UMamba-MoE : A Framework for Enhanced Image Segmentation

Pradeep Rajan

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Master of Science in Artificial Intelligence
of the
University of Aberdeen.

Department of Computing Science

2024

Declaration

No portion of the work contained in this document has been submitted in support of an application for a degree or qualification of this or any other university or other institution of learning. All verbatim extracts have been distinguished by quotation marks, and all sources of information have been specifically acknowledged.

Signed: PRADEEP RAJAN

Date: 2024

Abstract

Medical image segmentation is required for accurate anatomical segmentations for disease diagnosis and treatment planning, one of the methods to achieved it is with the help of deep learning models. One of the traditional deep learning models that was widely utilized for Medical Image Segmentation was the Convolutional Neural Networks (CNNs), which produced great results. However, due to their local receptive field constraint, they were not able to capture global context information within an Image effectively. Hence, Transformers were introduced to overcome their drawbacks and limitations. The use of Transformers led to a significant improvement in the prediction, accuracy and metrics of the results. Even though certain advancements, such as the self-attention mechanism, were introduced, they were unable to capture long-range sequences effectively as anticipated. To address these challenges, U-Mamba was produced as a solution. This bi-directional state space model architecture is believed to outperform both CNN and Transformers. By implementing, Mamba is able to capture the relationships between the distant parts of an image.

The U-Mamba architecture is integrated with ‘nnU-Net,’ a combination of the semantic segmentation CNN model and Mamba network. The nnU-Net, with the help of a dataset fingerprint, nnU-Net automatically adapts its architecture for preprocessing, training strategies, and inference to the specifics of the dataset. The name suggests that it is based on a U-Net model. With the help of this integration, long-range sequences can be easily captured effectively.

Even though it produces the required results, the novelty model that we tried to proposed has helped improve the inference by increasing accuracy and other metrics such as Dice Similarity Coefficient (DSC) and Normalized Surface Distance (NSD). It can clearly observe a difference in value for DSC, which improved by at least 5%, and NSD by at least 2% for 2D images. The novelty introduced enriches the existing model ‘U-Mamba’ by enhancing the system’s adaptability and performance across different datasets. This improvement is mainly facilitated by a concept known as the Mixture of Experts (MoE). The MoE deploys multiple experts, which are constructed around only one network, the Mamba network. When an expert of four is defined, about four Mamba networks are created and made to run in parallel, allowing us to achieve the desired outcome much efficiently. This novel strategy has attempted to introduce has facilitated the advancement of medical image segmentation, achieving significant advancements in medical image analysis through enhanced adaptability and accuracy across varied datasets.

Acknowledgements

I would like to use this opportunity to extend my warmest appreciation to my supervisor, Dr Mingjun Zhong for his guidance and support throughout my Dissertation. Thank you Sir.

I also wish to express my gratitude to the entire staff at the University of Aberdeen for their invaluable support, guidance, and the resources they provided, which were essential for completing both my project and my studies.

Additionally, I would like to thank my family and friends for their tremendous support and encouragement throughout my academic journey.

Contents

Chapter 1

Introduction

Medical image segmentation is being widely utilized by a wide range of professionals and researchers across various domains in the healthcare industry to enhance diagnostic accuracy and provide efficient treatment. They are widely being adopted, and due to their demand, they undergo rigorous development each day (Roy et al., 2018) . Medical image segmentation is primarily devised using deep learning models. Traditional deep learning models like Fully Convolutional Networks (FCNs) have achieved state-of-the-art image and video Segmentation. For instance, they could detect tumor cells and pave the way for efficient treatment planning on where each tumor cell is located. Furthermore, modern applications in the medical field extend the need for dynamic treatment planning and monitoring disease progression, leading to the need for more adaptable and efficient models. The continuous development and rigorous advancements of these technologies are critical due to the increasing complexity of medical diagnostics. Several notable models have consistently produced satisfactory results in medical image analysis. One of the earliest and most widely used deep learning networking models for Medical Image segmentation is the "U-Net" model. It was introduced in 2015, and it is purely based on the Fully Convolutional Network (FCN) architecture. FCN was primarily used for Semantic Segmentation, it helps to retain spatial information throughout the network. Spatial information in terms of image processing means the arrangement and relationships of the pixels of an image, in other words they assign class or category to each pixels in an image, making them ideal for segmenting different objects. FCN replaces the convolutional layers to output spatial maps instead of classification scores. These maps preserves spatial information and highlights the entire regions corresponding to different objects or features in the image. U-Net is an enhanced network using an encoder-decoder type architecture and skip connections to combine low-level and high-level features from different layers. This enables precise localization and segmentation of the anatomical structures. Notably, FCN was introduced even before U-Net, and it was introduced in 2014. It is a foundation model still being used for medical image segmentation and other segmentation models (Luo et al., 2021). Thus, the introduction of FCN and its evolution into U-Net have greatly advanced the field of medical image segmentation, providing robust frameworks for various diagnostic tasks.

U-Net was successfully extended for various medical image segmentation tasks, such as segmenting organs, blood vessels, cells. They were able to support other medical imaging modalities from Computed Tomography (CT) scans, Magnetic Resonance Imaging (MRI) scans, to X-rays. The U-Net architecture consists of encoders, decoders, and skip connections. The encoder is mainly responsible for capturing the contextual and semantic information by downsampling the feature maps available in an image. Feature Maps are the results that are obtained after the image passing through each Convolutional layers and Pooling layers. At the last module of the encoder which is the bottleneck, the feature maps will have the lowest spatial resolution yet the highest number of channels, each capturing rich contextual features. At the same time, at the bottleneck, the encoder forwards the feature maps to the decoder, which upsamples the feature maps and concatenates the corresponding feature maps from the encoder through skip connections (Ronneberger et al., 2015). This concatenation helps the decoder recover lost spatial and contextual information that was lost during the encoder phase. The final phase of the decoder contains a convolutional layer, that layer produces a specific number of channels corresponding to the classes to be segmented, ultimately generating the precise segmentation masks.

Despite the good results achieved with CNNs, they inherently struggle to capture the global context of the input data. This limitation has prompted the introduction of the Transformer models, which excel in understanding the entire image as a whole and their comprehensive data analysis capability. Initially developed for natural language processing, the Transformers excelled in producing the desired results. This includes improving machine translation quality and improved performance due to scalability (parallel running). Given their performance, they were slowly being adapted to image classification. For instance, the first network to be introduced is the Vision Transformer Model (ViT), it was widely sought after as it produced remarkable results in image recognition tasks when compared with the traditional CNN models. Based on this foundation, specialized models such as the Semantic Segmentation Transformer (SETR) and Segmenter have been developed. The Transformers also faced limitations, quadratic complexity, which leads to computation and memory complexities and also the need for large amount of data, otherwise it will cause the model to overfit. Hence, hybrid models such as Trans U-Net, UNETR, SegFormer, SwinUNETR, and TransFuse have been developed, combining the detailed local processing of CNNs with the global data understanding of Transformers, thus enhancing both the accuracy and efficiency of the Medical Image Segmentation.

1.1 Problem Statement

The hybrid architecture of both CNN and Transformers models is designed to leverage the strengths of both models, improving the overall performance and overcoming the challenges in each network. To understand this problem statement, detail observation needs to be performed on how each of those networks complement each other and identify all the drawbacks that they still face. Transformers perform well in capturing long-range dependencies. However, they are computationally expensive. They suffer from high quadratic complexity due to the self-attention mechanism and if there are very few images in the dataset, it will cause the model to overfit due to their high capacity and parameters. Therefore, to overcome these particular computation problems, CNN was integrated. Utilizing CNN, the model benefits from obtaining spatial and contextual data. Another main factor is that they help reduce the dimensionality of the input data with the help of pooling layers, thus reducing the computational burden and preventing overfitting by reducing the computational cost to some extent.

Now, the problem statement is that even with the help of hybrid architectures, these models' results were inefficient. This is particularly true in terms of capturing long-range information, computational speed, and resource utilization. Hence, a completely different approach was sought out. The approach undertaken to follow is a new concept known as Mamba, a linear sequencing modeling block that is assessed to outperform the Transformers.

The Mamba architecture consists of a module known as the Structured State Space Model. It is commonly known as SSM or Structured State Space sequence model (S4) (Gu et al., 2022). SSMs are designed to capture long-range sequence dependencies much more effectively than Transformers, making them well-suited for tasks that involve processing high-dimensional data, such as images and videos. One of the innovative features of this Mamba architecture is the selective mechanism introduced by the SSM, which allows the model to ignore specific inputs based on the relevancy of the data. It dynamically adjusts the parameters based on the input and has a high memory capacity, which helps to faster training and inference. In other words, the Mamba model introduces a mechanism that allows the model to selectively focus on relevant information in an input-dependent manner. This selective mechanism enables the model to efficiently process high-resolution inputs by attending to the most informative regions, reducing computational complexity and improving performance. However, the Mamba model is not a standalone solution, it requires integration with architectures like CNNs, RNN to fully capture and utilize spatial information effectively. When others initially tested it, Mamba was able to perform better than Transformers on a large scale on both language processing and image classification tasks.

In this dissertation, an hybrid architecture, U-Mamba has been integrated with another trending concept known as the Mixture of Experts (MoE). U-Mamba is an hybrid architecture that involves the use of both a CNN based architecture and Mamba architecture. The base network architecture of U-Mamba is the nnU-Net network, an advanced version of U-Net designed to automatically adapt to the specific dimensions of 2D or 3D datasets, preprocessing, training processes, network architecture and post processing. The nnU-Net is also called a "no-new-net." Notably, there are three distinct types of U-Net available in that architecture: 2D U-Net, 3D U-Net, and U-Net Cascade. Each of them has their own purpose, as its name suggests. Next, the novelty part attempted to be introduced can be discussed, which is the Mixture of Experts(MoE). The Mixture of Experts is widely used in the field of deep learning. Applications such as ChatGPT and other large language models use it for computational efficiency. The MoE defines a number of experts and each experts process the data given, then combines the output of those experts and passes it to the next layer. With the help of this combined architecture, the model can capture both local and global image features and long-range dependencies with less computational burden, significantly improving training time and inference.

1.2 Motivation

Accurate medical image segmentation plays a vital role in various clinical applications, such as disease diagnosis and treatment planning for patient monitoring and surgical guidance. The primary motivation for this research is to develop a state-of-the-art model by addressing the existing limitations and challenges. This includes the inability to capture long-range dependencies, high computation costs, and the need for accurate image segmentation. The limitations can be overcome by leveraging the power of SSM, together with the selective mechanism of the Mamba model and the multiple experts' method that specializes in handling different aspects of the input data.

