, d)*, where d represents the number of features, w represents the width, and h represents the height. Subsequently, the feature maps f_A and f_B were downsampled into a smaller dimension representation to reduce computational costs. These downscaled feature maps were then input into a correlation layer, which computes the dot product of input features to quantify the similarity between the two images. This correlation layer combines f_A and f_B to generate a correlation map C_{ab} of the same size. The computation of the correlation map is expressed as:

$$C_{ab}\big(i, j, k\big) = f_B\big(i, j\big)^T * f_A\big(i_k, j_k\big) \tag{6.1}$$

The equation $k = h(j_k - 1) + i_k$ is used to calculate the index variable k based on the indices i_k and j_k, with h representing the width of the feature map. The resulting correlation map C_{ab} indicates the similarity of features from f_b at position (i, j) with all features from f_a. To address potential ambiguous matches, normalization is applied to obtain the correspondence map f_{ab}. This map then undergoes processing by a feature-matching network, which is a regression network responsible for estimating the parameters of the geometric transformation associated with the input MRI and pathology images. Following the architecture described in paper [86], the regression network comprises two layers. Each layer begins with a convolutional unit, followed by batch normalization and ReLU activation. A final fully-connected (FC) layer performs the regression of parameters for the geometric transform, outputting the affine matrix θ. The matrix θ serves as the affine transformation for the registration process.

6.2.4 Loss Function for Feature Matching

The loss function was determined as the sum of squared differences (SSD) between the original input MRI and the deformed pathology image. The formula of the loss function was shown as:

$$loss = \sum_{i}^{H} \sum_{i}^{W} \|I_A(i, J) - I_B(i, J) \bullet \phi_\Theta(i, j)\|^2 \tag{6.2}$$

where $\phi_\Theta(i, j)$ is the related transformation vector from the output of feature matching sub-network, H is the height of the image and W is the weight of the image.

6.2.5 Fusing Pathology and MRI

After completing the image registration process, the pathology images, MRI scans, and annotated lesions were aligned with the corresponding MRI slices using the estimated composite affine transformation θ. It is important to note that pathology images typically have larger dimensions than sliced MRI images, which are usually

512×512. Consequently, the deformed pathology images maintain their original size as high-resolution images, as the affine transformation is only applied to the original image. Once the affine transformation matrix θ is determined based on the input image, it remains fixed. During inference for high-resolution pathology images in gigapixel scale, we first resize them to the same smaller size as the MRI slides. These resized images are then fed into our network to obtain the transformation matrix θ. Finally, we apply θ to the original resolution pathology image to obtain the registered high-resolution pathology image. To effectively generate our fusion results, we deployed a post-processing Python script. This script upscales the MRI scans to a larger dimension and then utilizes the mask image to determine the position of the MRI, the registered pathology images, and fuse all the images together. An example of this process is provided in the results section to demonstrate the functionality of our post-processing script.

6.3 Evaluation

In this section, we demonstrate the effectiveness of our proposed registration framework on two prostate cancer datasets [96]. Our evaluation consists of three parts. In the first part, we evaluate the fusion image from the resolution comparison, mutual information (MI) evaluation, and Structural Similarity (SSIM). In the second part, we will display some qualitative results for our fusion image. In the last part, we will have a quantitative evaluation of downstream lesion segmentation tasks.

6.3.1 Comparison of Fused Image and Original MRI

In this section, We evaluated the difference between the fusion image and the original MRI both from the resolution enhancement, MI score, and SSIM on the prostate cancer dataset [96].

6.3.1.1 Resolution Enhance

Our research aims to improve resolution enhancement through post-processing fusion, and to strengthen the quality of medical imaging data, notably by leveraging high-resolution pathology images to high-resolution Magnetic Resonance Imaging (MRI). The reconstruction resolution change for each test case is displayed in the table 6.1. Through the reconstruction, our hyper-resolution image enhanced the resolution 39 times than original MRI image with rich information.

Table 6.1: Resolution of original MRI, original pathology image, and registration fused image of all 6 test cases.

Case ID	Resolution (pixel * pixel) ↑		
	MRI	Pathology image	Superegstration
aaa0060 C1C2C3C4	320*320	860*860	2000*2000
aaa0060 D1D2D3D4	320*320	860*860	2000*2000
aaa0060 E1E2E3E4	320*320	860*860	2000*2000
aaa0069 CSlides	320*320	860*860	2000*2000
aaa0069 DSlides	320*320	860*860	2000*2000
aaa0069 ESlides	320*320	860*860	2000*2000

6.3.1.2 Mutual Information(MI)

We also evaluated the similarity of fusion image and MRI image by Mutual Information (MI). MI is a measure of the mutual dependence between two random variables. It quantifies the amount of information obtained about one random variable through the other random variable. In the context of image processing, MI can be utilized to assess the similarity between two images by comparing the statistical dependencies between their pixel intensities. The mutual information between two discrete random variables X and Y is defined as:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right)$$

where:$I(X;Y)$ is the mutual information between X and Y. $p(x,y)$ is the joint probability mass function of X and Y. $p(x)$ and $p(y)$ are the marginal probability mass functions of X and Y, respectively.

Higher MI values indicate a greater similarity between the images, while lower MI values suggest dissimilarity. The result of MI information is show in the table 6.2. As shown in the table, our MI score is close to the upper bounder of MI score which is 4 on average. We also display the normalized MI score as a reference. The score suggested our fusion image has high similarity to the upscaled MRI.

Table 6.2: MI information and score of all six test cases.

Case ID	Mutual Information ↑		
	MI	MI Upper boundary	MI(Normal)
aaa0060 C1C2C3C4	4.112	15.6	6.80e-7
aaa0060 D1D2D3D4	4.142	15.6	6.82e-7
aaa0060 E1E2E3E4	4.033	15.6	6.71e-7
aaa0069 CSlides	4.161	15.6	6.93e-7
aaa0069 DSlides	4.011	15.6	6.69e-7
aaa0069 ESlides	4.032	15.6	6.70e-7

6.3.1.3 Structural Similarity Index (SSIM)

We also evaluated the structural similarity of the fusion image and MRI image by Structural Similarity Index (SSIM). SSIM is a metric used to measure the similarity between two images. It takes into account three aspects of image quality: luminance, contrast, and structure. The formula for SSIM is given by:

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{6.3}$$

where: μ_x is mean of x, μ_y is mean of y, σ_x is standard deviation of x, σ_y is standard deviation of y, σ_{xy} is covariance of x and y, c_1 is constant to stabilize the division with weak denominator, c_2 is constant to stabilize the division with weak

denominator. The SSIM value ranges from -1 to 1, where a value of 1 indicates perfect similarity between the two images.

The result of SSIM in all six cases is shown in Fig. 6.2. As the value shows, the lowest SSIM score we have for aaa0069 is still 0.933 which is close to the max SSIM value which is 1. That suggested that our fusion image has high similarity with the up-scaled MRI in both luminance, contrast, and structure.

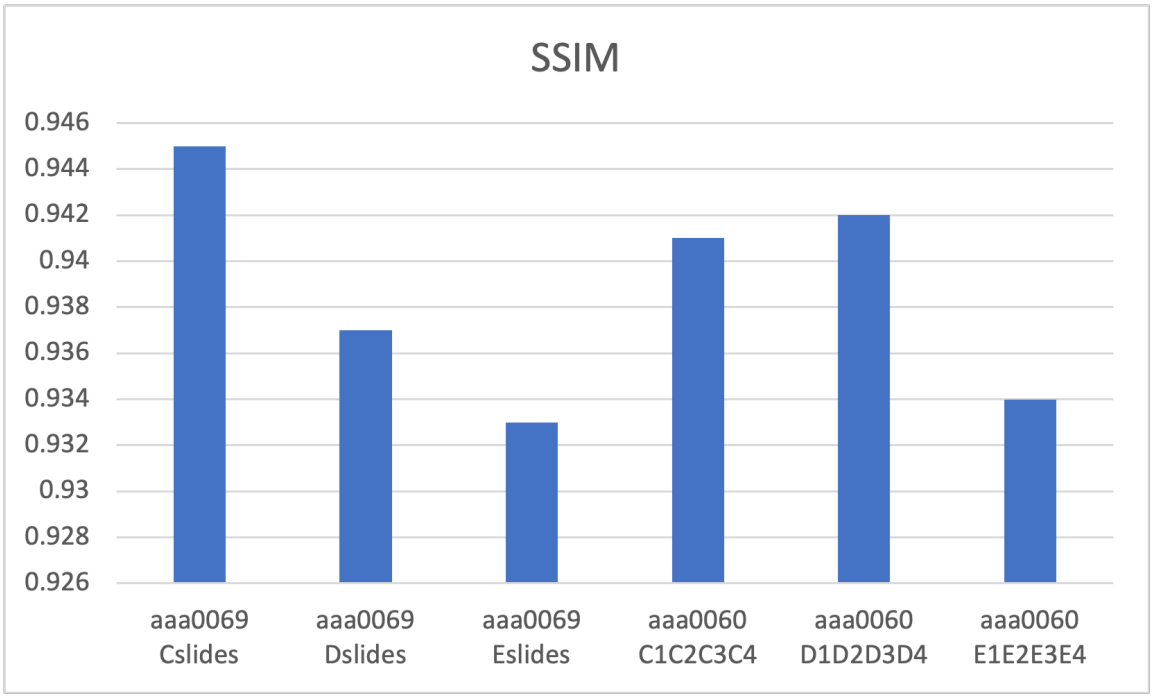


Figure 6.2: SSIM score of all six test cases

6.3.2 Quantitative and Qualitative Result on Downstream Segmentation Task

6.3.2.1 Quantitative results

We assessed the quantitative results for lesion segmentation using two metrics: the Euclidean distance and the Dice score, applied to MRI and registered pathology images across six test cases. A higher Dice score indicates increased similarity, while a lower Euclidean distance(ED) signifies reduced separation. In this evaluation, lesions manually annotated on both MRI and pathology images served as landmarks for assessing the segmentation quality. These lesion annotations were solely used

for evaluation purposes and were not incorporated into the training process. To ensure consistency, the same transformation was applied to the original lesions on the pathology images. The Euclidean distance measured the distance between landmarks on MRI and registered pathology images, while the Dice score indicated the overlap of lesions between the two images. The initial findings, summarized in Table 6.3, revealed an average Dice score of $64.0\% \pm 4.1\%$ across the six cases, with an average Euclidean distance of $2.074, \text{mm} \pm 0.776, \text{mm}$. Compared to the original work by Shao et al. [86], our method demonstrated improvements, increasing the average Dice score from 57.3% to 64.0% and reducing the average distance from 5.42 mm to 2.074 mm. Since the original paper did not report the Dice score and Euclidean distance (ED) for each case, our comparison in this paper focuses on the mean Dice score and mean ED across all cases.

Table 6.3: Quantitative result of registering pathology images with MRI image on all the six test cases. The Euclidean Distance and the Dice score metrics are calculated between the ground truth of lesions on MRI and fused pathology images.

Case ID	Euclidean Distance (mm)↓		Dice ↑	
	Original	Our	Original	Our
aaa0060 C1C2C3C4	N/A	1.663	N/A	0.653
aaa0060 D1D2D3D4	N/A	1.872	N/A	0.632
aaa0060 E1E2E3E4	N/A	1.763	N/A	0.665
aaa0069 CSlides	N/A	1.532	N/A	0.681
aaa0069 DSlides	N/A	2.851	N/A	0.600
aaa0069 ESlides	N/A	2.767	N/A	0.609
Mean	5.42	**2.074**	0.573	**0.640**

6.3.2.2 Qualitative results

We present the visualization of our registered results in Fig. 6.3. The figure showcases the following components: the upscaled MRI in section (a), the pathology image with the related mask in section (b), the fusion image in section (c), and the lesion labels in section (d). Each row corresponds to a visualized patient, with cases aa0069

Cslides, aa0069 Dslides, and aa0069 Eslides displayed from top to bottom, respectively. The lesion labels were annotated by an expert pathologist reported on the original dataset. We applied the same transformation to the labels and registered them with the fusion image, as illustrated in sub-figures (d) and (e). Given the limited existing work on fusing low-resolution MRI images with high-resolution pathology images on the same dataset, we did not visualize other frameworks for comparison since adopting such methods for our paper would have required substantial additional effort and resources.

