Friedrich-Alexander-Universität

Attention-based networks for brain segmentation in k-space

Master's Thesis in Data Science

submitted by

Erik Gösche

born 02.10.1999 in Halberstadt

Written at

Computational Imaging Lab
Department Artificial Intelligence in Biomedical Engineering
Friedrich-Alexander-Universität Erlangen-Nürnberg

in Cooperation with

Center for Intelligent Imaging
Department of Radiology and Biomedical Imaging
University of California, San Francisco.

Advisors: Prof. Florian Knoll,
Dr. Andreas Rauschecker

Started: 08.05.2023
Finished: 08.11.2023

Declaration of Originality

I, Erik Gösche, student registration number: 22971879, hereby confirm that I completed the submitted work independently and without the unauthorized assistance of third parties and without the use of undisclosed and, in particular, unauthorized aids. This work has not been previously submitted in its current form or in a similar form to any other examination authorities and has not been accepted as part of an examination by any other examination authority.

Where the wording has been taken from other people's work or ideas, this has been properly acknowledged and referenced. This also applies to drawings, sketches, diagrams and sources from the Internet.

In particular, I am aware that the use of artificial intelligence is forbidden unless its use an aid has been expressly permitted by the examiner. This applies in particular to chatbots (especially ChatGPT) and such programs in general that can complete the tasks of the examination or parts thereof on my behalf.

Furthermore, I am aware that working with others in one room or by means of social media represents the unauthorized assistance of third parties within the above meaning, if group work is not expressly permitted. Each exchange of information with others during the examination, with the exception of examiners and invigilators, about the structure or contents of the examination or any other information such as sources is not permitted. The same applies to attempts to do so.

Any infringements of the above rules constitute fraud or attempted fraud and shall lead to the examination being graded "fail" („nicht bestanden").

Place, date	Signature

Abstract

Attention-based networks show cutting-edge performance in medical image segmentation due to their ability to capture long-range spatial relationships. However, since they do not have a receptive field like convolutional neural networks (CNNs), they suffer in the modeling of local features. Images, that are in the frequency domain, might be more suitable for the attention mechanism. By transforming images into the frequency domain, local features are represented globally. Moreover, considering the convolution theorem, the attention operation could intuitively be viewed as a convolution. Due to the properties of MRI data acquisition, these types of images are particularly suitable. The goal of this work is to investigate how the choice of the image domain (pixel domain or frequency domain) affects the segmentation results of deep learning models. Attention-based networks are in particular focus here. Furthermore, it is to be examined whether additional positional encoding is necessary when an attention-based network is used and the input images are in the frequency domain. For the evaluation of these research questions, a skull stripping task and a brain tissue segmentation task are posed. The attention-based models used in this work are the PerceiverIO and a Transformer encoder. To provide a comparison to non-attention-based models, an MLP and the ResMLP are additionally trained and tested. As a reference and for a better placement, the results will be compared with those of the nnU-Net. It was experimentally shown that the choice of input and label domain have significant effects on the segmentation results. Also, additional positional encoding does not seem to be beneficial for attention-based networks if the input is in the frequency domain. Even though none of the models used reached the performance of the nnU-Net, the rather non-complex models showed promising results.

Contents

Chapter 1

Introduction

1.1 Background and Motivation

With the increasing amount and complexity of medical imaging data available within the healthcare system, Artificial Intelligence (AI) has been increasingly used to process this data in a reasonable amount of time. In particular, Machine Learning (ML) and Deep Learning (DL) techniques deliver superior performance in assessing brain-related problems using image data [Seg+20, p.1].
To generate such data, there are imaging modalities such as computed tomography (CT), magnetic resonance imaging (MRI), ultrasound, positron emission tomography (PET), single-photon emission computed tomography (SPECT) or combinations of previously mentioned such as positron emission tomography–computed tomography (PET-CT), or positron emission tomography–magnetic resonance imaging (PET-MRI). One of the most important tasks in medical imaging is segmentation. Medical image segmentation describes the process of separating selected regions and structures such as specific tissue types from other structures, and the background. For this purpose, individual pixels or voxels are classified manually, semi-automatically, or automatically [Rog+09, p.71] [Li+21, p.2].
From the segmented image, relevant information can be extracted, which is crucial for the diagnosis of diseases but also for the preparation, guidance, and subsequent analysis of treatments [Seg+20, p.2] [Kap+14, p.79]. For example, DL models are being developed that determine the volume of a brain tumor and enable therapy response assessment [Gut+23]. Medical image segmentation plays a key role in research in addition to the clinical field. The analysis of volume and pathological processes by segmentation serve as biomarkers, and are used for clinical drug or therapy trials [Rog+09, p.113].

With the success of the Transformer model [Vas+17] and later the Vision Transformer [Dos+20], which was further developed for image processing, the attention mechanism became increasingly successful in the domain of DL. Today, attention-based networks outperform previous architectures in vision tasks such as image classification, object detection, or semantic segmentation [Str+21] [Li+22] [Rya+23]. Various attention-based models are also being developed in the area of medical vision tasks, which also deliver state-of-the-art performance [Lin+22] [Hat+22] [Cao+23]. With the attention mechanism, models have an excellent way to represent long-range dependencies within the input. Based on the recent development of architectures, this is shown to be beneficial in vision tasks and an advantage over the convolution operations that were prevalent in the past [Zha+22, p.1-2]. However, the Transformer attention mechanism also has weaknesses. Without modifications, it is not applicable to large-scale inputs, since its complexity is quadratic to the number of input values [Vas+17, p.6].

Furthermore, a larger amount of data is required to reach a similar performance as convolutional neural networks (CNNs). This is most likely because Vision Transformers do not have inductive biases like local receptive fields or shared weights which support feature learning from images. For pixel data, the resulting explicit relating of neighboring pixels, which are usually strongly correlated and the translation in-variance are desirable [Wu+21, p.1-2].

Another effect of not having a receptive field is that all input values are used to calculate attention. This means each value is related to the others. This is beneficial for global feature extraction, but disruptive for capturing local features [Che+21, p.2].

This work aims to turn what initially appear to be weaknesses of attention-based models into strengths. By using image data in the frequency domain instead of the time or pixel domain, local features are spread over the entire input. Local problems like segmentation tasks become global problems which makes the task unnecessarily difficult at first glance. However, for a Transformer-based architecture, the problem could become much more suitable. The absence of the mentioned inductive distortions is also not a problem in the frequency domain but is even necessary. Since there is usually no correlation between adjacent values in the frequency domain, such properties would be misleading. Also, the concept of translation in-variance only makes sense in pixel images.

On top of that, an additional positional encoding of the input, which is usually necessary for attention operations, might not be needed if the input is in the frequency domain. Such an observation has already been made for natural language processing (NLP) tasks [Lee+22].

MR scan data is perfectly suited for this approach, as it is already in the frequency domain (called k-space) due to its acquisition characteristics. For this reason, the object of this work is to investigate attention-based models that perform segmentation tasks using k-space data from MRI scans.

1.2 Research Objectives

The goal of this work is to find out to what extent the domain of the input values has an influence on DL-based segmentation models. Of particular interest here are attention-based models, as they could benefit from input data in the frequency domain due to their properties. The advantages and disadvantages of transforming input images or volumes into the frequency domain have to be identified. Of central importance here is the segmentation result. Segmentation success is to be measured using common metrics (such as the Dice score). There will also be a qualitative look at the segmentation, highlighting eventual patterns in the nature of the segmentation depending on the input domain. For a closer look at attention-based models, the PerceiverIO model [Jae+22] and a custom Transformer encoder model are used. The latter is the fundamental component of the successful BERT model [Dev+19]. Using these two models, the aim is also to check whether the input, if it is in the frequency domain, needs to be additionally contacted with a positional encoding. Two non-attention-based models are also used to obtain meaningful results regarding the selection of the input domain. More specifically, an multi-layer perceptron (MLP) and the ResMLP model [Tou+21]. The MLP acts as a proof-of-concept model. Then, to further frame the results, the nnU-Net [Ise+21] as a leading-edge model will perform the same segmentation tasks in the pixel domain.
The entire research questions are to be examined based on reconstructed MR images. Since MRI data is already generated in the frequency domain (k-space) anyway due to the way MRI scanners work, this data source is particularly suitable for this work. Two brain segmentation problems are used as tasks for the models: skull stripping and brain tissue segmentation.

1.3 Scope

To meet the research objectives, all experiments are evaluated regarding the segmentation results. Therefore, several objective metrics were used. There are no measurements

regarding the complexity, CPU usage, GPU usage, or training time reported. To measure the performance of the DL models, two different datasets were used to obtain results that are as meaningful as possible. However, all datasets are from one imaging modality, namely MRI. Among the datasets, the acquisition parameters vary. Using three-dimensional data, this work differs from others not only in terms of subject matter but also in terms of methodology. Therefore, the evaluation in this paper is based on several models and datasets. For the evaluation of attention-based models, only models that do not divide the input image into patches or use a hierarchical approach were considered. This limits the choice considerably but is useful in that it ensures that the models used here can also work with k-space data from MRI scans.
Also, to create a meaningful result, all training procedures were using a hyperparameter framework. The used grid search strategy was thereby performed in parallel. However, the training and testing procedure of one single trial was done on one single GPU. Due to these limited computational resources, the input size must be limited as well. Resize, crop and pad operations are much more straightforward in the image domain than in the frequency domain. Because of that, the preprocessing steps of the input are applied in the image domain. Then, the data is transformed into frequency space using the 2-dimensional discrete Fourier transformation applied to the real part of the data. The used transformation omits the negative frequencies in the last dimension since the signal is Hermitian-symmetric. However, this procedure may not be applicable for raw k-space data because of phase shifts.

1.4 Thesis Structure

After the introduction chapter gave the main reasons why it is worth further investigating the usage of frequency data in combination with deep learning and attention-based models, the following literature review helps to situate this work within existing research. Thus, the traditional brain segmentation techniques and their evolution towards ML and DL based methods are presented. Insights will also be provided into current research on attention-based networks for medical use cases and on networks that operate on data in the frequency domain.
The next chapter contains the theoretical foundations on which this thesis is based. It is about the Transformer architecture and its further developments as well as about the characteristics of the attention operation in frequency domain representation.
In the methodology chapter, the methods and materials used to address the questions in

this thesis are explained and justified. Also described here are the specific experiments that eventually produce results. In the results chapter, these results are presented and discussed. The conclusion chapter summarizes the collected results and highlights key findings. Furthermore, a critical analysis of the present work and a short outlook on possible future research is given.

Chapter 2

Literature Review

2.1 Approaches for Brain Segmentation

There is a very broad landscape of image segmentation algorithms, which take different approaches. Over the years, many researchers have variously attempted to categorize these algorithms [Har+85][Pal+93][Yad+22]. In the following, the classification according to Rajapakse et al. is used [Raj+00, p.1]. Here, the segmentation techniques are divided into classical, statistical, fuzzy, and neural network techniques. Classical techniques include edge- and region-based methods, which will be explained in more detail below.
Behind the statistical approach is a model that describes the segmentation as a conditional probability. More precisely, the segmentation S is searched for, which has the highest probability given image I. Using a statistical model such as Markov random fields (MRFs), segmentation based on selected image features are thus created using Maximum A Posteriori (MAP) method [Ant+22].
Fuzzy image segmentation methods are based, as the name suggests, on fuzzy logic [Zad94].

