Master's thesis
Master's Programme in Data Science

Vision Transformers in Brain Image Segmentation

Jussi-Jaakko Mankki

February 18, 2025

Supervisor(s): Professor Laura Ruotsalainen

Examiner(s): Professor Laura Ruotsalainen
Dr. Klavdiia Bochenina

University of Helsinki
Faculty of Science

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Degree programme
Faculty of Science	Master's Programme in Data Science

Tekijä — Författare — Author

Jussi-Jaakko Mankki

Työn nimi — Arbetets titel — Title

Vision Transformers in Brain Image Segmentation

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidantal — Number of pages
Master's thesis	February 18, 2025	63

Tiivistelmä — Referat — Abstract

Image segmentation is a challenging task that involves partitioning an image into regions based on a specific criteria. For brain images, segmentation methods can be used for visualization of structures, delineation of injuries, analysis of brain development, and planning of surgical procedures. These methods aid medical professionals with early detection of diseases, precise diagnoses, and effective treatment planning.

Approaches based on Convolutional Neural Networks (CNNs) have shown significant progress in automating brain image segmentation. However, these methods often struggle with capturing long-range dependencies and contextual information, leading to limitations in segmentation accuracy. Recent advances in Vision Transformers (ViTs) have demonstrated superior performance in computer vision tasks by modeling global spatial relationships.

This thesis reviews the application of Transformer-based architectures in brain image segmentation, focusing on their integration with existing CNN models, particularly the U-Net. A review of state-of-the-art Transformer-based segmentation models, including UNETR, TransBTS, Swin-UNETR, and VT-UNet, is presented. The advantages of these models over CNN-based approaches, particularly in brain tumor and brain tissue segmentation, are also discussed.

An experiment is conducted to evaluate the impact of Transformer integration using a small brain MRI dataset. The findings indicate that the Transformer-integrated architectures may not necessarily improve the performance of the U-Net architecture on limited data. However, with further refinement, including more suitable loss functions, larger datasets, and improved model designs, Transformer-based models could still show potential advantages in small-scale testing.

ACM Computing Classification System (CCS):
Computing methodologies → Artificial intelligence → Computer vision → Computer vision problems → Image segmentation
Applied computing → Life and medical sciences → Computational biology → Imaging

Avainsanat — Nyckelord — Keywords

computer vision, transformer, vision transformer, brain, semantic segmentation, medical imaging

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1. Introduction

In medical imaging it is common that a specialist examines the human brain slice-by-slice to form an understanding of the subject as a whole [53]. However, the large and complex data produced by medical imaging devices makes manual data extraction a laborious, time-consuming, and challenging task, where even small errors in analysis become the responsibility of an individual clinician [13]. Consequently, there is a growing demand for algorithmic methods that assist with analysis and diagnostics.

Image segmentation is the process of dividing an image into distinct regions based on specific criteria, represented as masks or contours. In semantic segmentation, each pixel is assigned a label corresponding to a specific category. The goal of segmentation is to divide an image into semantically meaningful, cohesive, and distinguishable regions that share common characteristics such as intensity, depth, color, or texture [13, 35].

Medical image segmentation involves identifying lesions and organs, which aids medical experts with early support, accurate diagnoses and treatment planning [67]. With brain images specifically, segmentation methods are used in visualization of structures, delineation of injuries, analysis of brain development, and planning of surgical procedures [13]. Some studies target specific conditions, such as neonatal brain development, traumatic brain injury, and brain tumor segmentation [63]. Among these, brain tumor segmentation has received the most attention, with a wide range of deep learning methods proposed for segmenting tumor images [59], while the segmentation of brain tissues and structures has received comparatively less attention in research.

In recent years, Convolutional Neural Networks (CNNs) have been the primary architecture for computer vision tasks [15], significantly advancing medical image analysis [6]. However, their use in medical imaging still faces challenges, including limitations with annotated datasets, demands for computational power, and issues with interpretability. Various techniques have been developed to address these challenges, and ongoing research continues to introduce more effective methods [51]. More recently, the Transformer architecture, originally designed for machine translation tasks, has gained traction in computer vision field. The Transformer demonstrates improved performance in image segmentation compared to earlier deep learning approaches, and offers benefits such as reduced inductive bias and lower computational costs [52].

The objective of this thesis is to study the application of Transformer architectures in brain image segmentation, focusing on their architectural design and accuracy improvements over traditional CNN-based approaches. This thesis provides a background on medical imaging and brain image segmentation, summarizes the the theoretical framework of Transformer-based models for computer vision, reviews recent state-of-the-art Transformer-based architectures designed for brain image segmentation, and evaluates the performance of Transformer-based architectures in brain segmentation tasks using a small-scale experiment. Since most state-of-the architectures extend U-Net, a widely used CNN-based segmentation model, the research question in this thesis is to study how the integration of Transformer with U-Net impacts brain image segmentation performance.

This thesis continues with a background on medical imaging and brain image segmentation in Chapter 2. Chapter 3 discusses the theoretical foundation of the Transformer architecture and its extensions to computer vision tasks. Recent state-of-the-art architectures designed for brain image segmentation that utilize the Transformer are reviewed in Chapter 4. Experimental comparisons of different architectures are presented in Chapter 5. Chapter 6 concludes with a summary of the key findings and final remarks on the topic.

2. Background

This chapter covers the main principles of non-invasive medical imaging, particularly Magnetic Resonance Imaging (MRI) and Computerized Tomography (CT), which serve as the primary imaging modalities to provide imaging data for the segmentation models. A brief summary of brain segmentation methods is presented, including some traditional approaches prior to the rise of deep learning.

2.1 Medical Imaging

MRI and CT are the primary imaging modalities used in brain studies, each offering specific imaging qualities. MRI is especially suited for imaging soft tissue contrast, which is essential for distinguishing brain structures, while CT provides high-resolution imaging for detecting brain hemorrhages and fractures. Understanding these modalities helps in adapting segmentation models to specific tasks, as each imaging type introduces unique challenges, such as their difference in noise and artifacts. Images from both MRI and CT have been used for training brain segmentation models.

Magnetic Resonance Imaging

MRI is a medical imaging technique based on nuclear magnetic resonance, which produces detailed cross-sectional images of the human body, referred to as slices. Most MRI scans focus on the central nervous system, but MRI methods can also be used for real-time imaging of the body's moving structures [19]. MRI is highly suitable for studying brain structures and damage, as it allows for the contrasting of soft tissues in the body [42].

A typical MRI scan includes multimodal information, where images taken with different sequences interact with each other. Pulse sequences refer to a collection of imaging settings used to contrast various body tissues [60]. A key factor affecting the sequence is relaxation time, which is a tissue-specific constant that cannot be changed during imaging [19]. Relaxation time affects the magnitude of the measurable magnetic resonance signal, which in turn creates contrast differences between tissues in the MRI

image. In this context, contrast differences refer to the variation in signal intensity between different areas of the image. High intensity, or hyperintensity, appears bright in the MRI image, whereas low intensity, or hypointensity, appears dark [12]. The choice between relaxation times depends on the imaging goal and the properties of the tissue being studied. Multimodal information (using multiple images with different relaxation times) provides a clearer and more comprehensive view of the target compared to an image with single modality.

Relaxation times are typically categorized into T1- and T2-weighted relaxation. T1-weighted images provide a detailed anatomical view of the tissue and are suitable for forming an overall perspective. T1 with contrast material gadolinium (T1c) can help in distinguishing small pathologies, such as tumors, from the image [47]. T2-weighted images reduce tissue visibility but highlight fat and fluid in the image. T2-weighted images are often used to emphasize pathological changes with increased fluid, such as tumors. Additionally, the signal from cerebrospinal fluid (CSF) can be suppressed using Fluid-Attenuated Inversion Recovery (FLAIR) to highlight structures near fluid from a T2-weighted image [60, 62]. Figure 2.1 shows the difference between T1-weighted, T2-weighted, and FLAIR brain images.

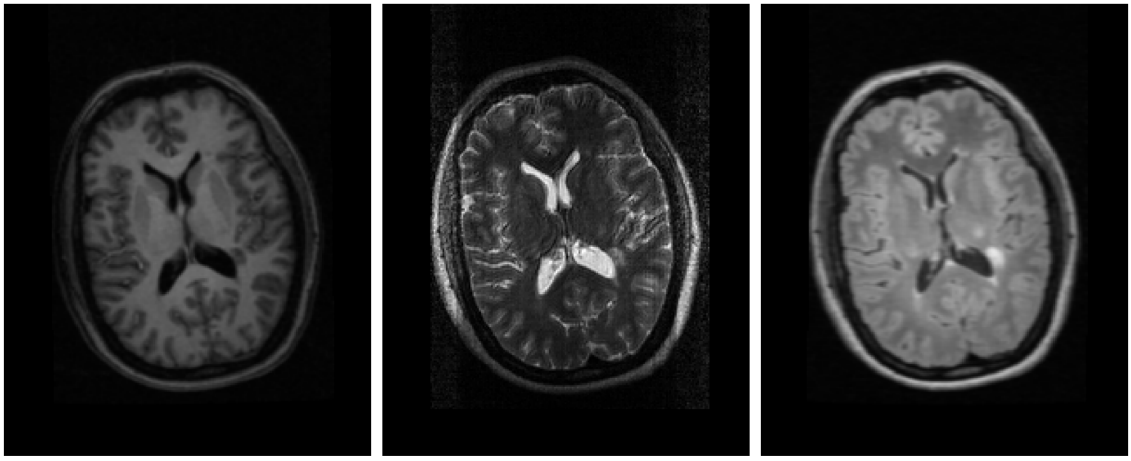

Figure 2.1: Images of MRI with different modalities [42]. T1-weighted image (left) highlights fat as bright and fluid as dark. T2-weighted image (middle) highlights both fat and water bright while keeping tissues dark. The FLAIR scan (right) suppresses CSF and emphasizes pathological fluid as bright.

Artifacts, which are distortions that degrade image quality and interfere with analysis, are often present in MRI [13]. MRI artifacts can be roughly categorized into two types: artifacts caused by the equipment and those caused by the patient or the equipment operator [37]. Artifacts caused by the equipment are relatively rare, although they can be difficult to diagnose. Artifacts resulting from the patient or operator are common but are often preventable and correctable once detected. The

most common source of artifacts is caused by movement, for example breathing by the patient or pulsation of blood flow. Movement can cause repetitive patterns (ghosting), blurred areas, or noise in the images.

Computerized Tomography

Computerized Tomography (CT) scans are a non-invasive tool for examining internal organs and tissues [51]. For segmentation tasks, CT is preferred for multiorgan abdominal segmentation and hard tissue segmentation, such as bones. Along with MRI, CT is a preferred choice in medical diagnosis due to its versatility.

In CT scanning, X-ray beams pass through a subject and measure the attenuation of different tissue densities across multiple angles to create cross-sectional images [45]. CT gathers attenuation coefficients from many angles, which are then processed to reconstruct images. Each cross-section represents a thin slice of the body, typically ranging from 1.5 to 10 mm, depending on the application [4]. For clearer imaging, thin contiguous slices reduce the loss of depth dimension, but may increase the image noise and artifacts [3, 44]. Thicker slices, on the other hand, reduce noise but lower the resolution of the image.

The properties of CT images are influenced by various parameters that affect resolution and contrast [44]. Resolution, in particular, is determined by detector size and scan parameters, such as the number of projections and the reconstruction filter. Smaller detectors produce higher resolution but increase the number of partitions and decrease detection efficiency, while a larger number of projections provides finer resolution. High-resolution reconstruction filters improve spatial resolution but are prone to producing more noise. Contrast, essential for distinguishing tissues, depends on inherent tissue properties, with higher beam energy generally reducing contrast. Contrast media injected into the bloodstream enhances the contrast for specific structures, such as the surrounding tissue of blood vessels.

As with MRI, CT head scans have limitations that affect imaging accuracy, including off-center spiral issues, patient radiation exposure, and artifacts that can obscure or mimic pathology [24]. Common artifacts include ring artifacts, noise, beam hardening, and scatter [3]. Ring artifacts are caused by a miscalibrated or defective detector, resulting in rings centered on the center of rotation. Noise, resulting from low photon counts, appears as bright or dark streaks. Beam hardening and scatter produce dark streaks between high attenutation objects (such as metal or bone). Ring artifacts can be often fixed by recalibrating the detector, while noise, beam hardening, and scatter can be mitigated through iterative reconstruction.

MRI and CT in Segmentation

Most brain segmentation research uses MRI over CT, as MRI is preferred for brain imaging due to its excellent soft tissue contrast [29]. However, MRI is not suitable for certain patients, such as those who are too large, claustrophobic, have implants, or cannot stay still. In these cases, CT is often used while being faster, more accessible, and cost-effective [24]. There have been examinations to segment soft tissue from CT images, although brain segmentation is more commonly studied using MRI rather than CT for better soft tissue contrast [63].

Lauric and Frisken [29] have shown that brain segmentation is feasible on CT data, but lacks the detail needed to differentiate gray and white matter compared to MRI, as shown in Figure 2.2. Segmentation methods can also vary based on imaging modality and target tissue. For instance, thresholding works well for bones in CT images due to high contrast but fails for soft tissues.

Over the years, there have been extensive research on MRI-based brain segmentation, while studies on CT brain segmentation have been limited [24]. Lenchik et al. [31] found that 94% of neurological segmentation studies used MRI, while only 5% used CT. While CT-based segmentation typically underperforms compared to MRI-based segmentation, some studies have applied deep learning for brain segmentation in CT with promising results, as shown by [25], [63], and [70].

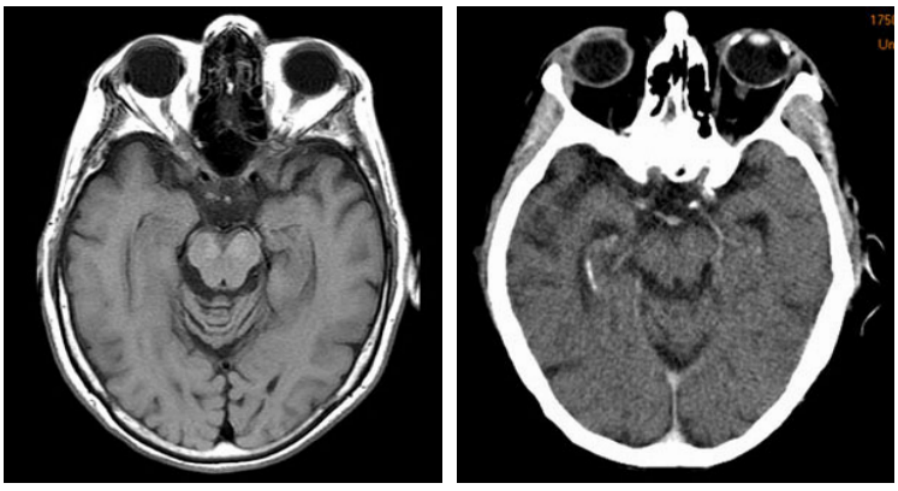

Figure 2.2: The MRI image (left) differentiates gray and white matter well, while the CT image (right) highlights bone structure but lack distinction of soft tissue [29].

2.2 Brain Image Segmentation

Semantic segmentation involves labeling each pixel or voxel in a data set according to specific image criteria, such as tissue type or tumor in medical images [29]. Traditional brain image segmentation methods, such as thresholding, clustering, manual feature extraction, and contour techniques [6], often rely on conventional algorithms that fail to capture the complex patterns in MRI images [52]. These methods mainly targeted segmenting three brain tissue types, which are white matter (WM), grey matter (GM), and cerebrospinal fluid (CSF) [63]. While effective in specific contexts, challenges with noise, variability in image quality, and lack of adaptability across varying imaging conditions degrade their applicability.

In addition to the three tissue types, many other brain tissues and anatomical regions, such as cerebellum, brain stem, and basal ganglia, hold significant physiological and pathological relevance [63]. These regions should receive attention for many reasons, such as advancements in pathological studies, improvement of deep learning models, enhanced diagnostic accuracy, early detection capabilities, and opportunities for cross-modal comparisons. Accurate segmentation of various brain structures is essential in applications such as radiation therapy planning, where precise knowledge of both tumor and surrounding healthy tissue locations ensures targeted radiation without harming unaffected areas [42]. In addition, quantitative measures of lesions and surrounding structures enable clinicians to explore disorder impacts on brain anatomy, potentially aiding in identifying biomarkers for early diagnosis and disease progression monitoring.

Recent advances in deep learning, particularly those involving Convolutional Neural Networks (CNNs), have significantly enhanced the accuracy of medical image analysis. Among these, U-Net has become a leading model for medical image segmentation [67] and is discussed in greater detail in Section 4.1. More recently, Transformer models have gained attention for their superior ability to capture complex spatial features in medical images, often outperforming previous state-of-the-art models based on CNNs [52]. In medical image segmentation, the Transformer is able to capture long-distance dependencies and global context, and to minimize feature loss to preserve relevant local information and integrity.