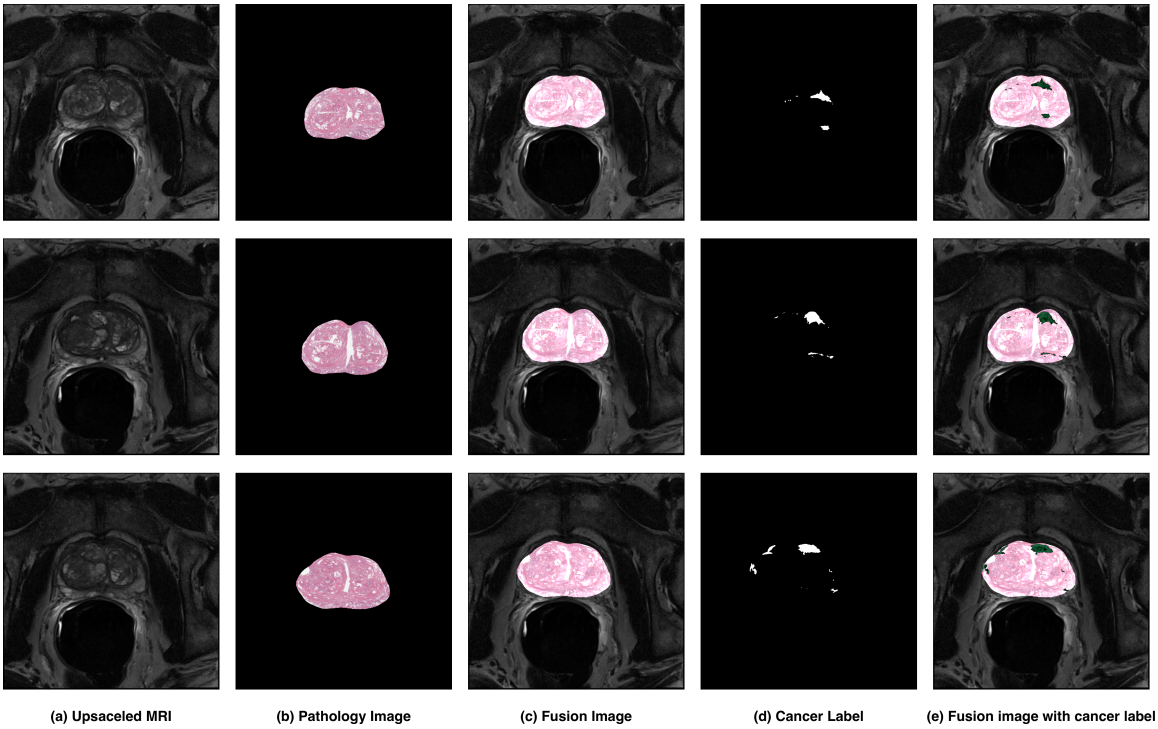


Figure 6.3: Visualization of Qualitative Result. (a) is the visualization of three MRI sample slides that aligned with the axial view and resized to (2000,2000). (b) is the visualization of pathology images. All sides were combined by four WSIs. (c) is fused pathology-MRI image. (d) is the visualization of registered lesions segmentation. (e) is the visualization of registered lesions with fusion images.

6.3.3 Ablation Study

In our methodology, we utilized the DINO self-supervised learning (DINO-SSL) framework for pre-training on diverse datasets. To measure the effectiveness of our

pre-training strategy using self-supervised learning, we employed the downstream lesion segmentation task as our evaluation metric. The results are summarized in Table 6.4. All experiments utilized the same architecture, leveraging the ViT small backbone within the DINO framework with an output dimension of 348. The first row of the table corresponds to our framework without pre-training, trained with feature matching network on two training datasets. Here, we observed a Dice score as low as 55%. However, when employing pre-trained model weights from ImageNet, as provided by the official DINO paper, for both MRI and pathology feature extractors, we observed a notable improvement, with the Dice score increasing to 61%. Furthermore, adopting separate pre-training on TCIA and TCGA datasets for MRI and pathology images can led to further enhancement, with the Dice score reaching 64%. This highlights the significance of pre-training on distinct datasets as a crucial factor in enhancing the performance of existing methodologies. Through our ablation study, we identified key factors contributing to the overall efficacy of our proposed framework.

Table 6.4: Dice Scores of Various Training Configurations using Pre-Trained Feature Extractors on Different Datasets for Downstream Prostate Cancer Segmentation

Arch	SSL Method	Dataset	Epochs	Dim	Dice Score
ViT-S/16	Dino	scratch	N/A	348	0.55
ViT-S/16	Dino	ImageNet	100	348	0.61
ViT-S/16	Dino	TCIA/TCGA	100	348	0.64

6.4 Related Work

In this section, we briefly discussed image super-resolution in medical domains, self-supervised learning methods, multi-resolution networks, and fusion and registration in pathology images.

6.4.1 Fusion and Registration of Multi-modal Medical Images

Existing methodologies mainly focus on aligning images of same-level precision, such as MRI, CT, or PET scans, leaving absence of solutions tailored to the challenge of different resolution registration between pathology and radiology modalities [84, 85, 86, 29, 87, 88, 89]. Traditional techniques often entail laborious 3D reconstruction of histopathological sections followed by iterative refinement of image alignment, while emerging machine learning-driven approaches, exemplified by Pros-RegNet, offer promising avenues for enhanced accuracy but are constrained by the necessity for copious annotated data. Few research studies have delved into the domain of different resolution registration between pathology and radiology images [86]. One approach involves reconstructing the 3D features of pathology slides and subsequently employing a traditional grid-like search to optimize the transformation between pathology and radiology images [91]. Another method, known as RAPSODI, initiates with the 3D reconstruction of histopathological specimens through the registration of each histopathology slice to its adjacent slice. Subsequently, 2D rigid, affine, and deformable transformations are iteratively estimated between each histopathology image and its corresponding MRI slice using gradient descent. However, these conventional methods exhibit lesser computational efficiency. On the other hand, machine learning-based approaches like ssEMnet [87] have demonstrated the capability to achieve more accurate registrations, albeit being limited to specific datasets. In contrast, our approach tried to use the registration of high-resolution pathology images to correlate position at MRI to high-resolution fusion image.

6.4.2 Super-Resolution in Medical Domains

Single image super-resolution (SISR), which refers to the process of recovering high-resolution (HR) images from low-resolution (LR) images, is an important class of image processing techniques in computer vision and image processing. In the real-

world application, SISR is important to provide rich information, especially in the medical image domain. High-resolution medical images can provide more riched information about the human tissue which can help the diagnosis processing. In recent, Chen et al. proposed a Multi-level Densely Connected Super-Resolution Network with GAN network to generate high-resolution MR images, which can achieve 6 times more faster than 3D FSRCNN both in training and inference [99]. For CT image, a 3D Super-Resolution Convolutional Neural Network (3DSRCNN) is proposed to use the convolution network to restore single low-resolution CT image to high-resolution 3D-CT volumetric images [100]. To further solve the problem of lacking high quality and effective training samples, Zhao et al. proposed a deep Channel Splitting Network (CSN) to use a series of cascaded channel-splitting blocks with two hierarchical feature branches with different information propagations [101]. The proposed CSN can achieve a more accurate SR image. In [102], Peng et al. introduced a SpatiallyAware Interpolation Network (SAINT) for medical slice synthesis to generate 6 times SR CT with promising results. However, our approach can achieve 39 times resolution enhancement and provide extra information from different modalities by using the registration of mask region pathology images.

6.4.3 Self-Supervised Learning for Pathology Images

In recent years, researchers have begun to explore self-supervised learning techniques in the context of pathology image analysis. These methods aim to exploit the abundant unlabeled pathology images available to learn representations that capture relevant biological and morphological characteristics. DINO [95], a wide self-supervised learning framework that form of self-distillation with no labels. It provides a way to facilitate the extraction of features containing explicit information about an image's semantic segmentation. In [103], Chen et al. extended the DINO framework and trained various self-supervised models, finding that Vision Transformers, particularly with DINO-based knowledge distillation, effectively learned interpretable

features in histology images. In the following work[104], they proposed a new Hierarchical Image Pyramid Transformer (HIPT) architecture utilizes this hierarchical nature in WSIs by employing a two-level self-supervised learning approach to effectively learn high-resolution image representations. Our work uses dino as a feature extractor.

6.4.4 Multi-Resolution Networks

Several investigations, including those by [105, 106, 107, 31, 108], have delved into the wealth of multi-resolution data inherent in pyramidal Whole Slide Images (WSIs). DSMIL [107] explored the fusion of features across different resolutions, while [105] utilized patches from various resolutions within the same bag during Multiple Instance Learning (MIL). Rijthoven et al. Recently introduced a hooking mechanism [106] that links two distinct encoder-decoder networks operating with input from lower and higher resolutions. This 'hook' involves cropping contextual features and concatenating them with the feature maps in the target branch, thereby aligning pixels from different resolutions in a shared semantic space. While this mechanism is tailored for semantic segmentation tasks, it lacks applicability to classification tasks. In contrast, [31] proposes a hierarchical approach to learning from multiple resolutions. It first applies Semi-Supervised Learning (SSL) to learn features for higher resolutions, capturing finer details, before employing SSL on spatially aggregated detailed features to capture context. CD-Net [108] focuses on jointly integrating contextual features with an aggregated detailed feature representation captured at a higher resolution. Compared to those multi-resolution networks, our approach is to get the feature of different-resolution images separated by two pre-trained feature extractors.

6.5 Discussion, Limitation and Future Work

In this paper, we proposed a registration framework for aligning radiological and pathological images. Through our experimental analysis, we have demonstrated the

capability of self-supervised learning approach to bridge the resolution gap between these two modalities, facilitating accurate registration without the need for labeled training data. The fusion of high-resolution pathology data with low-resolution radiological scans have shown great promise in enhancing the diagnostic potential of medical imaging, particularly in tasks such as cancer segmentation. However, it is important to acknowledge the limitations of our work. Our experiments were conducted on limited datasets, and the fused high-resolution image was not directly generated but created through the registration process. These limitations highlight areas for future research and improvement. Moving forward, further research and validation studies will be essential to validate the clinical utility and robustness of our framework across diverse datasets and clinical scenarios. Nonetheless, the results presented here signify a significant step forward in the quest to leverage advanced ML techniques for improving medical image registration and diagnosis.

Part III

Transfer To Real Clinical Application

Overview

This part focuses on integrating deep learning models into clinical workflows and exploring federated learning to enhance collaborative model training across institutions while preserving data privacy. After extensively exploring various methods to improve segmentation accuracy, we transitioned our academic research into real-world applications in two key aspects:

- **Chapter 7: Enhancing RayStation with a Pluggable Deep Learning Framework for Inference and Training Auto Tumor Segmentation** This chapter introduces DeepRaySeg, a pluggable deep learning framework integrated with RayStation for automatic tumor segmentation. The framework includes a standalone RayStation script, a cloud drive for data exchange, and an HPC server for model training and inference. This integration is designed to streamline clinical workflows, improving the accuracy and efficiency of tumor segmentation in radiotherapy planning.

- **Chapter 8: Experimenting with FedML and NVFLARE for Federated Tumor Segmentation Challenge [109]** This chapter investigates the use of federated learning frameworks, FedML and NVFLARE, for collaborative tumor segmentation across multiple institutions. The proposed approach leverages federated learning to train models without sharing sensitive patient data, enhancing model robustness and generalizability. Evaluation results on the FeTS dataset demonstrate the potential of federated learning to improve tumor segmentation accuracy while maintaining data privacy.

Each chapter in this part details the design, implementation, and evaluation of

the proposed frameworks, highlighting significant improvements in integrating deep learning methods into clinical practice and enhancing collaborative model training through federated learning.

CHAPTER 7: Enhancing RayStation with a Pluggable Deep Learning Framework for Inference and Training Auto Tumor Segmentation

7.1 Introduction

In treatment planning, accurate tumor segmentation plays a crucial role. Precise tumor segmentation can enhance the quality of treatment plans and improve patients' survival rates. With the emergence of deep learning, the field of medical image analysis has undergone a revolutionary transformation, promising heightened accuracy and efficiency in this critical task. However, for physicians, training their own models can be challenging due to the significant gap between computer science and the medical domain.

To address this challenge, RayStation, an advanced radiation therapy planning system, provides its own machine learning training and inference capabilities. Physicians can train their models using an internal plug-in. However, this training plugin has several limitations. Firstly, it still relies on the U-net architecture, which is outdated. Secondly, training on RayStation, a machine dedicated to radiation therapy planning, increases the workload and can slow down the daily clinical workflow. Thirdly, the computational resources on the RayStation machine are limited, resulting in time-consuming model training. Lastly, it lacks the ability for physicians to customize their training preprocessing and fine-tune their models.

While RayStation provides built-in machine learning capabilities for medical image segmentation and treatment planning, there are significant reasons why these tools might not be suitable for medical professionals, especially physicians.

Firstly, RayStation's pre-defined machine learning models, although designed for general-purpose organ segmentation, present a major challenge. These models are

not continuously updated; model weights are only periodically refreshed, usually a few times a year. This can be problematic because the field of medical imaging and artificial intelligence is rapidly evolving, and new, more accurate models and techniques are being developed constantly. Physicians and healthcare professionals need access to the latest advancements in machine learning to ensure the best patient care.

Secondly, the machine learning tools in RayStation are not user-friendly for medical professionals. Physicians, who may not have a strong background in computer science or machine learning, face a steep learning curve when trying to use these tools. The lack of clarity regarding the network structures used in RayStation's machine learning models further complicates the situation. This ambiguity in model architecture and performance transparency makes it difficult for physicians to fully understand and trust the results generated by these models.

Thirdly, the one-size-fits-all approach of RayStation's machine learning may not work well for various healthcare institutions. Different medical facilities often have their own standards for medical images in terms of dimensions, formats, and annotations. This diversity of data requires customized and flexible machine learning solutions, which can adapt to the unique characteristics of each institution's data. RayStation's machine learning tools, designed for general use, cannot easily accommodate these diverse data requirements.

In summary, while RayStation offers machine learning features, the limitations in terms of model updates, user-friendliness, and adaptability to individual healthcare institutions' needs can hinder its effectiveness for medical professionals. These limitations underscore the importance of flexible, user-friendly, and up-to-date machine learning solutions in the healthcare field.