2.1.1 Classical Approaches

The following explanations of the classical approaches are based on the work of Fawzi et al. and Rogowska et al.
Region segmentation methods divide the image into enclosed areas that share pre-defined characteristics. By doing so, these created region exhibit homogeneity with respect to certain criteria, such as intensity. Thresholding, as a simple example of such methods, defines a value according to which all voxels are divided into two groups. Thus, voxels

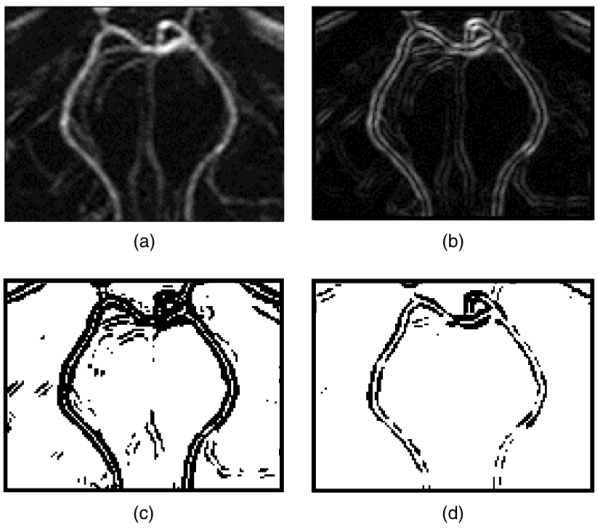

Figure 2.1: Example of segmented blood vessels using edge detection and thresholding. (a) Original image, (b) edge magnitude image obtained by a Sobel operator, (c) edge image after a low threshold applied, (d) edge image after a higher threshold [Rog+09, p.79].

whose value is smaller than the defined value are divided into one group, and all others into the second group. In addition to this so-called global thresholding, there are also other variants. Some are based on local features such as the average gray values within a sub-region in the image. In 1979, Otsu demonstrated a method to determine the threshold that optimally distinguishes an object from the background [Ots79]. Thresholding methods are usually very simple in design and therefore have low computational costs. However, the disadvantage of these algorithms is that they give poor results when regions of a label are inhomogeneous or affected by noise. In addition, the segmentation depends on the pre-defined threshold. In addition to thresholding, clustering algorithms and region growing algorithms are part of region segmentation methods as well [Faw+21][Rog+09].
In contrast to region segmentation techniques, edge segmentation techniques do not work with intensity values per se but with their changes. The intensity changes are determined with the help of voxel intensity gradients. Gradient operators such as the Sobel operator or Robert's operator are used for this purpose. These operators use convolution operations to generate edge magnitude images. Thresholding can then be applied to this if needed (see Figure 2.1). The obtained edges are merged so that whole objects can be segmented. Like region segmentation techniques, edge-based techniques are implemented simply and have low computational costs. Unfortunately, the detected edges often do not reliably enclose the resulting regions, so further processing steps such as edge merging are necessary. This

often makes this approach computationally expensive in the end. Moreover, edge-based segmentation is also sensitive to noise [Faw+21][Rog+09].
Besides these classical methods just mentioned, there are also atlas-based segmentation algorithms. Atlas-based segmentation uses template data of already segmented brains to segment other brain data. The manually segmented dataset (called atlas) contains the corresponding label for each voxel. The goal is to find the optimal non-linear spatial wrapping transformation that maps the values of the atlas to the brain data to be segmented. Thus, the segmentation problem is transformed into a registration problem. In an approach presented by Collins et al., this happens in three steps: First, those features are extracted from the brain to be segmented which have already been extracted from the atlas. With this feature, the algorithm is independent of the atlas used since it can be easily exchanged. For feature extraction the image becomes blurred and convolution is used. This is followed by the next two steps: linear and non-linear registration. The feature extraction and both registrations are performed in a hierarchical iterative manner. This is done by first registering the image with a high degree of blur and then registering it stepwise with less blurred versions of that image. The reason for this is to avoid local minima in the objective function. The linear registration ensures atlas and brains to be segmented globally match through translation, rotation, and scaling. The fine adjustment regarding morphological differences in the anatomy can be taken into account using non-linear registration. For this purpose, local rather than global deformations are applied. Thus, the selected distance between the atlas and the brain to be segmented is gradually reduced. This technique can be used to classify the different types of tissue in the brain. There are also extensions with which it is possible to segment tumors in this way [Bau+10][Col+95][Rog+09].
ML techniques gained popularity as they are more robust to noise and intensity inhomogeneities. For example, constraints are added to the optimization task to ensure that unlikely segmentations (e.g. regions consisting of one voxel) do not occur. Moreover, they are able to combine different feature representations of multiple MR modalities. Common algorithms are K-means, MRF, random forest (RF) or Gaussian mixture models (GMMs) [Faw+21][Rog+09].

2.1.2 Neural Network Techniques

The difficulty with these traditional methods is that features must be manually selected or created. Thus, the performance is strongly dependent on the skill of the user and requires domain knowledge. DL methods can extract features from the input themselves. These

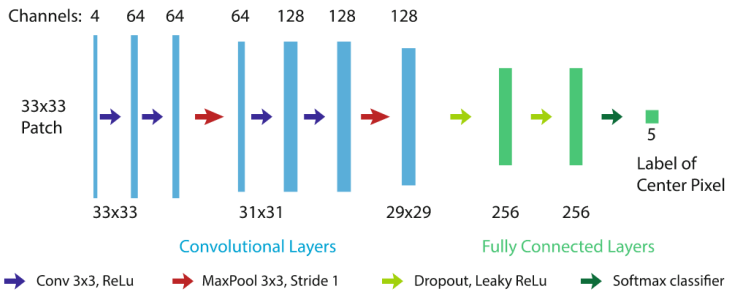

Figure 2.2: Network architecture of the single-path CNN presented by Randhawa et al. [Ran+17, p.68].

features can be highly complex, resulting in more robust learning. The models exclusively learn how to find underlying patterns using examples in the form of a dataset. No heuristics or handmade rules are implemented. A large amount of training data is often required, more data usually also increase the performance of the DL model. Especially in the clinical environment, however, it is often cumbersome to generate suitable data. One of the most widely used brain segmentation datasets is the BraTS dataset [Men+15], which is public and comes with an annual brain tumor segmentation challenge. CNNs are particularly suitable for vision tasks and deliver promising segmentation results. The concept of shared weights enables CNNs to have more layers and to process the sometimes high-dimensional input in its full resolution. CNNs are nowadays the standard when it comes to visual tasks like object detection, registration or segmentation in medical imaging. A distinction is made between networks that can process 2D or 3D inputs. Probably the most significant CNN for segmenting medical data is the U-Net [Faw+21][Seg+20][Bal22].

The ability to maintain spatial neighborhood relationships between individual pixels using a kernel makes CNNs particularly powerful. Due to pooling operations that take place between individual convolution layers, noise is not as significant, making training more robust. Also, the concept of dropouts, makes CNNs less prone to overfitting. Judging by the top CNN models in the BraTS challenge of recent years, four different CNN architectures predominantly stand out. These are single path, multi path, cascaded and U-Net CNNs. Single path and multi path CNNs were the architectures that delivered the best performance for brain segmentation tasks, especially until 2016. Single path CNNs, as the name suggests, have one path along which the data is processed. The path consists of several alternating convolution, pooling, activation, and dropout layers.

An example is the 2D CNN presented by Randhawa et al. which consists of 8 layers and can handle $4 \times 33 \times 33$ large inputs (see Figure 2.2) [Bal22]. To segment an MRI, the image

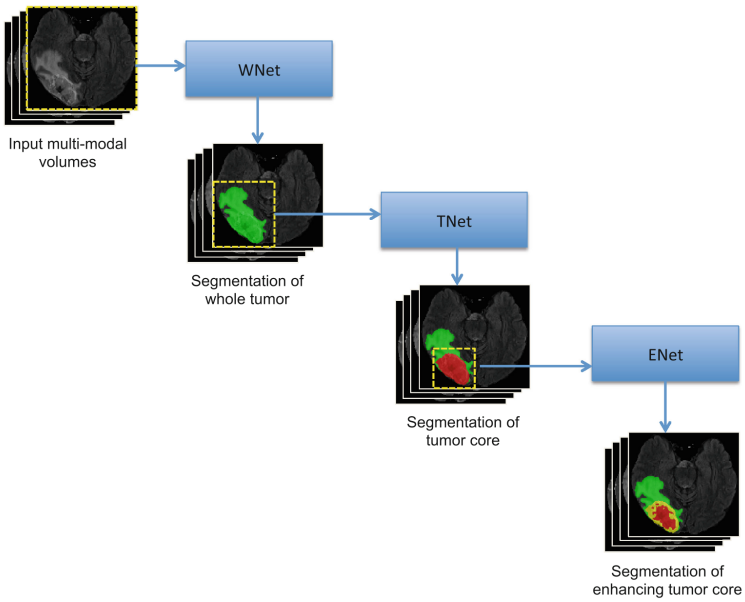

Figure 2.3: Network architecture of the cascaded anisotropic CNN by Wang et al. [Wan+18, p.3].

must be divided into patches of 33×33, with the center pixel in each patch classified by the network. Along the second dimension, differently weighted MRIs (e.g. T1, T2) can be used. Using 3×3 convolutions and max pooling operations as well as dropout layers, a bottleneck is created which results in the classification of a single pixel.[Ran+17]
Single-path CNNs have few parameters and are therefore fast, but this also limits their capacity. Multi-path CNNs consist of multiple processing paths that extract features at different scales. This ensures that local features such as edges or textures and global features such as objects as components of the image are captured. The output of each path is finally concatenated and fed into a fully connected layer. This is followed by a classification layer if the fully connected layer does not already act as a classifier [Ran+17].
Cascaded segmentation CNNs are hierarchical, with each level segmenting a sub-region of the level above it. The advantage of this approach is that multiple specialized networks can be used instead of one that must segment the entire input. They also work better with imbalanced problems such as segmenting anomalies [Ran+17].
For example, the cascaded anisotropic CNN presented by Wang et al. uses three different networks for brain tumor segmentation (see Figure 2.3). Each of them creates a binary mask and using this the input is masked and fed to the next network. The WNet segments the whole tumor, the TNet segments the tumor core based on the tumor region and the ENet segments the enhancing tumor core. Within these sub-networks, techniques such as

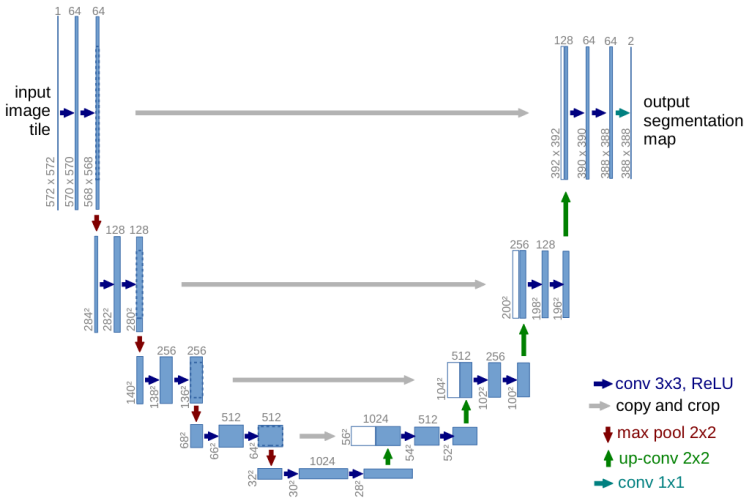

Figure 2.4: Network architecture of U-net by Ronneberger et al. [Ron+15, p.2].

dilated convolution and residual connections are applied [Wan+18].

Especially U-Net CNNs are popular lately. The name goes back to the U-Net model [Ron+15] with the same name. This consists of two interconnected paths, the first of which uses down-sampling techniques to spatially reduce the input. This path acts as an encoder. The size of the feature maps is increased during this process. Once the feature maps are transformed into the final latent space, the second path acts as a decoder and expands the feature maps back to the initial spatial size. The down-sampling and up-sampling are done by convolutions, non-linear activation functions (mostly ReLU functions) and max-pooling. Most U-Net CNNs are fully convolutional-based networks. This means that they do not use fully connected layers for the final segmentation output, but also convolution layers. In the case of the original U-Net, a 1×1 convolution is used over the feature maps, which eventually generates the segmentation mask. Another important feature of U-Net CNNs is skip connections at different spatial levels between the two paths. These allow the decoder to capture spatial information of different scales. As illustrated in Figure 2.4, the architecture is often described as a U. This architecture formed a model for many subsequent models. Thus, variants such as 3D U-net architectures or the nnU-Net were created. The latter is able to adapt itself to new data sets by determining parameters such as network topology and batch size based on the characteristics of the input. It also supports 2D and 3D inputs as well as a cascading approach. Thus, it is very versatile and still delivers state-of-the-art performance among convolution-based networks [Ise+21].

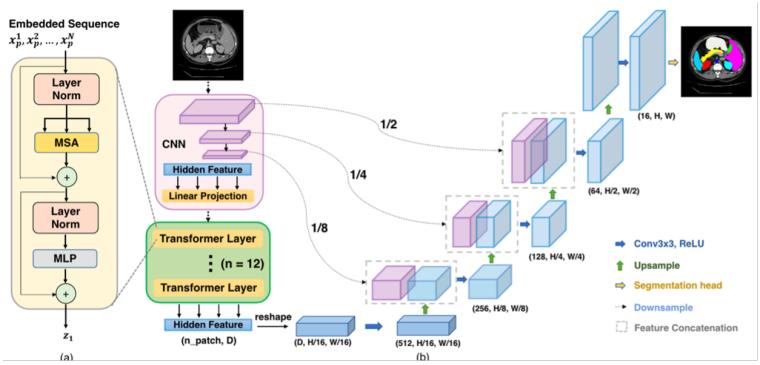

Figure 2.5: Architecture of the TransUNet [Che+21].