This model aims to help healthcare professionals, such as doctors and lab technicians, by precisely segmenting anatomical structures that can help them isolate and highlight specific regions of interest. This, ultimately, leads to the best possible patient care. Beyond patient care, it can help contribute to advancements in medical research and education, leading to the development of scientific discoveries and therapeutic techniques. Moreover, the visual representation of the segmented medical images can also be greatly beneficial for the patients. It can also help them recognize or enhance their understanding of their body conditions and can help them engage more actively in their healthcare journey.

1.3 Goals

For this dissertation, two main goals are strived to be achieved here,

- Reduce Computation Cost: The first goal is to reduce the computation cost significantly by reducing the required training steps. This is achieved with the help of the MoE approach. The MoE leverages parallel computational capabilities by concurrently processing inputs across multiple expert networks. This strategy not only accelerates the segmentation process but also increases the scalability by allocating resources dynamically.

- Enhance Segmentation Accuracy: The second goal is to improve the segmentation accuracy. Integrating MoE exclusively into the Mamba model helps identify intricate patterns and long-range spatial dependencies efficiently. This focused integration is designed mainly to focus on the Mamba model's generalization power, improving its performance and the accuracy of its segmentation outputs.

1.4 Outline

Chapter 1: Stating the challenges that the project aims to address by discussing the significance and relevance of the research.

Chapter 2: Talks about the Image segmentation used in the medical field and the different types of computer vision tasks to review the use of CNN and transformers, according to the segmentation tasks that is being used. Literature Review is also done here.

Chapter 3: Briefly explains the base architecture in which our project is based on and reviews the integration of mamba network and Mixture of Experts. The review includes the details about how this integration has enhanced the performance of the model.

Chapter 4: This is the experiment section where the focus is turned to preprocess, training and inference. It explains each procedure and the results given by them. A brief description of the datasets is also described here.

Chapter 5: Compares and provides a detailed review about the performance of the different datasets across different network architectures.

Chapter 6: It summarises the research findings and also talks about the limitations that the project faced when doing the project. It also explains the future work on how the model can be improved even further still.

Chapter 2

Background

2.1 Overview of Medical Image Segmentation

As discussed, medical image segmentation plays a critical role in diagnosing a disease and in planning treatment for that particular disease. They also play a major role in identifying and labeling anatomical structures accurately. It helps outline regions of interest such as Teeth, organs, tumors, cells in dental X-rays, CT scans, MRI scans and microscopy images. In modern days, various medical imaging modalities are employed, including X-rays, MRI, CT, Positron Emission Tomography (PET), and Ultrasound imaging, each with its unique strengths and applications. While X-rays and ultrasound images are 2D, the remaining generate 3D data, necessitating specialized segmentation algorithms for each task.

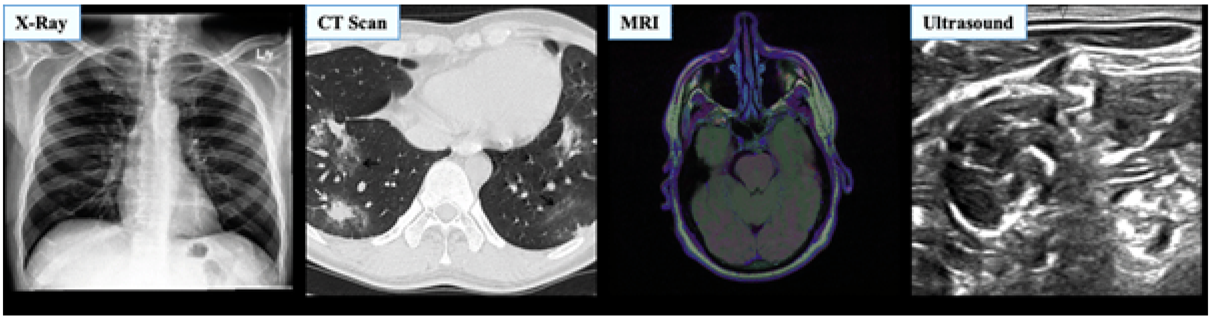

Figure 2.1: Different types of Modalities

2.2 Overview of Different tasks in Computer Vision

Initially, manual segmentation was used to segment images. However, as technology advanced, automatic segmentation using deep learning techniques gained popularity. Image Segmentation is a well known method in computer vision process, it helps in segmenting images into regions of interest or objects that helps to identify and analyse. The computer vision has five main tasks, Image Classification, Object Detection, Segmentation, Image Generation, and Pose Estimation (Paneru and Jeelani, 2021). There are several different types of Segmentations available, and the most commonly used segmentations are, Semantic Segmentation, Instance Segmentation, and Panoptic Segmentation.

2.2.1 Semantic Segmentation

It involves assigning class labels to every pixel available in an image. All pixels that belong to the same object type will be assigned the same class label. It provides categorical information at pixel level, (Hao et al., 2020) hence, it can be used to identify objects and analyse those objects. For instance, applications such as autonomous driving (Cheng and Zheng, 2021) can be used for identifying pedestrians, roads, and other vehicles. Primarily, Semantic Segmentation datasets were mostly utilized by Convolutional Neural Networks. The reason is that Semantic Segmentation plays a vital role in identifying and utilizing spatial information effectively. Nowadays it is being used by several other deep learning models. The datasets that we are using are also based on this type of segmentation.

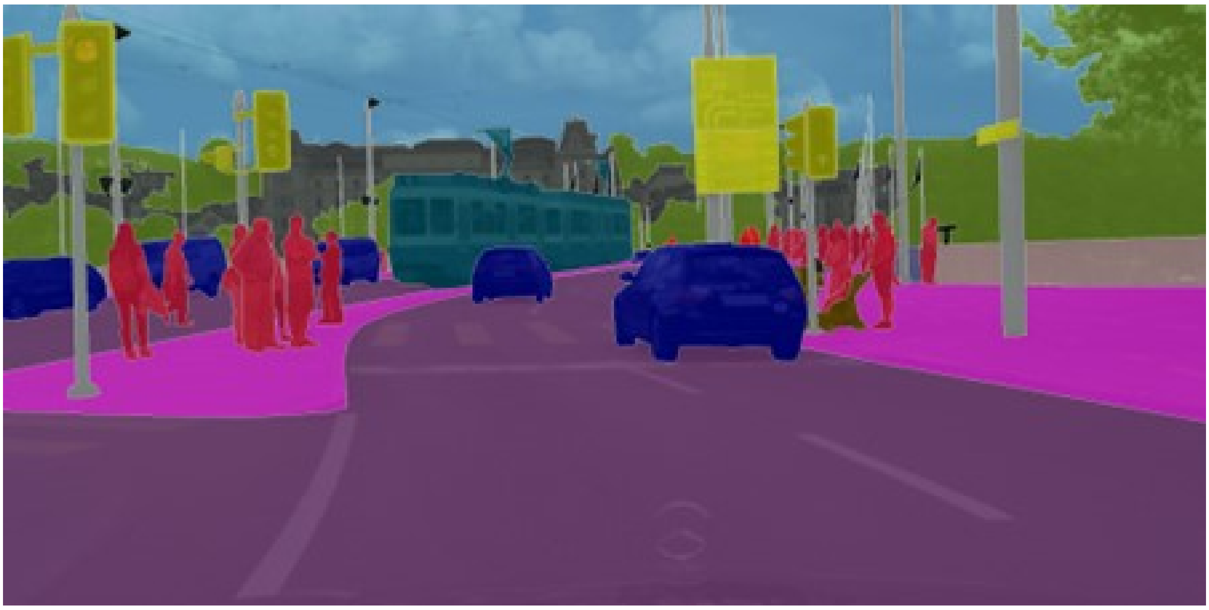

Figure 2.2: Example image for Semantic Segmentation

2.2.2 Instance Segmentation

It involves the use of objection detection to identify the objects available in an image and then segments the image. It labels every pixel that helps in identifying the object, but it also employs additional information to differentiate each similar object available (Hao et al., 2020). It distinguishes between different instances of the same class within the same image. For instance, in autonomous driving, it not only identifies the pedestrians and the vehicles, but it also distinguishes between the different types of pedestrians going down the street and different cars that are on the road. Instance segmentation labels each of those objects differently, which helps in even more object identification and thorough analysis.

2.2.3 Panoptic Segmentation

It is the combination of both Semantic and Instance Segmentation. This type of segmentation was proposed due to the inherent disadvantages and limitations seen in both of the above segmentation methods. Panoptic Segmentation operates by assigning a label to every pixel in an image and distinguishing the difference between instances of the same object category or same class (Kirillov et al., 2019). It just categorizes and labels the needed objects differently while just categorizing

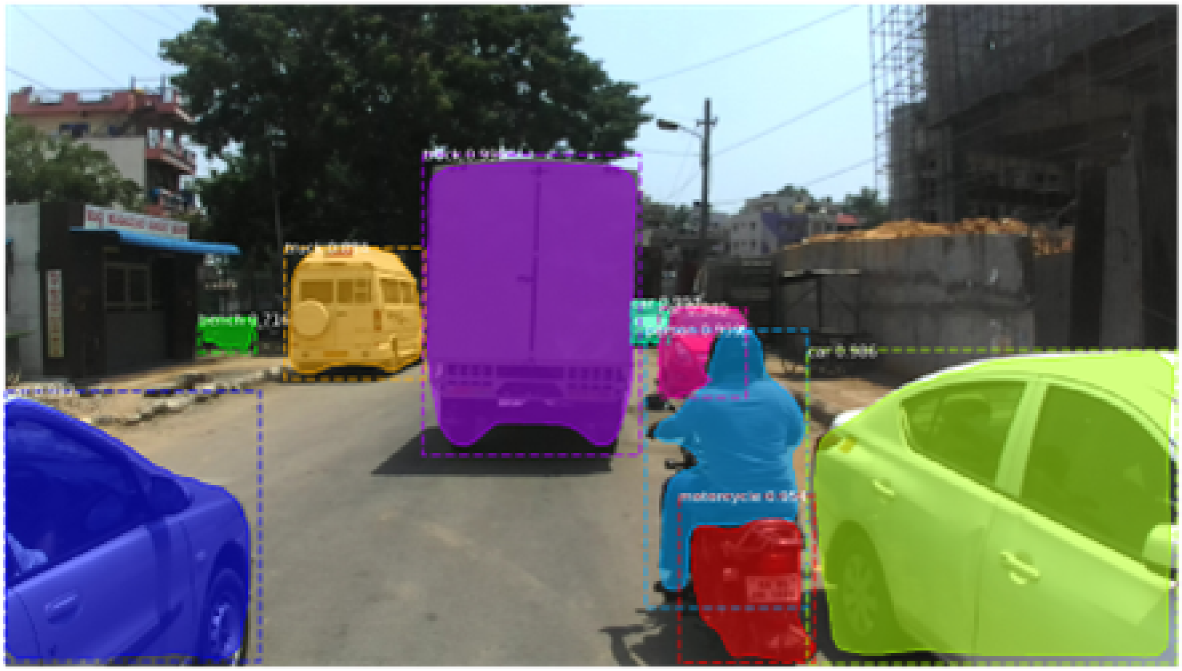

Figure 2.3: Example image for Instance Segmentation

objects and labelling them as a whole that doesn't require instance-level segmentation. For example, people having a picnic, where you can see the sky and the different trees are just categorized into one class label category.