To overcome these challenges, we propose DeepRaySeg, a pluggable deep learning framework seamlessly integrated with RayStation. DeepRaySeg consists of two

components: a set of Python scripts that can be easily integrated into any RayStation installation, requiring no third-party libraries for training and inference, and a High-Performance Computing (HPC) system for accelerated training and inference. By harnessing the power of deep learning and combining it with RayStation's clinical capabilities, we introduce a novel approach that not only automates tumor segmentation but also provides medical professionals with an intuitive interface for seamless integration of these advanced techniques. Our contributions can be summarized as follows:

1. We provide a standalone DeepRaySeg script with a user-friendly interface, allowing physicians to train models and perform segmentation with a simple click.

2. We design an HPC system that automates training and inference without impacting RayStation's performance. This system also offers a distributed data parallel (DDP) training framework for deep learning.

3. We evaluate state-of-art machine learning methods for tumor segmentation and incorporate them into clinical practice. Physicians can now train their models using the latest machine learning network structures, and we assess the performance of these methods on clinical data.

4. We containerize trained models for inference and future use.

In this paper, we embark on a journey to explore the architecture, functionality, and impact of DeepRaySeg in the field of tumor segmentation, bridging the gap between artificial intelligence and clinical practice. We believe our approach can significantly reduce the divide between research and clinical application.

7.2 Design of the Plug-able Deep Learning Framework

7.2.1 Overall Architecture

Our project design integrates state-of-art machine learning techniques into the RayStation platform to improve the precision and efficiency of tumor segmentation and medical image registration. The overall architecture shown as 7.1. The architec-

ture comprises three key components, each meticulously designed to ensure seamless integration and robust performance: 1. Standalone RayStation Script for Machine Learning Training and Inference 2. Cloud drive for Data Exchange and Communication 3. High-Performance Computing (HPC) Server. We will introduce the details of each component in the following subsection.

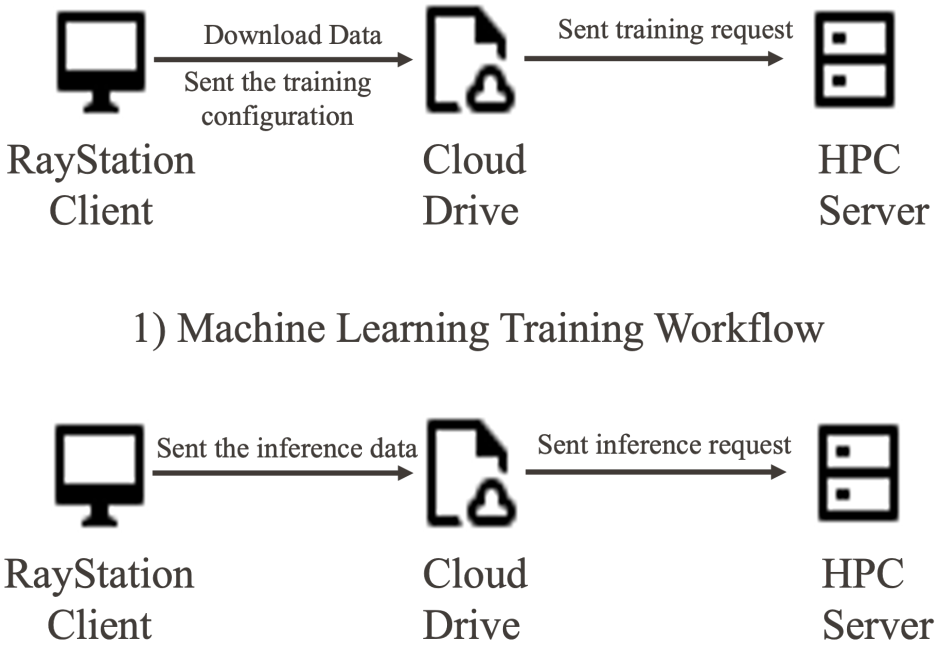


Figure 7.1: Overall Architecture Design of Our Auto Machine Learning Framework

7.2.2 Standalone RayStation Script for Machine Learning Training and Inference

The standalone RayStation script is a pivotal element in our architecture, enabling the execution of sophisticated machine learning tasks directly within the RayStation environment. It serves dual purposes:

- **Training**: The script prepares and packages the training configuration and associated datasets. These datasets are transmitted to the HPC server via the cloud drive. Leveraging its extensive computational resources, the HPC server

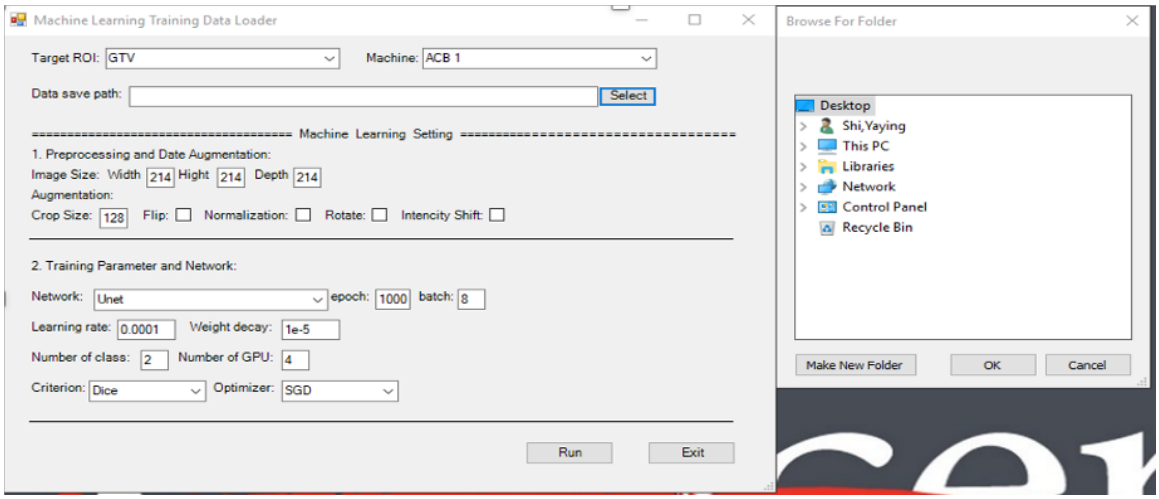


Figure 7.2: The design of training Windows UI

executes the training process, utilizing advanced algorithms to learn from the provided data.

- **Inference**: For inference tasks, the script facilitates the transfer of necessary data to the HPC server. The server processes this data using pre-trained models and returns the inference results back to RayStation for further analysis and integration, allowing real-time decision-making in clinical settings.

This integration ensures that training and inference workflows are tightly coupled with the RayStation platform, providing a user-friendly interface for medical professionals and enhancing the overall usability of the system.

7.2.3 Cloud Drive for Data Exchange and Communication

The cloud drive acts as a crucial intermediary that manages the seamless exchange of data and communication between RayStation and the HPC server. Its primary functions include:

- **Data Transfer**: Efficiently managing the upload and download of training and inference data between the RayStation client and the HPC server. The cloud drive ensures data integrity and security during transfer.

- **Training Configuration Transmission**: Sending the training configurations from RayStation to the HPC server, ensuring that the models are trained with the correct parameters and datasets.

- **Request Management**: Handling training and inference requests in a prioritized and orderly manner, ensuring that they are processed promptly and accurately.

The cloud drive is designed to ensure secure, reliable, and efficient data flow, minimizing latency and maximizing throughput, which is critical for handling large volumes of medical data.

7.2.4 High-Performance Computing (HPC) Server

The HPC server is the computational backbone of our architecture, equipped with state-of-art hardware and software to handle the demanding computational requirements of machine learning tasks. Its responsibilities include:

- **Running Training Jobs**: The server processes training requests sent from RayStation, utilizing high-performance GPUs and CPUs to train deep learning models rapidly and accurately. This involves complex operations such as data preprocessing, model training, and validation.

- **Inference Processing**: The server handles inference requests, running trained models on new data to generate predictions. These predictions are then sent back to RayStation for integration and analysis, aiding in clinical decision-making.

The HPC server's robust computational capabilities ensure that even the most complex and resource-intensive tasks are executed efficiently, providing rapid turnaround times for both training and inference.

7.2.5 Machine Learning Training and Inference Workflows

7.2.5.1 Machine Learning Training Workflow

The training workflow begins with the RayStation client sending the training configuration and data to the cloud drive. The cloud drive transmits this information to the HPC server, where the training process is executed. Upon completion, the trained model is stored on the HPC server, ready for future inference tasks.

Steps:

1. The RayStation client sends the training configuration and data to the cloud drive.
2. The cloud drive forwards the training request to the HPC server.
3. The HPC server performs the training using the provided configuration and data.
4. The trained model is stored on the HPC server for subsequent use.

7.2.5.2 Machine Learning Inference Workflow

The inference workflow is initiated by the RayStation client sending inference data to the cloud drive, which then forwards the request to the HPC server. The HPC server processes the data using the trained model and returns the predictions to the RayStation client for analysis and integration.

Steps:

1. The RayStation client sends inference data to the cloud drive.
2. The cloud drive forwards the inference request to the HPC server.
3. The HPC server processes the inference using the trained model.
4. The predictions are sent back to the RayStation client for integration and analysis.

7.2.6 Training and Inference UI Design

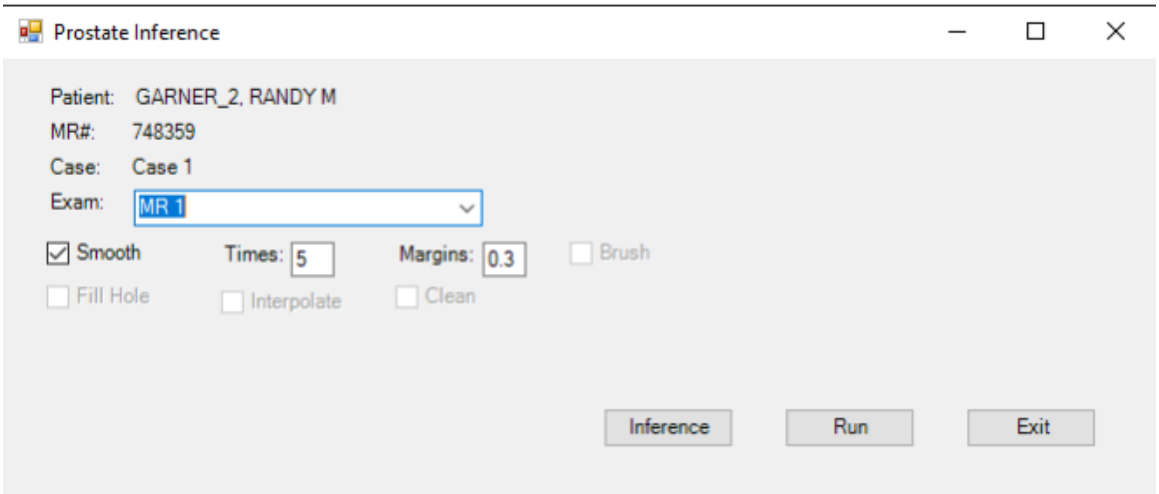


Figure 7.3: The design of Inference Windows UI

7.2.6.1 Training Window UI

The training window UI includes several key components to facilitate efficient and user-friendly machine learning training for medical image segmentation as shown in Fig 7.2. The **Region of Interest (ROI)** is the target area you want to contour in the medical images. Users can select specific regions to focus on, ensuring that the training process is tailored to the most relevant parts of the images. **Machine Selection** allows users to choose the machine or computing resource that will be used for the training process, whether it's a local machine or a high-performance computing (HPC) server.

Users can specify the **Data Path**, which is the location for storing input images, annotations, and other necessary files, ensuring organized and efficient data management. **Machine Learning Training Settings** enable users to set various parameters, such as image size, preprocessing method, network architecture, and other training-related configurations. These settings can be customized to optimize the training process for different types of medical images and segmentation tasks.

Additionally, **Training Parameters** allow users to fine-tune the training process to

optimize the model's performance and convergence. This includes adjusting learning rates, batch sizes, number of epochs, and other hyperparameters to achieve the best possible results. These elements together provide a comprehensive and user-friendly interface for conducting machine learning training for medical image segmentation.

7.2.6.2 Inference Window

Similar to the training window, we also developed an inference window as shown in Fig. 7.3. The inference window includes essential information and options for performing segmentation on a patient's MRI. **Patient Information** displays relevant details about the current patient, such as their name, age, and other medical identifiers, ensuring that the segmentation results are correctly associated with the right patient.

MRI Selection allows the user to choose the specific MRI scan that requires segmentation. Users can select from a list of available scans, ensuring that the correct data is used for analysis. Additionally, **Postprocessing Options** provide a selection of techniques that can be applied to refine the segmentation results and improve accuracy. This may include smoothing, filtering, or other image enhancement techniques.

With these components, the inference window facilitates seamless and precise segmentation of the MRI data, enabling medical professionals to obtain accurate and reliable insights for diagnosis and treatment planning.

7.3 Evaluation

We conducted a comprehensive evaluation of our plan using a private dataset for thyroid and prostate tumor segmentation. The dataset comprised 134 training cases and 44 test cases, providing a robust basis for assessing the performance of our model.