2.2 Attention-based Networks in Medical Imaging

Hierarchical vision transformers dominate the medical image segmentation [Che+21][Hat+22] [Cao+23]. In contrast to the original vision Transformer, these learn feature representations of different dimensions. Some important representatives will now be briefly presented.
TransUNet takes the architecture of the successful U-Net and extends it with Transformer components. In this way, the strengths of convolution and attention are to be combined. As shown in Figure 2.5, the model uses an encoder which consists of a CNN and a Transformer which uses self-attention. The CNN extracts feature maps which become tokenized into vectorized patches x_p. Using a trainable linear projection, these patches get embedded into D-dimensional latent space. Also, a positional encoding is added. This approach comes from the vision transformer. With the embedded sequence, a Transformer encoder then learns the hidden features. Just like the original Transformer encoder, the one used here consists of L layers, which include a multi-head self-attention (MSA), a MLP block, layer normalization and residual connections. With this encoder concept, the poor ability of CNNs to capture long-range relations is compensated by transformers. Transformers are excellent at capturing global context due to their attention operation. It is expected that therefore structures that show a strong variance between patients will be segmented more reliably [Sch+19]. The decoder consists of multiple upsampling steps which are called the cascaded upsampler (CUP). In addition to the hidden feature representation of the transformer, the CNN feature maps of different resolutions are also used in the decoding path. Before the CUP is used, the hidden feature representation is reshaped to the shape of $\frac{H}{P} \times \frac{W}{P} \times D$ where $H \times W$ is the original image shape and P the patch dimension. Each cascade stage consists of a 3×3 convolution with ReLU activation function and 2 upsampling operations. In addition, the already mentioned feature maps from the

encoder are concatenated in the CNN with their respective resolutions. Finally, the original resolution is reached. The TransUNet was evaluated using the Multi Atlas Labeling Beyond The Cranial Vault (BTCV) dataset [Lan+15] and the Automated cardiac diagnosis challenge dataset [Ber+18]. On both data sets, the TransUNet was able to outperform previous state-of-the-art models such as the U-Net or AttnUNet [Sch+19][Che+21].
Another model that combines Transformer and convolution is the UNETR [Hat+22]. The UNETR architecture is very similar to that of TransUNet. Here, the U-shape of the U-Net is also adopted. However, UNETR claims to be able to segment 3D inputs directly. In addition, this approach differs in that no CNN is used for feature extraction. Instead, only a stack of transformers is used as an encoder (see Figure 2.6). As with TransUNet, the input is embedded in a 1D sequence and then projected linearly into a latent space, in the spirit of the vision transformer. The difference here, however, is that the patches into which the input image is divided are 3-dimensional. Using several cascading transformer blocks, the hidden features are learned. The representations between the individual blocks are connected to the decoder using skip connections. Using a varying number of blocks consisting of a $2 \times 2 \times 2$ deconvolution, a $3 \times 3 \times 3$ convolution, batch normalization and ReLU activation layers, different resolutions of the extracted features are made available to the decoder. Even the raw input is connected to the decoder via skip connections and two blocks of $3 \times 3 \times 3$ convolution, batch normalization and ReLU activation layer. Similar to the TransUNet, the hidden representation is upsampled by a cascade in the decoder. However, only $3 \times 3 \times 3$ convolutions and $2 \times 2 \times$ deconvolutions with batch normalization and ReLU are used for this. In the end, the output is reshaped to the original resolution using a $1 \times 1 \times 1$ convolution. UNETR was also tested on the BTCV dataset as well as the MSD dataset [Sim+19]. On the BTCV dataset, UNETR outperforms all selected state-of-the-art models in terms of average dice score and most segmentation class-wise dice scores. On the MSD data set, UNETR delivers the best performance compared to the available models in all aspects. Even the TransUNet performs slightly worse [Hat+22].

2.3 Networks working with Frequency Representations

Due to the convolution theorem, fast Fourier transforms (FFTs) are used in CNNs to speed up calculations [Gol+20] or to reduce the input data size of networks [Xu+20].
In 2022, Lee-Thorps et. al. were able to show that the self-attention layers in the Trans-

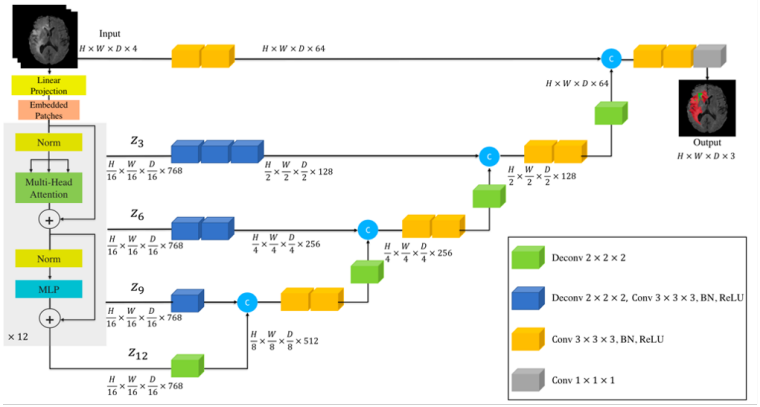

Figure 2.6: Architecture of the UNETR [Hat+22].

former encoder can be replaced by simpler mechanisms, with nearly the same performance on NLP problems [Lee+22]. According to this, attention does not have to be decisive for the success of the Transformer in NLP tasks. It is only important that a hidden representation mixing mechanism is available. In the model presented here, called FNet, attention layers are replaced by Fourier layers. The Fourier transform acts as a mechanism for mixing tokens since single values are related to the whole input sequence in the frequency representation. Accordingly, the architecture is identical to the Transformer encoder except for the replacement of the MSA layer by a Fourier transform. After N encoder blocks follow a dense linear layer and an output projection layer. The Fourier layer performs your 2D discrete Fourier transform (DFT) along the sequence length and the hidden dimension. The imaginary part of the resulting complex number is discarded. The authors intuitively describe the concatenation of multiple FNet encoder blocks as alternating multiplication and convolution, due to Fourier transforms with subsequent feed-forward layers. Measured on the GLUE benchmark [Wan+19], FNet achieves 92% of BERT performance while being significantly faster in training. In the "Large" BART configuration, FNet even achieves 97% of BART's performance [Lee+22].

Frequency representations are not only used in NLP, but also in time series forcasting [Zho+22][Wu+22]. The FEDformer is presented below as an example. In the area of long-term time series forecasting, the quadratic complexity of the Transformers in terms of input length also causes problems. To make the use of the Transformer for long time series feasible, similar to computer vision, techniques like patching or types of local attention like sparse attention were used [Li+20][Nie+23]. All these techniques have in common that the self-attention layers are not applied over the entire input but only a part of it.

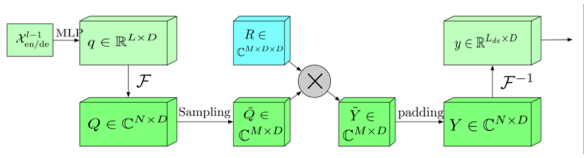

Figure 2.7: Frequency Enhanced Block with Fourier transform (FEB-f) structure [Zho+22].

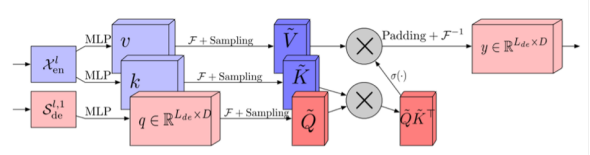

Figure 2.8: Frequency Enhanced Attention with Fourier transform (FEA-f) structure [Zho+22].

The problem here, however, is that global characteristics and statistics can thus not be captured. Zhou et al. therefore propose to use the Fourier transform in combination with self-attention. Hence, so-called frequency enhanced blocks with Fourier transform (FEB-f) are introduced in the FEDformer architecture (see Figure 2.7). In these blocks, the input is first linearly projected before being transformed into the frequency domain using FFT. From this frequency representation, only a few frequency components are randomly sampled. Random selection of frequency components ensures that both high-frequency and low-frequency components are retained. This allows for capturing trend changes while avoiding an excessive amount of noise. The undersampled output $\tilde{Q}$ is then multiplied by a parameterized kernel R in the following way: $Y_{m,d_o} = \sum_{d_i=0}^{D} Q_{m,d_i} \cdot R_{d_i,d_o,m}$, where $d_i = 1, 2, ..., D$ is the input channel and $d_o = 1, 2, ..., D$ the output channel. The product is then zero-padded so that it has the original shape of the input. At the end, the sequence is transformed back into the time domain using inverse Fourier. Using the Frequency enhanced attention blocks also presented, self-attention is applied in the frequency domain. As can be seen in Figure 2.8, the structure of these is very similar to the self-attention layers from the Transformer encoder. Here the queries, keys and values, which serve for the computation of the attention as input, are Fourier transformed before. As with the frequency enhanced blocks, the frequency representations are undersampled. Either softmax or tanh is used as the activation function. The result of this attention calculated in frequency domain is then also padded to obtain the initial shape and finally transformed back to the time domain. With this architecture and a wavelet-based variant, the FEDformer achieved state-of-the-art performance on popular datasets such as the ETTm2 dataset [Zho+21] or the Electricity dataset [Tri15]. In fact, it outperforms the Autoformer model [Wu+22], which had the best performance in all these benchmarks so far [Zho+22].

Chapter 3

Theoretical Foundations

3.1 Overview of Transformer Architecture

The attention mechanism is a way to learn features by focusing differently on certain parts of the input. The widely used self-attention does this by relating all values of the input to each other. The mechanism is applied only to itself, so to speak. The Transformer architecture presented by Vaswani et al. uses self-attention to translate text [Vas+17]. What is innovative about this is that it relies entirely on self-attention instead of mechanisms such as recurrence or convolution. In this way, dependencies between input values can be mapped independently of their distance. One of the main limitations of recurrent neural networks (RNNs) that Transformers want to address is the sequential computation of the hidden state, which makes it difficult to parallelize the process. The Transformer architecture consists of an encoder, which encodes the input once, and a decoder. The decoder generates the output autoregressively, using the encoder output and the decoder output from the previous step at each step.

The input is first projected linearly into an embedding dimension of fixed size. This projection is learned. Then the embedded input is concatenated with a positional encoding. This is important because the self-attention operation does not by itself encode the sequence order. Therefore, the relative positions of the input values are added explicitly. The transformer uses sine and cosine functions with different frequencies for this purpose. The encoder consists of 6 identical blocks, which are MSA layers, feed forward layers, and normalization layers. Around the MSA layer and the feed forward layer there are residual connections. The decoder is almost identical. It has an additional masked MSA layer, which is also surrounded by a residual connection with subsequent normalization. This

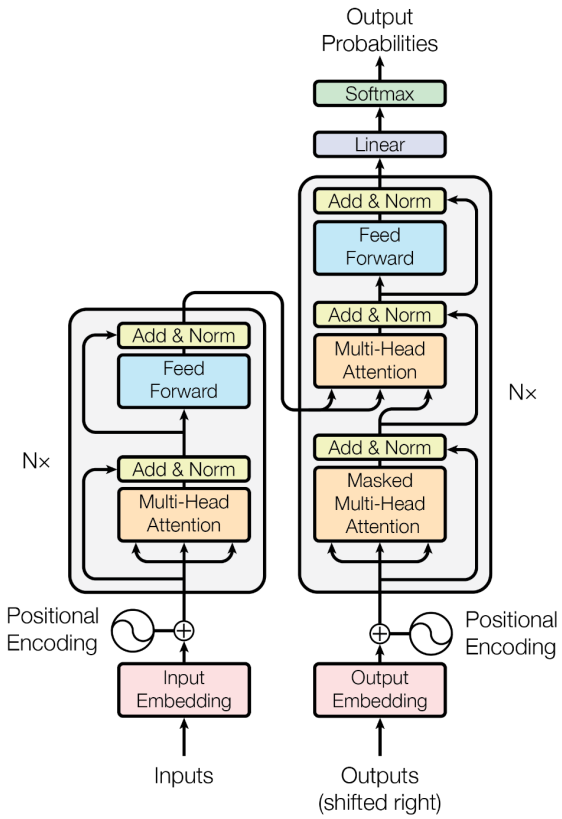

Figure 3.1: Architecture of the Transformer [Vas+17].

masked MSA layer takes as input the output from the previous steps or the ground truth output in case of training. This output was embedded identically to the encoder input and concatenated with positional encoding. The paper calls it attention mechanism "Scaled Dot-Product Attention". It is defined as follows:

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

where Q is the query matrix, K the key matrix and V the value matrix. Before calculating the attention, the queries, keys and values are computed. This is done by multiplying the embedded input using three different learned weight matrices. Then, the attention scores matrix A is calculated using the Query Matrix and the Key Matrix. This process is similar to information retrieval. A query can be understood as a request and the keys as available information. The scaled dot-product acts as a similarity measure between the two vectors. The closer the query is to the available keys, the higher the weighting in the attention scores. [Bra+23] By multiplying the inputs by the attention scores, unimportant parts of

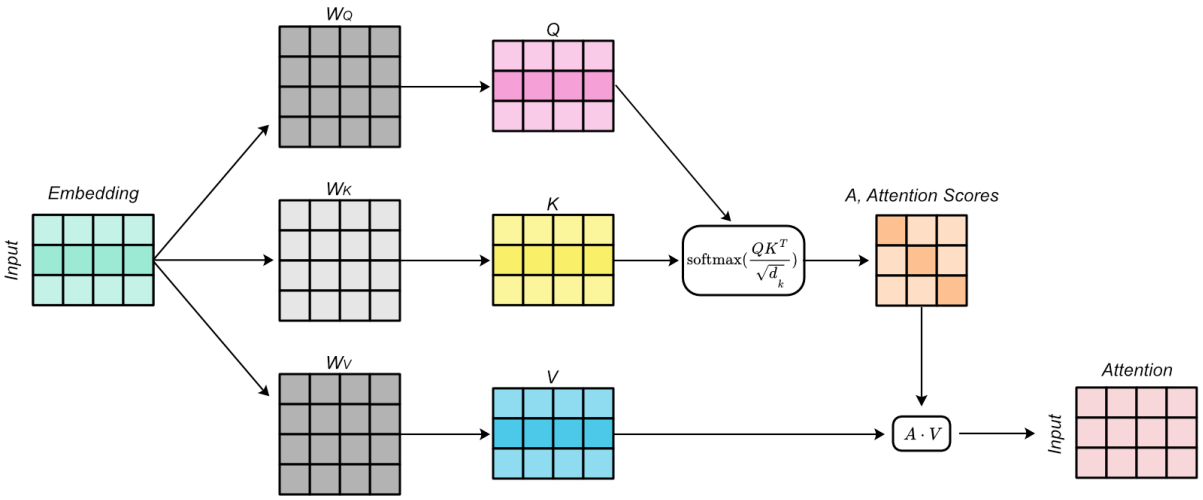

Figure 3.2: Scaled Dot-Product Attention (own illustration).

the input are suppressed and important ones are retained. Thus, the attention of the model is directed. The entire process is illustrated in detail in Figure 3.2. The Transformer uses MSA, which means that Q, K and V are created not only once by linear projection but N times. Each time, different weight matrices are used for this purpose. The variable N is called heads. In the Transformer paper 8 heads are used. Thus, the model is able to efficiently attend to information at different positions, which is in different representations. The linear layer at the end of the decoder projects the feature representation into word scores. Using the softmax function, these scores are converted into the probability of how likely each word in the vocabulary comes at the current location in the sentence [Vas+17].

Modern architectures such as BERT [Dev+19] or GPT [Rad+18] are all derived from the Transformer. BERT is based only on the encoder of the Transformer. This means that it is not autoregressive, but provides the complete output in one step. GPT, in contrast, is based on the decoder of the transformer and is consequently also autoregressive.

3.2 Attention in Frequency Domain Representation

For this work, two Fourier properties are particularly important. One is the convolution theorem. The convolution theorem states that if two functions $f(x)$ and $g(x)$ each have a Fourier transform $F(x)$ and $G(x)$, the convolution of these two functions is equivalent to the

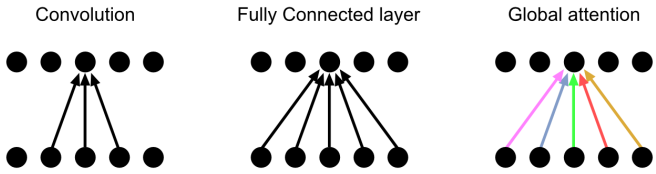

Figure 3.3: Illustration of the different operation of convolution, fully connected layers and attention [Ada+20].

pointwise product of their Fourier transforms. In an equation, it is defined like this [Bra00]:

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\} \tag{3.2}$$

The second property used is that each value of the function $f(x)$ is related to all values of the Fourier transform $F(x)$. The self-attention mechanism weights the relevance of a value of an input sequence on the basis of the entire sequence. There is no kind of receptive field as there is with CNNs. The advantage of this is that long-range relationships within the input sequence can be modeled effectively. However, CNNs introduced receptive fields because they allow better extraction of local features and lower computational complexity. Thus, it was possible to build even deeper networks and achieve promising performance. Intuitively, this also makes sense, especially for segmentation tasks. Once the location of a target region has been identified on an input, it is only a matter of distinguishing the region from the background at the local level. Values that are spatially far away no longer need to be considered. Of course, it is also plausible that as long as local features can be reliably extracted, the more of the input sequence that is used to learn the feature representation, the better the performance. This has already been empirically confirmed by Araujo et al. who could show a logarithmic correlation between the size of the receptive field and the classification accuracy [Ara+19]. However, as CNNs shows, it is simply not necessary to directly relate each value in the input but is unnecessarily complex. For this reason, using self-attention on images without further modification is not optimal.
However, this is different if the input is in the frequency domain. Since each of the original input values is now connected to all frequency components, it makes sense to always use all Fourier-transformed input values. Local problems are thus always globally represented. Moreover, intuitively, this could also create a form of convolution operation. Self-attention can be viewed as an MLP whose weights vary depending on the input. Thus, a multiplication

in the frequency domain takes place, which is equivalent to a convolution in time or pixel domain. Another reason why attention in the frequency domain instead of in the time domain or pixel domain is advantageous is related to the absence of inductive bias like it can be found in CNNs. The inductive bias in CNNs is the receptive field to effectively model the neighborhood relationships between spatially close pixels and the principle of shared weights to achieve translation invariance. With this bias, a priori knowledge about the nature of images in the pixel domain was added to the model. The goal is to achieve a faster and more stable convergence. Transformer models are more generic and do not have this bias. This sounds like a disadvantage at first but turns out to be an advantage for input data in the frequency domain. Indeed, the inductive biases do not make sense on Fourier-transformed data, since there is no such neighborhood relationship in the frequency domain. Also, a translation invariance is rather a hindrance than a benefit. In conclusion, attention in the frequency representation is very suitable.

Chapter 4

Methodology

4.1 Data

To obtain the most meaningful results, all models studied in this work were evaluated using two tasks. Multi-parametric magnetic resonance imaging (mpMRI) for de novo glioblastoma (GBM) patients from the University of Pennsylvania Health System (UPENN-GBM) [Bak+22] was used for the skull stripping task. OASIS-1 dataset [Mar+07] was used for the evaluation of the brain tissue segmentation task. Both datasets are freely available to ensure the reproducibility of this work.

The UPENN-GBM dataset contains T1-weighted cross-sectional MRI scans of the brain from 630 subjects. Brain data were collected at the University of Pennsylvania Health System during routine clinical radiology examinations. All subjects included in this data set were diagnosed with de novo glioblastoma. The magnetic field strength was 3T during the scan. The raw scan data size was $256 \times 256 \times 196$. The data was resampled to a size of $256 \times 256 \times 256$ and an isotropic voxel resolution at 1 mm. The data was also pre-processed and de-faced. For these brain scans, there are skull stripped versions generated using an in-house deep learning model. All brain masks were manually reviewed and accepted or corrected if necessary. Follow-up scans are excluded from this work.

The OASIS-1 dataset consists of cross-sectional T1-weighted brain MRI scans of 416 subjects ranging in age from 18 to 96 years. The matrix size of the raw scan data was $256 \times 256 \times 128$ with a voxel resolution of 1 mm $\times$ 1 mm $\times$ 1.25 mm. This data is resampled to a matrix size of $256 \times 256 \times 256$ and isotropic voxels at 1 mm resolution. Included subjects who were diagnosed with early-stage Alzheimer's Disease. All data have been anonymized. Only the FreeSurfer output of this dataset was used. Unfortunately, only 407 of the 416 subjects

can be used for this work, because due to technical problems on the OASIS Brain dataset servers, part of the dataset could not be downloaded at the time of writing. The following transformations were applied to the dataset: All subject data except for the brain mask and tissue segmentation created by FreeSurfer were removed. Follow-up scans were also deleted. Finally, the two FreeSurfer outputs for each subject were converted to NIfTI format.
As part of the pre-processing stage to use these two datasets, various transformations were equally applied. The samples consist of brain MRI data and skull stripped brain MRI data or skull stripped brain MRI data and brain tissue segmented brain MRI data, depending on the task. All samples are reordered to canonical (RAS+) orientation and sampled into the same physical space with an isotropic voxel size of 3 mm. Then all samples are cropped to a size of $64 \times 64 \times 64$ and z-normalized. After preprocessing, the following augmentation transformations are applied: random affine transformation, random contrast change, random Gaussian noise, random MRI motion artifact, and random MRI bias field artifacts. The random affine transformation is always used. All others are applied with a probability of 10%. In the last stage, the samples are transformed into their target domain. Depending on the experiment, the samples or parts of the samples (input or label) are transformed into the frequency domain using FFT. For both segmentation tasks, the training dataset consists of 80% of the total data, the validation set consists of 10%, and the test set consists of the remaining 10%. Accordingly, for the skull stripping task, the training dataset consists of 504 samples, the validation dataset consists of 63 samples and the test dataset consists of 63 samples. For the brain tissue segmentation task, the training dataset consists of 326 samples, the validation dataset consists of 41 samples and the test dataset consists of 40 samples.

4.2 Implementation Details

In this work, all implementations and analysis were performed using Python (version 3.10.12), PyTorch (version 2.0.1), TorchIO (version 0.18.92), Ray (version 2.5.1) and PyTorch Lightning (version 2.0.5). The data is loaded using the PyTorch DataLoader and the TorchIO SubjectsDataset. This combination allows efficient loading and online preprocessing as well as data augmentation. For transforming the samples into the frequency domain the 2D real FFT PyTorch implementation was used. This function assumes that the result of the Fourier transformation is Hermitian-symmetric. In the case of the reconstructed MRI data used here, this is true. The 2D real FFT uses this symmetry and omits the

redundant values in the output. This will almost halve the input (only almost because all frequencies must be kept up to the Nyquist frequency). Then, the complex values of the resulting tensors are split up into real and imaginary parts, which are stored separately in a new dimension. The size of the samples thereby remains almost the same. The final shape of all input and label tensors in a sample is the same regardless of the segmentation task or target input and label domain. It can be described as (b, c, v, x, y, z), where b is the batch, c is the class, v is the part of the complex number and the remaining three dimensions are the spatial dimensions of the MRI data. The class dimension results from the fact that all samples get one-hot encoded. For the input, this dimension has always a length of 1. The length of the class dimension varies for the labels depending on the segmentation task (e.g., for brain segmentation, the length is 2). The vector dimension holds the real and imaginary parts of complex numbers. However, it has a length of 1 if the corresponding tensor is in the image domain. While this design may seem inflexible and verbose, it has the advantage that the number of dimensions is always predictable, making the implementation of subsequent transformations easier and more efficient. The FreeSurfer brain tissue segmentation in the OASIS-1 dataset includes a large number of classes and is very detailed. To simplify the segmentation problem to some degree, only 6 classes (excluding the background) are used in this thesis. To achieve this, label mapping takes place when loading the samples from the OASIS-1 dataset. This is done online in order to be able to keep the data set in its original state on the hard disk as much as possible. The corresponding mapping is documented in Table 4.1. In the OASIS-1 dataset, the segmentation classes are also imbalanced, which can lead to unwanted effects in the training of the models. To solve this problem, the individual classes are weighted in the calculation of the loss. For the case where frequency data is used, a weighted MSE loss has been implemented. By slicing along the class dimension of the prediction tensor, each class can get weighted with a particular value. For data in the pixel domain, the cross-entropy loss implementation of PyTorch is used, which already offers weights as parameters. To calculate the weight for class i this formula was used:

$$\text{Class Weight}_i = \frac{\text{Total Number of Samples}}{\text{Number of Samples in Class}_i} \tag{4.1}$$

Five DL models were used to answer the research questions: an MLP, a Transformer encoder, the PerceiverIO model, the ResMLP and the nnU-Net. With this selection, attention-based models and non-attention-based models are equally represented. Convolution-based models were not chosen because they are not suitable for inputs in the frequency domain (for more

Custom labels used in this project	**FreeSurfer labels**
0 - Background	Unknown and everything else not mentioned in the next rows
1 – CSF	Left-Lateral-Ventricle, Left-Inf-Lat-Vent, 3rd-Ventricle, 4th-Ventricle, CSF, Left-vessel, Right-Lateral-Ventricle, Right-Inf-Lat-Vent, Right-vessel, 5th-Ventricle
2 – Cortical Gray Matter	Left-Cerebral-Cortex, Right-Cerebral-Cortex
3 – White Matter	Left-Cerebral-White-Matter, Right-Cerebral-White-Matter, Left-WM-hypointensities, Right-WM-hypointensities
4 – Deep Gray Matter	Left-Thalamus, Left-Caudate, Left-Putamen, Left-Pallidum, Left-Hippocampus, Left-Amygdala, Left-Accumbens-area, Left-VentralDC, Right-Thalamus, Right-Caudate, Right-Putamen, Right-Pallidum, Right-Hippocampus, Right-Amygdala, Right-Accumbens-area, Right-VentralDC, Left-non-WM-hypointensities, Right-non-WM-hypointensities
5 – Brain Stem	Brain-Stem, Optic-Chiasm
6 – Cerebellum	Left-Cerebellum-White-Matter, Left-Cerebellum-Cortex, Right-Cerebellum-White-Matter, Right-Cerebellum-Cortex

Table 4.1: Mapping of FreeSurfer segmentation labels to custom labels

details, see section 3.2).