2.3 Challenges

Brain image segmentation faces multiple challenges, largely due to variability in image quality and the limitations of training data. Brain images often contain noise, artifacts, and inconsistent contrast due to differences in imaging devices and settings. This

variation complicates the segmentation of brain tissues, especially in clinical contexts where high precision is required. In addition, the availability of diverse, high-quality training data is limited, as medical images are difficult to gather in large quantities due to privacy concerns, high costs, and the need for expert annotation [1, 18].

Although there has been significant progress in automatic segmentation of brain tissues in recent years, many of the methods developed are not easily applicable to clinical settings due to the variable quality of imaging scans and the presence of pathologies, such as tumors or injuries [52]. As a result, clinicians must manually analyze the information, which is a time-consuming task and introduces variability based on the clinician's subjective judgment [42]. It is not expected that the results provided by algorithmic segmentation methods would replace clinicians' diagnoses; rather, these methods can be used to alleviate workload, provide a second opinion, or assist in modeling brain development [14]. This highlights the need for interdisciplinary collaboration between clinicians, radiologists, data scientists, and engineers to ensure robust solutions suitable for real-world medical applications.

One of the issues with traditional segmentation methods is that they have been developed to segment based only on specific intensity distributions [42]. However, sequences, imaging devices, and imaging methods all impact how a brain image appears. MRI scans used in studies may, for example, use the same sequence or come from the same manufacturer's imaging device, which can reduce the performance of segmentation methods when the intensity characteristics of the MRI being studied differ from those used in developing the method. Additionally, limiting training data to, for instance, only T1-weighted MRIs might slow down the integration of promising studies into clinical use. In the clinical setting, it is typical to take MRIs of the same region using multiple different sequences. These MRIs of the same region contain complementary information, which can and should be utilized for the development of segmentation methods.

Although methods utilizing deep learning have only been developed for a few years, they have proven to be more reliable than previously used segmentation techniques [1]. Brain image analysis has been a major challenge for computer-assisted techniques due to the brain's complex anatomy and variability in appearance, non-standardized sequences resulting from variations in imaging protocols, imperfections in image acquisition, and the presence of pathologies. More general techniques, such as deep learning, are best suited to handle the variability found in datasets. However, while deep learning has advanced the brain image segmentation models, they often struggle with limited data, imbalanced classes, and making predictions that are interpretable [70]. Additionally, fine-tuning these models also demands high computational power and expertise [52].

While deep learning models have shown promise in addressing some segmentation challenges, their effectiveness is held back by limited training datasets. Larger datasets are essential, as model accuracy has shown to improve with larger data size [1]. Outside of brain tissue segmentation methods, deep learning has shown significant results when there are millions of images available as training data. As deep learning methods improve when the size of the training dataset increases, the growing demand for brain images is justifiable.

There are several reasons for the lack of high-quality imaging data [18]. First, securing funding for the creation of quality teaching data is challenging, as developing such material requires medical expertise and various types of brain imaging. Additionally, privacy-related factors complicate the sharing of medical data. Despite these challenges, progress has been made in data availability. Studies routinely use publicly available datasets for the development and validation of novel methods.

In brain image segmentation, obtaining even thousands of training images is usually challenging, however, data augmentation has led to promising results for deep learning based models [1]. By adjusting the contrast, rotation, shape, and exposure of an MRI, neural networks can adapt so that the segmentation result is not significantly affected by the imaging method, imaging device, noise, or sequence. Efforts using data augmentation have helped the network learn invariance and robustness, even with limited training samples [49]. Yet, reliable segmentation models that perform consistently across various imaging devices and patient conditions are still under development [16, 42].

It will still take years before segmentation methods can be applied in clinical environments. A reliable method that is independent of the imaging device, sequences, imaging technique, or noise has yet to be developed. Additionally, integration into clinical settings requires collaboration with hospitals and changes in radiologists' working methods. The use of segmentation methods in clinical environments is also affected by the lack of regulations that would govern, for instance, the purpose of use, ethical considerations, and responsibility in diagnostics. Nevertheless, there is significant demand for segmentation methods. Computer-based segmentation methods are expected to become an essential tool in clinical environments, especially in qualitative diagnosis and studies where 3D reconstruction and visualization of anatomical structures are important [13].

3. Transformers in Computer Vision

The Transformer is a neural network architecture, originally developed for Natural Language Processing (NLP) tasks, particularly machine translation. It surpassed architectures based on Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), setting a new state-of-the-art baseline. The Transformer's success inspired researchers to adapt the architecture for image classification tasks, achieving results similar to or better than CNNs and RNNs [20]. This chapter begins by presenting the encoder-decoder structure, which sets the foundation for understanding the fundamental design of the Transformer, and the extensions that enabled the Transformer's application to image classification.

3.1 Encoder-Decoder

The encoder-decoder framework is a neural network architecture that is used for sequential data processing and generation. It is widely applied across natural language processing, computer vision, speech processing, and interdisciplinary fields [39], and commonly used for tasks such as image and video captioning, question answering for text and images, text summarization, anomaly detection, and image segmentation.

The encoder-decoder splits the neural network into an encoder and a decoder. The encoder extracts a fixed-length feature vector from the variable-length input to capture the most relevant patterns and relationships within the data [8]. The feature vector is passed to the decoder, which then generates a relevant variable-length output based on the features extracted by the encoder [39]. The encoder and decoder structures can be adapted with other deep learning models for more flexible feature extraction, for example, CNNs are commonly used for images and videos, while RNNs have commonly been used for sequential or structured data [43]. A basic encoder-decoder model is illustrated in Figure 3.1

A limitation of the baseline encoder-decoder approach is that the encoder must compress all input information into a single fixed-size vector, which can make it chal-

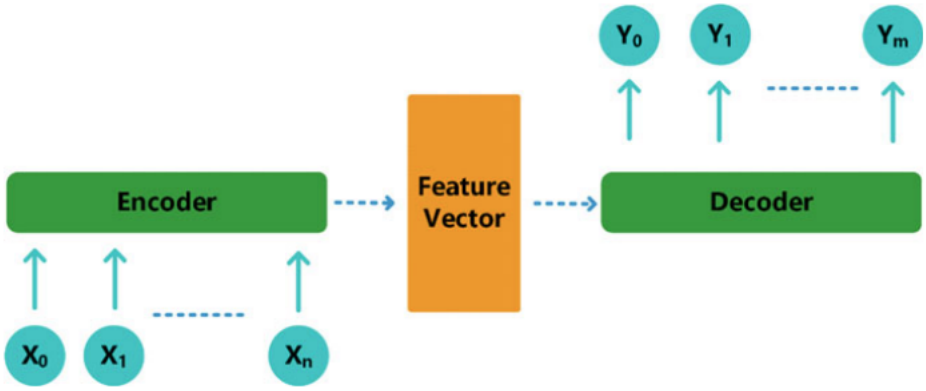

Figure 3.1: The encoder-decoder model [39]. $X = \{X_0, X_1, ..., X_n\}$ denotes the input and $Y = \{Y_0, Y_1, ..., Y_m\}$ denotes the output. The encoder creates a fixed-length feature vector from the input, which the decoder then uses to produce an output.

lenging for the model to capture long-term dependencies [39]. The attention mechanism addresses this limitation by extending the encoder-decoder model by encoding the input into a sequence of annotation vectors, which the decoder can adaptively use to generate the output. The attention mechanism is discussed more closely in Section 3.2.

3.2 Transformer

The Transformer model is an encoder-decoder based neural network architecture, introduced by Vaswani et al. [61] for NLP tasks, particularly machine translation. Before Transformers, existing models that convert input sequences into output sequences relied on RNNs or CNNs with encoder-decoder structures, often enhanced by attention mechanisms which connect the encoder and the decoder. The Transformer, however, uses only the attention mechanisms to capture global dependencies between inputs and outputs, eliminating the need for recurrence or convolutions. This results in an improved performance and reduced training complexity compared to prior models.

There are two key elements that make Transformers work well in NLP tasks - positional encoding and the attention mechanism, specifically Multi-Head Attention (MHA) and Self-Attention (SA). In the context of machine translation, positional encoding assigns order to input data for the network to learn and interpret the significance of word order. The attention mechanism allows a text model to focus on specific words in the input sentence when generating the output. MHA allows the model to attend to multiple representation subspaces simultaneously, capturing complex relationships within the data. SA is a refinement of traditional attention, where the attention is turned on the text itself to understand the context of each word within the input.

Positional encoding

Language models break down text into tokens, which are the smallest units that can represent words, subwords, or characters. However, as the Transformer lacks inherent mechanisms to recognize the order of tokens (contrary to recurrence or convolution), it uses positional encodings to include order information of a sequence [61]. These encodings are added to the input embeddings for both the encoder and the decoder, and share the same dimensions as the respective embeddings to allow summation.

The positional encodings can be learned or fixed [61]. Learned positional encodings are parameters that are trained as part of the model. These encodings are flexible, as they adapt during training to represent positional information. Fixed positional encodings use predefined functions, such as sinusoidal functions, to encode positional information. These encodings remain constant and do not change during training.

Attention

As discussed in Section 3.1, the encoder-decoder models transform input into a fixed-length feature vector, with the decoder generating output based on this encoded information. However, this approach has inherent limitations. Certain output elements rely on specific parts of the input, requiring detailed information about them, while irrelevant input data can mislead the model [39]. Moreover, the fixed-length vector representation can restrain the neural network's ability to process long sentences effectively [2]. The attention mechanism addresses these issues by enabling models to focus more precisely on relevant sections of the input during output generation.

The attention mechanism architecture was first introduced by Bahdanau et al. [2] to improve the performance in machine translation tasks. The input is encoded into a sequence of vectors, from which the decoder adaptively selects subsets to generate the output [39]. Their proposed model extends the basic encoder-decoder architecture by allowing the decoder to focus on relevant parts of the input sequence, eliminating the need to compress all input information into a fixed-length vector. This mechanism supports handling long-term dependencies and producing more detailed outputs.

An extension of the attention mechanism was introduced in the Transformer model by Vaswani et al. [61], demonstrating that the attention mechanisms can replace convolutional and recurrent layers in network architectures to improve performance and reduce computational complexity. An attention function maps a query and key-value pairs to an output, all represented as vectors. The output of the attention function is the weighted sum of the values, with weights determined by a compatibility function between the query and the corresponding key.

To get a better intuition of the queries, keys, and values used in the attention

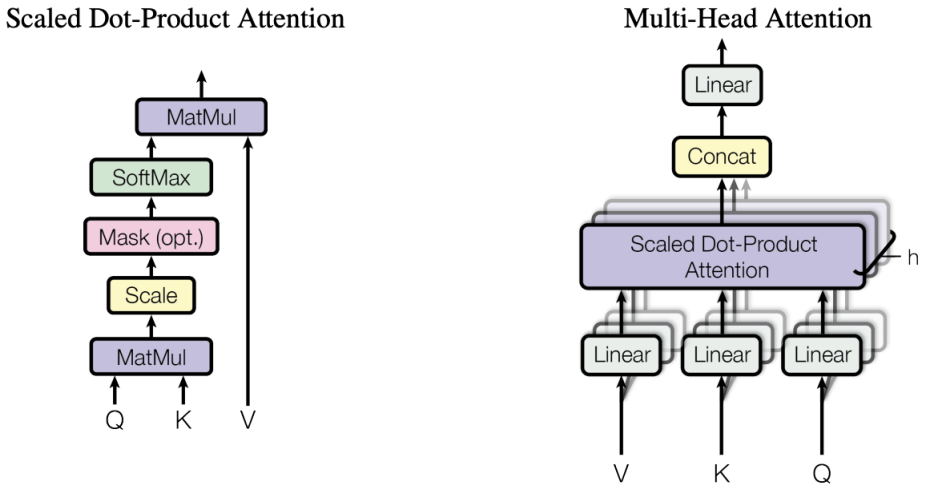

Figure 3.2: Scaled Dot-Product Attention (left) and Multi-Head Attention (right) [61]. The Scaled Dot-Product Attention inputs the queries Q, keys K, and values V. The dot product between Q and K^T measures the similarity between queries and keys, scaled by $\sqrt{d_k}$ to stabilize gradients, normalized with a softmax to produce probabilities, and then multiplied by the values V to generate the output. The Multi-Head Attention divides the queries, keys, and values into h heads, where each head processes a different subspace of the input. The output of all heads is concatenated and then combined into a single output using linear transformation.

mechanism, the interaction between these elements in the context of processing an input sequence can be considered as follows. Tokens, which are the smallest units of the input sequence (e.g. words), are mapped to a vector representation, known as an embedding, which encodes its meaning in a high-dimensional space. The query (Q) is used to represent what is being searched and is compared to all keys (K) in the sequence to measure similarity. The corresponding values (V) capture the meaning or context of the tokens. This process is analogous to information retrieval, where the query is used to identify the target, keys are treated as the search space, and values are used based on the relevance scores.

In the Transformer model, attention is computed using Scaled Dot-Product Attention, which acts as the compatibility function by calculating the dot product between the query and all keys. The result of the dot product is scaled by dividing by the square root of the key's dimension (d_k) to stabilize gradients and generate meaningful probabilities. The scaling prevents issues with overly large dot products, which could produce extremely small gradients in the softmax function. The matrix of the output is

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{3.1}$$

where queries are packed into a matrix Q, keys to a matrix K, and values to a matrix

V. The Scaled Dot-Product Attention provides the basic mechanism for focusing on specific input elements relevant to a given context. The process of the Scaled Dot-Product Attention is illustrated on the left side of Figure 3.2.

Multi-Head Attention

As an extension of the single-layer Scaled Dot-Product Attention, the Transformer uses Multi-Head Attention (MHA) in the model's encoder-decoder attention layers [61]. This mechanism enables the model to capture complex relationships by attending to multiple representation subspaces simultaneously. Instead of directly applying attention to the original query, key, and value vectors, these vectors are first linearly projected into different subspaces called "heads". Using multiple heads allows the model to simultaneously attend to information from different parts of the sequence or focus on various features of the same sequence. This design avoids potential information loss caused by compressing all attention into a single representation.

In MHA, the queries come from the prior decoder layer, while keys and values come from the encoder's output, enabling the decoder to focus on all input sequence positions. The process involves dividing the model into multiple attention heads, where each head projects the queries (Q), keys (K), and values (V) into distinct subsets, performs the Scaled Dot-Product Attention calculations independently, and combines their outputs. This process is illustrated on the right side of Figure 3.2.

For a single head i, the transformed vectors are computed as follows:

$$Q_i = QW_i^Q, \quad K_i = KW_i^K, \quad V_i = VW_i^V$$

where the projections are performed using learned weight matrices (W) specific to each head. These weight matrices, trained during the model's learning process, are defined as:

$$\begin{aligned} W_i^Q &\in \mathbb{R}^{d_{\text{model}} \times d_k}, && \text{which projects the query vectors,} \\ W_i^K &\in \mathbb{R}^{d_{\text{model}} \times d_k}, && \text{which projects the key vectors,} \\ W_i^V &\in \mathbb{R}^{d_{\text{model}} \times d_v}, && \text{which projects the value vectors.} \end{aligned}$$

The concatenated outputs of all heads are projected back to the model's dimension using the matrix $W^O \in \mathbb{R}^{h \cdot d_v \times d_{\text{model}}}$, which projects concatenated outputs of all heads back to the dimension of the model. The formulation of the MHA process is

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \\ \text{where head}_i &= \text{Attention}(Q_i, K_i, V_i) \end{aligned} \tag{3.2}$$

Self-Attention

The attention mechanism can be considered as a general technique to focus on specific parts of the input while processing a sequence. Self-Attention (SA) is a specific type of attention, that computes relationships within a single sequence [61]. The Transformer model uses only SA to understand its input and output.

The key characteristic of the SA mechanism, compared to the general attention mechanism, is that it operates within the same sequence to get a more precise understanding. SA is essentially the same as the attention mechanism, except that the Q, K, and V vectors are all derived from the same input sequence. This results in the SA calculating attention weights for each token relative to every other token in the sequence.

Both the encoder and the decoder in the Transformer model use SA, specifically MHA, but with subtle differences. The encoder includes SA layers, where Q, K, and V originate from the output of the previous encoder layer. Each position in the encoder can focus on all positions in the prior layer. In the decoder, SA layers ensure each position attends only to itself and preceding positions, maintaining the auto-regressive property where the current value is a function of the past values. This is achieved by masking invalid connections in the Scaled Dot-Product Attention, setting these connections to $-\infty$ before applying the softmax.