7.3.1 Accuracy Results

During the training process, the DICE score reached an impressive 0.992, indicating highly accurate segmentation of the training data. This high score reflects the model's

capability to learn and delineate tumor boundaries precisely. For the test cases, the DICE score remained consistently high at 0.93, demonstrating the model's robust performance on previously unseen data. This consistency underscores the model's generalizability and reliability in clinical scenarios where it encounters new patient data.

7.3.2 Efficiency Analysis

Training on a single GPU took approximately 23 hours, indicating the significant computational requirements of the deep learning model. This duration highlights the intensive nature of the training process, necessitating considerable computational resources. When scaled up to 2 GPUs, the training time was reduced to 12 hours. This reduction showcases the advantages of parallel processing, which allows for more efficient use of computational power and faster training times. With a powerful setup of 8 GPUs, the training time decreased further to just 5 hours, indicating the potential for even faster model convergence and deployment in a clinical setting. This scalability is crucial for practical implementation, where rapid model training can significantly impact the timeliness of treatment planning.

During the inference stage, our model exhibited impressive efficiency, requiring only 1 to 2 minutes for segmentation on a single instance. This quick turnaround time is vital for clinical applications, where timely decisions can significantly affect patient outcomes. The ability to perform rapid and accurate segmentation in a matter of minutes enhances the practicality and usability of our model in real-world settings.

These results collectively demonstrate the effectiveness and efficiency of our proposed plan for thyroid and prostate tumor segmentation, showcasing its potential for impactful applications in medical imaging and radiation therapy planning.

7.3.3 Case Study

We conducted an in-depth case study for prostate segmentation, focusing on five key regions within one patient's MRI to demonstrate the effectiveness and robustness of our framework. As illustrated in Fig. 7.4, the input patient MRI is visualized using the Raystation system, providing a detailed view of the anatomical structures.

After running our standalone inference script, which leverages advanced machine learning algorithms, the segmented image is produced as shown in Fig. 7.5. This figure highlights the automatic region of interest (ROI) contouring capabilities of our method. Specifically, the segmentation delineates the prostate in blue, the EUS in cyan-blue, the seminal vesicle in yellow, the rectum in green, and the bladder in red, demonstrating precise and accurate segmentation of these critical regions.

To further refine the segmentation results for clinical use, we applied a post-processing script designed to smooth the boundaries of the segmented regions. This step is crucial for ensuring the segmentation contours are suitable for clinical workflows and practical application in treatment planning. The visualization of these smoothed segmentation boundaries is shown in Fig. 7.6. This figure demonstrates the enhanced smoothness and clinical readiness of the segmented contours after post-processing.

This visualization case study underscores the capability of our framework to accurately segment and refine prostate and surrounding structures within MRI images. The precise segmentation and subsequent boundary smoothing are critical steps towards integrating advanced machine learning techniques into clinical radiotherapy workflows, ultimately aiming to enhance patient outcomes through improved treatment planning.

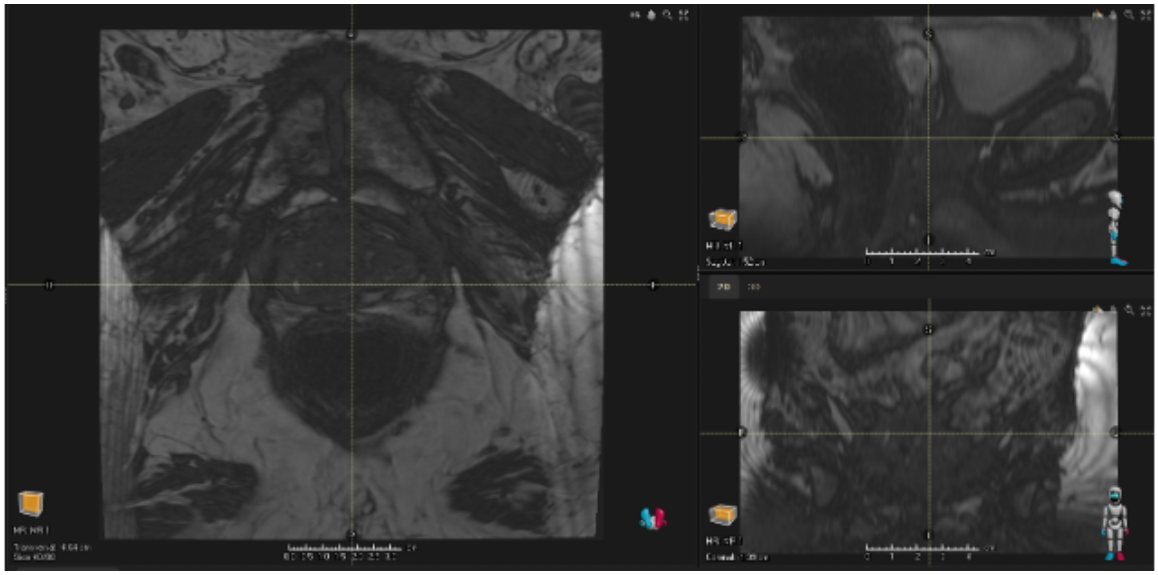


Figure 7.4: Visualization of the patient's MRI in Raystation, providing a detailed anatomical view.

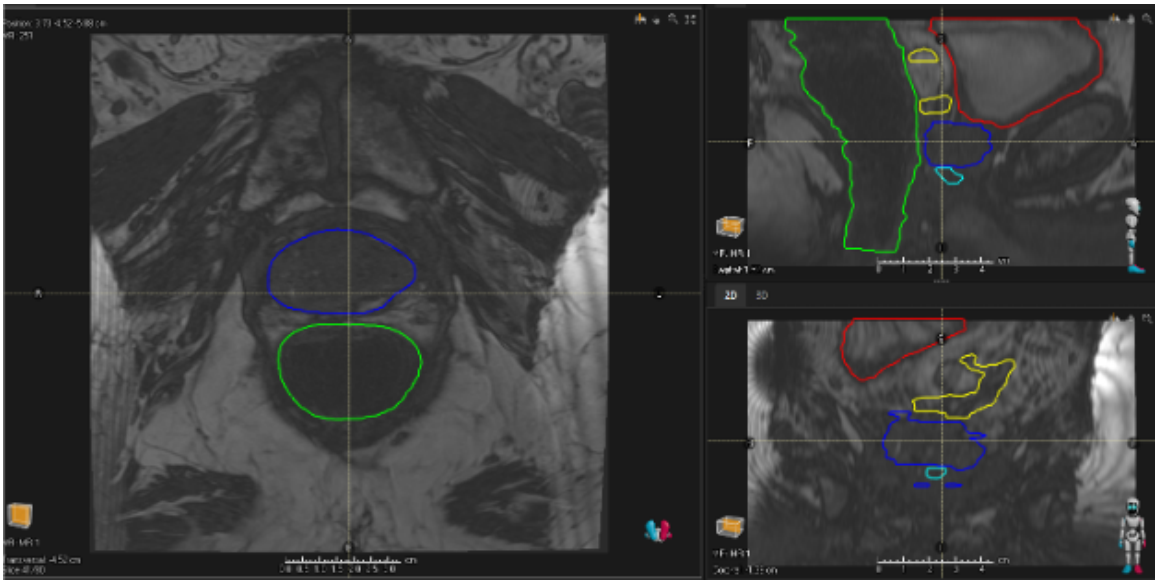


Figure 7.5: Visualization of the patient's MRI with automatic ROI contouring in Raystation. The segmentation delineates the prostate (blue), EUS (cyan-blue), seminal vesicle (yellow), rectum (green), and bladder (red).

7.4 Discussion

In this paper, we have introduced DeepRaySeg, a pluggable deep learning framework seamlessly integrated with RayStation to address the limitations of existing machine learning tools in radiation therapy planning. Our proposed solution lever-

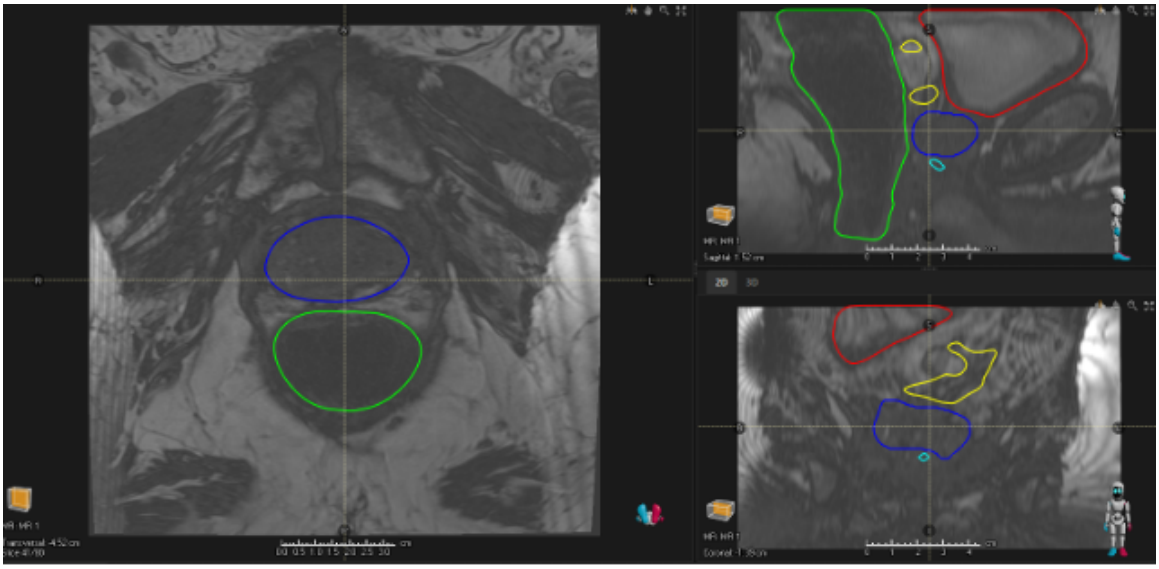


Figure 7.6: Visualization of the patient's MRI after applying smoothing post-processing in Raystation. The segmentation contours are refined for clinical application, showing the prostate (blue), EUS (cyan-blue), seminal vesicle (yellow), rectum (green), and bladder (red).

ages the latest advancements in deep learning to provide medical professionals with accurate, efficient, and user-friendly tools for tumor segmentation.

Through our evaluation, we demonstrated that DeepRaySeg achieves high accuracy in tumor segmentation, with a DICE score of 0.992 on training data and 0.93 on test data. These results underscore the robustness and generalizability of our model, ensuring reliable performance in clinical scenarios. Additionally, our framework significantly reduces training time, showcasing the advantages of utilizing high-performance computing resources.

The detailed case study on prostate segmentation further validated the effectiveness of our approach. The automatic ROI contouring and subsequent boundary smoothing processes highlighted the precision and clinical readiness of the segmented contours, making them suitable for integration into treatment planning workflows.

By addressing the challenges of model updates, user-friendliness, and adaptability to diverse data requirements, DeepRaySeg bridges the gap between computer science and the medical domain. Our standalone script and HPC system enable physicians to

train and fine-tune their models without disrupting daily clinical operations, providing a scalable and flexible solution for various healthcare institutions.

In conclusion, DeepRaySeg represents a significant step forward in the integration of artificial intelligence and clinical practice. By providing state-of-art machine learning capabilities within the RayStation environment, we empower medical professionals to enhance the precision and efficiency of tumor segmentation, ultimately improving patient outcomes in radiation therapy. Future work will focus on further optimizing the framework, expanding its capabilities to other types of medical imaging tasks, and continuing to incorporate the latest advancements in deep learning to stay at the forefront of medical technology.

CHAPTER 8: Experimenting FedML and NVFLARE for Federated Tumor Segmentation Challenge

8.1 Introduction

Brain tumor segmentation is one of the most difficult segmentation challenges in the medical image domain. It is hard to find a uniform pattern to segment tumors since tumors vary in size, type, and shape. In medical institutions, physicians use magnetic resonance imaging (MRI) to locate, track, diagnose and treat the tumor. However, physicians still need to manually contour the boundary of tumor to conduct a high-quality treatment plan. For a 3D MRI, it is time consuming for physicians to manually segment tumor slides by slides. Recently, the RSNA-ASNR-MICCAI Brain Tumor Segmentation Challenge (BraTS) is one of the competitions which provides a well-labeled brain tumor dataset for competitors to find out the best segmentation method for brain tumor [3]. It shows great quality segmentation results of CNN based deep learning method. Meanwhile, it also provided various state-of-art brain tumor segmentation algorithms and demonstrated the potential of deep learning methods.

Recent post-challenge proceedings in BraTS challenge show that most of the methods are based on UNet [8]. UNet, one of the famous methods in the medical image segmentation domain, is an encoder-decoder based deep neural network. It is widely used as a baseline architecture for most segmentation in other image segmentation methods. Many variants of UNet were proposed to be used for brain tumor segmentation by adding additional blocks. Residual 3D UNet was implemented by adding residual block on 3D UNet [55]. Densely connected UNet introduces a new densely connected layer [110]. Vnet is another approach to improvement UNet by introducing Dice Coefficient loss which is an innovation loss function broadly used in the segmen-

tation method![9]. Other approaches combined UNet with self-attention mechanisms, such as TranBTS [4] and TransUNet [14].