The MLP is the simplest of the four models and is primarily used as a proof-of-concept model. The MLP consists of linear input embedding, N hidden fully connected layers and a linear output embedding. The embeddings are linear layers that project the input into or out of the latent space. After each hidden layer follows a tanh activation layer. The input is reshaped so that it can be fed sagittal into the network to reduce complexity. The length of the latent space dimension M as well as the number of hidden layers N are hyperparameters. The Transformer encoder also has linear embeddings for input and output. Besides that, this model has N encoder components from the Transformer model of Vanswani et al.. The structure is thus very similar to the successful BERT model. The number of encoder blocks and dimension of latent space are again hyperparameters. For

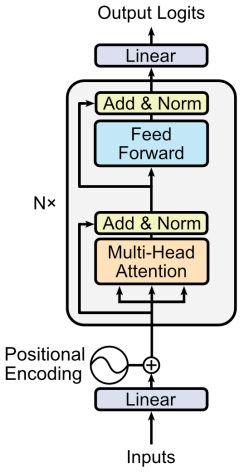

Figure 4.1: Architecture of the Transformer encoder used in this work. Inspired by [Vas+17].

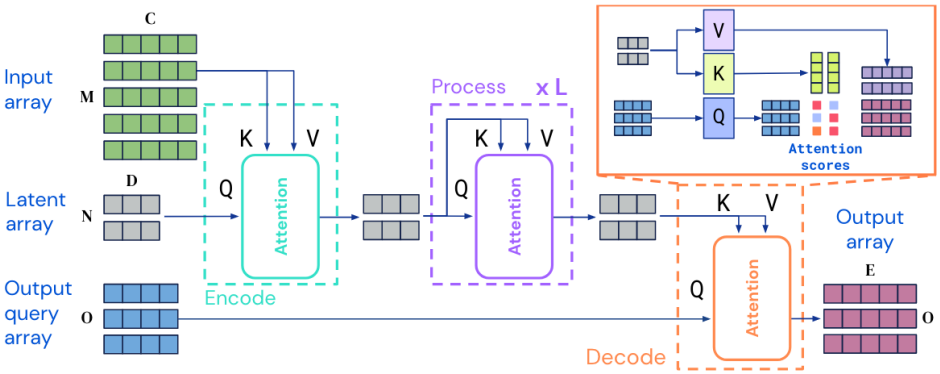

Figure 4.2: Architecture of the PerceiverIO [Jae+22].

the implementation of the Transformer encoder, the corresponding PyTorch class was used. The number of attention heads was set to 2 and the dropout to 0.2. In addition, Fourier position encoding is concatenated with the embedded input if needed. The architecture of this model is shown in Figure 4.1. For the integration of the PerceiverIO model, the corresponding Python module by Krasser and Stumpf was used [Kra+23]. The module offers the possibility to connect the components presented in the PerceiverIO paper with each other in any way. The PerceiverIO implemented here follows the original architecture. The model is characterized by the fact that it uses the cross-attention mechanism, which allows mapping the input into a latent space. This latent space is smaller and therefore the quadratic complexity of the attention can be reduced to a linear one.

The ResMLP model was implemented following the documentation from the original paper. The structure of the ResMLP is similar to that of the vision transformer, but it does not use attention at all. Attention layers were replaced by linear layers. Furthermore, normalizations like BatchNorm or LayerNorm were removed. Instead, simple affine transformations are used. [Tou+21] As can be seen in Figure 4.3, a ResMLP layer consists of a cross-patch sublayer and a cross-channel sublayer. The intuition behind this is that linear layers should be applied to all channels independently and then independently to all patches. For this work, the ResMLP was implemented so that the input is not divided into patches and

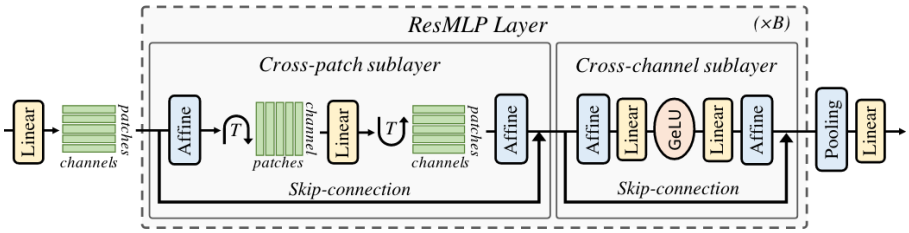

Figure 4.3: Network architecture of ResMLP by Touvron et al [Tou+21, p.2].

embedded using a linear projection. Instead, the sagittal slices of the MRI brain data serve as channels and the remaining dimensions are flattened. Through the alternating cross-patch and cross-channel operations, the individual sagittal slices of the brain MRI interact with each other.

The nnU-Net was retrieved in version 2 from the official GitHub repository. It has not been modified and was run in the default configuration. It comes with its own training and testing procedures, so the two datasets were converted into the format specified by the nnU-Net.

The training procedure for all other models is written in such a way that at startup the target domain for the inputs and labels can be set independently. Therefore, all possible combinations of input and label domains can be specified. If the labels are in the pixel domain, it is a classification problem. If they are transformed into the frequency domain, it is a regression problem. This also means that all DL models must output raw logits instead of probabilities to support both cases. Only when calculating the loss, a Sigmoid layer is used if necessary. Regardless, a Sigmoid layer is always used after calculating the loss to calculate metrics such as the Dice score and to display qualitative segmentation results. At this point, inverse fast Fourier transform (iFFT) is necessarily used to transform the output back to the pixel domain if the target domain of the label was the frequency domain. All quantitative metrics and selected qualitative segmentation results are logged in both training and testing using the tensorboard.

4.3 Evaluation Metrics

The following metrics were used quantitative assessment of the selected model-domain combinations:

- Loss (Binary Cross Entropy, Cross Entropy or Mean squared error)

- Dice score per class
- Average Dice score over all classes
- Recall per class
- Average recall over all classes
- Specificity per class
- Average specificity over all classes

All of these metrics were utilized in this work at the epoch level and using only the test dataset. The values are reported for the epoch that achieved the highest average Dice score. The Dice score is reported since it is the most common reported metric for image segmentation, thus it provides the ability to compare this work with most of the other research. To report the recall and specificity, the torchmetrics (version 0.9.3) library was used. The class-wise Dice score for a prediction y_{pred} and a ground truth y_{true} was implemented following these equations:

$$\text{Intersection}_c = \sum_{v,x,y,z} y_{\text{pred}}[c, v, x, y, z] \cdot y_{\text{true}}[c, v, x, y, z] \tag{4.2}$$

$$\text{Union}_c = \sum_{v,x,y,z} (y_{\text{pred}}[c, v, x, y, z] + y_{\text{true}}[c, v, x, y, z]) \tag{4.3}$$

$$\text{Dice score}_c = \frac{2 \cdot \text{Intersection}_c + \epsilon}{\text{Union}_c + \epsilon} \tag{4.4}$$

where

- c is the segmentation class.
- v is the dimension that holds the real and imaginary part of the number.
- x, y and z are the spatial dimensions of the brain MRI scan data.
- ϵ is a small constant to avoid division by zero.

The average Dice score across all classes is then calculated by taking the average of these individual c Dice scores.
In addition to these quantitative metrics, qualitative segmentation results are also reported.

4.4 Experiments

As mentioned earlier, two tasks are used for evaluation: skull stripping and brain tissue segmentation. The reason for this is firstly to obtain the most meaningful results and secondly, because the skull stripping task may be too simple to see differences between the various approaches. To answer the research questions mentioned at the beginning of this thesis and for better clarity, several experiments are defined.

Experiment 1: In the first experiment the impact of the input and label domains on the performance of different DL models will be investigated. Both segmentation tasks and all implemented models are used for this purpose. The models are trained, validated, and finally tested with three different input-label-domain constellations. Once the input and label domain is the pixel domain, then for both the k-space domain and finally for the input domain k-space and for the label domain pixel. The three cases are referred to as pixel, k-space, and k-space to pixel in this order.

Experiment 2: The second experiment focuses on the question of whether positional encoding is necessary for attention-based models when the input data is in k-space. For this purpose, the two attention-based models (PerceiverIO and Transformer encoder) are used. Using the k-space domain, the two models are trained and evaluated on both datasets once with positional encoding and once without.

Experiment 3: In the last experiment, all results from experiment 1 are additionally compared with the nnU-Net as a state-of-the-art medical segmentation model, which is based on convolution. An unmodified version of the publicly available nnU-Net implementation is used for this purpose. The implementation uses cross-validation, but in this experiment, only one fold is used to be more comparable with the other models. The nnU-Net is trained and evaluated only in the pixel domain.

With the help of the framework ray, hyperparameter tuning was used for all experiments. The hyperparameter space used for each model is recorded in Table 4.2. The ray framework offers different trail schedulers, which stop trails early if they do not look promising. Although the ASHA scheduler was selected for hyperparameter tuning, the grace period was set to 100 for all experiments, which is equal to the maximum number of epochs for training. Thus, no trails are stopped early. Nevertheless, an early stopping criterion has been added, which stops any training process if the dice score does not improve within 10 epochs. For the skull stripping task, the dice score of the target class is monitored and for the tissue segmentation task the average dice score. The models were trained on different high-performance cluster nodes. For the skull stripping task, a node with 64

Intel(R) Xeon(R) CPU E5-2683 v4 @ 2.10GHz processors, each having 16 cores, was used. This node had 264 GB of RAM and four NVIDIA TITAN XP graphics cards with 12 GB of memory each. Whereas, for the brain tissue segmentation task, a second node with 32 Intel(R) Xeon(R) Gold 6234 CPU @ 3.30GHz processors with 8 cores each and a memory of 264GB was used. This second node was equipped with four NVIDIA TITAN RTX GPUs (24 GB of memory each). The only exception is the nnU-Net, which was trained on another separate node. This node was equipped with an AMD EPYC 7262 8 core processor, had 500 GB of RAM and an NVIDIA A100-SXM GPU with 40 GB of memory. However, only one GPU was used for all experiments. All three nodes are running with Ubuntu (Release 20.04).

Model	**Hyperparameter**	**Values/Range**
MLP	Hidden layer	(1, 3, 6)
	Hidden factor	(1, 2)
	Learning rate	(0.01, 0.005)
	Loss	(unweighted, weighted)
	Step size	(300, 600)
	Optimizer	(Lamb, Adam)
ResMLP	Hidden layer	(3, 6)
	Hidden factor	1
	Learning rate	(0.01, 0.005)
	Loss	(unweighted, weighted)
	Step size	(300, 600)
	Optimizer	(Lamb, Adam)
PerceiverIO	Num latents	(512, 1024)
	Num latent channels	(1024, 2048)
	Num cross attention layers	(4, 8)
	Num self-attention layers per block	(3, 6)
	Num self-attention blocks	(4, 8)
	Dropout	(0.0, 0.1, 0.2)
	Learning rate	(0.01, 0.005)
	Loss	(unweighted, weighted)
	Step size	(300, 600)
	Optimizer	(Lamb, Adam)
Transformer	Hidden layer	(2, 6)
	Hidden factor	1
	Learning rate	(0.01, 0.005)
	Loss	(unweighted, weighted)
	Step size	(300, 600)
	Optimizer	(Lamb, Adam)

Table 4.2: Hyperparameter space for different models

Chapter 5

Results

5.1 Quantitative Results

In this section, the quantitative segmentation results of all experiments are presented. The hyperparameter configuration with which the following results were obtained are recorded in the appendix. First, the results of experiment 1 are presented, with Table 5.1 showing the measurements for the skull stripping task. Looking only at the pixel domain, it is noticeable that the Dice scores of almost all models are similar. Only the MLP performs slightly worse, the same applies to recall and specificity. In k-space, MLP delivers worse performance compared to the pixel domain. All other models perform about the same compared to the pixel domain. In the k-space domain, the MLP is also the weakest model. In the k-space-to-pixel domain, the MLP has about the same performance as in the pixel domain. All other models again achieve approximately the same results. In the following three tables, the results of the previously evaluated models are presented again for the 3 different domains. Now, these results are for the brain tissue segmentation task. Table 5.2 shows the results for the pixel domain, Table 5.3 for the k-space and Table 5.4 for the k-space-to-pixel domain.