Model Architecture

Architecture of the Transformer model illustrated in Figure 3.3. The encoder of the Transformer model consists of a stack of $N = 6$ identical layers, each containing two sub-layers. The first sub-layer is a Multi-Head Self-Attention mechanism, and the second is a position-wise fully connected Feed-Forward Network (FFN), which is a two-layer neural network that processes each element of the input sequence independently. The term "position-wise" refers to the Transformer's ability to process input sequences in parallel, with the FFN operating on individual input vectors independently. Each sub-layer is followed by a residual connection and layer normalization for stabilization. The residual connection mitigates the vanishing gradient problem by summing the original input of a sub-layer to its output, which would otherwise lead to diminishing gradients during the training of the network. The residual connection and layer normalization are depicted as the "Add & Norm" in the sub-layers of Figure 3.3.

The decoder is similar to the encoder, as it also consists of a stack of $N = 6$ identical layers, but adds a third sub-layer. The Masked MHA ensures that the decoder only attends to earlier positions in the sequence to preserve the auto-regressive property. The second sub-layer performs MHA over the output of the encoder stack where the

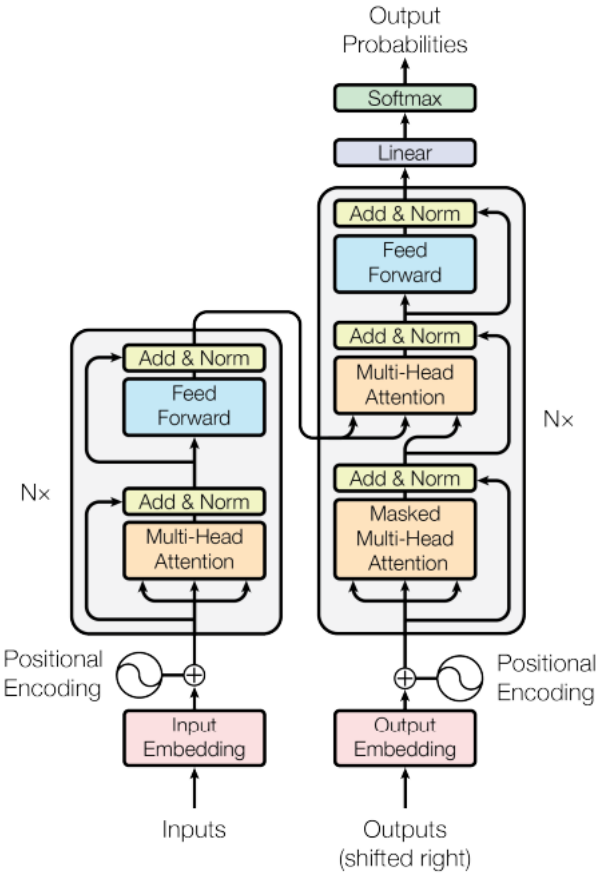

Figure 3.3: The Transformer model architecture by Vaswani et al. [61]. The encoder is illustrated on the left half of the picture, and the decoder is illustrated on the right.

keys and values come from the encoder's output, and the query comes from the previous decoder layer. This allows the decoder to focus on relevant parts of the encoded input sequence while generating the output. Similarly to the encoder, the position-wise FFN processes each position independently.

3.3 Vision Transformer

While the Transformer model has established itself as the state-of-the-art in NLP, convolutional architectures have remained dominant in computer vision. Before the introduction of the Transformer model, attention mechanisms in computer vision were typically integrated into CNNs or used to replace specific components without altering the overall CNN architecture. Building on the success of the Transformer, Dosovitskiy et al. [15] introduced the Vision Transformer (ViT), which adapts the Transformer architecture for computer vision tasks.

ViT divides images into patches, models their global relationships using Transformer layers, and applies the resulting representations to the corresponding tasks.

Unlike the original Transformer, which uses both encoder and decoder stacks for sequence-to-sequence tasks, the ViT architecture only uses the encoder, as the ViT initially experimented with image classification tasks. However, segmentation tasks require a decoder to convert the feature representation back to the spatial domain, which is explored in more detail in Chapter 4.

Transformers lack the inductive biases of CNNs, such as translation equivariance and locality. Inductive bias refers to the assumptions a learning algorithm makes to generalize from training data to unseen data. As a result, Transformers are less effective with limited training data [15]. However, when training with large datasets, the training surpasses the inductive bias. For this reason, ViT is typically pre-trained on large datasets before being fine-tuned for smaller, target-specific tasks [50]. This type of strategy has achieved excellent results, as ViT scales well with increased model size and dataset size, with larger models showing significant improvements when trained on extensive datasets. ViT outperforms CNN-based baselines in accuracy across most benchmarks, while requiring less computational resources.

Directly applying SA to images would involve every pixel attending to all others, resulting in quadratic complexity that does not scale for realistic input sizes [15]. To address this, various approximations have been proposed for adapting Transformers to image processing. These include limiting SA to local pixel neighborhoods, replacing convolutions with localized MHA blocks, using sparse approximations to scaled global SA, and structuring attention in blocks or along single axes. While these specialized attention architectures have achieved strong results, they are often complex and challenging to implement efficiently.

In contrast, ViT simplifies image processing by representing an image as a sequence of patches [15]. These patches are extracted and converted into linear embeddings, which are then processed by the standard Transformer encoder. ViT treats patches similarly to word tokens in NLP tasks, with no additional image-specific biases apart from patch extraction. This process is illustrated in Figure 3.4.

To adapt a standard Transformer for image processing, fixed-size image patches are flattened into 1-dimensional patch embeddings, which serve as an input to the Transformer. For an image with resolution $H \times W$ and C color channels, the image is divided into patches of size $P \times P$, resulting in $N = HW/P^2$ patches. Each patch is flattened and passed through a trainable linear projection to produce a D-dimensional patch embedding, where D is the consistent latent vector size maintained across all Transformer layers. To encode positional information and account for patch arrangement, positional embeddings are added to the patch embeddings.

The original ViT architecture uses one-dimensional positional embedding, which considers the inputs as a sequence of patches in raster order. The positional encoding

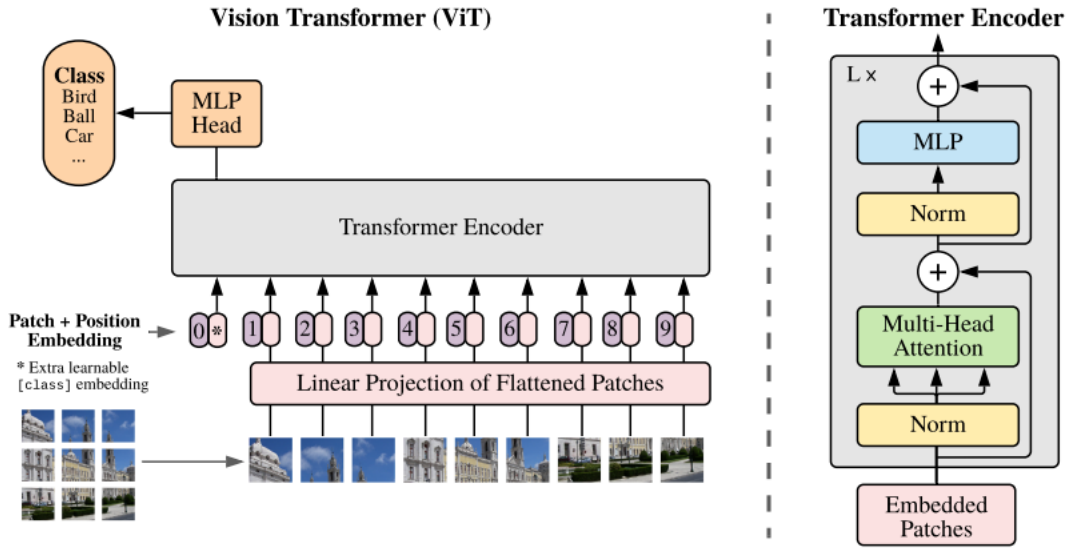

Figure 3.4: Architecture of the Vision Transformer (ViT) by Dosovitskiy et al. [15]. The left side of image shows the process of the ViT. The ViT splits an image into fixed-size patches, embeds them linearly, adds positional embeddings along with a class token used for classification, and processes the sequence with a Transformer encoder. The right side of the figure shows the encoder of the ViT. The encoder processes input image patches as a sequence of tokens. The encoder consists of alternating layers of Multi-Headed Self-Attention and Multi-Layer Perceptrons (MLPs), with layer normalization (LN) and residual connections.

may also be two-dimensional, have relative positional embedding or no positional information at all. In the case of two-dimensional positional embeddings, the input is treated as a grid of patches. Two separate embeddings are learned for each axis (X and Y), each with a size of $D/2$. The positional embedding for each patch is then constructed by concatenating its corresponding X and Y embeddings based on the patch's coordinates. Relative positional embeddings, on the other hand, encode the spatial relationships between patches by focusing on their relative distances rather than their absolute positions. When no positional information is used, the model processes the input as an unordered collection of patches.

The encoder consists of alternating layers of Multi-Head Self-Attention (MSA) and Multi-Layer Perceptrons (MLPs), with layer normalization and residual connections. MSA captures global relationships between the patches. The MLP blocks contain two layers with a Gaussian Error Linear Unit (GELU) non-linearity as activation for capturing complex relationships. Layer normalization ensures stability, while residual connections improve learning.

In image classification tasks, an additional learnable embedding, called the "class token", is added to the beginning of the sequence of each patch embedding to aggregate information for the final classification task. Initially, the class token is a randomly

initialized placeholder with no inherent meaning. As the Transformer processes the sequence, the class token gathers information from the other patches using SA. After the encoding process, the class token's output is processed by a small MLP called the "MLP Head", which contains a single hidden layer using the hyperbolic tangent function (tanh) for non-linearity to produce the class prediction.

3.4 Extensions

ViT currently serves as a backbone for various computer vision tasks [20]. However, ViT poses inherent challenges, such requirements for large datasets and fixed patch sizes. To address these challenges, researchers have proposed modifications to the ViT's architecture and mechanisms. Notably the Data Efficient Transformer (DeiT) by Touvron et al. [58], the Swin Transformer by Liu et al. [33], and the Transformer in Transformer (TNT) by Han et al. [21] have built upon the original model.

DeiT

ViT typically requires large datasets and significant computational resources for effective training. The Data Efficient Transformer (DeiT) by Touvron et al. [58] addresses this challenge by demonstrating that Transformers can be trained to be robust on medium-sized datasets through distillation. DeiT utilizes a teacher-student strategy specific to Transformers, which relies on a distillation token. The distillation token complements the traditional class token in the student model (ViT) to reproduce the label that is estimated by the teacher model (typically a pre-trained CNN). The student learns from the teacher through the attention mechanism and leverages soft labels (teacher's probabilistic predictions) to allow the model to learn from the hard labels (teacher's class predictions) using the distillation token. This approach improves generalization and training efficiency, making ViT's more suitable to use with smaller datasets.

Figure 3.5 illustrates the distillation process. At the bottom of the diagram, three types of tokens are shown: class token, patch tokens, and the distillation token. The class token, depicted as a circle on the left, enables the model to make predictions in a way similar to traditional classification methods. The patch tokens, represented by squares in the center, correspond to parts of the input image. Finally, the distillation token, shown on the right, transfers knowledge from the teacher model during training. As these tokens pass through layers of self-attention and an FFN, they interact and exchange information, which enables the Transformer to understand both global and local features within the image. At the top of the image, two loss functions guide the

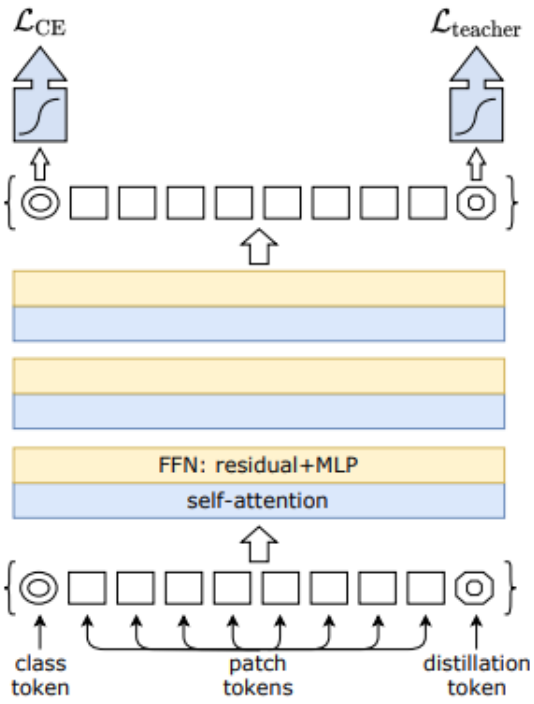

Figure 3.5: The distillation procedure [58]. A distillation token is added to engage with class and patch tokens through self-attention. Unlike the class token, the distillation token aims to reproduce the teacher's predicted label (rather than the true label) at the network's output. Both class and distillation tokens are learned using back-propagation.

learning process. The cross-entropy loss (L_{CE}) uses the class token to compare the model's predictions to the true labels so that the model learns to classify images correctly. The distillation loss ($L_{teacher}$) uses the distillation token to compare the model's output to the predictions of the teacher model. This setup allows the Transformer to learn from both the true labels and the teacher's predictions, while also receiving inductive biases of the teacher model.

Swin Transformer

Adapting Transformers from language to vision is a challenging task due to differences between domains, including varying scales of visual entities and higher pixel resolution in images compared to words in text. The Swin Transformer by Liu et al. [33] addresses these challenges by using a sliding windowing scheme. This approach improves efficiency by restricting self-attention to local, non-overlapping windows while allowing cross-window connections. The hierarchical design supports modeling at multiple scales and maintains linear computational complexity relative to image size. This concept is illustrated in Figure 3.6.

The Swin Transformer's main design feature is shifting the window partition be-

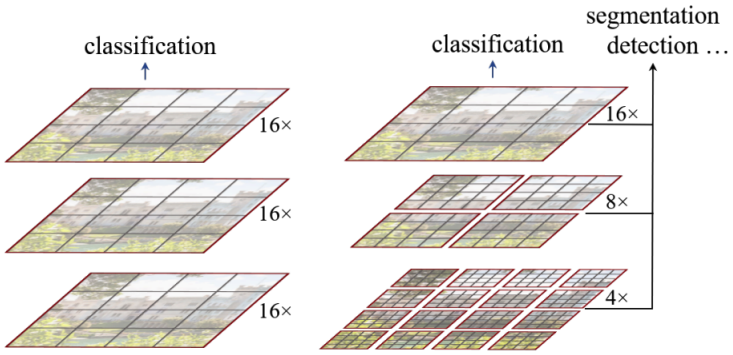

Figure 3.6: Compared to the original ViT (left), Swin Transformer (right) builds a hierarchical representation by merging small patches progressively in deeper layers [33].

tween consecutive self-attention layers, as demonstrated in Figure 3.7. This shift connects windows across layers to improve the modeling power and has shown excellent results in image classification, object detection, and semantic segmentation, outperforming ViT by a significant margin [33]. The Swin Transformer is suitable for various downstream tasks in which extracted features can be used for further processing [22].

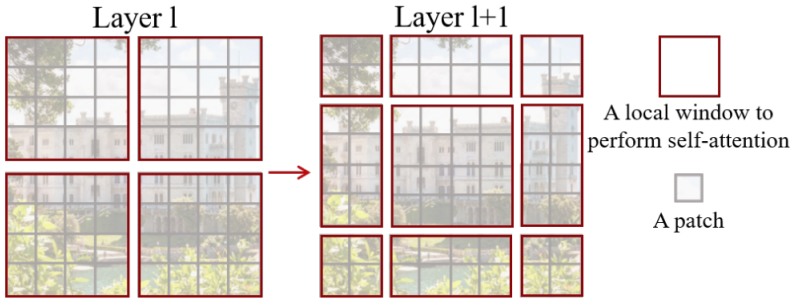

Figure 3.7: In the Swin Transformer, self-attention is computed within regular windows at layer l, and by shifting the windows in layer $l+1$, connections across previous window boundaries are created for a broader self-attention [33].