However, while existing works show great potential and demonstrate their capability for medical image segmentation in research and experiment studies, there are still challenges in how to effectively harness the distributed medical data and apply ML/DL in clinical applications due to data privacy [111] and data scarcity [112]. Federated learning (FL) for healthcare [113] has recently been recognized as a promising solution to address privacy and data governance challenges by enabling ML from non-co-located data. Federated learning (also known as collaborative learning) was introduced in 2017 as a deep learning technique that trains an algorithm across multiple decentralized edge devices or servers holding local data samples without exchanging them [114]. The Federated Tumor Segmentation (FeTS) 2022 challenge which is the first challenge to ever be proposed to address brain tumor segmentation by using federated learning.

In this work, we implement the training and evaluation of UNet for FeTS 2022 challenge using two federated learning frameworks, FedML [115] and NVIDIA FLARE (NVFLARE) [6]. The UNet baseline got mean dice scores of 0.734, 0.763, and 0.827 of Enhanced Tumor (ET), Tumor Core (TC), and Whole Tumor (WT) of the FeTS validation data. The FedML with FedOPT policy got 0.724, 0.701, and 0.760 of ET, TC, and WT respectively. NVFLARE with FedAVG policy got 0.724, 0.723, and 0.784 of ET, TC, and WT respectively. To compare with the best network and model, we have optimized the UNet network and training hyperparameters, with centralized training, the model of the optimized 3D UNet achieved mean dice scores of 0.811, 0.848, and 0.910 on ET, TC, and WT respectively. We however were not able to use either FedML or NVFLARE to train the models from the optimized network and hyperparameters because both FedML and NVFLARE requires much more more computing resources to train this network. This indicates the limitations

of the current FL framework of handling complicated network structures and more computation-intensive training tasks.

8.2 Design of the Baseline Network and Federated Learning Framework

In this section, we will first introduce the baseline UNet model and discuss various optimization and modification techniques. As mentioned in last section, we will use FedML and NVFLARE for federated training. We will present each federated framework, along with its training details, in the following subsections. We will also provide detailed descriptions of the spilled methods, training parameters, and aggregation policies used for the two different federated learning frameworks.

8.2.1 Baseline Network

To begin with, we implemented a baseline UNet model that is based on the original UNet architecture [8] and extended to 3 dimensions. The network architecture, shown in Fig.8.1, has a typical "U" shape with a decoder, encoder part, and a bottom layer. No additional variant blocks, such as dense blocks, residual blocks, or self-attention blocks, were included to extract features. The encoder part consists of three down-sampling blocks, each of which uses a 3D convolution operation with a ReLu activation function. The decoder part has an up-sampling block that uses transposed convolution.

The input consists of four 3D MRI images, each with a size of 128×128×128. We applied three down-sampling operations, each of which includes a convolution operation to obtain a low-resolution feature map. After each convolution operation, we applied a ReLU activation function, normalization, and zero padding. The initial filter size was set to 32. At the bottleneck, we mirrored the feature maps from the encoder to the decoder part. With three up-sampling operations, we restored the segmentation output into four class labels. The initial learning rate was set to 2e-4 and was reduced according to the formula below:

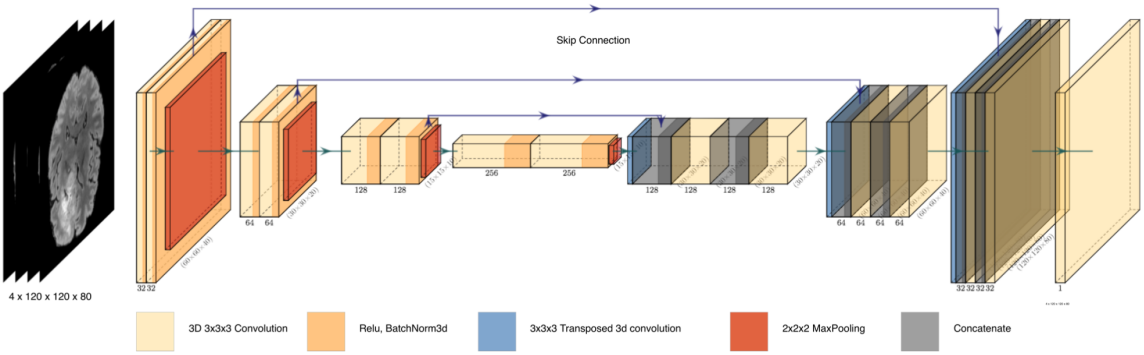


Figure 8.1: 3D UNet Network Architecture. For FeTS 2022, we used an input patch size of 128×128×128 and four MRIs as four modalities. The network has a U-shaped architecture with three down blocks for down-sampling. Each down block uses 3D convolution with a 3x3x3 filter for each modality. Up-sampling is performed using convolution transpose. The size of the feature map is shown in the encoder part. At the bottom of the U shape, the feature maps are directly copied into the encoder part.

$$a = a * \left(1 - \frac{e}{es}\right)^{0.9} \tag{8.1}$$

where e is current epochs, es is total number of epochs. We used L2 norm regularization on the convolutional kernel parameters with a weight of 1e−5. The learning rate is decayed with a schedule at 2e−4. We ran the training for a total of 1000 epochs, with each epoch having 52 iterations. We also applied pre-processing and data augmentation to the training data.

8.2.2 Optimization and Modifications

We optimized and modified the baseline UNet according to the guideline from nnUNet[74].

Firstly, we applied region-based training optimization. The training dataset has three labels: edema (label 2), necrosis (label 1), and enhancing tumor (label 4). However, the evaluation of the segmentation results is based on three regions: enhancing tumor, tumor core (enhancing tumor with necrosis), and whole tumor (enhancing tumor, necrosis, and edema). According to previous BraTS challenge meth-

ods [49, 50, 12, 54], performance can be improved if we optimized the region rather than optimized the label. Therefore, we used a sigmoid function on the three tumor regions at the last layer of the 3D UNet in order to obtain a better segmentation result.

Secondly, we increased the batch size from 2 to 24. According to previous BraTS challenge conclusions, using a lower batch size on a larger dataset can produce noisy gradients, which may reduce the over-fitting issue while also influencing performance. This is a bias-variance trade-off when choosing the batch size [42].

Thirdly, we applied more data augmentation to the training dataset. We first applied Z-score normalization to all modality 3D MRIs using the mean and standard deviation. In addition, we applied several aggressive augmentation techniques to the training data to obtain a more stable model. These techniques included:

1) random mirror flipping across the axial, coronal, and sagittal planes with a probability of 0.5;

2) random rotation with a probability of 0.5;

3) random intensity shift between [-0.1, 0.1] and scale between [0.9, 1.1];

4) random cropping of MRIs from a size of 240×240×155 to 128×128×128;

Lastly, we used batch normalization instead of instance normalization. Based on previous experiences in the BraTS challenge, the performance of the test dataset in terms of dice score was significantly lower than that of the training and validation datasets, likely due to existing domain gaps [12]. To reduce these gaps, we applied more data augmentation techniques and used batch normalization.

8.2.3 Federated Learning Framework 1: FedML

FedML is an open-source research library and benchmark for federated machine learning [115]. It provides a federated learning framework that covers several domain topics such as computer vision, natural language processing, medical image processing, finance, and more. FedML supports various different computing paradigms: on-

device training, distributed training, standalone simulation, and so on. The overview architecture of FedML is shown in Fig. 8.2.

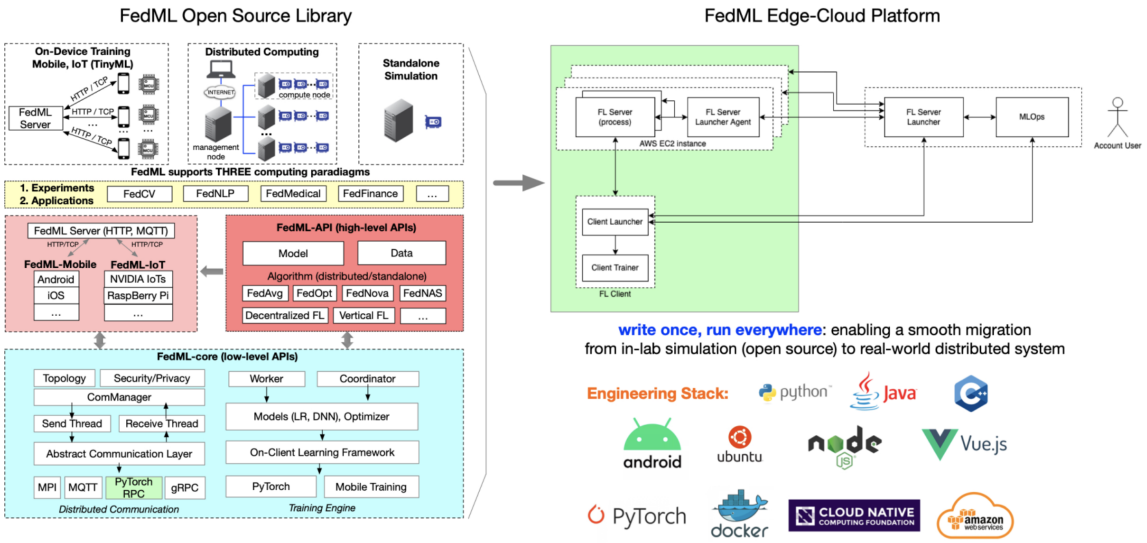


Figure 8.2: FedML Core Architecture Design [5]

In this work, we adapted FedML to train FeTS 2022 challenge data on one computer for a standalone federated learning simulation. To use FedML, we split the challenge dataset into several sites based on the medical institution ID provided by the official FeTS training dataset. Each site represents a distinct medical institution during the simulation training process. We also randomly spilt the dataset into train and test at a ratio of 0.8. Then we applied the same pre-process to the training dataset that is the same as centralized training on 3D UNet. We applied two aggregation policies (FedAvg [114] and FedOPT [116]) for two training. We ran the training for 1000 rounds. The learning rate was set as 1e−4. All the other training parameters were consistent with those used in centralized UNet training.

8.2.4 Federated Learning Framework 2: NVFLARE

NVFLARE is an open-source Federated Learning framework that allows researchers to adapt their ML/DL methods to a federated paradigm and build a distributed collaboration [6]. It provides a robust SDK for users to deploy a real-world federated

learning framework with high privacy security and supports federated training simulations on a standalone computer. However, users must assign one site per GPU, which means the number of sites is limited to the number of GPUs when compared to FedML. The overall architecture of NVFLARE is shown in Fig. 8.3

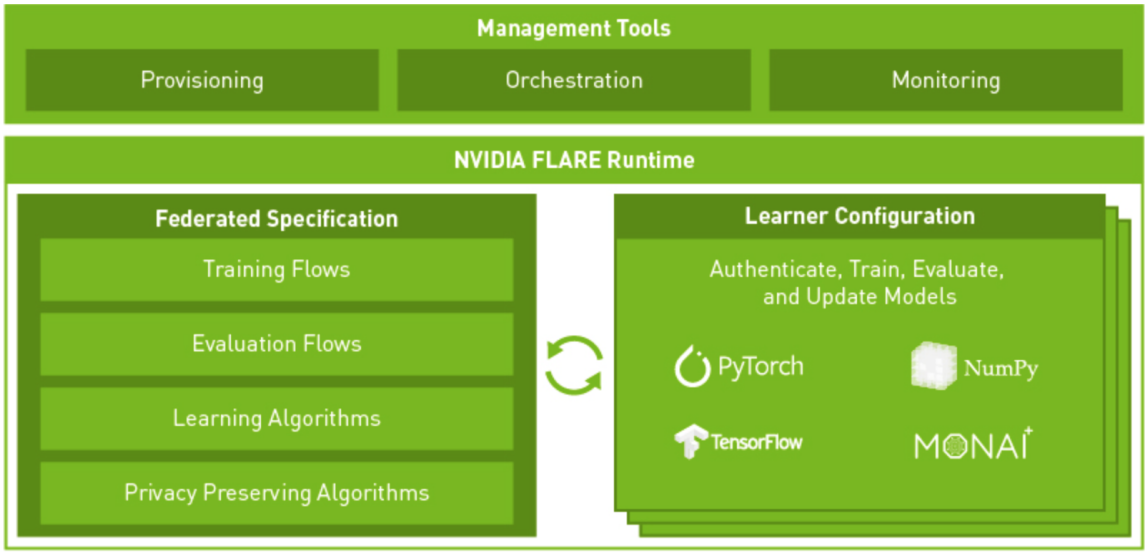


Figure 8.3: Overall Core Architecture Design of NVFLARE [6]

In this work, we made several changes to NVFLARE for one computer standalone federated learning simulation on the FeTS 2022 challenge data. Similar to FedML, we randomly split the challenge dataset according to the number of GPUs on their machine, which is 4. Thus, the number of sites for NVFLARE training is 4. We also applied the same pre-processing and augmentation to the training data, used FedAvg [114] as the aggregation policy for training, and ran the training with the same round number as FedML. The learning rate was set as 1e−4 and the rest of the training parameters were the same as centralized UNet training.