Domain/Metric	**Pixel**			**K-space**			**K-Space → Pixel**		
	Dice	**Recall**	**Spec.**	**Dice**	**Recall**	**Spec.**	**Dice**	**Recall**	**Spec.**
MLP	0.964	0.960	0.992	0.898	0.888	0.976	0.966	0.968	0.990
ResMLP	**0.978**	0.973	0.995	**0.978**	0.980	0.993	**0.976**	0.977	0.993
PerceiverIO	0.930	0.923	0.984	0.927	0.919	0.984	0.929	0.919	0.985
Transformer encoder	0.971	0.971	0.992	0.971	0.969	0.993	0.972	0.973	0.993

Table 5.1: Performance metrics in pixel domain, k-space and k-space-to-pixel domain for different models on skull stripping.

Domain		Pixel			
Anatomy	Metric	MLP	ResMLP	PerceiverIO	Transformer
CSF	Dice	0.562	**0.868**	0.078	0.686
	Recall	0.816	0.871	0.046	0.870
	Specificity	0.995	0.999	0.999	0.997
Cortical Gray Matter	Dice	0.491	**0.773**	0.263	0.558
	Recall	0.737	0.764	0.192	0.774
	Specificity	0.911	0.985	0.981	0.930
White Matter	Dice	0.602	**0.874**	0.470	0.673
	Recall	0.564	0.884	0.458	0.648
	Specificity	0.982	0.992	0.971	0.984
Deep Gray Matter	Dice	0.645	**0.828**	0.317	0.699
	Recall	0.876	0.834	0.266	0.909
	Specificity	0.993	0.999	0.997	0.995
Brain Stem	Dice	0.506	**0.883**	0.405	0.589
	Recall	0.951	0.885	0.331	0.962
	Specificity	0.995	0.999	0.999	0.996
Cerebellum	Dice	0.623	**0.916**	0.443	0.723
	Recall	0.967	0.926	0.382	0.970
	Specificity	0.980	0.998	0.994	0.987
All	Dice	0.625	**0.876**	0.418	0.698
	Recall	0.829	0.879	0.379	0.864
	Specificity	0.979	0.987	0.926	0.984

Table 5.2: Performance metrics in the pixel domain for different models on brain tissue segmentation.

It can already be seen in Table 5.2 that the results of the models are now significantly different than they were with skull stripping. Looking at the overall values first, it is noticeable that the ResMLP achieves the best performance by far. The ResMLP reaches a Dice score of 0.876. This is followed by the transformer encoder and the MLP, with the transformer encoder being about 12% better than the MLP. The PerceiverIO model brings up the rear, achieving a Dice score of only 0.418. Looking at the individual segmentation classes in more detail, it is noticeable that the ResMLP achieves a Dice score of over 0.8 in each of the classes except cortical gray matter. The other models also had difficulties with the cortical gray matter but additionally with the brain stem. For example, the Dice score for the brain stem class for the Transformer encoder falls below 0.6. With a Dice score of 0.263 and a specificity of 0.981, the PerceiverIO shows that the cortical gray matter was not recognized and segmented as such. Compared with the pixel domain, the results of the ResMLP become worse in k-space. All other models improve, especially the Transformer encoder, which achieves an approximately 13% better dice score. Changing the domain to k-space has further degraded the classification of the cortical gray matter class for the ResMLP. The Transformer has interestingly benefited from this, even if not strongly. Even though the PerceiverIO does not reliably detect cortical gray matter, its performance for this class has improved significantly with the domain change (roughly 1.5 times better). In the k-space-to-pixel domain, the ResMLP has again the best results and even the best results over all domains. The model achieves a dice score of 0.883. The cortical gray matter class, which previously caused problems for ResMLP, has a significantly increased Dice score of 0.801 in the k-space-to-pixel domain. Except for the PerceiverIO, all models benefit from the k-space-to-pixel domain with respect to the segmentation of this class. The MLP has slightly deteriorated compared to k-space, but slightly improved compared to pixel domain. The Transformer encoder also achieves its best performance in the k-space-to-pixel domain with a dice score of 0.861.

The following two tables show the evaluation for experiment 2 and thus whether additional positional encoding is necessary for attention-based networks when the input is in k-space. Table 5.5 shows a comparison of the PerceiverIO and Transformer encode with and without additional positional encoding in the three different domains for the skull stripping task. As can be seen, no significant difference can be identified within the respective models. This applies to all three metrics. The same observation can be made for the brain segmentation task in Table 5.6. All metric values differ only marginally, if at all, between the two strategies.

Domain		K-Space			
Anatomy	Metric	MLP	ResMLP	PerceiverIO	Transformer
CSF	Dice	0.570	**0.844**	0.098	0.795
	Recall	0.469	0.820	0.061	0.738
	Specificity	0.999	0.999	0.999	0.999
Cortical Gray Matter	Dice	0.522	**0.581**	0.395	0.561
	Recall	0.680	0.585	0.512	0.575
	Specificity	0.935	0.970	0.924	0.967
White Matter	Dice	0.612	**0.723**	0.486	0.688
	Recall	0.609	0.707	0.545	0.670
	Specificity	0.977	0.985	0.959	0.984
Deep Gray Matter	Dice	0.668	**0.806**	0.333	0.774
	Recall	0.759	0.825	0.340	0.811
	Specificity	0.996	0.998	0.995	0.998
Brain Stem	Dice	0.752	**0.880**	0.471	0.860
	Recall	0.893	0.861	0.528	0.843
	Specificity	0.999	0.999	0.998	0.999
Cerebellum	Dice	0.745	**0.892**	0.510	0.874
	Recall	0.955	0.900	0.656	0.903
	Specificity	0.989	0.998	0.984	0.998
All	Dice	0.690	**0.815**	0.461	0.790
	Recall	0.757	0.811	0.507	0.788
	Specificity	0.979	0.975	0.960	0.974

Table 5.3: Performance metrics in k-space for different models on brain tissue segmentation.

Domain		K-Space → Pixel			
Anatomy	Metric	MLP	ResMLP	PerceiverIO	Transformer
CSF	Dice	0.645	**0.866**	0.054	0.853
	Recall	0.867	0.876	0.030	0.840
	Specificity	0.996	0.999	0.999	0.999
Cortical Gray Matter	Dice	0.539	**0.801**	0.270	0.761
	Recall	0.756	0.776	0.201	0.774
	Specificity	0.926	0.989	0.980	0.982
White Matter	Dice	0.645	**0.898**	0.464	0.858
	Recall	0.621	0.914	0.451	0.841
	Specificity	0.982	0.993	0.971	0.993
Deep Gray Matter	Dice	0.686	**0.826**	0.323	0.804
	Recall	0.917	0.828	0.277	0.832
	Specificity	0.994	0.999	0.997	0.998
Brain Stem	Dice	0.568	**0.880**	0.410	0.863
	Recall	0.956	0.870	0.339	0.879
	Specificity	0.996	0.999	0.999	0.999
Cerebellum	Dice	0.696	**0.919**	0.466	0.902
	Recall	0.965	0.912	0.421	0.903
	Specificity	0.986	0.999	0.993	0.998
All	Dice	0.676	**0.883**	0.419	0.861
	Recall	0.856	0.881	0.385	0.865
	Specificity	0.983	0.987	0.927	0.986

Table 5.4: Performance metrics the in k-space-to-pixel domain for different models on brain tissue segmentation.

Domain	**K-Space**		
Metric	**Dice**	**Recall**	**Specificity**
PerceiverIO	0.927	0.919	0.984
PerceiverIO (no pos. enc.)	0.930	0.920	0.985
Transformer	0.971	0.969	0.993
Transformer (no pos. enc.)	0.970	0.967	0.993

Table 5.5: Quantitative comparison of attention-based models with and without additional positional encoding on skull stripping in k-space.

Domain	**K-Space**		
Metric	**Avg Dice**	**Avg Recall**	**Avg Specificity**
PerceiverIO	0.461	0.507	0.960
PerceiverIO (no pos. enc.)	0.463	0.491	0.957
Transformer	0.790	0.790	0.974
Transformer (no pos. enc.)	0.790	0.788	0.974

Table 5.6: Quantitative comparison of attention-based models with and without additional positional encoding on brain tissue segmentation in k-space.

Domain	**K-Space**		
Metric	**Dice**	**Recall**	**Specificity**
nnU-Net	0.986	0.987	0.996

Table 5.7: Quantitative results of the nnU-Net in the pixel domain for skull stripping.

The last part of this section presents the results of experiment 3 and with it the results of the nnU-Net. Table 5.7 shows the skull stripping segmentation results of the nnU-Net measured by the selected metrics. The results were obtained in the pixel domain. As can be seen, the nnU-Net provides slightly better segmentations than ResMLP, which was the best of all evaluated models in skull stripping (see Table 5.1). However, the difference is actually marginal. In the diagram below, the results of the nnU-Net can be further classified in comparison to the models evaluated so far. Figure 5.1 shows a quantitative comparison of all models including the nnU-Net as the baseline model on skull stripping. Clearly, the choice of domain has no relevant influence on the Dice score with the exception of the MLP. Neglecting the MLP in k-space, it can be seen how the PerceiverIO provides significantly worse segmentations than the other models. The nnU-Net also delivers excellent results in the brain tissue segmentation task. Table 5.8 shows that it achieves an overall Dice score of 0.956. Relatively slight performance drops were only seen in deep gray matter and cortical gray matter. Figure 5.2 also shows the performance of all models including the nnU-Net for the different domains for the brain tissue segmentation task in a visually clear form. Here it is clear that the nnU-Net has the best performance for brain tissue segmentation. Nevertheless, the Transformer encoder in the k-space-to-pixel domain and the ResMLP provide competitive segmentations. Here it also emerges that for brain tissue segmentation the selection of the domain influences the segmentation results.

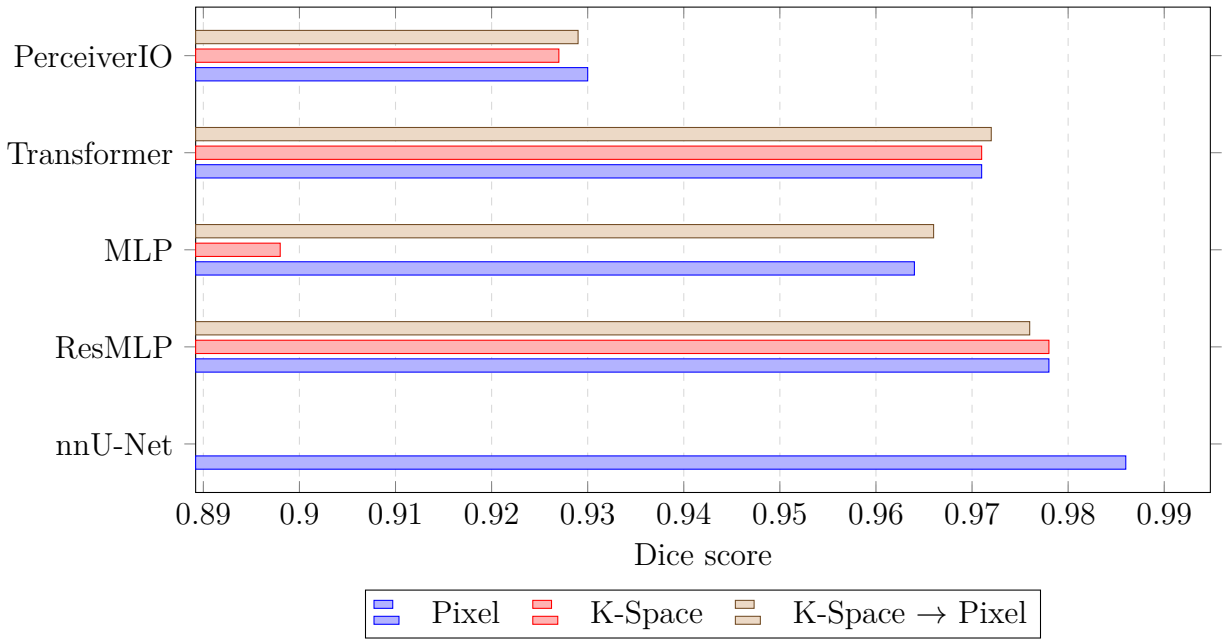

Figure 5.1: Quantitative comparison of all models on skull stripping in various domains.