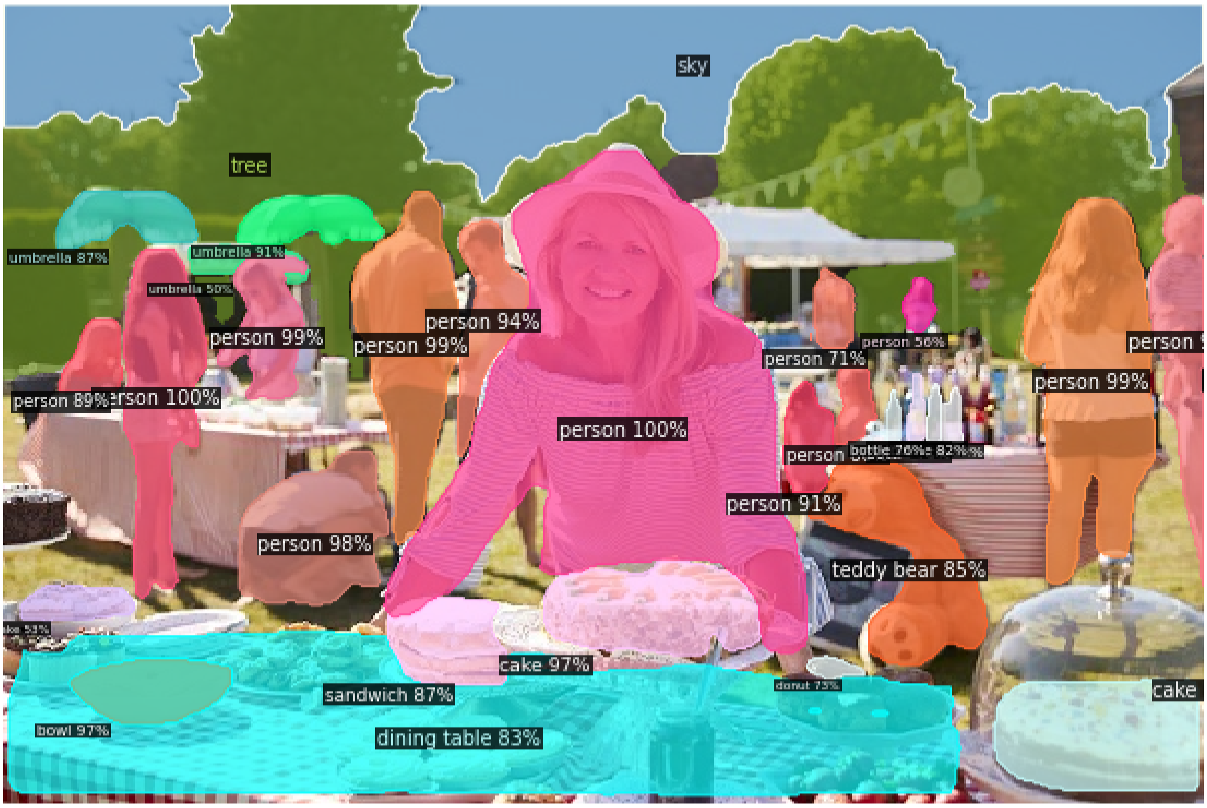

Figure 2.4: Example image for Panoptic Segmentation

2.3 Overview of CNNs Used in Medical Image Segmentation

Convolutional Neural Networks (CNNs) is a type of deep learning algorithm that plays a vital role in the advancement of image classification. These networks have been very helpful in contributing to the progression of various computer vision tasks (Kayalibay et al., 2017). It helped in object identification, and with the help of those results, intensive analysis was made possible. As advancements were being slowly made, CNNs were gradually adapted for analysis in the medical field. They also played an integral role in the detection and identification of tumors, viruses, micro-bacteria, medical equipment, and organs.

CNNs have demonstrated their versatility by effectively analyzing various imaging modalities such as Computed Tomography (CT), ultrasound, and X-rays (Kayalibay et al., 2017). Other than Image classification they are being actively used in the field of image segmentation tasks. With the help of different modalities, segments are created, and as a result, categorization of different anatomical structures is made possible.

As discussed before, FCN was the first network that was used for the image segmentation process. They are made up of only convolutional layers and were not suitable for accurate segmentation tasks as they could not preserve spatial dimensions through the network, which is crucial for semantic segmentation that uses pixel-wise prediction. (Lin et al., 2013) proposed a simple 1 x 1 convolutional layer and the inclusion of some pooling as a solution. This allowed the network to predict multiple pixels simultaneously in a forward pass. The convolutional layer helped in feature learning and the pooling layer helped to reduce the computation power by reducing the spatial size. This in turn resulted in the loss of image resolution due to the reduction of input size by those layers. Hence, U-Net (Ronneberger et al., 2015) was introduced. U-Net is improved upon FCNs by incorporating two key enhancements, using the concept of upsampling and skip connections, that concatenate feature maps from downsampling layers, thus preserving and recovering the information lost or replace the resolution lost during the downsampling process. They could efficiently capture spatial and contextual information with the help of their local receptive fields.

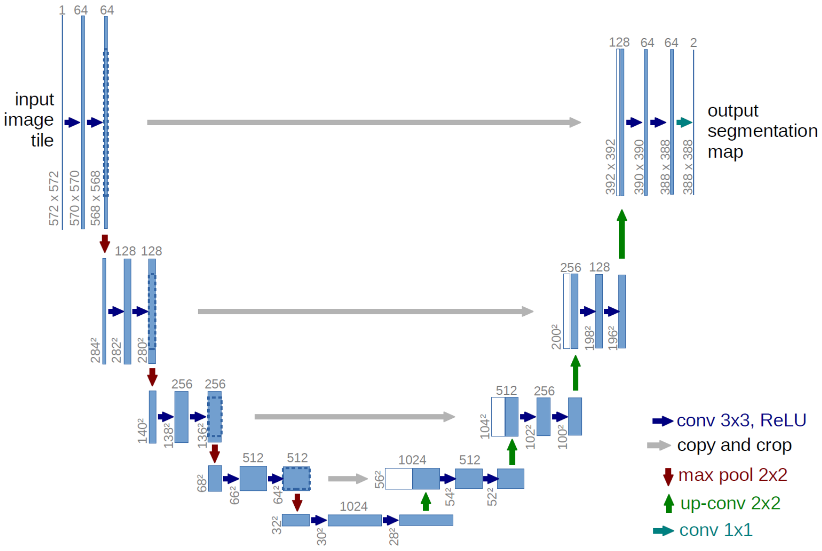

Figure 2.5: U-Net Architecture

Upon the introduction of the U-Net architecture, researchers have continued to develop and enhance the CNN architecture by improving its efficiency and accuracy across different imaging modalities. This led to the development of many variants of CNN-based image segmentation models such as V-Net, U-Net++, SegNet, SegResNet, and nnU-Net. U-Net, which was initially designed for segmenting 2D images. Nevertheless, as 3D medical imaging became more prominent, in a sense that modern CT and MRI scans were capable of producing 3D scans. The U-Net architecture underwent several modifications and enhancements to accommodate volumetric data and improve segmentation accuracy.

2.4 Overview of Transformers in Medical Image Segmentation

The Transformer network was initially created for the Natural Language Processing community (Vaswani et al., 2017). It helped several large language models to capture global information, which was pivotal for tasks such as translation, summarization, communication, and text generation. One of the most significant aspects of the Transformer model is the self-attention mechanism, coupled with its ability to run in parallel (Vaswani et al., 2017). It also has a feed-forward network that helps in capturing long-range dependencies. These advantages made the image classification community to adapt to the use of Transformers. Many researchers have successfully introduced Transformers in image classification models to predict accurate image segmentation results.

The first ever model to successfully adapt the Transformer network is the Vision Transformer Model (ViT). The ViT (Dosovitskiy et al., 2020) made use of a patch-based approach to learning. It involves dividing the image into fixed-size patches, which are then linearly embedded and then processed as sequences by the Transformers. From those embedded patches, the self-attention mechanism could capture the global information effectively. Many advancements were made using the Transformers, some of them are SegFormer, DeiT, Swin-Transformer, UNETR, and Swin-UNETR. Swin-Transformer (Liu et al., 2021) is different from other Transformer models, instead of using patch-based approach it uses something called as sliding window approach which merges image patches to build feature maps.

Transformers offers a significant advantage in the advancement of medical image segmentation. It effectively improves the accuracy and efficiency of the segmentation process. The only disadvantage to the Transformers (Xu et al., 2021) is the significant use of high computational power for processing and the need for data requirements, if the input data is less, then the model tends to overfit.

2.5 Literature Review

The SegResNet is proposed by (Myronenko, 2018), which has an encoder-decoder architecture in which the network incorporates the VGG16 network. The SegResNet modifies the deep residual network, adopting three key techniques. Firstly, it employs dilated convolution to expand the receptive field of the network, enabling it to capture broader contextual information. Secondly, Long short-term memory (LSTM), it is a type of Recurrent Neural Network (RNN) for its capability to capture long-range dependencies and, finally, multi-scale prediction, which helps mitigate the loss of spatial information during the encoding process. By combining these techniques, the SegResNet aims to improve segmentation performances. With the help of PASCAL VOC12 data, they achieved an accuracy of 1.6% more than that of the other state-of-the-art models when compared with the same dataset. Thus, SegResNet has proven effective in various medical image segmentation tasks, including brain lesion segmentation and organ delineation.

The Transformer-based Nested U-Net (UNETR) model, introduced by (Hatamizadeh et al., 2021), is also an encoder-decoder architecture. By incorporating a Vision Transformer encoder into a U-Net structure, UNETR captures long-range dependencies and global context more effectively than traditional CNNs. The drawbacks to this model were its high requirement of computation power, potentially leading to increased memory requirements and longer training times, especially for large 3D volumes. For the evaluation, they used the medical decathlon dataset, which seems to achieve 85.3% accuracy, outperforming other state-of-the-art models by at least 2.125%.

Building upon the success of UNETR, the Swin-UNETR model (Hatamizadeh et al., 2022) further improves performance by incorporating the Swin Transformer backbone. It also has the same self-attention mechanisms with just one additional add-on, which is that it utilizes a shifting window approach. It partitions the input into overlapping windows, allowing it to capture multi-scale features while maintaining computational power, especially for 3D images. This model was developed for a challenge named BraTS Challenge, which involves 3D brain tumor images. The Swin-UNETR, when executed, demonstrated some promising segmentations, and it was able to capture complex patterns and model long-range dependencies. It has outperformed its competition in terms of Dice score by 0.5%, by nnU-Net and 0.6% when compared to SegResNet method. Hence, the Swin-UNETR has demonstrated superior performance compared to UNETR, particularly in segmenting small and intricate structures.