8.3 Evaluation

8.3.1 FeTS Dataset

We used the FeTS 2022 challenge datasets for evaluation [117, 118, 3, 119]. The FeTS 2022 training dataset comprises of 1251 cases. Each training case includes one

segmentation ground truth, and four modalities of 3D MRIs namely T1-weighted, T2-weighted, T1 contrast-enhanced, and T2 FLAIR. Our model was trained solely with the FeTS training dataset, without incorporating any other public or private datasets. In this work, we treated four 3D MRIs as four input channels in the computer vision tasks, and each volume has dimensions of $240 \times 240 \times 155$. The ground truth labels have 4 classes: label 0 represents the background, label 1 represents enhancing tumor, label 4 represents necrosis, and label 2 represents edema. The validation dataset consists of 219 cases with no ground truth provided. For the evaluation metrics, we use Dice score and Hausdorff distances as required by the challenge.

8.3.2 Implementation Details

Our model was implemented by Python 3.8.5 with PyTorch 1.9.0. For each training case, we applied several data augmentations including linear normalization, random crop, random clip, random intensity shift, and so on. Each image was preprocessed by cropping to a size of $128 \times 128 \times 128$ before being fed into the segmentation model. The network was trained from scratch using four NVIDIA A100 GPUs (40GB VRAM) for 1000 epochs, with a batch size of 24. The same environment was employed for both federated and centralized training, which were carried out on the same machine. PyTorch Distributed Data Parallel was used for centralized training. The entire training process took approximately 25 hours for 1000 epochs. Moreover, we applied Test-Time Augmentation (TTA) for the challenge.

8.3.3 Federated Validation Result

The performance of different training methods on FeTS 2022 validation data is reported as Table 8.1. The centralized method yielded a better performance than the two Federated Learning frameworks. Among these frameworks, NVFLARE had better performance than FedML. We speculated that the reason for this could be related to the data splitting methods. FedML split the data based on institution

ID, which is more representative of real-world scenarios. However, the datasets split by FedML were more unbalanced compared to NVFLARE, which evenly split the data into 4 sites. In terms of the aggregation policy, FedOPT performed slightly better than FedAvg. The optimizations and modifications applied resulted in a significant improvement in performance compared to the model that was submitted to the challenge.

Table 8.1: Dice score and Hausdorff distance on FeTS 2022 validation dataset. ET, TC, WT present enhancing tumor, tumor core, and whole tumor respectively.

	Dice			Hausdorff95 (mm)		
validation dataset	ET	TC	WT	ET	TC	WT
Unet baseline	0.7335	0.7633	0.8266	32.6349	25.3428	17.4926
FedML(FedAvg)	0.7267	0.6772	0.7492	34.6926	40.0021	26.0513
FedML(FedOPT)	0.7235	0.7012	0.7597	33.5342	37.9832	25.7833
NVFLARE	0.7245	0.7231	0.7847	32.8643	34.1002	17.9282
Challenge	0.8113	0.8482	0.9101	18.0867	11.4861	4.39933

8.3.4 Validation Phase Result

In terms of challenge, we have this section specially listed for the competition result. The performance of our model on FedTS 2022 validation data is reported as Table 8.2. Our model reached the mean dice scores of 0.81, 0.84, and 0.91 on ET, TC, and WT respectively. From the deviation, it is evident that the enhancing tumor and tumor core have high variation. By analyzing the median and 25th quantile results, we can see that our model is stable across the entire validation dataset. Upon reviewing the complete validation score table, we found that our model had a score of 0 for enhancing tumor on some validation cases, which also had lower dice scores such as cases 1689, 1797, and so on. We will investigate further the reasons for the differences in performance on these cases.

8.3.5 Qualitative Results

We selected some validation cases that were close to the dice scores we presented in the last section for visualization. These validation cases were chosen as the best,

Table 8.2: Dice score and Hausdorff distance on FeTS 2022 validation dataset. ET, TC, and WT present enhancing tumor, tumor core, and whole tumor respectively.

	Dice			Hausdorff95 (mm)		
validation dataset	ET	TC	WT	ET	TC	WT
mean	0.8113	0.8482	0.9101	18.0867	11.4861	4.3993
stdev	0.2477	0.2265	0.0870	73.6223	50.1051	7.0679
median	0.8922	0.9326	0.9357	1.4142	2.2361	2.4495
25th quantile	0.8266	0.8534	0.8935	1.0000	1.0000	1.7321
75th quantile	0.9416	0.9629	0.9586	2.8284	4.3298	4.2426

worst, median, and 75th and 25th percentiles based on their Dice scores. As shown in Fig. 8.4, the overall segmentation quality is high. The best case is FeTS2022_00153 with dice scores of 0.983, 0.992, 0.988 on ET, TC, and WT. The 75th percentile case is FeTS2022_00129 with dice scores of 0.942, 0.960, 0.953 on ET, TC, and WT. The median case is FeTS2022_00256 with dice scores of 0.869, 0.882, 0.895 on ET, TC, and WT. The 25th percentile case is FeTS2022_00129 with dice scores of 0.824, 0.866, 0.905 on ET, TC, and WT. The worst case is FeTS2022_00213 with dice scores of 0.081, 0.081, 0.506 on ET, TC, and WT.

8.3.6 Test Phase Result

The performance of our model on FedTS 2022 test data is reported as Table 8.3. In the test phase, our model was evaluated on 30 different sites, representing 30 different medical institutions. Each site has an unknown number of test cases with undisclosed ground truth. The performance that we listed in Table 8.3 is the mean value of all test cases for each site including mean value, the standard deviation, the median value, the best, worst, 25th quantile, and 75th quantile. The total test cases on each local site are still unknown to us. Compared to validation datasets, our model had better performance in terms of mean value among the real test cases. Our model also reached the mean dice scores of 0.854, 0.869, and 0.913 on ET, TC, and WT respectively. From the standard deviation perspective, the values indicate that the performance is stable across all 30 sites, with standard deviations of only 0.06,

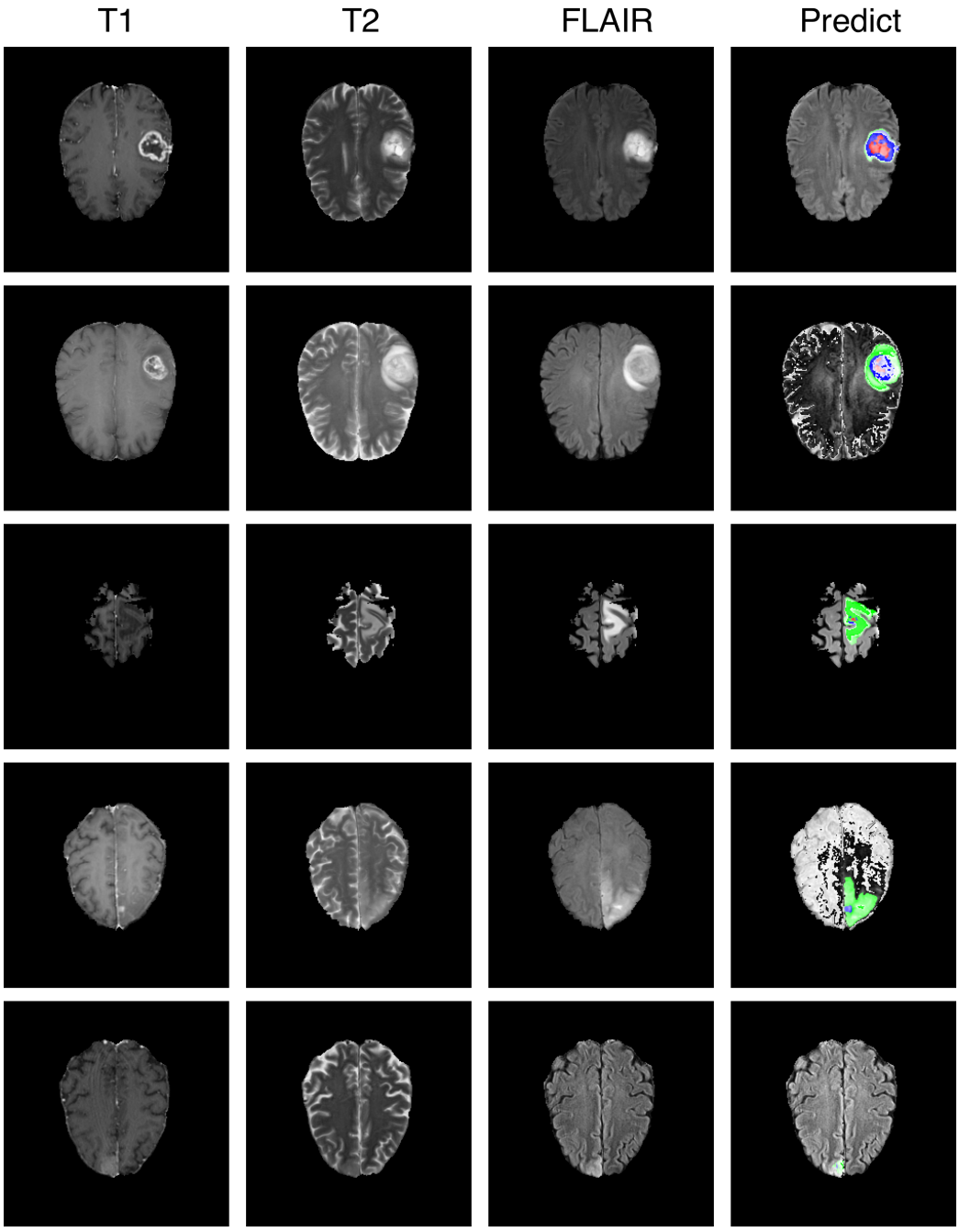


Figure 8.4: The visualization of qualitative results is presented as follows: For each row, the raw T1 image is shown in the first left column. The second column to the left is the raw T2 image. The T2 Flair image is next to the T2 image. The predicted outcome is in the last right column. Edema is shown in green, enhancing tumor in red, and necrosis/non-enhancing tumor in blue. From the first row to the last row, we have displayed the best, 75th percentile, median, 25th percentile, and worst validation cases, respectively.

0.08, and 0.04 for ET, TC, and WT, respectively. By analyzing the median, worst site, and 25th percentile results, we can see that our model is stable in most medical institutions. However, the values of the worst sites decrease the overall performance of the global model. In the future, we will work to optimize the performance of the worst sites.

Table 8.3: Dice score and Hausdorff distance on FeTS 2022 test dataset. ET, TC, and WT present enhancing tumor, tumor core, and whole tumor respectively.

	Dice			Hausdorff95 (mm)		
test dataset	ET	TC	WT	ET	TC	WT
mean	0.8543	0.8688	0.9133	11.7867	12.2827	6.3178
stdev	0.0683	0.0806	0.0470	12.8352	14.5118	5.3278
median	0.8677	0.8931	0.9253	4.7401	5.9342	4.4666
min	0.7151	0.6416	0.7822	47.5965	58.6337	25.0028
max	0.9477	0.9626	0.9650	1.2649	1.8225	1.6413
25th quantile	0.8016	0.8288	0.9053	21.3416	16.4353	7.8095
75th quantile	0.9075	0.9268	0.9460	2.2018	2.9832	3.3202

8.4 Discussion

In this work, we presented an optimized and modified UNet model for improved segmentation on multi-modality 3D MRI. Our results in the validation phase were generally positive. We also adapted our method to two Federated Learning frameworks, FedML and NVFLARE, and both frameworks performed well and showed potential for federated training on the FeTS challenge data.

In the future, there are several improvements we can make to our method. One aspect we can focus on is implementing a post-processing method for cases where the enhancing tumor is empty. These cases can result in undefined dice scores due to division by 0. To address this, we can apply an algorithm that replaces all empty enhancing tumor predictions with tumor core labels. By removing the empty enhancing tumor and replacing it with necrosis, we can ensure that these voxels are still considered part of the tumor core. This will not affect the performance of the tumor core and improve our Rank score by removing those true positive predictions.

Additionally, we can improve our method by using batch dice instead of mini-batch dice. By using batch dice, we can consider the entire dataset as one large sample that is trained in one batch. This approach balances the trade-off between bias and variation.

In conclusion, we achieved mean dice scores of 0.854, 0.869, and 0.913 on ET, TC, and WT, respectively. We will address the proposed improvements in the next challenge.

CHAPTER 9: Conclusion and Discussion

In this dissertation, we addressed the critical tasks of medical image segmentation and registration, aiming to enhance the precision and efficiency of radiotherapy planning. By leveraging advanced deep learning techniques, we developed and integrated novel frameworks and models into the RayStation platform, specifically designed to overcome existing limitations in clinical workflows.

In the field of segmentation, we developed an ensemble approach for 3D medical image segmentation of brain tumors inspired by stacking ensemble learning. We extended the U-Net architecture into a stacking feature U-Net using different scales of input images. Furthermore, we developed the SMOE-MPLS network, which incorporates sparse Mixture of Experts (MoE) with Vision Transformers (ViT) for tumor segmentation in the brain and Head and Neck regions. Our proposed methods not only improved the accuracy of tumor segmentation tasks but also increased the speed of training and inference for machine learning models.