TNT

Natural images are highly complex with rich details and colors, making coarse patch division insufficient at capturing features at varying scales and locations. The Transformer in Transformer (TNT) by Han et al. [21] describes patches as "visual sentences" and divides them into smaller sub-patches, described as "visual words". An inner Transformer block is used to model the relationship between the sub-patches and an outer Transformer block is responsible for patch-level information exchange. In other words, the inner Transformer computes relationships within visual words of a single visual sentence to capture local details, and the outer Transformer processes relationships

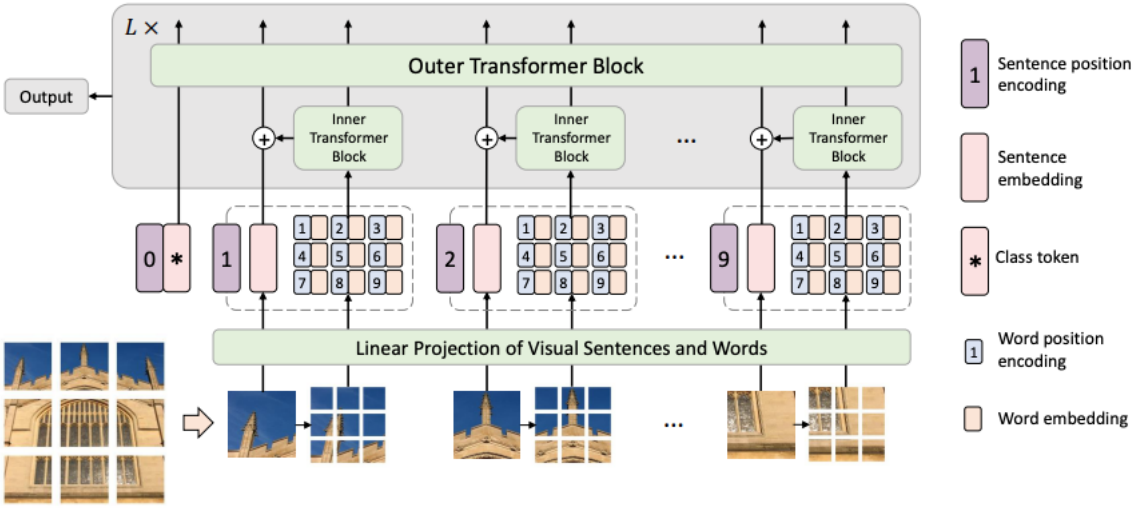

Figure 3.8: The TNT architecture by [21].

across visual sentences to capture the global context. Unlike the original ViT, which disrupts the local structure of an image with fixed-size patches, the TNT architecture preserves and models local and global information more effectively.

The TNT architecture builds on the basic configuration of ViT and DeiT, as illustated in Figure 3.8. The input image is split into larger patches (visual sentences) and each patch is further split into smaller sub-patches (visual words). Positional encoding is applied at two levels: sentence position encoding is added for the entire patches and word position encodings are shared across visual sentences.

The architecture includes two types of Transformer blocks. The inner Transformer blocks are shared in the same layer and operate on the visual words within a visual sentence. These blocks capture local relationships and detailed features by modeling the relationship between visual words within the same sentence. The outer Transformer block processes the visual sentences and models relationships between them by augmenting the sentence embedding with the sequence of word embeddings. The output values of each attention head are concatenated and linearly projected to produce the final output.

The TNT has shown superior performance on several benchmarks, achieving higher results than comparable models, such as ViT and DeiT. The hierarchical approach of TNT has shown an improvement for efficiency by offering a better accuracy-to-computation trade-offs.

4. Transformers in Brain Image Segmentation

Segmentation is a key focus in medical image analysis, as evidenced by the large volume of research and the rich research landscape of studies experimenting with various model architectures [51]. Recent Transformer-based approaches, including DeiT and TNT, show promising results in generic vision tasks, but adapting them to domain-specific medical imaging remains challenging [50]. Nonetheless, models that have utilized Transformer in their components have demonstrated success in medical image segmentation with competitive performance and great potential compared to CNNs [20].

Transformer-based segmentation models have been widely applied to segmentation tasks in different regions of the human body, including the brain, lungs, heart, and abdominal organs [67]. In these tasks, the Transformer encoder divides the input image into patches and processes them to generate high-level feature embeddings. To achieve pixel-level (2D) or voxel-level (3D) predictions, these abstract features are transformed back into the input domain using custom segmentation decoders, which map the encoded features to spatially detailed outputs that align with the input resolution [56].

Recent research has increasingly focused on hybrid approaches that integrate CNNs for modeling local features and Transformers for capturing global dependencies [62]. In particular, U-Net-based Transformer architectures, which incorporate Transformers in the encoder or between the encoder and the decoder, have shown to be well suited for segmentation tasks [67]. However, segmentation requirements often require modifications to these architectures. Among brain segmentation methods, brain tumor segmentation has received the most attention for utilizing Transformers, while research on brain tissue segmentation and whole-brain segmentation using Transformers is still limited.

This chapter provides an overview of recent state-of-the-art architectures for brain image segmentation and highlights their key characteristics that contribute to their success. Architectures commonly used in medical image segmentation are introduced in Section 4.1, emphasizing their general applicability for medical image segmentation.

Section 4.2 focuses on state-of-the-art methods for brain tumor segmentation, followed by an overview of architectures for brain tissue segmentation in Section 4.3. Although these architectures are primarily designed and trained for specific tasks, such as tumor or tissue segmentation, they have also been applied to other segmentation tasks and evaluated against alternative architectures with various datasets. The chapter ends with discussing future directions in brain image segmentation.

4.1 General Architectures

This section introduces U-Net, a type of CNN designed specifically for image segmentation tasks, along with notable extensions and state-of-the-art adaptations of this architecture. These architectures have been evaluated on various medical image datasets, including those for brain image segmentation.

4.1.1 U-Net

The U-Net, introduced by Ronneberger et al. [49] is a CNN architecture that has significantly advanced image segmentation performance, particularly in medical applications due to its performance with limited number of training images compared to traditional CNNs. U-Net has been adopted across various imaging modalities, including CT scans, MRI, X-rays, and microscopy [51]. While originally developed for biomedical image segmentation, it has also been successfully applied to other tasks, such as image reconstruction and object detection.

The architecture of the U-Net follows the encoder-decoder structure, as illustrated in Figure 4.1. The encoder (contracting path) learns global contextual representation using downsampling and the decoder (expansive path) upsamples this representation back to the input resolution with pixel-wise semantic prediction [23]. The architecture also utilizes skip connections to retain and reuse information from the encoder, such as lost spatial information that is caused by downsampling. The network is trained in a supervised manner using input images and corresponding segmentation masks [49].

The encoder is a typical CNN that downsamples the image into a compressed representation, capturing the most essential features of the image. The image is downsampled by repeatedly applying two unpadded convolutions, each followed by ReLU and max pooling. At each downsampling step, the number of features is doubled to preserve the representational capacity of the network.

The decoder upsamples the compressed representation back to the original input dimension. Every upsampling step is followed by an up-convolution that halves the feature channels. During each upsampling step, a skip connection is utilized, which

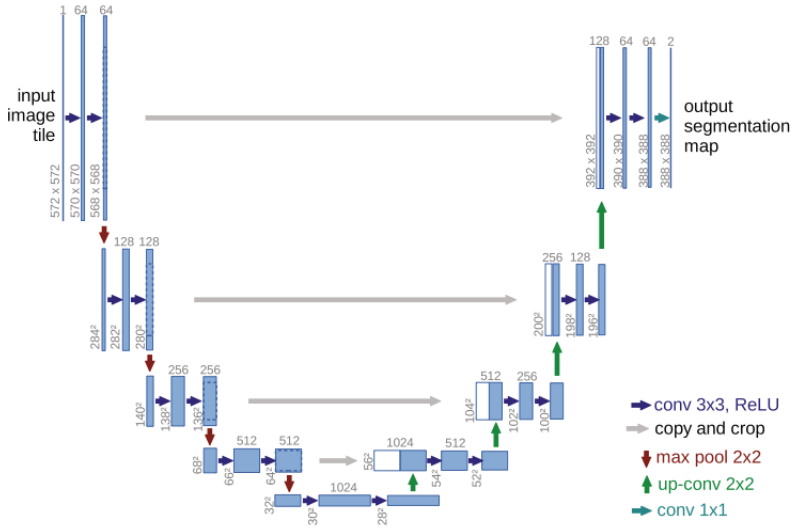

Figure 4.1: The U-Net architecture [49]. Blue boxes represent multi-channel feature maps. Arrows illustrate operations, such as convolutions, pooling, and upsampling. The number of channels is displayed above each box, and the spatial dimensions are noted at the lower left. White boxes indicate feature maps from the encoder (contracting path), which are concatenated to the decoder (expansive path) using skip connections (grey arrows).

concatenates a cropped feature map from the encoder. After the concatenation, two convolutions are applied, each followed by ReLU activation. A final convolution maps features to target classes.

While the U-Net architecture performs well with limited datasets, training the U-Net effectively requires data augmentation especially in medical image segmentation, where annotated datasets are often limited [49]. Data augmentation techniques apply shifts, rotations, and intensity variations to improve the model's invariance and robustness even with limited datasets. However, significant deformations can cause substantial damage and introduce noise, leading to unrealistic and unnatural images [17].

The U-Net and architectures based on it have achieved state-of-the-art performance benchmarks on different 2D and 3D semantic segmentation tasks across various imaging modalities [22]. The structural design and success of U-Net has inspired the development of modified architectures, including 3D U-Net [9], UNETR [23] and TransAttUNet [7].

4.1.2 3D U-Net

Volumetric data is widely used in biomedical analysis, but labeling it raises significant challenges due to the limitations of displaying only 2D slices on screens [9]. Annotating slice by slice is both time-consuming and inefficient, as adjacent slices are nearly iden-

tical. For machine learning, which requires extensive labeled data, fully annotating 3D volumes is not a practical approach for building large, diverse datasets necessary for effective generalization.

To address the challenges of annotating volumetric data, the 3D U-Net was introduced by Çiçek et al. [9] as an extension of the original U-Net. The 3D U-Net modifies the original architecture by replacing all 2D operations with 3D operations, enabling detailed volumetric segmentation while requiring only a limited number of annotated 2D slices for training. 3D volumes are processed with corresponding 3D operations, in particular, 3D convolutions, 3D max pooling, and 3D up-convolutional layers. The network is able to segment images using minimal annotated examples, as 3D images often contain repeating structures and shapes. This characteristic also enables faster training process, even with a limited amount of labeled data [51].

4.1.3 UNETR

UNEt Transformers (UNETR), proposed by Hatamizadeh et al. [23], is designed for 3D medical image segmentation and marks as the first architecture to use the Transformer as its encoder without relying on a CNN-based feature extractor. UNETR formulates volumetric segmentation as a sequence-to-sequence prediction problem by directly leveraging the Transformer's ability to model long-range dependencies, which CNNs struggle with. The architecture has been validated on both CT and MRI datasets, including datasets for brain tumor and spleen segmentation, and achieves superior accuracy and efficiency across diverse tasks [22], outperforming previous state-of-the-art methods.

Similar to the U-Net, UNETR follows a U-shaped structure that consists of an encoder (contracting path) and a decoder (expanding path), as illustrated in Figure 4.2. Transformer encoder is used for learning sequence representations of embedded input patches, reframing 3D segmentation as a 1D sequence-to-sequence prediction problem while capturing global, multi-scale contextual information [23]. The representations learned by the Transformer are combined with a CNN-based decoder through skip connections across multiple resolutions to generate segmentation outputs. A CNN-based decoder is preferred over Transformers for this purpose, as Transformers are highly effective at capturing global information but are less suited for processing localized information.

The encoder consists of a stack of Transformers that are connected to the decoder via skip connections. The 3D input volume is first flattened to a 1D input that contains uniform non-overlapping patches. The linear layer then projects these patches into a K dimensional embedding space (constant feature size) with 1D learnable positional

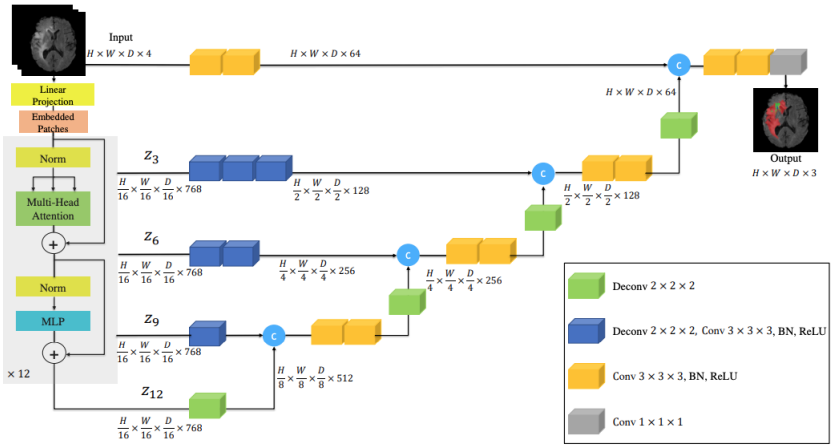

Figure 4.2: UNETR architecture [23]. A 3D input volume with dimensions $H \times W \times D$, such as a multi-channel MRI image with $C = 4$ channels, is split into uniform, non-overlapping patches and transformed into an embedding space through a linear layer. Position embeddings are added to the sequence, which is then used as an input to a Transformer model. The architecture uses 12 Transformer layers, and encoded representations from various layers (Z_3, Z_6, Z_9, Z_{12}) are extracted and concatenated (blue circles) with the representations in the decoder using skip connections.

embedding that is used for preserving the spatial information between the patches. After the embedding, the Transformer layers, consisting of MSA and MLP sublayers, are used for feature extraction.

The decoder applies a deconvolutional layer to increase the resolution by a factor of 2. This resized feature representation is then concatenated using skip connections from the Transformer's output in the encoder. Consecutive convolutional layers are applied and the output is upsampled using a deconvolutional layer. These steps are repeated for all the subsequent decoding steps and a final output is generated using a $1\times1\times1$ convolutional layer with softmax activation to generate voxel-wise class predictions.

4.1.4 TransAttUnet

One of the most recent state-of-the-art architectures is a Transformer-based attention guided network called TransAttUnet by Chen et al. [7], which uses multi-level guided attention and multi-scale skip connection to improve the segmentation performance of a 2D image. Multi-level guided attention models concurrently global spatial relationships among encoder semantic features. Multi-scale skip connections aggregate residual or dense contextual feature maps from decoder blocks of varying semantic scales to generate more discriminative feature representations. TransAttUnet has shown significant improvement of the medical image segmentation quality and outperforms existing U-Net-based state-of-the-art methods.

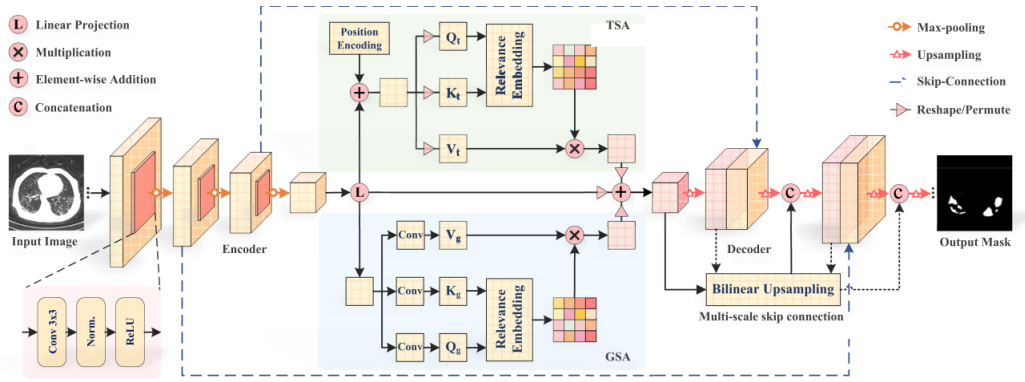

Figure 4.3: The architecture of TransAttUnet [7]. The architecture builds on a standard U-Net backbone by integrating a Self-Aware Attention (SAA) module between the encoder and the decoder. This module combines Transformer Self-Attention (TSA) (green) and Global Spatial Attention (GSA) (blue) mechanisms to capture long-range contextual relationships and enhance the representation of global features. Here, Q_t, K_t, and V_t represent the query, key, and value tensors for TSA, while Q_g, K_g, and V_g serve similar roles in GSA. Multi-scale skip connections merge semantic features from varying scales using bilinear upsampling (smoothing intermediate layers) and concatenation.

The architecture of TransAttUnet shows similarity to the U-Net architecture, as illustrated in Figure 4.3. Multi-level guided attention is implemented by using a Self-Aware Attention (SAA) module, that is placed between the encoder and decoder. The SAA module consists of two independent self-attention mechanisms: Transformer Self-Attention (TSA) and Global Spatial Attention (GSA). These mechanisms capture broader and more detailed contextual representations compared to the standard U-Net.

The TSA module uses MSA to extract semantic information from global representation subspaces to model global context of the input. Within this module, each attention head operates independently, and the outputs of these heads are combined through a subsequent embedding layer to form a unified representation. The TSA uses learned positional encodings, that are shared across all attention layers to capture absolute and relative position of information. By using query, key, and value embeddings, TSA computes attention maps that reflect relationships across different positions in the input feature map.