Next, let us examine the nnU-Net (No-New-Net) proposed by (Isensee et al., 2020). As discussed, this is the base network on which the novelty model is based on. It is a self-configuring deep network framework that only helps to streamline the process of preprocessing and training across diverse datasets. The model adapts according to the dataset used and dynamically adjusts its parameters according to the dataset's modality, size, and depth. However, the only drawback to this model is that it does not involve any novelty architecture or networks and only uses the base U-Net architecture. It does not do any comparison between any other models, it just evaluates 10 different datasets and produces the result.

The U-Mamba architecture (Ma et al., 2024) uses the nnU-Net as the base architecture and introduces a novelty called the Mamba network (Gu and Dao, 2023). It is being discussed that the Mamba network outperforms the Transformers to some extent and is computationally less expensive. By incorporating the Mamba network into the encoder module, the U-Mamba architecture aims to enhance the feature extraction process. This leads to better performance in multi-organ segmentation tasks. It also helps capture long-range sequences much more effectively. With respect to 3D images, it outperforms the the other SOTA models by 1.18% and 2D images by 2.18%.

Each of these models outperforms the other by a margin. However, their performance and suitability vary due to their own drawbacks and limitations. Some of the major factors that affect their performance are the imaging modality, segmentation of complex anatomical regions, and analysis of complex patterns in the images. Most medical images are full of noise, and they have high or low contrast, leading to limited visibility. Anatomical images are located close to each other, which makes segmentation difficult. Hence, the focus is on capturing long-range sequences. Even though the initial model attempted to solve this problem, the newly proposed model captures those sequences much more efficiently and with minimal training time.

Chapter 3

Method

In this chapter, the exploration delves deeply into the model architecture's construction. The architecture will be systematically broken down, highlighting how each component contributes to the model's enhanced capabilities. Let us observe each of them one at a time.

3.1 Base Network Architecture

As discussed before, U-Mamba and Mixture of Experts (MoE) were integrated to create a new novel model. With that information in context, let us first explore the details of the U-Mamba architecture. The base architecture of this U-Mamba is adapted from a model known as the nnU-Net (Isensee et al., 2018), also referred to as the No-New-U-Net. Now, let us delve into nnU-Net in detail and examine its contribution to enhancing the systems' efficiency.

3.1.1 Working of nnU-Net

The nnU-Net framework is designed to streamline the process of medical image segmentaion by automatically identifying the best possible configurations, including pre-processing, network architecture, training methods, and post-processing. The nnU-Net was initially designed for a challenge (Medical Decathlon Challenge) that involved the automatic adaptation of the model to the various types of datasets. It requires the model to dynamically adjust to each dataset's specifics, such as 2D or 3D, with their respective file formats. This includes png, obj, nii.gz, and tiff, to name a few. It has evolved from a solution for a challenge to a benchmark model being widely utilized in medical image segmentation. The nnU-Net uses a Dataset Fingerprint (Isensee et al., 2018), that analyses the datasets and adapts the model to find the best architecture and training steps that would best suit the dataset that is deployed. It analyses the spacing, dimensions, image resolution, class distribution (labels available in the dataset), spatial information and the modality (2D or 3D) of the dataset. The dataset fingerprint generates the pipeline fingerprints that decides the suitable network architecture. At its core, as the name suggests, nnU-Net is built upon the U-Net architecture. The dataset that involves 2D or 3D images employs the respective architecture (Isensee et al., 2018), such as 2D U-Net, 3D U-Net, and U-Net Cascade. These are just the basic U-Nets without complex additions, including dense connections, residual connections, or attention mechanisms. The 2D U-Net is used for the datasets that has Two Dimensional images and a slice wise segmentation is performed. The 3D U-Net is used for datasets that contains volumetric data. The 3D U-Net cascade is also used for segmenting three dimensional images, but it first uses a lower resolution 3D U-Net to capture the global context and then a higher resolution 3D U-Net for local segmentation process.

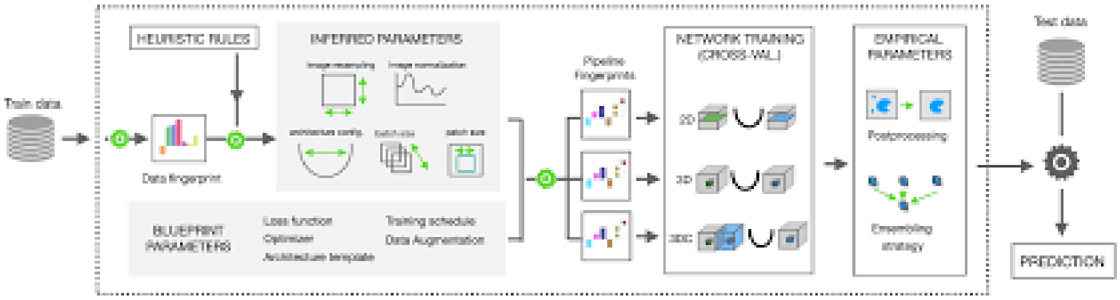

Figure 3.1: Network Architecture of nnU-Net

The U-Net's employed in this model are mostly kept true to their original configuration. The U-Net comprises an encoder and decoder, with skip connections that facilitate remapping features from the images, compensating for the information loss during the encoder's downsampling process. Notably, there will be at least two or three encoder blocks in each U-Net. Each block will downsample the image by a factor of 2, capturing spatial dimensionality and feature maps. Each encoder is made up of convolutional and pooling layers. Instead of using ReLU as the activation function, Leaky ReLU (Isensee et al., 2018) is used as the activation function and a pooling layer of 2x2 Kernel. This layer is responsible for downsampling the images and gives those samples as input to the convolutional layers of the next encoder block. At the same time, the deepest part of the encoder is called the bottleneck, where the last encoder block transfers the spatial information and local features to the decoder.

Now, finally, let us examine the decoder. It also contains the same layers as the encoder. The only difference is that instead of desampling, it upsamples the feature maps and gradually increases the spatial dimension of the image by reconstructing the original image. In addition to this upsampling, it also concatenates with the feature maps retrieved from the encoder through skip connections. The final layer of the last decoder is the convolution layer that reduces the number of output channels to the number of classes in the segmentation tasks.

The nnU-Net prefers to uses a batch-based training for the 2D images and patch-based training (Isensee et al., 2020) due to the presence of 3D images. If the model processes the 3D images as a batch, the system will not be able to handle it and will become more computationally expensive. Hence, the input is divided into small overlapping patches, which are then processed independently and reassembled. After the segmentation process is completed and the results are obtained for the training data, the interference is achieved. During the interference process, the images are predicted using the testing data with a sliding window approach. Further details about the internal workings will be covered in Chapter 4.

3.2 Mamba-Integration

Now, let us discuss the U-Mamba architecture. According to the original paper (Hendrycks and Gimpel, 2020), two types of U-Mamba architecture are developed. In one, the Mamba module is integrated at the bottleneck of the available U-Nets, and in another, the Mamba module is integrated in every available encoder block. The decoder is composed of the residual blocks, and there are skip connectors that go from the Mamba module to the decoder.

The State Space Model (SSM) (Gu et al., 2022) in the Mamba employs an input-dependent

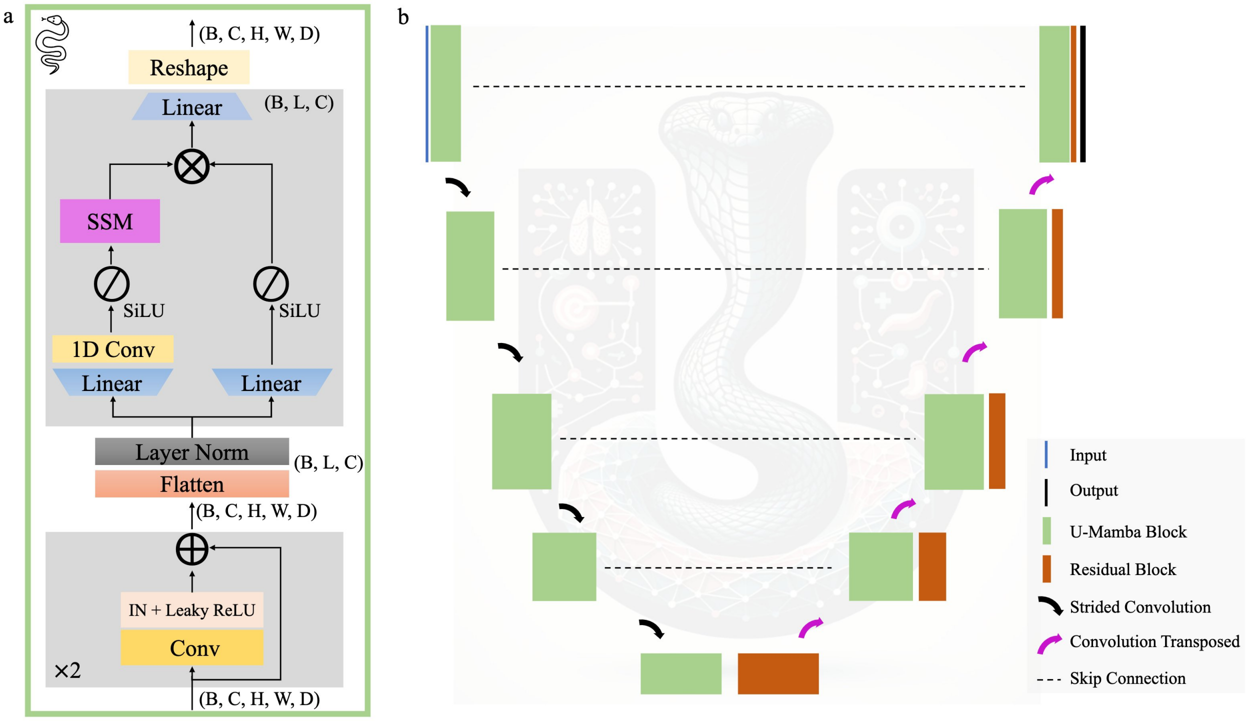

Figure 3.2: U-Mamba Network Architecture

selection mechanism, and it permits efficient information filtering of relevant information from inputs. This is achieved by parameterizing the SSM parameters based on the input data itself. SSM dynamically adjusts everything according to the information which taken from the input images. Secondly, Mamba employs a hardware-aware algorithm that scales linearly with sequence length. These sequences are produced by flattening the images. The algorithm enables efficient recurrent computation through a scan operation, making it faster than previous methods on modern hardware architectures. With the help of some linear layers, it integrates the SSM blocks. The Mamba model has performed greatly in the domain of language modeling and genomics. In addition, it greatly leverages the capabilities of modern hardware and enables more effective capturing of long-range dimensions.