To further improve our segmentation results, we sought more precise medical images, such as pathology images. We began by performing registration between MRI and pathology images to use the detailed pathology images to guide tumor segmentation on MRI. To achieve this, we developed a dual-domain feature extractor for both pathology images and MRI. By utilizing a feature matching network, we registered images of different resolutions, thereby improving the accuracy of downstream segmentation tasks. Extending this work, we used this method to enhance MRI to cell-level precision. We evaluated this approach using SSIM, MI, and resolution comparisons, achieving a 39-fold improvement in MRI resolution and enhanced lesion segmentation accuracy.

One of our key contributions, DeepRaySeg, is a pluggable deep learning framework that facilitates seamless machine learning training and inference within RayStation. This framework includes a standalone script and an HPC system, enabling efficient and scalable model training without disrupting daily clinical operations. Through comprehensive evaluations, DeepRaySeg demonstrated high accuracy in tumor segmentation, achieving a DICE score of 0.992 on training data and 0.93 on test data. These results underscore the robustness and generalizability of our models, ensuring reliable performance in clinical scenarios.

Additionally, our framework significantly reduced training time by utilizing high-performance computing resources, showcasing the advantages of parallel processing. A detailed case study on prostate segmentation further validated the effectiveness of our approach, highlighting the precision and clinical readiness of the segmented contours, making them suitable for integration into treatment planning workflows.

Furthermore, we explored the potential of federated learning for secure and efficient data sharing across medical institutions, addressing privacy concerns and enhancing collaboration. This approach ensures that models can be trained on diverse datasets without compromising patient confidentiality, paving the way for more robust and generalizable solutions in medical imaging.

Overall, this dissertation provides comprehensive solutions to the challenges of medical image segmentation and registration using deep learning. The methodologies developed in this research hold significant potential to improve patient outcomes by enabling more accurate and individualized treatment plans. Future work will focus on further optimizing the framework, expanding its capabilities to other types of medical imaging tasks, and continuing to incorporate the latest advancements in deep learning to stay at the forefront of medical technology.

CHAPTER 10: Future Work

In my future work, I plan to continue advancing segmentation techniques by developing and applying a unified model that caters to various medical imaging needs. Since Vision Transformers (ViT) are the current foundational method, I will develop its variants to further improve segmentation accuracy. Recently, Meta has pioneered a unified model for all types of medical images based on the Segment Anything Model (SAM). I propose a similar concept of a unified model but with a different approach. My model will focus on optimizing the training and inference processes of transformer networks, aiming to make transformers parallelizable. The goal is to achieve real-time segmentation for various types of tumors, significantly enhancing post-treatment planning in clinical settings. By harnessing the power of parallel computing, I aim to reduce the latency in processing medical images, thereby providing faster and more accurate diagnoses.

I also intend to expand my research in advanced registration and reconstruction of pathology imaging. My previous work required external pathology data for fusion. By developing image registration methods, I aim to generate pathology images from radiology images to achieve cell-level precision. This approach will greatly improve tasks such as segmentation by providing more detailed and accurate imagery. The focus will also remain on providing a cost-effective, painless, and non-invasive solution for obtaining pathology images. Additionally, this method will generate a large number of pathology images, which are currently expensive to obtain. It will address data challenges in machine learning, boosting overall performance and ensuring that the models can handle diverse and complex datasets.

For smart clinical applications, I will maintain close collaboration with clinical

physicians to ensure that my innovations meet real-world needs. This collaboration is crucial for translating state-of-the-art computer science technologies into practical medical imaging solutions. My goal is to tackle specific clinical issues through creative and effective approaches, ensuring that the developed technologies are both usable and beneficial in a clinical setting.

Furthermore, I will explore the potential of integrating computer vision (CV) and large language models (LLMs) to enable automated diagnosis. This integration aims to streamline and improve clinical decision-making by providing comprehensive diagnostic tools that can interpret medical images and patient data with high accuracy. The synergy between CV and LLMs has the potential to revolutionize the way diagnoses are made, making them more efficient and reliable.

Through these initiatives, I aim to bridge the gap between research and clinical practice. By focusing on real-world applications and maintaining a patient-centered approach, I hope to advance patient care and outcomes. My future work will continue to push the boundaries of what is possible in medical imaging, ensuring that technological advancements translate into tangible benefits for patients and healthcare providers alike.

REFERENCES

[1] Y. Cai, Y. Long, Z. Han, M. Liu, Y. Zheng, W. Yang, and L. Chen, "Swin unet3d: a three-dimensional medical image segmentation network combining vision transformer and convolution," *BMC Medical Informatics and Decision Making*, vol. 23, no. 1, pp. 1–13, 2023.

[2] K. T. Islam, S. Wijewickrema, and S. O'Leary, "A deep learning based framework for the registration of three dimensional multi-modal medical images of the head," *Scientific Reports*, vol. 11, no. 1, p. 1860, 2021.

[3] U. Baid, S. Ghodasara, M. Bilello, S. Mohan, E. Calabrese, E. Colak, K. Farahani, J. Kalpathy-Cramer, F. C. Kitamura, S. Pati, *et al.*, "The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification," *arXiv preprint arXiv:2107.02314*, 2021.

[4] W. Wang, C. Chen, M. Ding, J. Li, H. Yu, and S. Zha, "Transbts: Multimodal brain tumor segmentation using transformer," *arXiv preprint arXiv:2103.04430*, 2021.

[5] C. He, "Fedml," 2022.

[6] NVIDIA, "Nvidia federated learning application runtime environment," 2022.

[7] F. Bray, M. Laversanne, H. Sung, J. Ferlay, R. L. Siegel, I. Soerjomataram, and A. Jemal, "Global cancer statistics 2022: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: a cancer journal for clinicians*, vol. 74, no. 3, pp. 229–263, 2024.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, pp. 234–241, 2015.

[9] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 fourth international conference on 3D vision (3DV)*, pp. 565–571, 2016.

[10] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.

[11] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pp. 3–11, Springer, 2018.

[12] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnu-net: a self-configuring method for deep learning-based biomedical image segmentation," *Nature methods*, vol. 18, no. 2, pp. 203–211, 2021.

[13] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *CVPR*, pp. 7794–7803, 2018.

[14] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "Transunet: Transformers make strong encoders for medical image segmentation," *arXiv preprint arXiv:2102.04306*, 2021.

[15] A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, B. Landman, H. R. Roth, and D. Xu, "Unetr: Transformers for 3d medical image segmentation," in *IEEE/CVF WACV*, pp. 574–584, 2022.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[17] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *NeurIPS*, vol. 34, 2021.

[18] A. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer, "How to train your vit? data, augmentation, and regularization in vision transformers," *arXiv preprint arXiv:2106.10270*, 2021.

[19] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, pp. 10012–10022, 2021.

[20] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-unet: Unet-like pure transformer for medical image segmentation," *arXiv preprint arXiv:2105.05537*, 2021.

[21] Y. Bengio, "Deep learning of representations: Looking forward," in *International conference on statistical language and speech processing*, pp. 1–37, 2013.

[22] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[23] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.

[24] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "Gshard: Scaling giant models with conditional computation and automatic sharding," *arXiv preprint arXiv:2006.16668*, 2020.

[25] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *arXiv preprint arXiv:2101.03961*, 2021.

[26] A. Abbas and Y. Andreopoulos, "Biased mixtures of experts: Enabling computer vision inference under data transfer limitations," *IEEE Transactions on Image Processing*, vol. 29, pp. 7656–7667, 2020.

[27] K. Ahmed, M. H. Baig, and L. Torresani, "Network of experts for large-scale image categorization," in *ECCV*, pp. 516–532, 2016.

[28] S. Pavlitskaya, C. Hubschneider, M. Weber, R. Moritz, F. Huger, P. Schlicht, and M. Zollner, "Using mixture of expert models to gain insights into semantic segmentation," in *IEEE/CVF Conference on CVPR Workshops*, pp. 342–343, 2020.

[29] X. Wang, F. Yu, L. Dunlap, Y.-A. Ma, R. Wang, A. Mirhoseini, T. Darrell, and J. E. Gonzalez, "Deep mixture of experts via shallow embedding," in *Uncertainty in Artificial Intelligence*, pp. 552–562, PMLR, 2020.

[30] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "Condconv: Conditionally parameterized convolutions for efficient inference," *arXiv preprint arXiv:1904.04971*, 2019.

[31] C. Riquelme, J. Puigcerver, B. Mustafa, M. Neumann, R. Jenatton, A. S. Pinto, D. Keysers, and N. Houlsby, "Scaling vision with sparse mixture of experts," *arXiv preprint arXiv:2106.05974*, 2021.

[32] R. Rasti, A. Mehridehnavi, H. Rabbani, and F. Hajizadeh, "Convolutional mixture of experts model: A comparative study on automatic macular diagnosis in retinal optical coherence tomography imaging," *Journal of medical signals and sensors*, vol. 9, no. 1, p. 1, 2019.

[33] R. Rasti, A. Mehridehnavi, H. Rabbani, and F. Hajizadeh, "Wavelet-based convolutional mixture of experts model: An application to automatic diagnosis of abnormal macula in retinal optical coherence tomography images," in *MVIP*, pp. 192–196, 2017.

[34] Y. Hiramatsu, K. Hotta, A. Imanishi, M. Matsuda, and K. Terai, "Cell image segmentation by integrating multiple cnns," in *IEEE Conference on CVPR Workshops*, pp. 2205–2211, 2018.

[35] P. Afshar, F. Naderkhani, A. Oikonomou, M. J. Rafiee, A. Mohammadi, and K. N. Plataniotis, "Mixcaps: A capsule network-based mixture of experts for lung nodule malignancy prediction," *Pattern Recognition*, vol. 116, p. 107942, 2021.

[36] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[37] R. Caruna, "Multitask learning: A knowledge-based source of inductive bias," in *Machine learning: Proceedings of the tenth international conference*, pp. 41–48, 1993.

[38] M. Long, Z. Cao, J. Wang, and P. S. Yu, "Learning multiple tasks with multilinear relationship networks," *Advances in neural information processing systems*, vol. 30, 2017.

[39] L. Duong, T. Cohn, S. Bird, and P. Cook, "Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser," in *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: short papers)*, pp. 845–850, 2015.

[40] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings 1*, pp. 1–15, Springer, 2000.

[41] J. Ma, Y. He, F. Li, L. Han, C. You, and B. Wang, "Segment anything in medical images," *Nature Communications*, vol. 15, no. 1, p. 654, 2024.

[42] Y. Shi, C. Micklisch, E. Mushtaq, S. Avestimehr, Y. Yan, and X. Zhang, "An ensemble approach to automatic brain tumor segmentation," in *International MICCAI Brainlesion Workshop*, pp. 138–148, Springer, 2021.

[43] Y. Shi, X. Zhang, and Y. Yan, "Stacking feature maps of multi-scaled medical images in u-net for 3d head and neck tumor segmentation," in *3D Head and Neck Tumor Segmentation in PET/CT Challenge*, pp. 77–85, Springer, 2022.

[44] A. Bousselham, O. Bouattane, M. Youssfi, and A. Raihani, "Towards reinforced brain tumor segmentation on mri images based on temperature changes on pathologic area," *International journal of biomedical imaging*, vol. 2019, 2019.

[45] H. S. Abdulbaqi, M. Z. M. Jafri, A. F. Omar, K. N. Mutter, L. K. Abood, and I. S. B. Mustafa, "Segmentation and estimation of brain tumor volume in computed tomography scan images using hidden markov random field expectation maximization algorithm," in *2015 IEEE student conference on research and development (SCOReD)*, pp. 55–60, IEEE, 2015.

[46] S. Bauer, C. Seiler, T. Bardyn, P. Buechler, and M. Reyes, "Atlas-based segmentation of brain tumor images using a markov random field-based tumor growth model and non-rigid registration," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pp. 4080–4083, IEEE, 2010.

[47] N. Mathur, S. Mathur, and D. Mathur, "A novel approach to improve sobel edge detector," *Procedia Computer Science*, vol. 93, pp. 431–438, 2016.

[48] M. Angulakshmi and G. Lakshmi Priya, "Automated brain tumour segmentation techniques a review," *International Journal of Imaging Systems and Technology*, vol. 27, no. 1, pp. 66–77, 2017.

[49] G. Wang, W. Li, S. Ourselin, and T. Vercauteren, "Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks," in *International MICCAI brainlesion workshop*, pp. 178–190, Springer, 2017.

[50] K. Kamnitsas, W. Bai, E. Ferrante, S. McDonagh, M. Sinclair, N. Pawlowski, M. Rajchl, M. Lee, B. Kainz, D. Rueckert, *et al.*, "Ensembles of multiple models and architectures for robust brain tumour segmentation," in *International MICCAI brainlesion workshop*, pp. 450–462, Springer, 2017.

[51] K. Kamnitsas, E. Ferrante, S. Parisot, C. Ledig, A. V. Nori, A. Criminisi, D. Rueckert, and B. Glocker, "Deepmedic for brain tumor segmentation," in *International workshop on Brainlesion: Glioma, multiple sclerosis, stroke and traumatic brain injuries*, pp. 138–149, Springer, 2016.

[52] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[53] F. Isensee, P. Kickingereder, W. Wick, M. Bendszus, and K. H. Maier-Hein, "No new-net," in *International MICCAI Brainlesion Workshop*, pp. 234–244, Springer, 2018.

[54] A. Myronenko, "3d mri brain tumor segmentation using autoencoder regularization," in *International MICCAI Brainlesion Workshop*, pp. 311–320, Springer, 2018.