The GSA component extends the feature representations from the SAA module with global context, which improves intra-class compactness (similarity within the same class) and optimizes feature representations. The GSA encodes broader contextual positional information into local features by first transforming feature maps from the encoder into two subspaces using convolutional layers. These subspaces are then processed to generate attention maps that encode positional dependencies among features.

The final feature representation is derived by combining the output of the GSA

module and the output of the TSA module with the encoder's original feature map. In the decoder's sub-network, multi-scale skip connection aggregates contextual information progressively by performing different operations, including upsampling, concatenation, and convolution.

The TransAttUnet focuses on two different multi-scale skip connection schemes: residual connections and dense connections. Residual connections aggregate features by progressively feeding the output of each decoder block into the next stage, while dense connections incorporate all previous decoder block features as input to the current block. While both connection schemes show improvements over traditional U-Net architectures and other advanced Transformer-based methods, the residual connections perform the best due to their ability to maintain accuracy while reducing redundant information.

The TransAttUnet outperforms previous methods and demonstrates strong generalizability in medical image segmentation across five different medical image datasets. These datasets include skin lesion segmentation on dermatoscopic images, lung segmentation on chest X-ray images, COVID-19 pneumonia lesion segmentation on chest CT images, nuclei segmentation on divergent images, and gland segmentation on histology images. While the original experiments did not contain brain image segmentation, other studies, such as [46], have shown the TransAttUnet to outperform previous state-of-the-art methods in brain tumor segmentation, demonstrating its potential for accurate and robust performance across anatomical regions and imaging modalities.

4.2 Brain Tumor Segmentation

Brain tumor segmentation is a challenging problem in medical imaging and computer-aided diagnosis, with relatively extensive research dedicated to it [34]. As manual segmentation of brain tumors is a tedious task, computer-based methods are increasingly essential in healthcare for tumor diagnosis and surgical planning [22]. Detailed tumor characterization, such as volumetric and texture analysis, helps monitor tumor progression, plan surgeries, and predict life expectancy. The complexity of brain tumor segmentation is caused by multiple factors, such as uncertain tumor locations and shapes, poor imaging contrast, inconsistent annotation, and imbalanced data.

Brain tumors are categorized into two types: primary and secondary [22]. Primary tumors are formed from brain cells, while secondary tumors spread to the brain from other organs. Gliomas, the most common primary brain tumors originating from glial cells, are classified into low-grade (LGG) and high-grade gliomas (HGG) based on their varying levels of malignancy and aggressiveness [64]. High-grade gliomas are aggressive, grow quickly, often need surgery and radiotherapy, and are associated with

poor survival outcomes. LGGs, on the other hand, are slow-growing.

Multiple 3D MRI modalities (including T1, T1c, T2, and FLAIR) are needed to highlight different tissue characteristics and regions of the tumor spread [22]. The regions of a brain tumor are typically classified into three sub-regions: Enhancing Tumor (ET), Tumor Core (TC), and Whole Tumor (WT) [62]. ET represents areas of active tumor growth. TC contains the enhancing tumor, necrosis, and non-enhancing tumor. WT includes all regions of a tumor: peritumoral edema, enhancing tumor, non-enhancing tumor, and necrosis. Figure 4.4 illustrates an example of brain tumor segmentation highlighting these sub-regions.

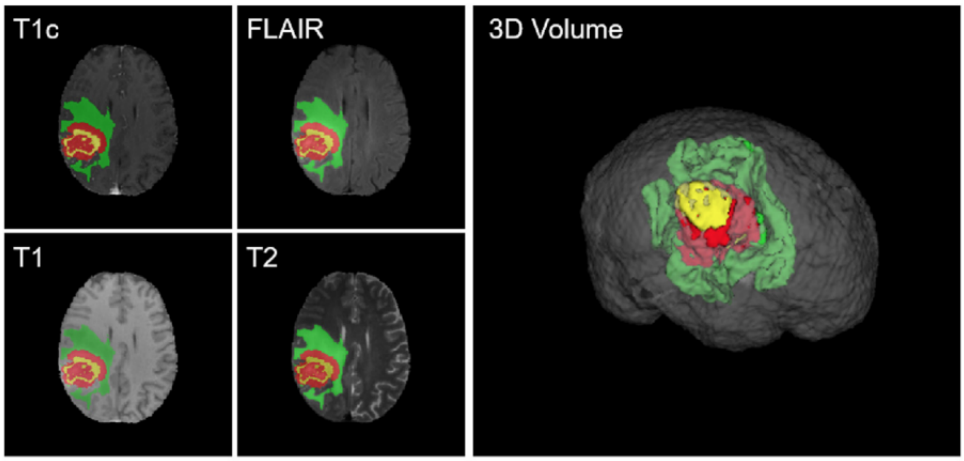

Figure 4.4: An example of volumetric brain tumor segmentation from MRI with different modalities (T1c, FLAIR, T1, T2), illustrating the three primary sub-regions of a brain tumor: ET (yellow), TC (red), and WT (green) [62].

The International Brain Tumor Segmentation challenges (BraTS), begun in 2012 and held annually, provide benchmark MRI datasets that contain images of LGG and HGG [11, 41]. The BraTS challenge evaluates advanced techniques for semantically segmentating brain tumors using a 3D MRI dataset, which includes voxel-level labels annotated by medical experts [22]. This challenge encourages researchers to develop and refine state-of-the-art algorithms for automated brain tumor segmentation.

The Dice Similarity Coefficient (Dice Score) and Hausdorff distance are the primary metrics used to evaluate brain tumor segmentation performance for ET, TC, and WT [28, 41]. The Dice Score is defined as

$$\text{Dice} = \frac{2\,|A \cap B|}{|A| + |B|}$$

where A represents the set of foreground pixels/voxels in the annotated image (ground truth) and B represents the set of foreground pixels/voxels in the segmentation result. The Hausdorff Distance measures the maximum distance between two sets. Figure

4.5 illustrates X and Y to be two different sets, where the Hausdorff distance d_H is defined as $d_H(X, Y) = \max(d_{XY}, d_{YX})$. However, the standard Hausdorff distance is highly sensitive to outliers and noise in the data. To address this issue, the 95 percentile Hausdorff Distance (HD95) is often used, where instead of considering the single worst-case point (maximum distance), it measures the 95 percentile of all distances, providing a more practical evaluation metric.

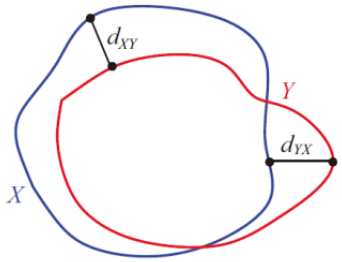

Figure 4.5: Hausdorff distance estimation curve [41]. The Hausdorff distance d_H is defined as $d_H(X, Y) = \max(d_{XY}, d_{YX})$, where X and Y are two different sets.

This section introduces state-of-the-art methods that have demonstrated excellent results in brain tumor segmentation. TransBTS is the first architecture to combine the Transformer and a 3D CNN for MRI brain tumor segmentation. BiTr-UNet gets its architectural inspiration from TransBTS and addresses some limitations that TransBTS and UNETR pose. Swin-UNETR extends the UNETR architecture with the shifting windowing mechanism and is specifically tailored for brain tumor segmentation. Finally, VT-UNet uses a Volumetric Transformer (VT) and improves on the segmentation results over TransBTS and UNETR.

4.2.1 TransBTS

Wang et al. [64] introduced TransBTS, the first architecture to integrate Transformers with 3D CNN for MRI brain tumor segmentation. By processing all image slices at once using 3D CNNs, TransBTS captures information from slices continuously (depth dimension) and is able to capture long-range dependencies across spatial dimensions. TransBTS achieves significant improvements over previous 3D segmentation models on brain tumor segmentation, such as the 3D U-Net.

TransBTS extends the encoder-decoder design and shares characteristics with the U-Net architecture, as illustrated in Figure 4.6. To encode compact volumetric feature maps that capture the local spatial context information, a 3D CNN extracts volumetric spatial features and downsamples the input data into a feature representation, which serves as input for the Transformer [64].

The distinguish feature of TransBTS are the four layers that connect the encoder

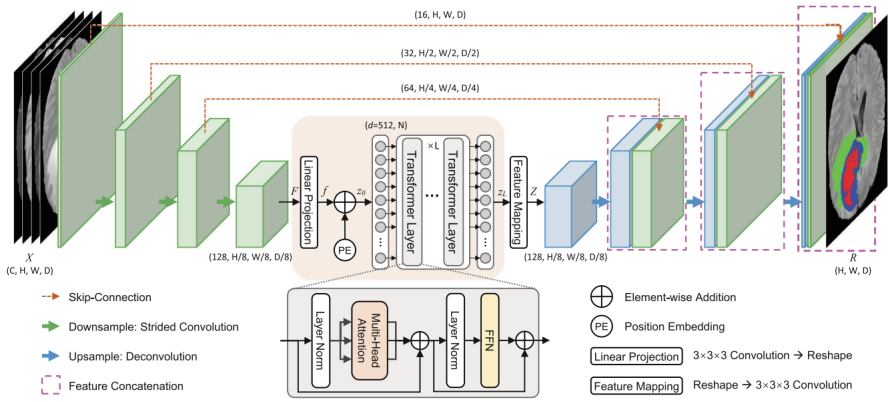

Figure 4.6: Architecture of TransBTS for brain tumor segmentation from 3D MRI scans [64]. The input MRI scan X has dimensions $C \times H \times W \times D$, where C is the number of channels (modalities), H and W represent spatial resolution, and D denotes the depth dimension (number of slices). The 3D CNN encoder extracts compact spatial and depth feature maps, while the Transformer encoder captures global dependencies. The decoder progressively upsamples these features to generate the segmentation output R, restoring the original resolution. Skip-connections improve recovering the spatial detail.

and the decoder: linear projection layer, patch embedding layer, Transformer layers and feature mapping layer [28]. The linear projection increases the channel dimension and the patch embedding layer collapses the spatial and depth dimensions into one dimension (as the Transformer expects a input sequence). The Transformer layers extract long-range dependencies and the feature mapping layer fits the sequence data back to the 3D CNN decoder. The output of the Transformer is reconstructed back to a full-resolution 3D segmentation map by progressive upsampling. Skip connections between encoder and decoder layers are used to merge encoder features with decoder features through concatenation.

Unlike most ViT-based segmentation approaches, TransBTS is trained from scratch on task-specific datasets without relying on pre-trained weights [50]. Compared to previous U-Net-based models, TransBTS demonstrates significant improvements in HD95 and achieves comparable results on the Dice Score. Test Time Augmentation, which applies random modifications to test images, has shown to further improve its performance.

4.2.2 BiTr-UNet

Inspired from the architecture of TransBTS, Jia and Shu [28] introduced Bi-Transformer UNet (BiTr-UNet) for 3D MRI Brain Tumor Segmentation. There are two distinguishing features from the architecture compared to TransBTS: two sets of Transformer layer connect the encoder and the decoder instead of one (which the prefix

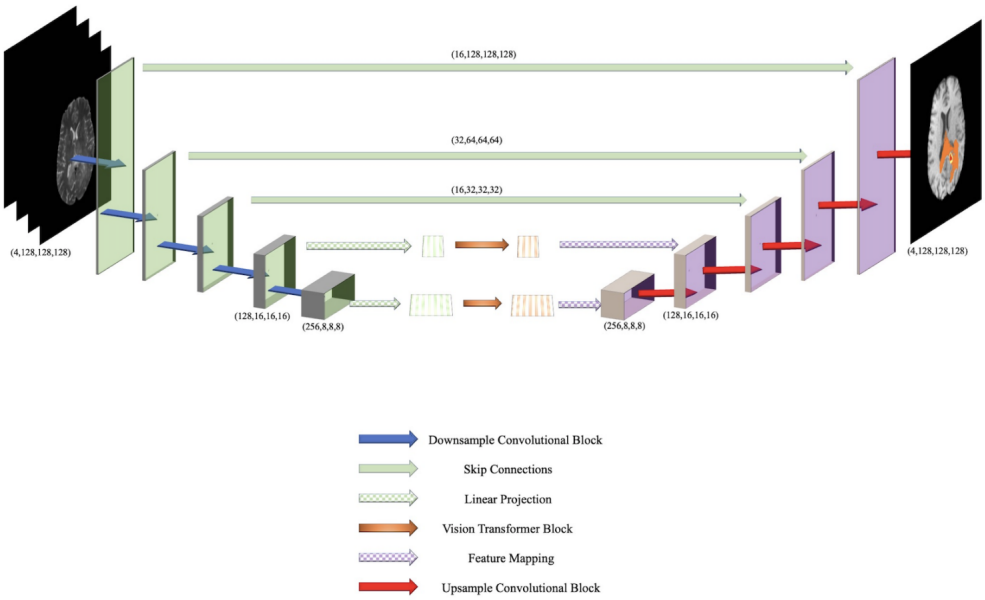

Figure 4.7: Architecture of BiTr-UNet [28], illustrating a CNN-based encoder, two ViT blocks for extracting long-range dependencies, and a 3D CNN decoder.

"Bi" reflects), and an attention module is added in the CNN encoder.

The architecture of BiTr-UNet is illustrated in Figure 4.7. The BiTr-Unet encoder uses 3D CNN layers for downsampling. An initial convolutional block increases the feature map dimensions, after which convolutional blocks extract feature representations and reduce the output size. To improve local context extraction, this output goes through a 3D Convolutional Block Attention Module (3D CBAM) [65], which sequentially infers a channel attention map and a 3D spatial attention map. Compared to TransBTS, the 3D CBAM blocks replace the 3D CNN blocks in the encoder.

The outputs from the initial three layers of the encoder are directly passed to the final three layers of the decoder using skip connections. The outputs from the fourth and fifth downsampling layers are processed through feature embedding of feature representation layer, Transformer layers, and a feature mapping layer before being passed to the corresponding upsampling layers. Four sequential convolutional blocks followed by a final convolutional block reduce the feature map dimensions in the decoder to produce the segmentation image.

The BiTr-UNet is specifically designed for the BraTS dataset, and uses postprocessing to achieve a higher Dice Score [28]. The postprocessing strategy used eliminates a volume of predicted segmentation if the volume is smaller than a threshold. To further improve accuracy, majority voting is used to combine predictions from multiple models to improve accuracy. Each voxel is assigned the category that receives the most votes as its final prediction.

The validation results of BraTS suggest that BiTr-Unet effectively extracts local

and long-range dependencies from 3D MRI scans. Compared to U-Net-based models without attention modules which already perform well in BraTS, BiTr-Unet benefits from adding a Transformer module. However, a dissimilarity between testing and training data highlights potential challenges for BiTr-Unet in handling unseen patterns.

4.2.3 Swin-UNETR

Hatamizadeh et al. [22] proposed Swin UNEt Transformers (Swin-UNETR), which combines a Swin Transformer encoder with a CNN-based decoder [50]. Unlike the UNETR model, Swin-UNETR is specifically designed for brain tumor segmentation. Feature representations are extracted at multiple resolutions with shifted windows, which are then used for computing self-attention. Compared to methods with fixed resolutions, the hierarchical Swin Transformer has shown to better manage long-range dependencies. As with other U-Net based architectures, the Swin-UNETR uses skip connections, where the features obtained by the encoder are passed to the decoder at each resolution.

In the encoder, the patch partition layer transforms multi-modal input data into a one-dimensional sequence of embeddings, which are then processed by a hierarchical Swin Transformer that serves as the encoder. Self-attention is computed from non-overlapping windows generated by the patch partition layer. Figure 4.8 illustrates the shifted windowing mechanisms for subsequent layers in a 3D image.

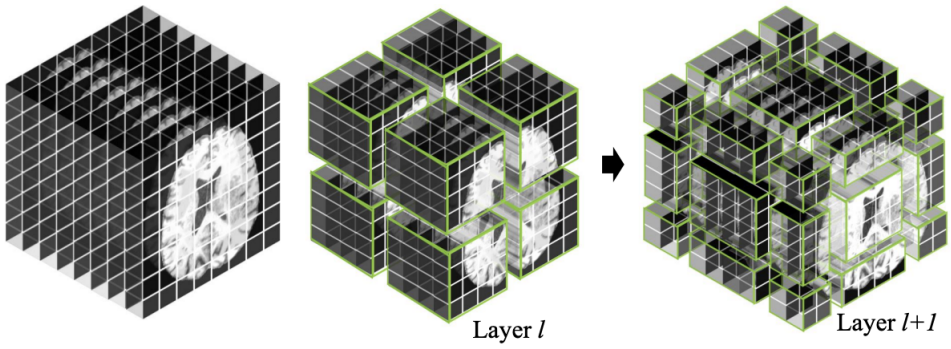

Figure 4.8: Overview of the shifted windowing mechanism of Swin UNETR, where 8×8×8 3D tokens and 4×4×4 window size are illustrated [22]. The green cubes illustrate self-attention units.