Now, let us examine the internal workings of the first model (U-Mamb_Bot) in which the Mamba module is integrated at the bottleneck. First, the downsampled image (3D image) from the residual encoder block is passed on to the next residual encoder block, and the image features are in the shape of (B, C, H, W, D). As discussed, the residual encoder block (He et al., 2015) contains 2x plain convolutional layers followed by an instance normalization and Leaky ReLU. After this process, the image features get flattened and transposed to (B, N, C) where N is (H x W x D). After obtaining this, the features are passed through layer normalization. After the layer normalization, the data then enters the Mamba block, where the data is passed in two parallel branches. In the first branch, the image feature is expanded to (B, 2N, C) and is passed to a linear layer followed by a 1D convolutional layer. The activation function used for this layer is SiLU, it is a combination of the sigmoid and ReLU activation function (Hendrycks and Gimpel, 2020). Together with the activation function, the image features are then passed to the SSM. .The second branch only contains the linear layer, and it also has an activation function, which is SiLU

(Hendrycks and Gimpel, 2020). Both the branches are then merged with the help of Hadamard products. Finally, the image features are projected to their original shape and then reshaped and transposed to (B, C, H, W, D). In the first U-Mamba model, this Mamba block is only used in the bottleneck. Therefore, after it gets reshaped and transposed to its original shape, they are then passed to the residual decoder block. In the second U-Mamba model, the outputs are passed on to the next available residual encoder blocks, and the image features are finally transferred to the decoder at the bottleneck. The decoder focuses on getting the local information and resolution recovery. The resolution recovery of the images is achieved with the help of a skip connection from the encoder blocks. The final segmentation is obtained after the final decoder image is passed to a 1x1x1 convolutional layer together with a SoftMax layer. The 2D images follow the same process, instead of an image shape of (B, C, H, W, D), as the 2D image does not contain any depth, it is represented as (B, C, H, W). The second model (U-Mamba_Enc), which introduces the Mamba network to every encoder (Ma et al., 2024), follows the same procedure as described above. Instead of passing the output to a decoder, it just passes the input shape to the next available encoder until it reaches the bottleneck, and then it passes the dimensions to the decoder, which helps produce the requires segmentation results.

3.3 Integration of Mamba-MoE

We have seen how the existing architecture works by integrating nnU-Net and Mamba network. A novelty is being introduced to the existing architecture, with an attempt to integrate a method called the 'Mixture of Experts' into the U-Mamba architecture. Here is an in-depth explanation of how this integration works. Similar to U-Mamba, the novelty architecture also consists of two models, where the integration occurs at the bottleneck of the U-Net for the first model and for the second model, it is done at every encoder available in the U-Net. The first model, after the input, with dimensions (B, C, H, W, D) underwent the encoder which has a series of convolutional and pooling layers, the feature maps are obtained, which are flattened and normalized and then submitted as input to the gating mechanism (Shazeer et al., 2017), which is a linear layer. Its output will be equal to the number of experts, which means that according to the number of experts, many branches will be created. The output of the gating network goes through the SoftMax layer to ensure that the weights across the experts sum up to 1. These weights determine how much each expert will contribute to the final output. Softmax is an important layer, it is particularly used in multi class classification where each class probability needs to be predicted and in Transformers (Shen et al., 2023) it is used to assigning probabilities (compute attention weights), which helps in focusing on important parts of the input data. The same process is being used here.

Now, let us observe how the experts work in our environment. Since the experts are applied to the Mamba module, according to the number of experts defined, multiple Mamba modules are created. No changes have been made to the Mamba module, it performs as usual. Note that every expert acquires input from the gating mechanism. They process it independently and in parallel, focusing on various aspects or features within the data due to the different learned parameters in each Mamba module. For instance, if the encoder has an output shape (B, H, W, D, C). After getting flattened and layer normalization, the gating mechanism has an input shape of (B, C, N) where N is (H, W, D). The flatten layer collapses all spatial features to one, which is N, the total

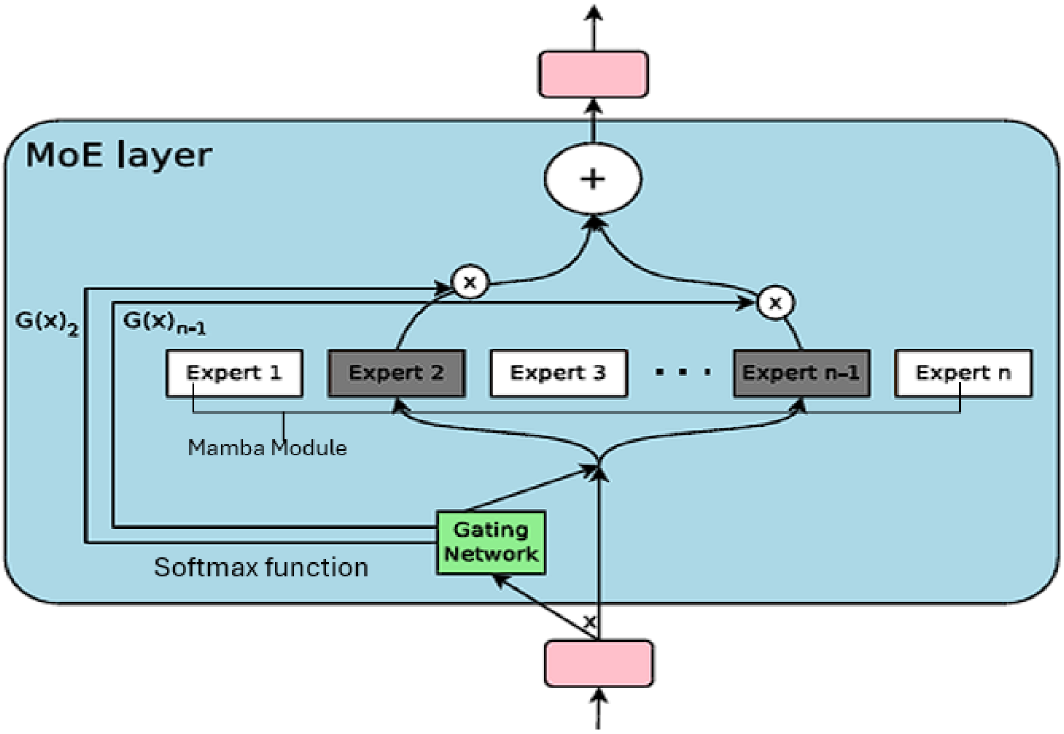

Figure 3.3: Mamba-Mixture of Experts Integration

number of spatial tokens. The layer normalization layer stabilizes the dimensions. If the number of experts available is defined as 2, then the Gating mechanism (Shazeer et al., 2017) produces an output of shape (2, B, N, C). The output shape consists of a pair of weights which are assigned, one for each expert. The probability distribution by the Softmax layer determines how much each expert should contribute for what input feature. After the mamba process, the outputs from all the experts are stacked together to form a shape of (experts_weight, B, N, C). The weights are then used to perform a weighted sum (Shazeer et al., 2017) over the experts' output. Consequently, the result is obtained as a single feature map of shape (B, N, C), which is then transposed and reshaped to its original spatial dimensions (B, H, W, D, C). This image feature is then passed to the next encoder or decoder block to process until the final segmentation map is obtained.

The same network design is adapted for the second model (MambaEnc_Moe). The mixture of experts is assigned to every Mamba network present in each encoder stage. The integration of the Mixture of Experts module with the U-Mamba model has allowed the model to dynamically adjust to different features in the input data and segment them. It leverages the experts to process the feature complexity and diversity, ultimately enhancing the segmentation accuracy and prediction. It also helps in greatly reducing the training time because of its parallel computing capabilities.

Chapter 4

Experiment

4.1 Dataset

4.1.1 Dental Dataset

The teeth dataset from Humans In the Loop, Bulgaria, consists of 112 images designed for semantic segmentation tasks. This dataset is smaller compared to the other datasets evaluated. It features 32 different classes representing various types of human teeth. Each class corresponds to a specific type of tooth, such as incisors, canines, premolars, molars, and wisdom teeth, each identified by its unique position and type in the mouth. Approximately 80 images were used for training and about 32 images for testing.

4.1.2 Microscopy Images

This dataset was from the NeurIPS 2022 Cell Segmentation Challenge, which focused on cell segmentation in various microscopy images. Here, 500 and 50 images were used for training and evaluation, respectively. This is also an instance segmentation task where algorithms are expected to assign a unique label for each cell instance. In these experiments, the instance segmentation was converted into a semantic segmentation task by predicting the cell boundaries and interior regions, this is done because nnU-Net doesn't support instance segmentation datasets.

4.1.3 Abdomen CT Images

This dataset was from the MICCAI 2022 FLARE Challenge that focused on the segmentation of 13 abdominal organs, which includes the liver, spleen, pancreas, right kidney, left kidney, stomach, gallbladder, esophagus, aorta, inferior vena cava, right adrenal gland, left adrenal gland, and duodenum. The training set contained 50 CT scans from the MSD Pancreas dataset, and the annotations were from AbdomenCT-1K. Another 50 cases from different medical centers were used for evaluation, and the annotations were provided by the challenge organizers.

4.1.4 Abdomen MRI Images

This dataset was from the MICCAI 2022 AMOS Challenge, which also focused on abdominal organ segmentation. The original dataset contains 40 and 20 MRI scans for training and validation, respectively. Since 20 cases are insufficient to draw statistically meaningful results, in these experiments, the original 60 labeled MRI scans were used for model training and annotated another 50 MRI scans were used as the testing set. In order to enable a modality-wise comparison of abdominal organs, the same 13 organs were focused on as the Abdomen CT dataset. The annotations were generated by radiologists with the assistance of MedSAM and ITK-SNAP. New annotated dataset were released to the community to promote the development of abdominal organ segmentation in MRI.

4.2 Preprocessing

The preprocessing step is where nnU-Net demonstrates its significant automation capabilities. This fully automated segmentation pipeline is based on the 'dataset fingerprint,' which analyzes the input data and selects the necessary parameters. It comprises several key steps, such as data resampling, intensity normalization, cropping, and data augmentation. Additionally, nnU-Net manages various image modalities, such as MRI or CT scans, which possess unique intensity distributions and require different processing approaches for each of them. It creates pipelines fingerprints for this process, it analyses the different modalities and creates a 2d pipeline which is used by 2D U-Net for training. 3d_fullres and 3d_lowres pipeline if the modality includes 3D images, it is used by either 3D U-Net or 3D cascade U-Net for training. Let us observe the preprocessing steps in detail to understand their contributions to the nnU-Net framework.