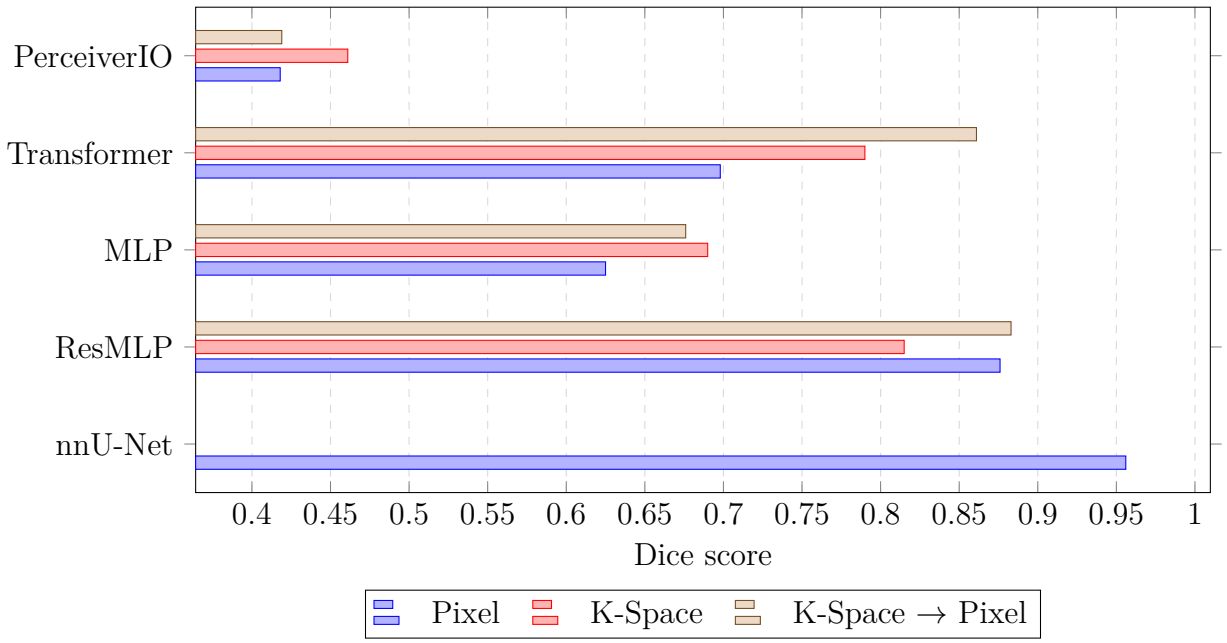

Figure 5.2: Quantitative comparison of all models on brain tissue segmentation in varies domains.

Domain	Pixel	
Anatomy	Metric	nnU-Net
CSF	Dice	0.942
	Recall	0.941
	Specificity	0.999
Cortical Gray Matter	Dice	0.930
	Recall	0.935
	Specificity	0.995
White Matter	Dice	0.970
	Recall	0.967
	Specificity	0.998
Deep Gray Matter	Dice	0.921
	Recall	0.928
	Specificity	0.999
Brain Stem	Dice	0.946
	Recall	0.947
	Specificity	0.999
Cerebellum	Dice	0.965
	Recall	0.968
	Specificity	0.999
All	Dice	0.956
	Recall	0.954
	Specificity	0.996

Table 5.8: Segmentation performance in the brain tissue segmentation task.

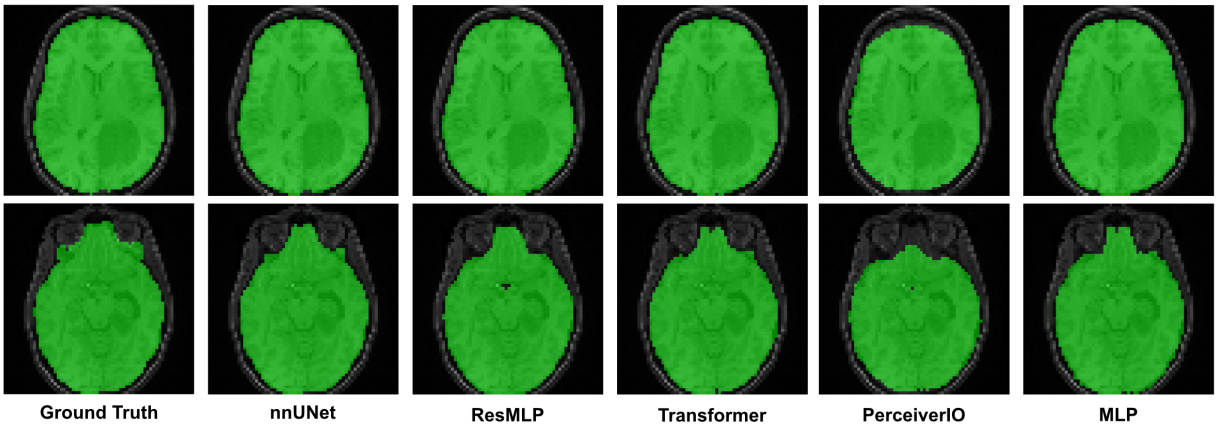

Figure 5.3: Segmentation masks of the different models for skull stripping in the pixel domain.

5.2 Qualitative Results

In this section, the segmentation masks created by the different DL models are visualized and compared. A set of axial slices from an MRI of the brain is always shown, consisting of the middle axial slice and a lightly lower axial slice, moving toward the cervical region. Figure 5.3, Figure 5.4, and Figure 5.5 show the skull stripping segmentation masks of all models in the pixel domain, in k-space, and in the k-space-to-pixel domain, respectively, in that order. With skull stripping segmentation masks in the pixel domain, it is difficult to detect qualitative differences. In Figure 5.3, it is only seen that there are differences in the area of the nose. In addition, the brain region is not completely filled in for the ResMLP and the PerceiverIO. For the k-space case of skull stripping in Figure 5.4, a similar picture emerges. Again, there are differences in the nose area and missing areas in the brain. In the frontal lobe, some areas are not correctly classified by the MLP. In particular, lateral artifacts are also noticeable with the MLP. Also in the k-space-to-pixel domain, besides the different nasal segmentations in all models, there are gaps in the brain region in the ResMLP and PerceiverIO. In this domain, however, the MLP was able to recognize the nose area better than the other domains. In brain tissue segmentation, differences in the quality of the segmentation are even more apparent.

As can be seen in Figure 5.6, the brain tissue segmentation masks of the models in the pixel domain are very different. The ResMLP is clearly the closest to ground truth segmentation and nnU-Net segmentation. The Transformer encoder and MLP were able to learn the coarse locations of each class but failed to learn fine structures. This is especially visible in the cortical gray matter, which is much too present.

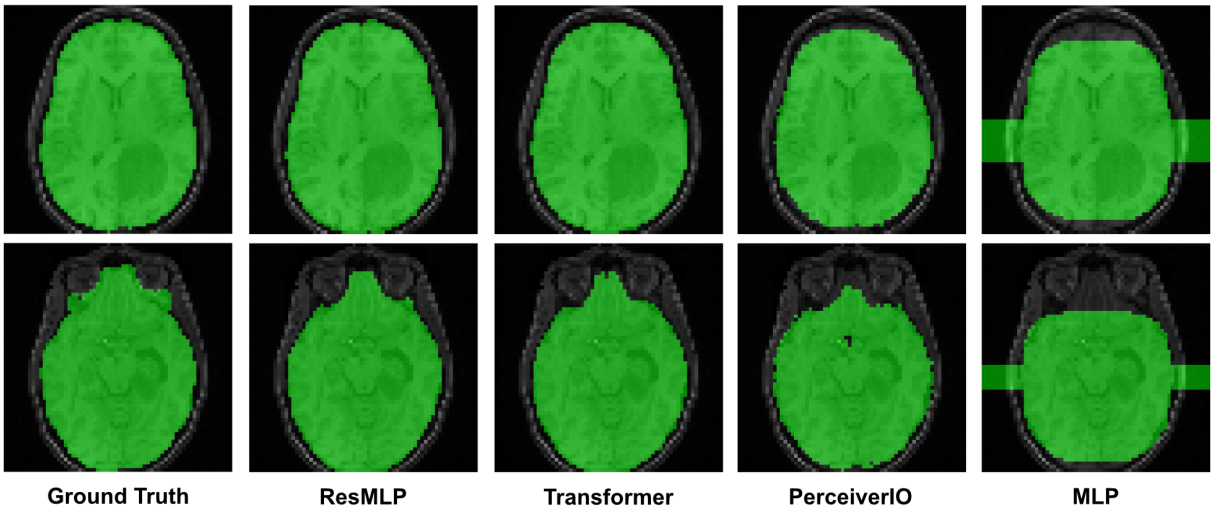

Figure 5.4: Segmentation masks of the different models for skull stripping in k-space.

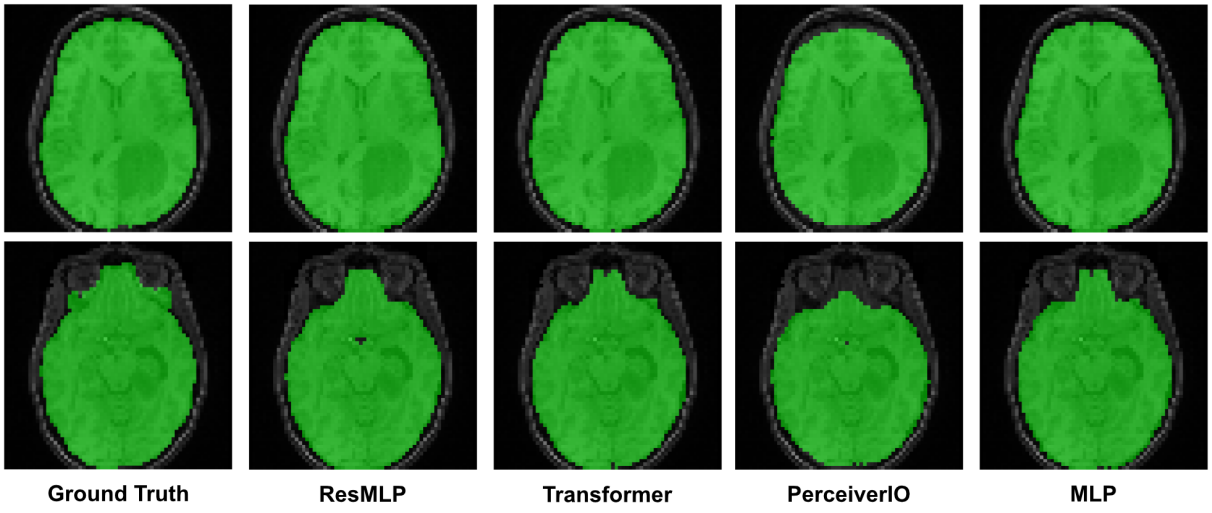

Figure 5.5: Segmentation masks of the different models for skull stripping in the k-space-to-pixel domain.

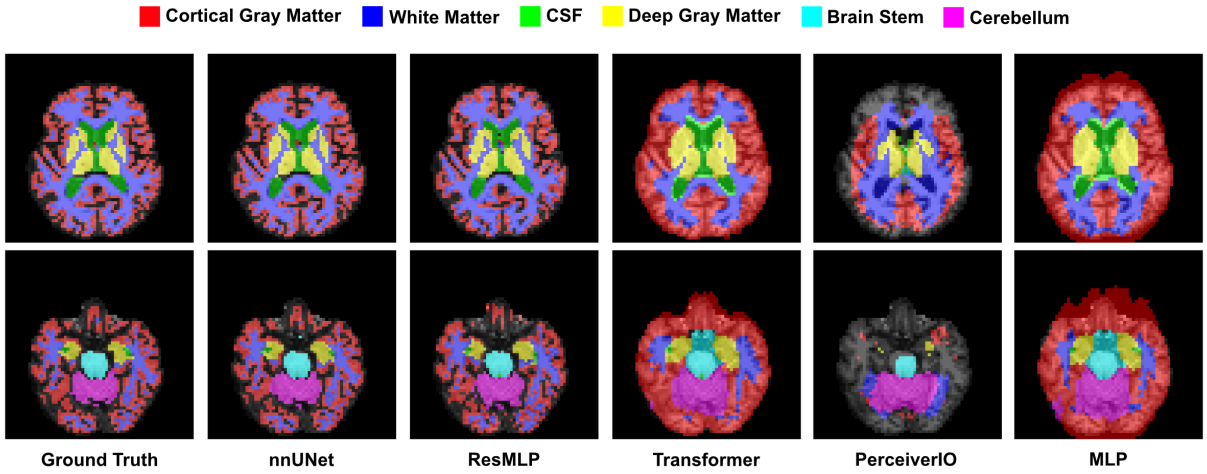

Figure 5.6: Segmentation masks of the different models for brain tissue segmentation in the pixel domain.