The encoder features are passed to the decoder through skip connections at each resolution. At each stage in the encoder and bottleneck, feature outputs are reshaped and processed by a residual block with two convolutional layers normalized by instance normalization. Instance normalization normalizes each element of the batch independently (unlike batch normalization which normalizes all images across the batch). The

resolution is doubled using a deconvolutional layer, and the results are concatenated with the outputs from the previous stage. These concatenated features are processed in a similar residual block as described earlier. Final segmentation outputs are produced using a convolutional layer.

Swin-UNETR achieves state-of-the-art performance on several 3D segmentation benchmarks and outperforms TransBTS in segmenting brain tumors [22, 30]. This improvement is likely due to Swin-UNETR's architectural design, which integrates the Swin Transformer features into a CNN decoder via skip connections across multiple resolutions, similar to the UNETR-architecture shown in Figure 4.2. In contrast, TransBTS employs a less efficient design, confining the Transformer between the encoder and the decoder as a standalone attention module without skip connections.

Although the original Swin-UNETR was tested only on the BraTS dataset without pre-training, Tang et al. [55] pre-trained the Swin-UNETR on CT images from multiple datasets covering multiple anatomical regions. This approach improved performance by approximately 10% compared to training the model from scratch, resulting in a pre-trained model with robust feature representations for various medical imaging tasks. The pretrained Swin-UNETR model, when fine-tuned, achieves higher accuracy, converges faster, and reduces annotation effort compared to training from scratch.

4.2.4 VT-UNet

Peiris et al. [40] introduced VT-UNet, a volumetric segmentation model for 3D image modalities, such as CT and MRI. Unlike many earlier 3D segmentation models that process 3D inputs as 2D slices, VT-UNet processes volumetric data in its entirety, fully encoding interactions between slices. As the model is built purely based on Transformers, it is convolution-free while still being capable of keeping volumetric input data intact. Additionally, VT-UNet limits model parameters using patch merging and Fourier Positional Encoding (FPE), which reduce computational complexity while maintaining high segmentation accuracy and robustness.

Designing a Transformer-based U-Net for volumetric segmentation is challenging for three main reasons [40]. First, capturing relationships between voxels across arbitrary positions is complex, as data in each slice is connected to three orthogonal views (axial, sagittal, and coronal). Second, preserving spatial information in volumes is hard, as breaking images into patches risks losing local structural cues, requiring effective encoding of local and global interactions along multiple axes. Third, the quadratic complexity of self-attention and the large size of 3D volume tensors demand computationally efficient design.

VT-UNet addresses these challenges through its architecture, which incorporates

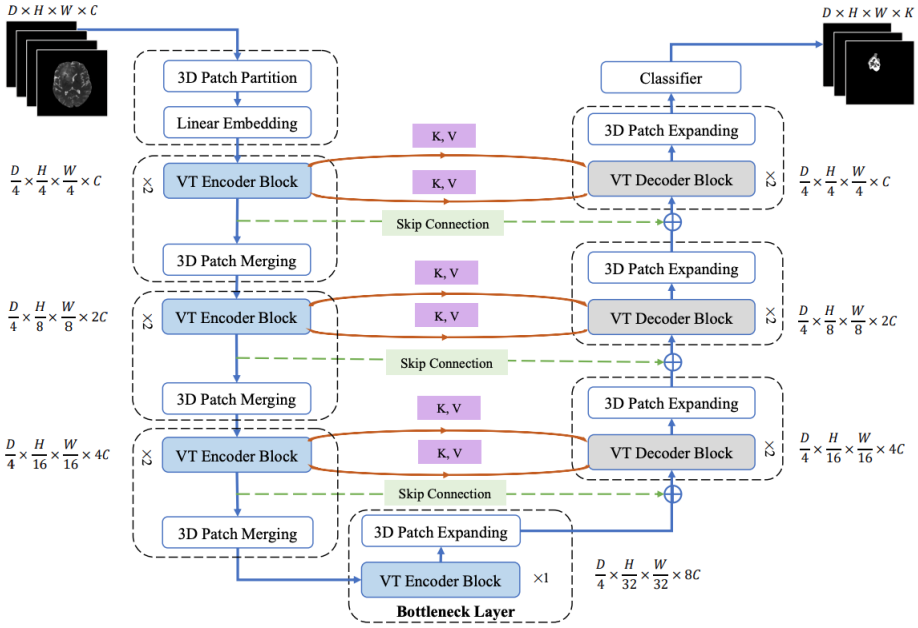

Figure 4.9: Illustration of the VT-UNet architecture [40]. The input 3D medical volume of size $D \times H \times W \times C$ is processed through VT Encoder and Decoder blocks, using keys (K) and values (V) for parallel self-attention and cross-attention. The output is a $D \times H \times W \times K$ volume, with K representing the number of segmentation classes.

two types of Transformer blocks: one in the encoder and another in the decoder. In the encoder, Transformer blocks hierarchically process 3D volumes to capture both local and global voxel relationships, similar to Swin Transformer blocks. The decoder uses parallel cross-attention and self-attention mechanisms, connecting queries from the decoder with keys and values from the encoder. This parallelization preserves global context and spatial information during decoding.

The absence of convolutions and attention outputs during decoding highlight the importance of sequence order. Relative positional encoding improves attention in each Transformer block, and complementary information from FPE is added during decoding. FPE uses sine and cosine functions with varying frequencies to generate unique encodings for each token by modulating high-frequency functions based on the token's location within the 3D volume.

The architecture of VT-UNet is illustrated in Figure 4.9. The encoder uses a 3D patch partitioning layer, liner embedding layer, 3D patch merging layer, and two successive VT encoder blocks. The patch partitioning layer creates a set of tokens by dividing the input into non-overlapping 3D patches. Linear embedding is used to map each token to a vector with fixed dimension. The 3D Patch merging blocks generate feature hierarchies, which are used to generate finer details in the output. The VT Encoder Blocks follow similar principle windowing scheme as the Swin Transformer,

but for volumetric data.

The bottleneck layer consists of a VT Encoder Block and a 3D Patch expanding layer. The 3D Patch Explanding layer "reverts" the patch merging effect, which involves creating new tokens in the decoder. In patch expanding, the dimension of an input token is increased by a factor of two using linear mapping. The resulting vector, with its dimensionality doubled, is reshaped into tokens.

The decoder uses successive VT Decoder Blocks with 3D Patch expanding layers and a classifier to produce the final voxel-wise predictions. The VT Decoder Blocks include self-attention and cross-attention. Each VT Decoder Block processes information from the preceding decoder stage while integrating spatial and contextual details from the VT Encoder Blocks through skip connections. After the final 3D patch expanding layer in the decoder, 3D convolutional classifier layer maps the features to classes.

VT-UNet achieves state-of-the-art results on the BraTS dataset. Specifically, it outperforms previous methods in Dice Score and HD95 across metrics for WT, TC, and ET. VT-UNet has a smaller model size and lower computational complexity compared to other state-of-the-art models, including TransBTS and UNETR, making it more practical for real-world clinical use. When subjected to synthetic artifacts such as motion, ghosting, and spike effects, VT-UNet consistently demonstrates higher robustness compared to other models.

4.3 Brain Tissue Segmentation

Segmentation of Gray Matter (GM), White Matter (WM), and Cerebrospinal Fluid (CSF) is essential for clinical and neuroscience studies, and supports the visualization of anatomical structures, aids surgical planning, and enhances image-guided interventions (medical procedures that use computer-based systems to provide virtual image overlays) [10, 66]. An example of a tissue segmentation from brain MRI is shown in Figure 4.10.

Manual segmentation of brain tissue is labor-intensive, difficult, and impractical for large datasets [48], while automated segmentation remains challenging due to inherent properties of MRI scans, as well as variability introduced by imaging devices, settings, and modalities. While CNNs have achieved significant progress solving this task, many solutions fail to generalize effectively to new datasets. Transformers, which have demonstrated success in image segmentation tasks, present an opportunity to enhance performance and generalization when integrated with CNNs for brain tissue segmentation. While the attention mechanisms have received some recognition in brain tissue segmentation, the Transformer architecture itself is less studied for this task.

This section introduces recent methods that have utilized the Transformer for

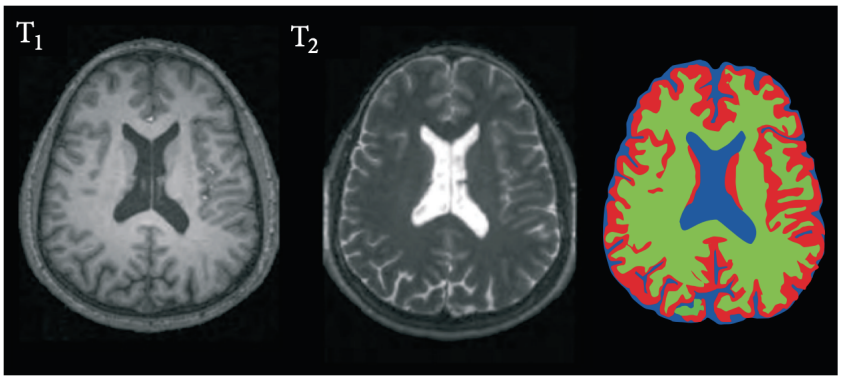

Figure 4.10: T1- and T2-weighted MRI images of the brain, along with the brain tissue segmentation results. Red indicates Gray Matter (GM), green represents White Matter (WM), and blue corresponds to Cerebrospinal Fluid (CSF) [13].

brain tissue segmentation. HybridCTrm compares results for two different encoder implementations. TABS demonstrates that the integration of a Transformer module between the encoder and the decoder improves brain tissue segmentation performance and generalizability. 3D UX-Net mimics the behaviour of a hierarchical Transformer but with reduced computational complexity.

4.3.1 HybridCTrm

Sun et al. [54] proposed HybridCTrm, a hybrid multimodal segmentation model combining Transformers and CNNs. The model encodes a 3D input using both a CNN encoder and a Transformer encoder. For the final segmentation, the decoder concatenates features from the CNN and the Transformer into a single matrix, which is processed through batch normalization layers, a Parametric Rectified Linear Unit (PReLU), and a $1\times1\times1$ convolution kernel. Softmax is applied to generate final segmentation probabilities.

The HybridCTrm can be implemented using one of two strategies: the single-path strategy and the multi-path strategy. The single-path strategy combines T1- and T2-weighted MRI inputs into a multichannel image, which is then encoded using convolutions and Transformers. In contrast, the multi-path strategy encodes the T1- and T2-weighted inputs separately with independent encoders, merging the encoded representations afterward. A multi-path network is designed to integrate and utilize the information and features from various modalities, whereas a single-path network emphasizes the interactions between these modalities.

Compared to previous CNN-based methods without Transformer blocks, HybridCTrm outperforms in nearly all metrics with both single-path and multi-path strategies. The single-path strategy outperforms CNN models across all three brain tissues

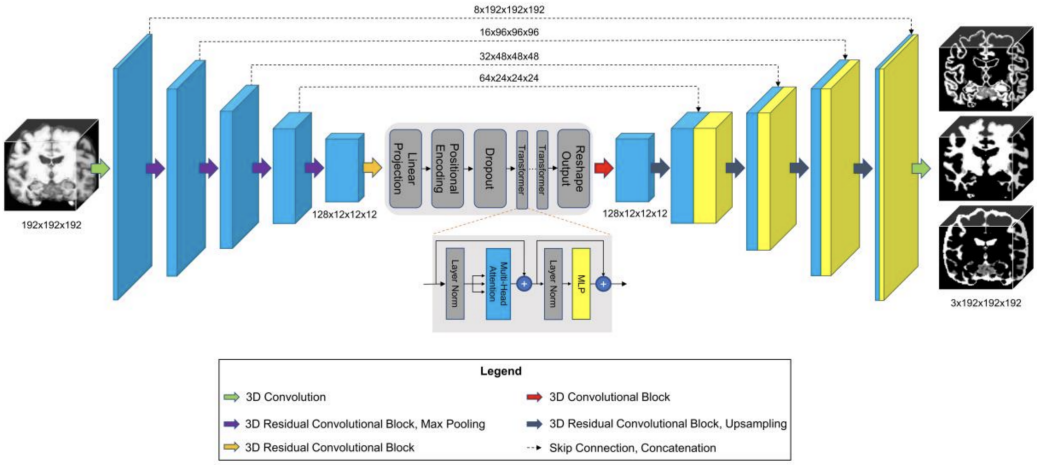

Figure 4.11: Architecture of TABS [48]. The model includes a 5-layer encoder-decoder with a Transformer between the encoder and the decoder. Input data is processed through the encoder to extract features, which are transformed into tokenized vectors for the Transformer module. The Transformer module captures long-range dependencies before passing data to the decoder for reconstruction. The final output consists of three channels corresponding to GM, WM, and CSF.

by integrating information from both modalities (T1 and T2) early on, enabling the Transformer to fully model their complexity. The multi-path strategy shows improvements on many metrics but is less effective than the single-path model. This is likely due to the limited information in each modality, which limits the Transformer's ability to learn features as effectively as the CNN's strong inductive bias. Nonetheless, the results show that the hybrid structure performs better than pure CNN models with greater stability.

4.3.2 TABS

Rao et al. [48] proposed the Transformer-based Automated Brain Tissue Segmentation (TABS), which is a 3D CNN-Transformer hybrid architecture for brain tissue segmentation. The TABS architecture follows the U-Net based architecture, and consists of 5-layers of 3D CNN encoder and decoder with a ViT between the encoder and the decoder, as illustrated in Figure 4.11. The input is first downsampled, and linear projection and learned positional embedding is used to convert the encoded feature tensor into tokenized vectors. These vectors are used as an input to the Transformer in the order that the positional embeddings are in. The decoder reconstructs the image back to the original input dimension using upsampling and skip connections, and a final convolution is applied for the final 3-channel output for each tissue type (WM, GM, and CSF).

TABS outperforms many benchmark U-Net-based models in segmentation accu-

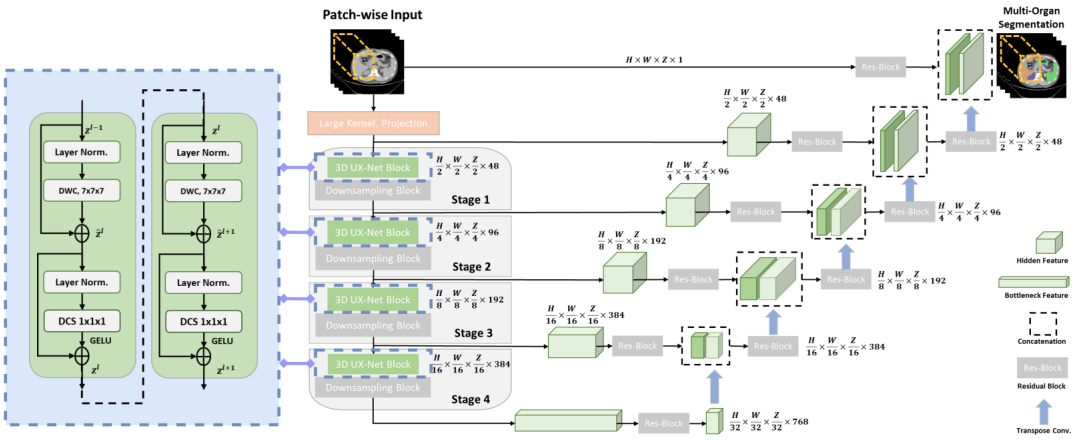

Figure 4.12: Overview of the 3D UX-Net architecture [30]. The depthwise convolutional blocks serve as the encoder backbone. Depthwise Convolution (DWC) applies a large kernel to effectively extract features. Depthwise Convolutional Scaling (DCS) uses a 1×1×1 kernel to independently scale each channel for improved feature representations with minimizing redundancy. H, W, and Z denote the height, width, and depth of the input patch dimensions. The LK projection layer generates patch-wise embeddings for hierarchical feature extraction.

racy across various datasets, showing robustness and generalizability in datasets with differing field strengths, scanner parameters, and patient conditions. However, as full-resolution MRI inputs were used for training TABS, memory constraints limited the batch size, potentially affecting its performance.