4.2.1 Data Resampling

During the preprocessing process, when we input an image from the database, the system scans the images and resamples them to ensure uniform voxel spacing and for 2D images it ensures uniform pixel spacing. This is performed on all the images in the database, mainly due to medical images derived from different scanners, such as MRI and CT, which often have varying voxel spacing. Voxel spacing refers to the distance between individual voxels (volumetric pixels) in a 3D image or volume. Consistent voxel spacing also helps them to learn the geometrical features accurately. To enable this model to learn and conduct segmentation accurately, those images are resampled to a common voxel spacing, typically the median voxel spacing of their respective dataset. Two types of interpolation methods were used for the input images. In contrast, 3rd order spline interpolation is used to preserve image quality and the nearest neighbor interpolation for the respective segmentation masks to maintain the discrete nature of the label classes.

4.2.2 Intensity Normalization

Normalization is vital in the segmentation process, particularly in medical images. As discussed, medical images have different modalities. In order to increase the learning efficiency of the models, normalization is performed. For CT images, the intensity scale is always absolute. Hence, they are normalized based on the statistics of their entire dataset. For CT, the input image intensities are clipped to a specific range, usually based on 0.5 and 99.5 percentiles of the intensity distribution, followed by a z-score normalization based on the mean and standard deviation of the entire collected dataset. For the MRI dataset, a normal z-score normalization is conducted. For 2D images, Unit Variance normalization is done, this involves in calculating the mean and standard deviation of the pixel intensities in the image and then using these values to standardize pixel intensities

4.2.3 Cropping

Cropping is mainly performed for one reason: to introduce uniformity across all the images with the aim that the model will face less computational burden, leading to optimized GPU usage. If an image is larger than a predefined size, the model identifies and removes non-essential regions, typically background areas, which are designated with a value of 0 in the JSON document. Conversely, if an image is smaller than the specified dimensions, it is padded to achieve the required uniformity. This preprocessing step ensures consistency across the dataset before the images are forwarded for further processing.

4.2.4 Data Augmentation

Data augmentation is performed when the available dataset is limited. This generalization technique is mostly done during the training phase. Medical images, particularly those with accompanying segmentation masks, are challenging to acquire. The datasets utilized in this research are primarily sourced from challenges, as they are not readily available. While Image Segmentation models can learn from a relatively small number of images, there are instances where overfitting occurs when training on limited data. To mitigate this risk, data augmentation is ensured to increase the diversity of samples in the training dataset. It includes a variety of geometric transformations, including horizontal rotations, vertical rotations, flipping, scaling, elastic deformations, mirroring, and adjustments in contrast and brightness. Specific parameters for data augmentation are defined according to the modality of the image, whether 2D or 3D, ensuring that the augmentation is tailored to the requirements of the image. This approach helps enhance the model's generalization capabilities, reducing the likelihood of overfitting in the context of limited data availability. Note that, during our training phase, we have disabled the use of Data Augmentation due to hardware constraints.

4.3 Training

Following the preprocessing stage, the subsequent phase is training the neural networks. In the training process, after the image configuration (Fingerprint Pipeine) and the dataset ID are set, all networks are trained using certain Epochs and minibatches. In the training process, a hybrid loss function comprising both cross-entropy and dice loss is employed to optimize the model. The optimization leverages stochastic gradient descent with Nesterov momentum, starting with a learning rate of 0.01, which is used for learning network weights. Nesterov momentum is an advanced version of the standard momentum used to optimise things in the deep neural network. Instead of using past gradients to push things forward, it calculates where the gradient is going by anticipating the future position. The learning rate is decayed using the 'poly' learning rate policy. In this policy, the learning rate follows a polynomial decay over the epochs, which helps in improving the model's convergence. After training, the network outputs which are the validation datasets are evaluated against the ground truths, and the loss function is calculated.

The loss function which is calculated is a combination of cross-entropy and Dice loss, effectively addressing the challenges posed by class imbalances within the dataset. This is further supported by a deep supervision mechanism, where losses are computed at multiple resolutions using downsampled versions of the ground truth segmentation masks. The total loss is then calculated as a weighted sum of these individual losses, with the weights decreasing by half with each subsequent resolution level, thereby emphasizing finer details in the early stages of the network.

To construct minibatches for training, a mix of random sampling and targeted oversampling is used. Specifically, two-thirds of the samples are randomly selected, while one-third are chosen to ensure the inclusion of foreground classes, which is essentially zero, addressing the potential issue of class imbalance. Moreover, data augmentation techniques such as rotations, scaling, and noise addition are applied dynamically during training to enhance the model's ability to generalize from the training data to unseen data, in our case we are not using it.

4.4 Inference

Inference is performed on the testing dataset, which always contains about 20% of the original dataset, before it is divided for testing and training. In the inference phase, the prediction is conducted using a method known as the sliding window approach. This is due to the fact that a patch-based approach was used to send the data during the training phase. Hence, it involves moving a window across the images, whose dimensions are identical to the patch size. At each step of the sliding window, a prediction is made.

To reduce the risk of missing important features, adjacent windows are made to overlap. They overlap at half the size of the patch, as a result each patch is covered multiple times. Even if this step helps in accurate prediction, there is a certain disadvantage which is the accuracy of segmentation can diminish near the edges of each window. To mitigate this issue, a Guassian importance weighting technique was employed. This technique assigns greater importance or weight to the predictions made for the centre voxels within each window compared to those near the edges when combining overlapping predictions. This weighting helps to smooth out the predictions and reduce stitching artifacts when merging the overlapping windows.

During the inference stage, testing time augmentation (TTA) is intentionally omitted from the evaluation process to maintain efficiency and reduce computational demands. This is despite its potential to improve performance, especially considering the significant increase in computational load it imposes on both 2D and 3D datasets.

Chapter 5

Evaluation and Result

5.1 Discussion

For the training and evaluating, three distinct methodologies are selected, which includes CNN-based networks (SegResNet and nnU-Net), transformer-based networks (UNETR and Swin-UNETR) and segmentation networks that is close to our method, the Mamba network integrated with the nnU-Net model, U-Mamba_Bot and U-Mamba_Enc. The U-Mamba_Bot network incorporates the Mamba network at the bottleneck of the U-Net architecture and the U-Mamba_Enc incorporates the Mamba network at every Encoder available within the U-Net architecture. These models are evaluated, with our newly introduced novelty, MambaBot_Moe and MambaEnc_Moe. The MambaBot_Moe architecture consists of the mixture of experts at the beginning of the Mamba Network at the bottleneck and the MambaEnc_Moe contains the Mixture of Experts at the beginning of the Mamba network at every encoder module in the U-Net Architecture.

For evaluation purpose, a Linux-based environment was required and CUDA version of 11.8. These are must needed requirements for the program to run. The hardware requirements were fulfilled by a cloud based platform called as "Paperspace". They host notebooks and virtual space to run and train AI models. The notebooks that I trained my models were equipped with 45 GB of RAM and an Nvidia RTX A6000 GPU with 16 GB of memory. The original base network, nnU-Net has suggested to run 1000 epochs which would require even a high-end GPUs such as multiple Nvidia A100s or RTX3090 and the time duration to run each model would approximately take 36 hours per model. Hence, we have tried to run the maximum number of Epochs that our hardware can utilize or handle.

The evaluation metrics that are used includes the Dice Similarity Coefficient (DSC), the Normalized Surface Distance (NSD), and F1 Scores which is applied across all four datasets to evaluate their test segmentation results. With the help of Dice Similarity Coefficient, the overlap between the predicted and ground truth is measured and the Normalized Surface Distance, evaluates the average surface distance between the two. Now, as we have seen an overview of the hardware details and the metrics to analyse our network model. We use it to assess the performance of the four datasets previously mentioned.

5.1.1 Comparison of all Networks with Dental Dataset

For this dataset, we have trained all models for 100 Epochs and about 100 minibatches were set. Due to the limited amount of dataset, we got the evaluation results against the test dataset rapidly. The number of experts is only set to one, as there is only one region of Interest, which is the teeths.

From the provided table, the CNN based methods have performed well when compared with the Transformer models. This can be mainly due to the limited size of the dataset, causing the model to overfit. An overall difference of about 8% can be seen in Dice Similarity Coefficient, between both the CNN based methods and the Transformer methods.

The U-Mamba models, the U-MambaEnc model has a Dice Similarity Coefficient of 0.79 and a Normalized Surface Distance of 0.84. At the same time, the U-MambaBot exhibits a superior performance of DSC 0.87 and an NSD of 0.92. This indicates the effectiveness of integrating the Mamba network within the U-Net architecture, especially at the bottleneck (as in U-MambaBot), where it seems more beneficial than within every encoder, as in U-MambaEnc.

Model	DSC	NSD	Time Taken for 1 Epoch (Seconds)
nnUNet	0.7445	0.7995	7-8
SegResNet	0.6442	0.6996	10-11
UNETR	0.5696	0.6455	19-20
Swin-UNETR	0.7925	0.8525	21-22
U-MambaBot	0.8708	0.9260	14-15
U-MambaEnc	0.7922	0.8434	23-24
MambaBot_MoE	0.8439	0.9000	11-12
MambaEnc_MoE	0.8257	0.8881	30-31

Table 5.1: Performance metrics of various models on the Dental Dataset

Focusing on the UMamba_MoE models, MambaBot_MoE has outperformed both the MambaEnc_MoE and U-MambaEnc by at least 2.16% and 6.13%, respectively, for DSC. Even over here, the MambaEnc_MoE has not outperformed the MambaBot_moe. As discussed before, one of the main reasons to assume the degraded performance of Enc models will be due to the lack of dataset. But even with all those drawbacks the models has performed well more than than we had anticipated.

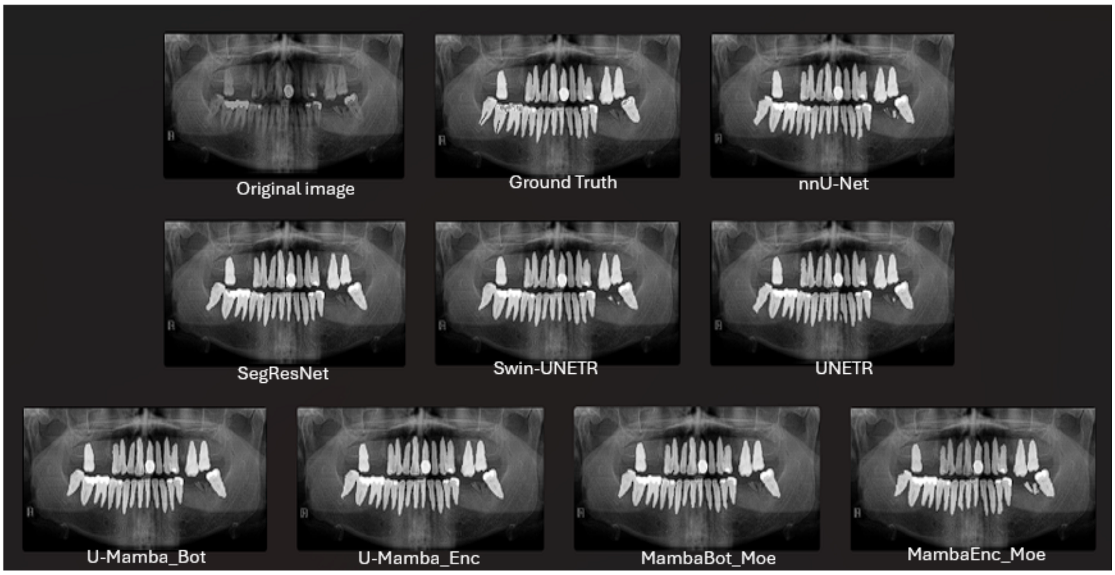

Figure 5.1: Result Visualization of an image from the Dental Dataset

In terms of computational efficiency, the training time taken by MambaBot_Moe ranges from

11 to 12 seconds per Epochs are lower and, the DSC and NSC values are also higher when compared to other models except U-MambaBot, which relatively indicates that the balancing power between computational demand and performance, even when the dataset is limited. The convergence data from the graph also states that only 60-70 epochs are enough to achieve the desired results, highlighting the computational advantage that the mixture of experts approach has achieved for both the models.