[55] A. Wolny, L. Cerrone, A. Vijayan, R. Tofanelli, A. V. Barro, M. Louveaux, C. Wenzl, S. Strauss, D. Wilson-Sánchez, R. Lymbouridou, S. S. Steigleder, C. Pape, A. Bailoni, S. Duran-Nebreda, G. W. Bassel, J. U. Lohmann, M. Tsiantis, F. A. Hamprecht, K. Schneitz, A. Maizel, and A. Kreshuk, "Accurate and versatile 3d segmentation of plant tissues at cellular resolution," *eLife*, vol. 9, July 2020.

[56] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*, pp. 424–432, Springer, 2016.

[57] K. Lee, J. Zung, P. Li, V. Jain, and H. S. Seung, "Superhuman accuracy on the snemi3d connectomics challenge," *arXiv preprint arXiv:1706.00120*, 2017.

[58] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[60] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, *et al.*, "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE transactions on medical imaging*, vol. 34, no. 10, pp. 1993–2024, 2014.

[61] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. S. Kirby, J. B. Freymann, K. Farahani, and C. Davatzikos, "Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features," *Scientific data*, vol. 4, no. 1, pp. 1–13, 2017.

[62] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos, "Segmentation labels and radiomic features for the pre-operative scans of the tcga-lgg collection," *The cancer imaging archive*, vol. 286, 2017.

[63] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos, "Segmentation labels and radiomic features for the pre-operative scans of the tcga-gbm collection. the cancer imaging archive," *Nat Sci Data*, vol. 4, p. 170117, 2017.

[64] J. Chang, X. Zhang, M. Ye, D. Huang, P. Wang, and C. Yao, "Brain tumor segmentation based on 3d unet with multi-class focal loss," in *2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–5, IEEE, 2018.

[65] S. Qamar, H. Jin, R. Zheng, P. Ahmad, and M. Usama, "A variant form of 3d-unet for infant brain segmentation," *Future Generation Computer Systems*, vol. 108, pp. 613–623, 2020.

[66] caner.net, "Head and neck cancer: Statistics," 2020.

[67] cancerresearchuk, "Survival for laryngeal cancer," 2020.

[68] S. D. Yanowitz and A. M. Bruckstein, "A new method for image segmentation," *Computer Vision, Graphics, and Image Processing*, vol. 46, no. 1, pp. 82–95, 1989.

[69] T. Leung and J. Malik, "Contour continuity in region based image segmentation," in *European Conference on Computer Vision*, pp. 544–559, Springer, 1998.

[70] A. A. Farag, "Edge-based image segmentation," *Remote sensing reviews*, vol. 6, no. 1, pp. 95–121, 1992.

[71] V. Oreiller, V. Andrearczyk, M. Jreige, S. Boughdad, H. Elhalawani, J. Castelli, M. Vallières, S. Zhu, J. Xie, Y. Peng, *et al.*, "Head and neck tumor segmentation in pet/ct: the hecktor challenge," *Medical image analysis*, vol. 77, p. 102336, 2022.

[72] V. Andrearczyk, V. Oreiller, S. Boughdad, C. C. L. Rest, H. Elhalawani, M. Jreige, J. O. Prior, M. Vallières, D. Visvikis, M. Hatt, *et al.*, "Overview of the hecktor challenge at miccai 2022: automatic head and neck tumor segmentation and outcome prediction in pet/ct images," in *3D Head and Neck Tumor Segmentation in PET/CT Challenge*, pp. 1–37, Springer, 2023.

[73] R. Beare, B. Lowekamp, and Z. Yaniv, "Image segmentation, registration and characterization in r with simpleitk," *Journal of statistical software*, vol. 86, 2018.

[74] F. Isensee, P. F. Jäger, P. M. Full, P. Vollmuth, and K. H. Maier-Hein, "nnu-net for brain tumor segmentation," in *International MICCAI Brainlesion Workshop*, pp. 118–132, Springer, 2020.

[75] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (bit): General visual representation learning," in *ECCV*, pp. 491–507, 2020.

[76] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:1910.10683*, 2019.

[77] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *NeurIPS*, vol. 33, pp. 1877–1901, 2020.

[78] H. Hu, Z. Zhang, Z. Xie, and S. Lin, "Local relation networks for image recognition," in *2019 IEEE/CVF ICCV*, pp. 3463–3472, 2019.

[79] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," in *NeurIPS*, 2019.

[80] Y. Tang, D. Yang, W. Li, H. R. Roth, B. Landman, D. Xu, V. Nath, and A. Hatamizadeh, "Self-supervised pre-training of swin transformers for 3d medical image analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20730–20740, 2022.

[81] Y. Ou, Y. Yuan, X. Huang, S. T. Wong, J. Volpi, J. Z. Wang, and K. Wong, "Patcher: Patch transformers with mixture of experts for precise medical image segmentation," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part V*, pp. 475–484, Springer, 2022.

[82] Y. Shi, A. Gottipati, and Y. Yan, "Path-ct registration with self-supervised vision transformer," in *2024 IEEE 21th International Symposium on Biomedical Imaging (ISBI)*, IEEE, 2024.

[83] Y. Shi, S. Das, and Y. Yan, "Upscaling prostate cancer mri images to cell-level resolution using self-supervised learning," in *The MICCAI Workshop on Computational Pathology with Multimodal Data*, Springer, 2024.

[84] D. Mahapatra, B. Antony, S. Sedai, and R. Garnavi, "Deformable medical image registration using generative adversarial networks," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 1449–1453, IEEE, 2018.

[85] A. Hering, L. Hansen, T. C. Mok, A. C. Chung, H. Siebert, S. Häger, A. Lange, S. Kuckertz, S. Heldmann, W. Shao, *et al.*, "Learn2reg: comprehensive multi-task medical image registration challenge, dataset and evaluation in the era of deep learning," *IEEE Transactions on Medical Imaging*, vol. 42, no. 3, pp. 697–712, 2022.

[86] W. Shao, L. Banh, C. A. Kunder, R. E. Fan, S. J. Soerensen, J. B. Wang, N. C. Teslovich, N. Madhuripan, A. Jawahar, P. Ghanouni, *et al.*, "Prosregnet: A deep learning framework for registration of mri and histopathology images of the prostate," *Medical image analysis*, vol. 68, p. 101919, 2021.

[87] I. Yoo, D. G. Hildebrand, W. F. Tobin, W.-C. A. Lee, and W.-K. Jeong, "ssemnet: serial-section electron microscopy image registration using a spatial transformer network with learned features," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pp. 249–257, Springer, 2017.

[88] B. D. de Vos, F. F. Berendsen, M. A. Viergever, M. Staring, and I. Išgum, "End-to-end unsupervised deformable image registration with a convolutional neural network," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pp. 204–212, Springer, 2017.

[89] H. Sokooti, B. De Vos, F. Berendsen, B. P. Lelieveldt, I. Išgum, and M. Staring, "Nonrigid image registration using multi-scale 3d convolutional neural networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 232–239, 2017.

[90] K. Bera, K. A. Schalper, D. L. Rimm, V. Velcheti, and A. Madabhushi, "Artificial intelligence in digital pathology - new tools for diagnosis and precision oncology," *Nature reviews Clinical oncology*, vol. 16, no. 11, pp. 703–715, 2019.

[91] M. Rusu, P. Rajiah, R. Gilkeson, M. Yang, C. Donatelli, R. Thawani, F. J. Jacono, P. Linden, and A. Madabhushi, "Co-registration of pre-operative ct with ex vivo surgically excised ground glass nodules to define spatial extent of invasive adenocarcinoma on in vivo imaging: a proof-of-concept study," *European radiology*, vol. 27, pp. 4209–4217, 2017.

[92] K. W. Clark, B. A. Vendt, K. E. Smith, J. B. Freymann, J. S. Kirby, P. Koppel, S. M. Moore, S. R. Phillips, D. R. Maffitt, M. Pringle, L. Tarbox, and F. W.

Prior, "The cancer imaging archive (tcia): Maintaining and operating a public information repository.," *J. Digital Imaging*, vol. 26, no. 6, pp. 1045–1057, 2013.

[93] N. C. I. C. for Cancer Genomics, "Tcga: National cancer institution center for cancer genomics."

[94] T. T. C. I. Archive, "Tcia: The cancer imaging archive." `https://www.cancerimagingarchive.net`.

[95] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

[96] R. M. . Madabhushi A, "Fused radiology-pathology lung (lung-fused-ct-pathology) (version 1) [data set]," 2018. The Cancer Imaging Archive.

[97] C. P(2016), "Data from prostate-mri."

[98] Madabhushi(2016), "Fused radiology-pathology prostate dataset (prostate fused-mri-pathology)."

[99] Y. Chen, F. Shi, A. G. Christodoulou, Y. Xie, Z. Zhou, and D. Li, "Efficient and accurate mri super-resolution using a generative adversarial network and 3d multi-level densely connected network," in *International conference on medical image computing and computer-assisted intervention*, pp. 91–99, Springer, 2018.

[100] Y. Wang, Q. Teng, X. He, J. Feng, and T. Zhang, "Ct-image of rock samples super resolution using 3d convolutional neural network," *Computers & Geosciences*, vol. 133, p. 104314, 2019.

[101] X. Zhao, Y. Zhang, T. Zhang, and X. Zou, "Channel splitting network for single mr image super-resolution," *IEEE transactions on image processing*, vol. 28, no. 11, pp. 5649–5662, 2019.

[102] C. Peng, W.-A. Lin, H. Liao, R. Chellappa, and S. K. Zhou, "Saint: spatially aware interpolation network for medical slice synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7750–7759, 2020.

[103] R. J. Chen and R. G. Krishnan, "Self-supervised vision transformers learn visual concepts in histopathology," *arXiv preprint arXiv:2203.00585*, 2022.

[104] R. J. Chen, C. Chen, Y. Li, T. Y. Chen, A. D. Trister, R. G. Krishnan, and F. Mahmood, "Scaling vision transformers to gigapixel images via hierarchical self-supervised learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16144–16155, 2022.

[105] N. Hashimoto, D. Fukushima, R. Koga, Y. Takagi, K. Ko, K. Kohno, M. Nakaguro, S. Nakamura, H. Hontani, and I. Takeuchi, "Multi-scale domain-adversarial multiple-instance cnn for cancer subtype classification with unannotated histopathological images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3852–3861, 2020.

[106] M. Van Rijthoven, M. Balkenhol, K. Siliņa, J. Van Der Laak, and F. Ciompi, "Hooknet: Multi-resolution convolutional neural networks for semantic segmentation in histopathology whole-slide images," *Medical image analysis*, vol. 68, p. 101890, 2021.

[107] B. Li, Y. Li, and K. W. Eliceiri, "Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14318–14328, 2021.

[108] S. Kapse, S. Das, and P. Prasanna, "Cd-net: Histopathology representation learning using context-detail transformer network," in *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5, IEEE, 2023.

[109] Y. Shi, H. Gao, S. Avestimehr, and Y. Yan, "Experimenting fedml and nvflare for federated tumor segmentation challenge," in *International MICCAI Brainlesion Workshop*, pp. 228–240, Springer, 2022.

[110] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, pp. 4700–4708, 2017.

[111] W. N. Price and I. G. Cohen, "Privacy in the age of medical big data," *Nature medicine*, vol. 25, no. 1, pp. 37–43, 2019.

[112] C. P. Langlotz, B. Allen, B. J. Erickson, J. Kalpathy-Cramer, K. Bigelow, T. S. Cook, A. E. Flanders, M. P. Lungren, D. S. Mendelson, J. D. Rudie, *et al.*, "A roadmap for foundational research on artificial intelligence in medical imaging: from the 2018 nih/rsna/acr/the academy workshop," *Radiology*, vol. 291, no. 3, p. 781, 2019.

[113] F. Zerka, S. Barakat, S. Walsh, M. Bogowicz, R. T. Leijenaar, A. Jochems, B. Miraglio, D. Townend, and P. Lambin, "Systematic review of privacy-preserving distributed machine learning from federated databases in health care," *JCO clinical cancer informatics*, vol. 4, pp. 184–200, 2020.

[114] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[115] C. He, S. Li, J. So, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu, L. Shen, P. Zhao, Y. Kang, Y. Liu, R. Raskar, Q. Yang, M. Annavaram, and S. Avestimehr, "Fedml: A research library and benchmark for federated

machine learning," *Advances in Neural Information Processing Systems, Best Paper Award at Federate Learning Workshop*, 2020.

[116] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," *arXiv preprint arXiv:2003.00295*, 2020.

[117] S. Pati, U. Baid, M. Zenk, B. Edwards, M. Sheller, G. A. Reina, P. Foley, A. Gruzdev, J. Martin, S. Albarqouni, *et al.*, "The federated tumor segmentation (fets) challenge," *arXiv preprint arXiv:2105.05874*, 2021.

[118] G. A. Reina, A. Gruzdev, P. Foley, O. Perepelkina, M. Sharma, I. Davidyuk, I. Trushkin, M. Radionov, A. Mokrov, D. Agapov, *et al.*, "Openfl: An open-source framework for federated learning," *arXiv preprint arXiv:2105.06413*, 2021.

[119] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen, *et al.*, "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data," *Scientific reports*, vol. 10, no. 1, pp. 1–12, 2020.