4.3.3 3D UX-Net

The 3D architectures that utilize the Transformer excel due to their large receptive fields for non-local self-attention, but require a substantial number of model parameters. To address this challenge, Lee et al. [30] proposed the 3D UX-Net, which uses CNNs with depth-wise convolution to emulate large receptive fields more efficiently. The 3D UX-Net does not directly use the Transformer architecture, but mimics the behaviour of a hierarchical Transformer. Specifically, it is inspired by the large receptive fields and hierarchical feature extraction, and uses Large Kernel (LK) depthwise convolutions instead of Multi-Head Self-Attention mechanisms to minimizes computational complexity.

The architecture of 3D UX-Net is illustrated in Figure 4.12. The Depth-wise Convolution encoder (DWC) processes 3D image volumes divided into random sub-volumes, with depth-wise convolution applied for feature extraction. Instead of flattening patches and applying a linear projection (similarly to most of the previously introduced architectures), LK convolutional layer computes partitioned feature maps that are projected into a 48-dimensional space. The LK convolutional layers are used

to create a large receptive field that mimics the behavior of self-attention mechanisms in hierarchical Transformers. Depth-wise Convolutional Scaling (DCS) process complements DWC by simulating global self-attention behavior. DCS improves feature learning by using 1×1×1 kernels to scale each channel feature independently, reducing cross-channel redundancy and preserving spatial resolution.

The encoder architecture includes four stages, each containing two LK convolution blocks, totaling eight layers. Each block consists of DWC layers along with DCS scaling and layer normalization. To reduce feature resolution by half and exchange information between channels, a standard 2×2×2 convolution block with stride 2 is used. This process is repeated to progressively reduce resolutions to extract multi-scale features hierarchically.

The multi-scale output from each encoder stage is connected to a CNN-based decoder via skip connections. The feature maps from each encoder stage are extracted and processed through a residual block with two post-normalized convolutional layers and instance normalization to stabilize features. These processed features are then upsampled with a transpose convolutional layer and concatenated with features from the previous stage. Residual features from input patches are also concatenated with upsampled features and passed through a residual block with 1×1×1 convolutional layer and softmax activation to predict segmentation probabilities.

3D UX-Net outperforms many of the Transformer-based architectures, including TransBTS, UNETR, and Swin-UNETR, with lower number of parameters and reduced training complexity. The results have been validated using three challenging public datasets, including for brain tissue segmentation, in both supervised training and transfer learning.

4.4 Future Directions

Transformer based models have significantly improved medical image segmentation, yet several challenges remain in regards of limited labeled data, model efficiency, and interpretability. Bringing these models into clinical workflows requires validation and explainability, as well as generalizability across different institutions, scanner types, and imaging protocols. In addition, these based models often require extensive computational power, making them difficult to deploy in real-world clinical settings with constrained resources.

Current methods for fully-supervised deep learning models heavily depend on large labeled MRI datasets. However, obtaining segmentation labels, particularly for brain tumors, is both costly and time-consuming [62]. In future research, the focus is to build models to train with fewer labels. This would be possible using semi-

supervised learning to train models with limited labeled and extensive unlabeled data. Semi-supervised learning uses the information gathered from unlabeled data to achieve effective segmentation with minimal manual labeling. Additionally, weak-supervised learning can be used for simpler box-level annotations instead of detailed pixel/voxel labels. This would result in efficient training while maintaining similar performance, significantly reducing annotation effort.

The datasets used for training often contain noisy annotations, particularly with tumors due to the complexity of marking tumor regions [62]. While some methods exist for handling noisy labels in image classification, they are challenging to adapt for segmentation. Additionally, brain tumors occupy only a small part of the brain, causing class imbalance in MRI data, which can bias segmentation towards healthy tissue and limit detection of small tumors. Adaptive class reweighting can improve the performance on small tumor regions, as it gives higher importance to the minority class, aiming to improve the model's focus on tumor regions.

Current brain segmentation methods typically assume complete MRI data, but their effectiveness drops significantly with incomplete inputs, such as missing T1, T2 or FLAIR images [62]. In clinical practices, obtaining multiple modalities is often challenging, as many institutions only have partial data due to equipment limitations. Although some recent studies have proposed solutions for missing modalities, such as [32], [68], and [69], they generally address only specific cases, limiting their applicability. Image segmentation competitions, such as BraTS, release well-annotated datasets, which serve as benchmarks for advancing semantic segmentation models [56]. Most research improvements rely on these datasets, making them important for progress in the field [50]. The BraTS challenge is likely the main reason brain tumor segmentation has attracted more attention in the research landscape compared to other brain segmentation tasks, such as brain tissue segmentation.

In clinical environments, deep models are deployed on resource-limited devices, requiring them to be efficient [62]. During training, models are expected to be lightweight and efficient. Compression techniques, such as pruning and quantifying weights, distillation, and low-rank approximations, can reduce memory and computational demands. Research on model efficiency and deployment for brain segmentation tasks is limited but considered essential for future clinical applications.

Deep learning models are often seen as "black-boxes" due to their lack of interpretability, which poses challenges in understanding their decision-making process [51]. Especially in clinical practice, it is important to understand what guides the model's decision process. One suggestion is to use visualizing feature maps to highlight dominant regions that influence model outputs [62]. Researchers have developed various techniques to analyze the intermediate layers of deep learning models and recent ad-

vancements adopt feature attribution methods to identify the most relevant features to a given prediction. Additionally, attention maps offer a way to visualize how a Transformer-based model processes and prioritizes different parts of an image.

Despite these challenges, Transformers have shown very promising results for medical image segmentation tasks. The hybrid Transformer architectures, as discussed in this chapter, show notable performance improvements (around 13% increase in overall Dice Score) over baseline Transformer models [50], highlighting rapid progress in the field. Architectures that are able to process high-dimensional volumetric information have demonstrated superior results, but improving on their efficiency is not yet extensively explored. Additionally, the Transformers have been applied for various tasks other than segmentation, including detection, classification, restoration, synthesis, and registration, which further justifies the Transformer's versatility and potential in medical imaging.

5. Experiments

U-Net has achieved significant success in segmentation tasks due to its effective capture of local features. As described earlier, recent advances in Transformer-based architectures offer new architectural designs for segmentation models. The purpose of this experiment is to study how the integration of Transformer models with U-Net impact segmentation performance, specifically examining whether a non-pretrained Transformer can enhance segmentation performance on a small 2D dataset.

The hypothesis is that the integration of the Transformer should result in segmentation performance that is comparable to or better than that of the conventional U-Net. The models' performance is validated using the Dice Score and the 95th percentile Hausdorff Distance (explained in Section 4.2). The code for the experiment is available at https://www.github.com/juicy-monkey/grad_thesis.

5.1 Setup

The code is written in Python, and main libraries used include PyTorch [38], Torchvision [36], seg-metrics [27], and Matplotlib [26]. PyTorch is used for developing the models by using predefined and custom components. Torchvision, an extension of PyTorch for computer vision applications, provides pre-coded models (such as the Vision Transformer) and functions for image transformations. The seg-metrics library evaluates image segmentation models, with Dice and the 95th percentile Hausdorff Distance used as validation metrics. Matplotlib is used to visualize model performance with different input images.

Dataset

The dataset used in this experiment was acquired from [5]. The dataset contains brain MR images with manually segmented FLAIR abnormality masks. The images were obtained from The Cancer Imaging Archive (TCIA), corresponding to 110 patients from The Cancer Genome Atlas (TCGA) LGG collection.

Each input image is a 3-channel RGB image, consisting of three MRI sequences

in the order of pre-contrast, FLAIR, and post-contrast. Pre-contrast (non-contrast) MRI is acquired before the injection of a contrast agent, such as gadolinium. FLAIR highlights structures near fluid, as discussed in Section 2.1. Post-contrast MRI refers to images captured after the injection of a contrast agent. The segmentation masks (target images) are 1-channel binary images.

For this experiment, 2D brain slices are considered to be independent, i.e. adjacent slices are not considered for segmentation. To align with the experiment's objectives, the dataset is filtered, as using the full dataset leads to poor predictions across all models. The original dataset consists of 3929 MRI slices, but since most do not contain brain tumors, only 1373 relevant slices are used for this experiment. Among the 1373 images, 1098 are randomly picked for training, and the remaining 275 images are reserved for performance comparison.

Models

The experiment compares three models:

U-Net The model introduced in Section 4.1. This model serves as the baseline architecture for comparison.

ViT-UNet A customized U-Net with Skip Connection from a ViT, designed specifically for this thesis. A ViT is used to transform the input image into a 784-length tensor with a patch size of 16×16, similar to the ViT-Base model in [15]. A learnable linear layer extends this tensor to the length of $1024 \cdot 16 \cdot 16 = 262144$, which is then reshaped into dimension $(B, 1024, 16, 16)$, where B is the batch size. This transformed tensor is used in the skip connection to concatenate features from the final down-convolution. The model has four encoder layers, with a Multi-Layer Perceptron (MLP) dimension of 2048 and 16 attention heads.

2D UNETR A modified version of the UNETR model introduced in Section 4.1, where all 3D operations are replaced with 2D operations. The code has been adapted and modified from [57] to be suitable for this thesis. Similar to the ViT-UNet, the Transformer encoder in the 2D UNETR outputs a 784-length tensor. The hyperparameters are the same as in the ViT-UNet, but the number of layers is increased from 4 to 12, similar to the original UNETR architecture.

For all models, a final Sigmoid activation maps predictions to probabilities, determining whether each pixel belongs to a tumor or not.

Training

The models are trained on the same dataset with the same parameters. Instead of processing all data at once, the training data is divided into smaller subsets (batches). Each batch, which is set to 32, is passed through the model, and the gradient of the loss function is computed to update the model weights, resulting in more efficient training compared to processing the entire dataset at once.

The optimization algorithm used is Stochastic Gradient Descent (SGD). Gradient Descent is an iterative method that updates model parameters by moving in the opposite direction of the gradient to minimize the loss function, improving accuracy on both training and test datasets. Unlike standard Gradient Descent, which processes the entire dataset at each step, SGD updates model parameters using small, randomly selected batches, introducing stochasticity into the optimization process. The learning rate, which scales the size of the step taken in the direction of the negative gradient, is set to 0.1. Momentum is set to 0.9, helping the optimizer gain speed in consistent gradient descent directions while reducing oscillations in less relevant ones.

The loss function used is Binary Cross Entropy Loss (BCE Loss), which is a widely used metric for binary classification tasks. BCE Loss measures the dissimilarity between the predicted probability and the ground truth label. For a single-channel output image, BCE Loss is defined as

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

where N is the total number of pixels, y_i is the ground truth label (0 for background, 1 for tumor), and $\hat{y}_i$ is the predicted probability (output from the Sigmoid function, ranging from 0 to 1). The log-function penalizes wrong predictions more heavily than correct predictions.

5.2 Results

Table 5.1 summarizes the results from the testing set of 275 images. The Dice Score represents the overlap between predicted and ground truth segmentation masks, with higher values indicating better performance. The 95th percentile Hausdorff Distance (HD95) measures the segmentation error, where a lower value suggests better model performance in terms of accurately detecting tumor boundaries.

As shown in Table 5.1, the standard U-Net model achieved the highest Dice Score (68.634%), demonstrating the best segmentation performance among the tested models. The ViT-UNet model, which incorporates a Transformer component, performed slightly worse with a Dice Score of 67.129%. Additionally, its HD95 value of 17.692 was higher

Model	Dice Score (%)	HD95
U-Net	68.634	15.982
ViT-UNet	67.129	17.692
2D UNETR	60.396	28.663

Table 5.1: Comparison of average Dice Score and average 95th Percentile Hausdorff Distance (HD95) between the models.

than that of the U-Net (15.982), meaning that its segmentation boundaries were less accurate. The 2D UNETR model, which is a more novel Transformer-based approach, had the lowest Dice Score of 60.396% and the highest HD95 value of 28.663. These results indicate that this model struggled the most with segmenting brain tumor regions effectively. The increased complexity of the Transformer-based architectures did not yield improvements in segmentation performance, and in fact, resulted in a decline in accuracy and an increase in boundary error.

Figure 5.1 further highlights the performance differences across models. In general, U-Net consistently produces segmentation masks that align with the ground truth. The tumor shapes are captured well with smooth and continuous boundaries, which aligns with its performance in quantitative metrics. ViT-UNet, while naive in its use of the Transformer, demonstrates slightly less reliable results. However, in some cases, it was able to locate brain tumor areas that the other models failed to detect. The 2D UNETR model, on the other hand, performs the weakest among the three. Its segmentation results are frequently noisy, with fragmented and disconnected regions that fail to accurately represent the tumor areas.

Overall, the experimental results and visualizations suggest that the addition of Transformer components to the U-Net architecture does not result in significant improvements in segmentation performance on a small dataset. The U-Net model remains the most effective, achieving the best balance between segmentation accuracy and boundary precision. The ViT-UNet and 2D UNETR models are affected by higher complexity and, most likely, the limitations of the dataset size. The additional computational complexity might have contributed to overfitting or difficulties in learning effective features.

5.3 Discussion

The objective of this experiment was to explore and study the implementations of Transformer components, and integrate these components to the U-Net to improve on its performance. The process involved multiple trials and errors, including experimental

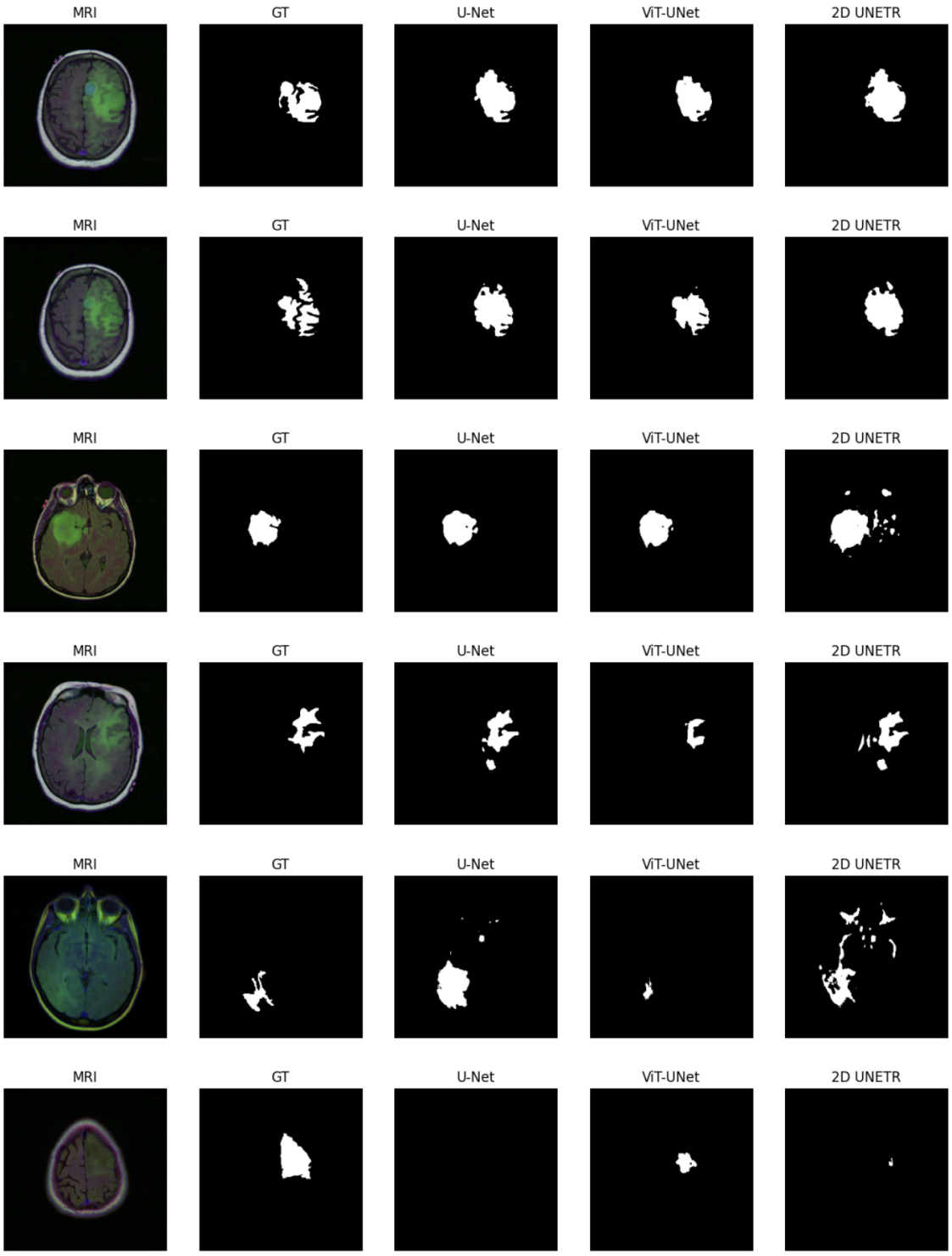

Figure 5.1: Visual comparison of segmentation results produced by the different models. Each row represents a different test case from the dataset, and the columns display the MRI input, ground truth segmentation (GT), and predictions from the U-Net, ViT-UNet, and 2D UNETR models.

models that integrated Transformers with multiple linear layers, significantly increasing computational cost. While these models had potential to outperform U-Net, their inefficiency and high training costs made them impractical.