5.1.2 Comparison of all Networks with Microscopy Dataset

For this dataset, we have trained each model for at least 350 Epochs, which took at least an average of 7 or more hours. Here, we use F1 scores to calculate and assess accuracy. The reason we are doing this instead of calculating DSC and NSD is because microscopy images contain a considerable amount of imbalance classes. For instance, One microscopy image might have a class label of 2 and another 20 or more. Our evaluation focuses solely on the Mean F1 scores, otherwise called the Hard Dice Score. The F1 Mean scores are a harmonic mean of precision and recall. For the Microscopy data, the experts are set to 2.

Let us focus on the results of the evaluated data, The MambaBot_Moe follows the SegResNet closely with an accuracy of 0.59. MambaEnc_Moe and u-MambaBot also display strong performances with scores of 0.58329 and 0.5882, respectively. To our surprise, nnU-Net has not performed well to its expectations, it just shows an moderate accuracy of 0.49. Compared with other models, it shows a difference of about at least 10.8%.

Model	F1 Mean	F1 Median	Time Taken for 1 Epoch (Seconds)
nnUNet	0.4943	0.4615	32-33
SegResNet	0.60099	0.6306	59-60
UNETR	0.4828	0.4640	82-83
Swin-UNETR	0.3643	0.2377	82-83
U-MambaBot	0.5882	0.5526	73-74
U-MambaEnc	0.5756	0.5900	85-86
MambaBot_Moe	0.59717	0.5749	41-42
MambaEnc_Moe	0.58329	0.6165	53-54

Table 5.2: Performance metrics of various models on the Microscopy Dataset

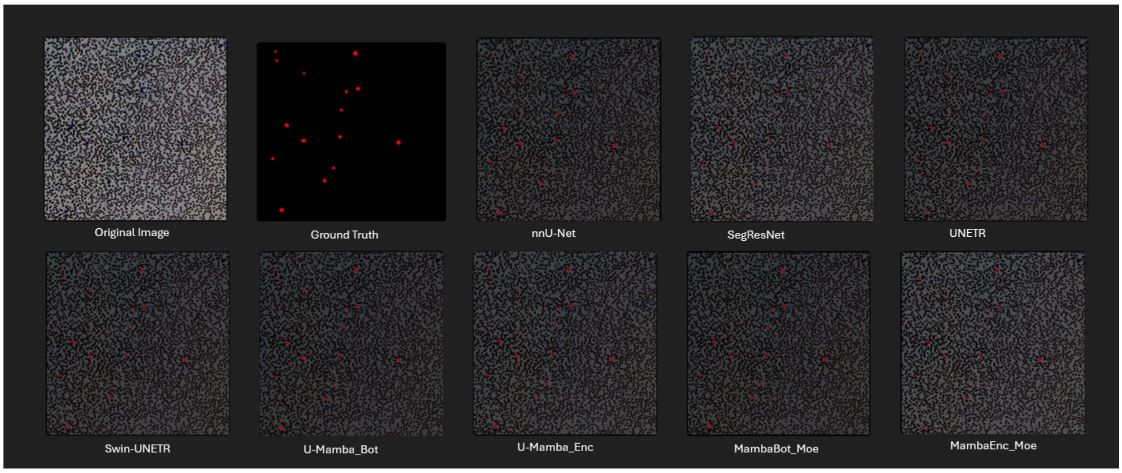

Figure 5.2: Result Visualization of an image from the Microscopy Dataset

Let us look at the computational efficiency for all the models that this dataset has achieved. The MambaBot_Moe is the most efficient with the shortest training time of 41-42 seconds per Epochs, followed by MambaEnc_Moe, about 53-54 seconds. Even though nnU-Net shows a faster training time, its low accuracy is a letdown. Segnesnet and Swin-unetr have taken the longest time per epoch, approximately 59-60 and 82-83 seconds, respectively. The convergence data from the graph also states that only 200-210 epochs are enough to achieve the desired results for both of the mixture of experts approaches, highlighting the computational advantage that the approach has achieved for both the models.

5.1.3 Comparison of all Networks with Abdomen CT Dataset

This dataset consists of 3D images, which poses a difficult challenge due to their extensive demand of computing resources. Initially, all the models were trained for only 100 epochs. When evaluated, the resulting DSC and NSD were too low, and the segmentation images were unsatisfactory upon evaluation with the ground truth. To address this issues, we made use of the checkpoint files which was available for each model except the ones that we have tried to propose. It was posted by the original paper, U-Mamba. These checkpoint files represent the state of the models after being trained for 1000 epochs. Checkpoint files are commonly used by machine learning to save the state of a model to a specific point of time, during training. Using these checkpoint files, each model is evaluated on a testing dataset, the evaluation is done with the testing dataset and then each of those testing dataset were evaluated with the ground truth to assess accuracy of the model.

Model	DSC	NSD	Time Taken for 1 Epoch (Seconds)
nnUNet	0.8614	0.8972	38-39
SegResNet	0.8084	0.8461	74-75
UNETR	0.6824	0.7004	82-83
Swin-UNETR	0.7594	0.7663	88-89
U-MambaBot	0.8718	0.9037	77-78
U-MambaEnc	0.8726	0.9041	81-82
MambaBot_Moe (100 Epochs)	0.8399	0.8647	51-52
MambaEnc_Moe (100 Epochs)	0.8425	0.8725	65-66

Table 5.3: Performance metrics of various models on the Abdomen CT Dataset

As discussed, both the Mixture of Experts approach model were run only for 100 Epochs. The MambaBot_Moe has been evaluated with a DSC of 0.83 and NSD of 0.86, which has always outperformed the SegResNet, UNETR, and Swin-UNETR. The MambaEnc_MoE also outperformed the above said models, its accuracy is just marginally higher than MambaBot_Moe. For 2D images, the MambaEnc produced results that were always moderate, but when working with 3D images, it produces some significant results. This suggests there might be an issue with, Data resampling function, it must be checked for the program to run 2D images effectively. It can be primarily due to issues in the preprocessing phase where the problem arises at the Data resampling step. Despite these hurdles, we have made significant strides towards our goal of reducing training times while maintaining high accuracy, and we are optimistic about the potential for further improvements. From the previous architecture, according to the convergence provided, it is seems that we can just run the training for 300 Epochs alone and achieve the optimal results for all the models.

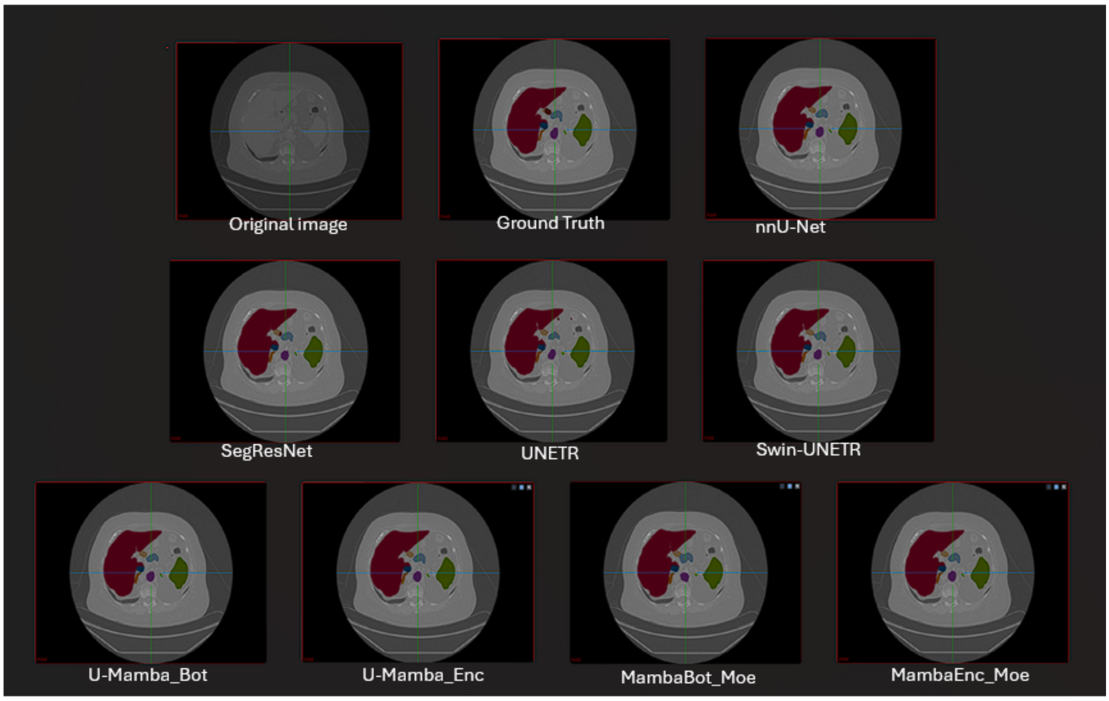

Figure 5.3: Result Visualization of an image from the Abdomen CT Dataset

As for the computation capacity, nnU-Net has the shortest epoch time of 38-39 seconds. The MambaBot_Moe and the MambaEnc_Moe also have a shorter training period of 51-52 seconds and 65-66 seconds respectively. According to the convergence, we can say that if the mixture of experts approach models were run for at least 150 epochs or closer to 200 epochs, we could have achieved our desired results. The less epoch time has highlighted the computational advantage the approach has achieved for both the models. Over here the experts are set to 13, because the dataset contains 13 regions of interest which are gall-bladder, stomach, pancreas and many more.

5.1.4 Comparison of all Networks with Abdomen MRI Dataset

The Abdomen MRI dataset also includes 3D datasets. Therefore, these datasets also follows the same methodology that was proposed above. The Mixture of experts model are run for only 100 Epochs and subsequently the segmentation results are evaluated with the ground truth to obtain the results of the metrics. The number of experts are set to 13, which is equal to their region of interest available.

Model	DSC	NSD	Time Taken for 1 Epoch (Seconds)
nnUNet	0.8303	0.8996	45-46
SegResNet	0.8114	0.8815	72-73
UNETR	0.6866	0.7440	81-83
Swin-UNETR	0.7564	0.8217	85-86
U-MambaBot	0.8569	0.9222	72-73
U-MambaEnc	0.8564	0.9205	80-81
MambaBot_Moe (100 Epochs)	0.7880	0.8568	57-58
MambaEnc_Moe (100 Epochs)	0.8052	0.8721	77-78

Table 5.4: Performance metrics of various models on the Abdomen MRI Dataset.

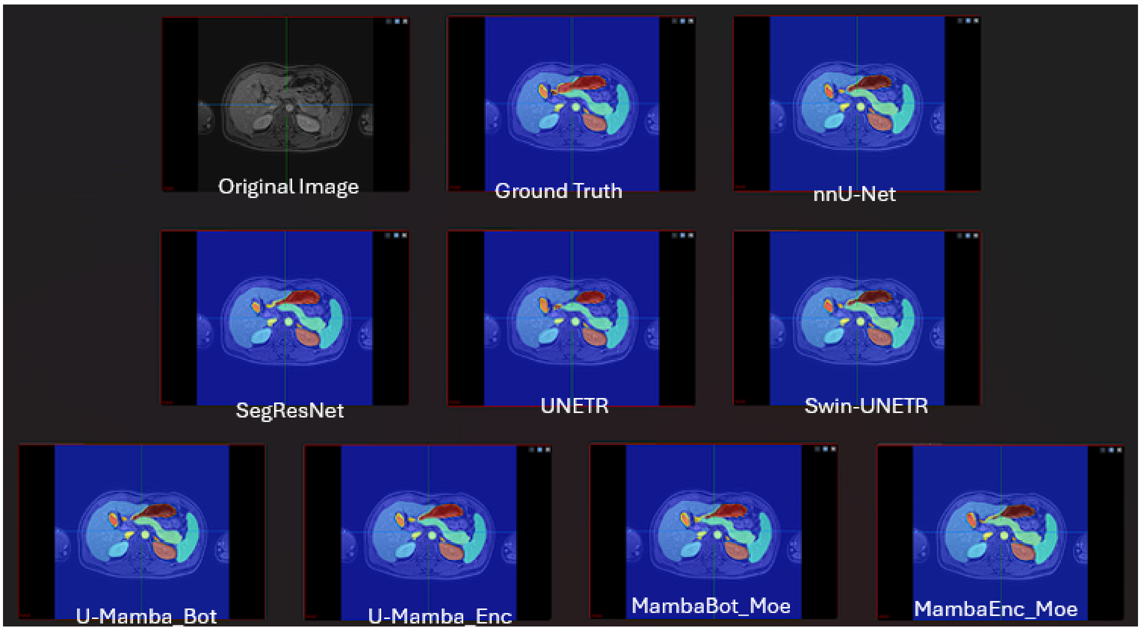

Figure 5.4: Result Visualization of an image from the Abdomen MRI Dataset

Even if they are trained for only 100 Epochs, the MambaEnc_Moe and MambaBot_Moe produce results of 0.805 and 0.788, respectively, for DSC. With respect to NSD, they have achieved scores of 0.87 and 0.85 for MambaEnc_Moe and MambaBot_Moe, respectively, which highlights their ability to perform competitively even with fewer training epochs.

In terms of computational efficiency, nnU-Net has achieved the shortest Epochs, which is about 45-46 seconds. The next model to achieve notable efficiency is the MambaBot_Moe which has achieved an Epoch duration of only 57-58 seconds to run one Epoch and then MambaBot_Moe has taken about 72-73 seconds. These two Mixture of expert models have outperformed the rest of the models except nnU-Net in terms of computational demand efficiency. According to the convergence, we could have achieved our desired results if the Mixture of expert models were run for at least 150 epochs or closer to 180 epochs.

Chapter 6

Conclusion

6.1 Summary

Despite some challenges of our models have shown promising results, particularly for 3D image datasets. We have achieved our segmentation results with just 100 Epochs of training. This efficiency in reducing the training time and achieving accurate segmentation results primarily aligns with our established goals. The performance with 2D image datasets could have been more effective when compared to the other models. It is assumed that there has to be some issues in the preprocessing phase, which is Data Resampling. The model adapts well to 3D images, but when it comes to 2D images, it is required to have some further refinements. From the original paper there has been many discussions about this particular thing in the GitHub platform and they recently released a new model that takes care of these issues and works properly. Despite these hurdles, we have made significant strides towards reducing training times while maintaining high accuracy, and we are optimistic about the potential for further improvements.

6.2 Limitations

From our evaluation, the novel integration of U-Mamba and the Mixture of Experts has shown significant changes in the performance of the image segmentation process. There are two main limitations that are easily resolvable, but we are facing these issues due to limited resources and time. Firstly, the number of experts is set to a maximum of 6, which is to likely limited the potential and major benefits of the Mixture of Experts approach. If multiple experts are defined, the system can easily leverage the mamba networks strengths in capturing long-range dependencies and improves the segmentation results in-tun. We could not increase the number of experts due to our system's lack of available CUDA memory. A GPU with at least 8 CPU cores and 50 GB of memory would be required to run the code smoothly and explore the implementation of additional experts. This issue arises only when we have a lot of class labels in a dataset.

Another system limitation we faced was that the code was only run for a maximum of 100 epochs for 3D datasets and the results presented in the tables above for the Mixture of Experts evaluation are from these limited training phase. The original paper (Isensee et al., 2018) recommended running each model for at least 1000 epochs to achieve accurate and reliable results. However, due to time constraints and system limitations it prevented us from conducting the experiments over such an extended duration.

As discussed in the previous paper, the purpose of the original study was to benchmark the performance of network architectures rather than pursuing state-of-the-art performance on this specific task. The U-Mamba architecture, which we used as a backbone, can serve as a backbone network in a state-of-the-art semantic segmentation framework. With the help of our novel integration, there is an opportunity to improve model performance even further.

6.3 Future Work

6.3.1 Longer Training

A key area for future work would be to extend the training duration from 100 epochs to 1000 epochs, as initially recommended. Longer training is likely to result in higher model stability, better accuracy and better prediction. By optimizing the code to be more computationally efficient or finding a system that has a great GPU core or using cloud-based platforms such as Google Cloud, AWS and Azure that provide scalable GPU resources for extended training periods could enable such lengthy training.

6.3.2 Increasing the number of Experts

Another important goal to work on in the future would be to increase the number of experts in the program by having better hardware to allow for the exploration of the use of multiple experts in the model. Increasing the number of experts is expected to take full advantage of the Mixture of Experts approach for certain datasets, potentially leading to significant improvements in segmentation accuracy and model robustness.

6.3.3 Improving the mixture of experts

There is a newly introduced concept called the Expert Choice Routing (Zhou et al., 2022). It involves in optimizing the allocation of tasks among the different experts defined based on the inputs specific abilities. The goal is to match each task in an input with the right expert, if it isn't then they tend to get under trained. In this new strategy, instead of the tasks choosing the experts, the experts chose the tasks. This method ensures that experts focus on the inputs that best match their skills. By utilizing this method, it is said to improve the performance by at least 2x.

Bibliography

Cheng, G. and Zheng, J. Y. (2021). Semantic segmentation for pedestrian detection from motion in temporal domain.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv:2010.11929 [cs].*

Gu, A. and Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces.

Gu, A., Goel, K., and Ré, C. (2022). Efficiently modeling long sequences with structured state spaces. *arXiv:2111.00396 [cs].*

Hao, S., Zhou, Y., and Guo, Y. (2020). A brief survey on semantic segmentation with deep learning. *Neurocomputing*.

Hatamizadeh, A., Nath, V., Tang, Y., Yang, D., Roth, H., and Xu, D. (2022). Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. *arXiv:2201.01266 [cs, eess].*

Hatamizadeh, A., Tang, Y., Nath, V., Yang, D., Myronenko, A., Landman, B., Roth, H., and Xu, D. (2021). Unetr: Transformers for 3d medical image segmentation. *arXiv:2103.10504 [cs, eess].*

Hendrycks, D. and Gimpel, K. (2020). Gaussian error linear units (gelus). *arXiv:1606.08415 [cs].*

Isensee, F., Jaeger, P. F., Kohl, S. A. A., Petersen, J., and Maier-Hein, K. H. (2020). nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18:203–211.

Isensee, F., Petersen, J., Klein, A., Zimmerer, D., Jaeger, P. F., Kohl, S., Wasserthal, J., Koehler, G., Norajitra, T., Wirkert, S., and Maier-Hein, K. H. (2018). nnu-net: Self-adapting framework for u-net-based medical image segmentation. *arXiv:1809.10486 [cs].*

Kayalibay, B., Jensen, G., and van der Smagt, P. (2017). Cnn-based segmentation of medical imaging data. *arXiv:1701.03056 [cs].*

Kirillov, A., He, K., Girshick, R., Rother, C., and Dollár, P. (2019). Panoptic segmentation. *arXiv:1801.00868 [cs].*

Lin, M., Chen, Q., and Yan, S. (2013). Network in network.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv:2103.14030 [cs].*

Luo, D., Zeng, W., Chen, J., and Tang, W. (2021). Deep learning for automatic image segmentation in stomatology and its clinical application. *Frontiers in Medical Technology*, 3.

Ma, J., Li, F., and Wang, B. (2024). U-mamba: Enhancing long-range dependency for biomedical image segmentation.

Myronenko, A. (2018). 3d mri brain tumor segmentation using autoencoder regularization. *arXiv:1810.11654 [cs, q-bio]*.

Paneru, S. and Jeelani, I. (2021). Computer vision applications in construction: Current state, opportunities challenges. *Automation in Construction*, 132:103940.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science*, 9351:234–241.

Roy, A. G., Navab, N., and Wachinger, C. (2018). Concurrent spatial and channel squeeze excitation in fully convolutional networks. *arXiv (Cornell University)*.

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

Shen, K., Guo, J., Xu, T., Tang, S., Wang, R., and Bian, J. (2023). A study on relu and softmax in transformer. *arXiv (Cornell University)*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need.

Xu, Z., Zhang, W., Zhang, T., Yang, Z., and Li, J. (2021). Efficient transformer for remote sensing image segmentation. 13:3585–3585.

Zhou, Y., Lei, T., Liu, H., Du, N., Huang, Y., Zhao, V., Dai, A., Chen, Z., Le, Q., and Laudon, J. (2022). Mixture-of-experts with expert choice routing.