Unfortunately, the models selected for the experiment showed that incorporating a Transformer into the U-Net architecture did not improve segmentation results. However, considering that the training set relatively small, further testing would be needed to determine whether these models can outperform the traditional U-Net. While the results of this experiment cannot lead to general conclusions, the results suggest that performance did not degrade significantly when including the Transformer, indicating potential for further exploration and optimization in future studies.

Several improvements could be made to improve the experiment. First, including cross-validation would reduce the small dataset size to better generalize results. Second, increasing the dataset size or applying data augmentation techniques could help reduce bias and improve generalizability. As the dataset contains only a small proportion of images with brain tumor areas, techniques such as synthetic data generation and image augmentation could improve model performance and robustness. Third, since individual image slices were treated as independent, there models lack spatial coherence across the whole brain area, limiting the models ability to consider surrounding brain structures. Treating the dataset as a collection of 3D models would address this issue, but it would also require greater computational resources and more advanced modeling techniques.

The training process used BCE Loss as the loss function for all models. While BCE Loss is effective in many applications, it has limitations in segmentation tasks when dealing with imbalanced datasets. In brain tumor segmentation datasets, the proportion of tumor pixels is significantly smaller than the background, making BCE Loss suboptimal. Many recent studies on medical image segmentation have proposed modified loss functions that incorporate Dice Loss in combination with BCE Loss. Dice Loss is specifically designed to address class imbalance by directly optimizing the Dice Score, which measures the overlap between predicted and ground truth masks. For example, the UNETR [23] architecture uses a combination of soft Dice Loss and Cross-Entropy Loss. Incorporating Dice Loss into the training process could potentially improve the segmentation performance of all models tested in this study. Specifically, it might address the deficiencies of 2D UNETR, where noisy and fragmented segmentations were produced. While weighted BCE Loss could have been used to address the class imbalance, it was not incorporated in this experiment to maintain a simple experimental setup.

Although Transformer-based architectures did not outperform the conventional U-Net, they did not significantly degrade segmentation performance. Notably, ViT-

UNet, while slightly underperforming in overall metrics, detected tumor regions that other models missed. This highlights its potential in capturing critical features that might be overlooked by traditional CNN-based architectures and suggests that Transformers may still offer advantages in specific cases. However, it remains unclear whether the overall underperformance of Transformer-based architectures comes from limitations in the loss function, model design, small dataset size, or a combination of these factors. With further refinements, the experiment could offer deeper insights into the Transformer's suitability and impact on segmentation performance.

6. Conclusion

This thesis provided background on medical imaging and brain image segmentation, and reviewed the theoretical foundation of the Transformer architecture and its extensions in computer vision. Recent state-of-the-art Transformer-based architectures for brain image segmentation were reviewed, and performance between a traditional U-Net and more novel models was experimentally compared on a small-scale setup.

With advances in deep learning, particularly CNNs and Transformers, the performance of automated segmentation has significantly improved in recent years. Recent state-of-the-art models, including UNETR, TransBTS, Swin-UNETR, and VT-UNet, integrate the local feature extraction capabilities of U-Net with global context modeling of Transformers. These models have surpassed previous CNN-based models, setting new benchmarks in brain image segmentation.

Experimental results showed that while Transformer-based models offer promising capabilities, they did not outperform the conventional U-Net. The U-Net model achieved the highest segmentation accuracy, while a custom ViT-UNet and 2D UNETR had slightly lower accuracy and increased boundary errors. These findings suggest that the added complexity of Transformer-based architectures does not necessarily enhance performance on small datasets. With further refinement, including more suitable loss functions, larger datasets, and improved model designs, Transformer-based models could still show potential advantages in small-scale testing.

This thesis contributes to the ongoing research on deep learning for brain image segmentation by evaluating the role of Transformer-based architectures. While Transformers have been shown to outperform many CNN-based architectures, their practical benefits in medical segmentation require further research. Addressing challenges regarding high computational costs, limited annotated datasets, and interpretability, as well as trade-offs between accuracy, efficiency, and real-world usability in medical image analysis workflows is required.

Bibliography

[1] Z. Akkus, A. Galimzianova, A. Hoogi, D. L. Rubin, and B. J. Erickson. Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions. *Journal of digital imaging*, 30:449–459, 2017.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate, 2016.

[3] F. E. Boas, D. Fleischmann, et al. CT artifacts: causes and reduction techniques. *Imaging in Medicine*, 4(2):229–240, 2012.

[4] D. Boyd and M. Lipton. Cardiac computed tomography. *Proceedings of the IEEE*, 71(3):298–307, 1983.

[5] M. Buda. Brain MRI segmentation. https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation/data. Accessed: 9.1.2025.

[6] 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. In L. Karlinsky, T. Michaeli, and K. Nishino, editors, *Computer Vision – ECCV 2022 Workshops*, pages 205–218, Cham, 2023. Springer Nature Switzerland.

[7] B. Chen, Y. Liu, Z. Zhang, G. Lu, and A. W. K. Kong. TransAttUnet: Multi-Level Attention-Guided U-Net With Transformer for Medical Image Segmentation. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8(1):55–68, 2024.

[8] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches, 2014.

[9] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pages 424–432, Cham, 2016. Springer International Publishing.

[10] K. Cleary and T. M. Peters. Image-guided interventions: technology review and clinical applications. *Annual Review of Biomedical Engineering*, 12:119–142, August 15 2010.

[11] M. C. de Verdier, R. Saluja, L. Gagnon, D. LaBella, U. Baid, N. H. Tahon, M. Foltyn-Dumitru, et al. The 2024 Brain Tumor Segmentation (BraTS) Challenge: Glioma Segmentation on Post-treatment MRI, 2024.

[12] Department of Radiology, University of Wisconsin. MRI Terminology. https://sites.google.com/a/wisc.edu/neuroradiology/image-acquisition/magnetic-resonance-imaging/mr-terminology, 2011. Accessed: 7.11.2024.

[13] I. Despotović, B. Goossens, and W. Philips. MRI Segmentation of the Human Brain: Challenges, Methods, and Applications. *Computational and Mathematical Methods in Medicine*, 2015(1):450341, 2015.

[14] L. Dora, S. Agrawal, R. Panda, and A. Abraham. State-of-the-Art Methods for Brain Tissue Segmentation: A Review. *IEEE Reviews in Biomedical Engineering*, 10:235–249, 2017.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, 2021.

[16] E. Goceri. Medical image data augmentation: techniques, comparisons and interpretations. *Artificial Intelligence Review*, 56(11):12561–12605, 2023.

[17] E. Goceri. Medical image data augmentation: techniques, comparisons and interpretations. *Artificial Intelligence Review*, 56(11):12561–12605, 2023.

[18] H. Greenspan, B. van Ginneken, and R. M. Summers. Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique. *IEEE Transactions on Medical Imaging*, 35(5):1153–1159, 2016.

[19] L. Hamberg and H. Aronen. Magneettikuvauksen perusteet ja tutkimusmenetelmät. *Duodecim*, 108(8):713–724, 1992.

[20] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao. A Survey on Vision Transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):87–110, 2023.

[21] K. Han, A. Xiao, E. Wu, J. Guo, C. XU, and Y. Wang. Transformer in Transformer. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 15908–15919. Curran Associates, Inc., 2021.

[22] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu. Swin UNETR: Swin Transformers for Semantic Segmentation of Brain Tumors in MRI Images. In A. Crimi and S. Bakas, editors, *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 272–284, Cham, 2022. Springer International Publishing.

[23] 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 *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 574–584, January 2022.

[24] Q. Hu, G. Qian, A. Aziz, and W. Nowinski. Segmentation of brain from computed tomography head images. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 3375–3378, 2005.

[25] S. Huisman, M. Maspero, M. Philippens, J. Verhoeff, and S. David. Deep learning-based brain segmentation model performance validation with clinical radiotherapy CT, 2024.

[26] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[27] J. Jia, M. Staring, and B. C. Stoel. seg-metrics: a Python package to compute segmentation metrics. *medRxiv*, pages 2024–02, 2024.

[28] Q. Jia and H. Shu. BiTr-Unet: A CNN-Transformer Combined Network for MRI Brain Tumor Segmentation. In A. Crimi and S. Bakas, editors, *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 3–14, Cham, 2022. Springer International Publishing.

[29] A. Lauric and S. Frisken. Soft Segmentation of CT Brain Data. *Tufts University*, 2007.

[30] H. H. Lee, S. Bao, Y. Huo, and B. A. Landman. 3D UX-Net: A Large Kernel Volumetric ConvNet Modernizing Hierarchical Transformer for Medical Image Segmentation, 2023.

[31] L. Lenchik, L. Heacock, A. A. Weaver, R. D. Boutin, T. S. Cook, J. Itri, C. G. Filippi, R. P. Gullapalli, J. Lee, M. Zagurovskaya, T. Retson, K. Godwin, J. Nicholson, and P. A. Narayana. Automated Segmentation of Tissues Using CT and MRI: A Systematic Review. *Academic Radiology*, 26(12):1695–1706, 2019.

[32] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda. Early-Learning Regularization Prevents Memorization of Noisy Labels. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20331–20342. Curran Associates, Inc., 2020.

[33] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, October 2021.

[34] Z. Liu, L. Tong, L. Chen, Z. Jiang, F. Zhou, Q. Zhang, X. Zhang, Y. Jin, and H. Zhou. Deep learning based brain tumor segmentation: A survey. *Complex & intelligent systems*, 9(1):1001–1026, 2023.

[35] P. K. Mallick, B. S. Satapathy, M. N. Mohanty, and S. S. Kumar. Intelligent technique for CT brain image segmentation. In *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, pages 1269–1277, 2015.

[36] S. Marcel and Y. Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 1485–1488, New York, NY, USA, 2010. Association for Computing Machinery.

[37] S. A. Mirowitz. MR IMAGING ARTIFACTS: Challenges and Solutions. *Magnetic Resonance Imaging Clinics of North America*, 7(4):717–732, 1999.

[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. *PyTorch: an imperative style, high-performance deep learning library.* Curran Associates Inc., Red Hook, NY, USA, 2019.

[39] W. Pedrycz and S.-M. Chen. *Deep learning: Concepts and architectures.* Springer Cham, 2020.

[40] H. Peiris, M. Hayat, Z. Chen, G. Egan, and M. Harandi. A Robust Volumetric Transformer for Accurate 3D Tumor Segmentation. In L. Wang, Q. Dou, P. T.

Fletcher, S. Speidel, and S. Li, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, pages 162–172, Cham, 2022. Springer Nature Switzerland.

[41] R. Preetha, M. J. P. Priyadarsini, and J. S. Nisha. Comparative Study on Architecture of Deep Neural Networks for Segmentation of Brain Tumor using Magnetic Resonance Images. *IEEE Access*, 11:138549–138567, 2023.

[42] O. Puonti. *Computational Analysis of Brain Images: Towards a Useful Tool in Clinical Practice*. PhD thesis, Technical University of Denmark, 2016.

[43] J. Qian, Z. Song, Y. Yao, Z. Zhu, and X. Zhang. A review on autoencoder based representation learning for fault detection and diagnosis in industrial processes. *Chemometrics and Intelligent Laboratory Systems*, 231:104711, 2022.

[44] Radiology Cafe. CT Image Quality. https://www.radiologycafe.com/frcr-physics-notes/ct-imaging/ct-image-quality/, 2021. Accessed: 8.11.2024.

[45] Radiopaedia.org. Computed Tomography. https://radiopaedia.org/articles/computed-tomography, 2024. Accessed: 8.11.2024.

[46] H. Ramamoorthy, M. Ramasundaram, R. S. P. Raj, and K. Randive. TransAttU-Net Deep Neural Network for Brain Tumor Segmentation in Magnetic Resonance Imaging. *IEEE Canadian Journal of Electrical and Computer Engineering*, 46(4):298–309, 2023.

[47] R. Ranjbarzadeh, A. Caputo, E. B. Tirkolaee, S. Jafarzadeh Ghoushchi, and M. Bendechache. Brain tumor segmentation of MRI images: A comprehensive review on the application of artificial intelligence tools. *Computers in Biology and Medicine*, 152:106405, 2023.

[48] V. M. Rao, Z. Wan, S. Arabshahi, D. J. Ma, P.-Y. Lee, Y. Tian, X. Zhang, A. F. Laine, and J. Guo. Improving across-dataset brain tissue segmentation for MRI imaging using transformer. *Frontiers in Neuroimaging*, 1, 2022.

[49] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[50] F. Shamshad, S. Khan, S. W. Zamir, M. H. Khan, M. Hayat, F. S. Khan, and H. Fu. Transformers in medical imaging: A survey. *Medical Image Analysis*, 88:102802, 2023.

[51] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni. U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications. *IEEE Access*, 9:82031–82057, 2021.

[52] B. Srinivas, B. Anilkumar, N. devi, and V. Aruna. A fine-tuned transformer model for brain tumor detection and classification. *Multimedia Tools and Applications*, pages 1–25, 2024.

[53] P. Suetens. *Fundamentals of medical imaging*. Cambridge university press, 2017.

[54] Q. Sun, N. Fang, Z. Liu, L. Zhao, Y. Wen, and H. Lin. HybridCTrm: Bridging CNN and Transformer for Multimodal Brain Image Segmentation. *Journal of Healthcare Engineering*, 2021(1):7467261, 2021.

[55] 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 (CVPR)*, pages 20730–20740, June 2022.

[56] H. Thisanke, C. Deshan, K. Chamith, S. Seneviratne, R. Vidanaarachchi, and D. Herath. Semantic segmentation using Vision Transformers: A survey. *Engineering Applications of Artificial Intelligence*, 126:106669, 2023.

[57] N. Tomar. Semantic-Segmentation-Architecture. https://github.com/nikhilroxtomar/Semantic-Segmentation-Architecture/blob/main/PyTorch/unetr_2d.py, 2023. Accessed: 28.1.2025.

[58] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou. Training data-efficient image transformers & distillation through attention. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10347–10357. PMLR, 18–24 Jul 2021.

[59] C. M. Umarani, S. Gollagi, S. Allagi, K. Sambrekar, and S. B. Ankali. Advancements in deep learning techniques for brain tumor segmentation: A survey. *Informatics in Medicine Unlocked*, 50:101576, 2024.

[60] S. Vaara, S. Syväranta, and J. Peltonen. Magneettikuvauksen ABC: T1, T2, fat sat, DWI ynnä muut: radiologin salakieli auki kirjoitettuna. *Duodecim*, 137(24):2681–2688, 2021.

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[62] P. Wang, Q. Yang, Z. He, and Y. Yuan. Vision transformers in multi-modal brain tumor MRI segmentation: A review. *Meta-Radiology*, 1(1):100004, 2023.

[63] T. Wang, H. Xing, Y. Li, S. Wang, L. Liu, F. Li, and H. Jing. Deep learning-based automated segmentation of eight brain anatomical regions using head CT images in PET/CT. *BMC Medical Imaging*, 22(1):99, 2022.

[64] W. Wang, C. Chen, M. Ding, J. Li, H. Yu, and S. Zha. TransBTS: Multimodal Brain Tumor Segmentation Using Transformer, 2021.

[65] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. CBAM: Convolutional Block Attention Module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[66] L. Wu, S. Wang, J. Liu, L. Hou, N. Li, F. Su, X. Yang, W. Lu, J. Qiu, M. Zhang, et al. A survey of MRI-based brain tissue segmentation using deep learning. *Complex & Intelligent Systems*, 11(1):1–16, 2025.

[67] H. Xiao, L. Li, Q. Liu, X. Zhu, and Q. Zhang. Transformers in medical image segmentation: A review. *Biomedical Signal Processing and Control*, 84:104791, 2023.

[68] Q. Yang, X. Guo, Z. Chen, P. Y. M. Woo, and Y. Yuan. D2-Net: Dual Disentanglement Network for Brain Tumor Segmentation With Missing Modalities. *IEEE Transactions on Medical Imaging*, 41(10):2953–2964, 2022.

[69] T. Zhou, S. Canu, P. Vera, and S. Ruan. Latent Correlation Representation Learning for Brain Tumor Segmentation With Missing MRI Modalities. *IEEE Transactions on Image Processing*, 30:4263–4274, 2021.

[70] J. Zopes, M. Platscher, S. Paganucci, and C. Federau. Multi-Modal Segmentation of 3D Brain Scans Using Neural Networks. *Frontiers in Neurology*, 12, 2021.