As a result, the white matter is underrepresented. For the PerceiverIO, it can be seen that the locations of structures such as cortical gray matter, white matter, brain stem, and cerebellum were approximately learned. However, their extension could not be properly captured. The PerceiverIO segmentation is thus the qualitatively poorest. The properties of the segmentation masks clearly change when looking into the k-space (see Figure 5.7). The ResMLP is no longer able to detect the fine structures of the cortical gray matter and therefore classifies these areas as too large. This is exactly the opposite with the Transformer encoder. The Transformer encoder benefits from the k-space and classifies the deep gray matter more accurately. In addition, the areas around the white matter are more accurate, as is the cerebellum. In the segmentation of the PerceiverIO, the cortical and deep gray matter is now better represented. However, the mask appears scattered. The MLP produced a similarly poor segmentation as in the pixel domain, but the area around the nose contains fewer errors. In the case of the k-space-to-pixel domain (see Figure 5.8), the mask looks a bit noisy for the ResMLP with respect to the deep gray matter. The Transformer encoder was able to delineate the individual classes more sharply, leading to a better segmentation result compared to the pixel domain. The segmentation of the PerceiverIO is almost identical to that in the pixel domain. The same applies to the MLP. The MLP was only able to differentiate the nose area from the background a bit better, although not perfectly.

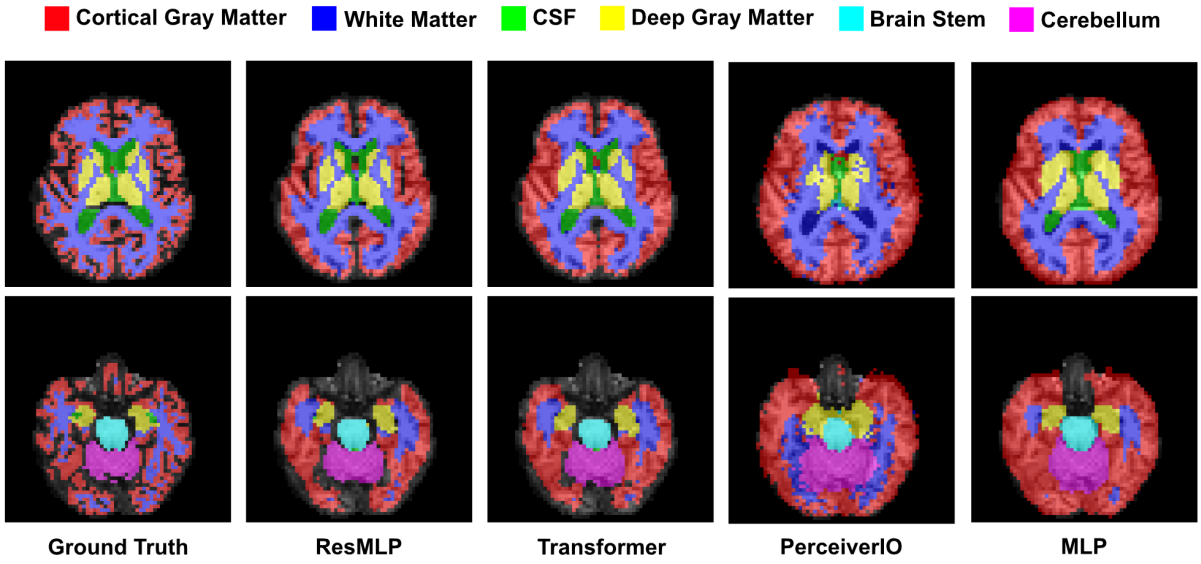

Figure 5.7: Segmentation masks of the different models for brain tissue segmentation in k-space.

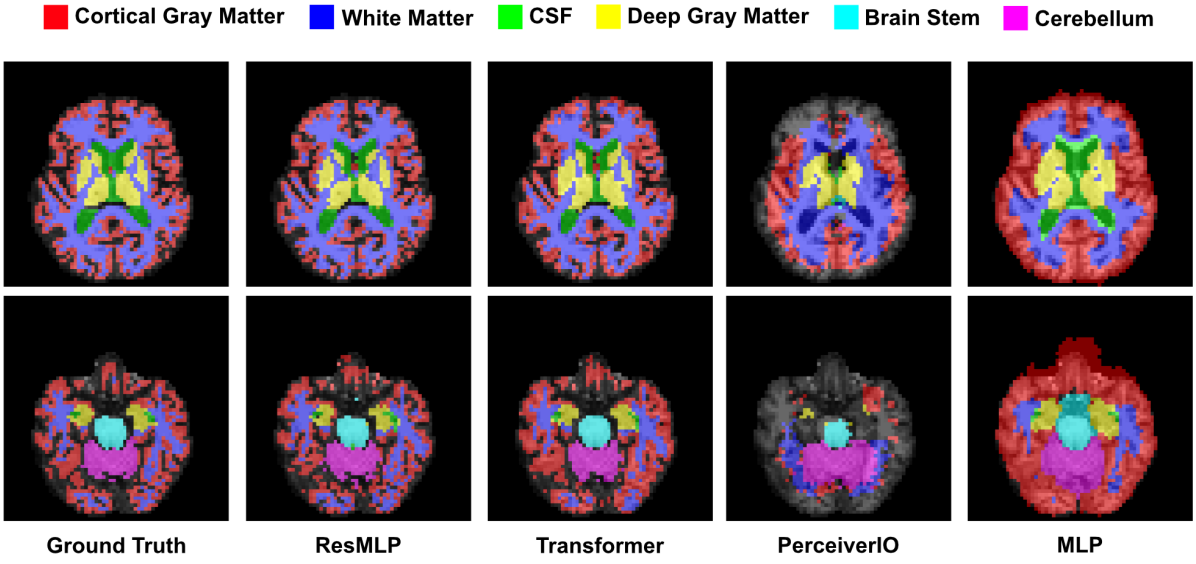

Figure 5.8: Segmentation masks of the different models for brain tissue segmentation in the k-space-to-pixel domain.

5.3 Discussion

On the basis of the quantitative and qualitative results, differences between the segmentations of the individual models could be identified, which can be attributed to the choice of domain. This was observable to a limited extent in the skull stripping task. The reason for this is most likely because skull stripping is not complex enough to see differences between domains. Only for the MLP a significant decrease in performance of the segmentation could be observed when used in k-space. In contrast, the brain tissue segmentation task clearly showed different segmentation results between the domains. Interestingly, all models except ResMLP produced better segmentations in k-space than in the pixel domain. For the ResMLP, it was more difficult to capture fine structures like the cortical gray matter in k-space. For the two attention-based models, however, these structures were captured more reliably in k-space. It may be that the hypothesis that the self-attention mechanism is more appropriate for data in the frequency domain applies here. In the k-space-to-pixel domain, on the other hand, the ResMLP achieves even better performance than in the pixel domain. The Transformer encoder also delivers the best performance by far in the k-space-to-pixel. One reason for this could be that the models generally find it difficult to output a segmentation mask in the frequency domain. A segmentation mask is sharply delineated in the pixel domain. This results in the Fourier representation of this mask having predominantly high frequency components and few low frequency components. This imbalance could be more difficult to learn. If instead the segmentation mask is predicted in the pixel domain, as is the case in the k-space-to-pixel domain, this problem does not arise. This could explain why some models perform best in the k-space-to-pixel domain.
The ResMLP as a non-attention-based DL model showed impressive results considering that mainly consists of MLPs. The ResMLP is the only model besides the PerceiverIO, one of the few models, which allows interactions between the channels. In the case of this work, the channels are the sagittal brain MRI slices. The interaction of these channels seems to be particularly beneficial for the brain tissue segmentation task. Additionally, the experiments in this work showed that for attention-based models in skull stripping and brain tissue segmentation, no additional positional encoding is necessary when the input is Fourier transformed. This observation, which could be shown using an NLP task, could also be transferred to computer vision tasks. However, even though some of the models tested here gave good results considering their complexity, the performance of the nnU-Net could not be reached.

Chapter 6

Conclusion

6.1 Summary

This work has investigated how DL models, and in particular attention-based DL models, perform in different domains on brain segmentation tasks. Besides the pixel domain, the focus of this work was on the frequency domain. It was shown how brain segmentation results vary when the inputs and labels in a supervised learning situation are independently in the pixel domain or transformed into the frequency domain. For this purpose, two attention-based models and two non-attention-based models were implemented, namely the PerceiverIO, a Transformer encoder, an MLP, and the ResMLP. The models were trained and evaluated under three different domain constellations. First, the inputs and labels are in the pixel domain. Second, the inputs and labels are in k-space, which is the frequency domain. And third, the inputs are in k-space, but the labels are in the pixel domain. Skull stripping and brain tissue segmentation were chosen as segmentation tasks. It could be shown experimentally that the choice of domain constellation has a significant influence on brain segmentation performance. In the case of the Transformer encoder, an increase in brain tissue segmentation performance of over 23% could be achieved by the choice of domain as measured by the Dice score. This observation supports the hypothesis that Fourier-transformed input data is more suitable than pixel data for attention-based networks, such as the Transformer encoder. It was also found that the Transformer encoder and ResMLP were able to achieve the best results in the k-space-to-pixel domain constellation. The reason for this could be that segmentation masks in the pixel domain are easier to predict, since they show an imbalance between high-frequency and low-frequency components in the frequency representation. Furthermore, it could be shown that an additional positional

encoding, as it is common for attention-based networks, is not necessary if the input is in frequency representation. Hence, an observation already made by researchers for NLP tasks was transferred to a computer vision task. Finally, the results were compared with the nnU-Net, which acted as a baseline model. Even though the nnU-Net performed better in both segmentation tasks, the ResMLP in particular achieved competitive performance in the k-space-to-pixel domain. Taking into account that the models implemented here, except of the PerceiverIO, are much less complex than the nnU-Net, impressive results were obtained.

6.2 Limitations

Some aspects of this work limit the expressiveness and applicability of the results obtained. For a better assessment of the thesis, these aspects will be explained in the following. To start with, the hyperparameter space for training the DL models was limited to essential parameters. The goal of hyperparameter training is to optimize the non-trainable parameters of the model so that the best possible configuration and thus the best possible segmentation result can be achieved for the respective model. With the help of a larger hyperparameter space, it might be possible to achieve better results than those presented in this work. Related to this is the fact that only the binary cross-entropy loss, cross-entropy loss and mean squared error loss were considered as loss functions in this work. Other loss functions such as Dice Loss or Focal Loss would also be worth considering.
It is also important to note that due to the use of k-space data, the choice of current attention-based networks is limited. Many architectures are based on patching or hierarchies, which are not compatible with the Fourier transform approach used. An adaptation of the code used to obtain the results would be necessary to be able to use these models as well. Furthermore, the approach used in this work cannot be directly translated into the clinical setting. By using the real 2D FFT it was assumed that the input is Hermitian symmetric. However, this assumption cannot be made for raw MRI k-space data. It should also be noted that the brain tissue segmentations from the OASIS-1 dataset were not made manually or manually corrected. This is only the FreeSurfer output, which may contain mistakes. Nevertheless, these data are suitable for comparing models and approaches among each other.

6.3 Outlook

Attention-based networks will certainly play an increasingly important role in the future, not only in medical image segmentation. With research regarding the usability of frequency data for medical image segmentation, the process of segmentation may be getting closer and closer to the MRI image reconstruction task. In the future, it may be possible for these two tasks to intertwine. For example, an MRI scan could be undersampled to omit those frequencies that do not contribute greatly to the segmentation task. An end-to-end DL solution would also be imaginable, which would take care of reconstruction and segmentation. Subsequent to this work, wavelet transforms could also be considered in addition to Fourier transforms. These have the advantage of preserving spatial neighborhood relations and dividing the input into frequency components. Furthermore, it is interesting to consider how the segmentation results change when the masked input is required as model output instead of a binary mask. This would have the advantage that a masked image or volume in its Fourier transformation represents a more normalized output than a binary mask in the frequency domain. In summary, this work provides a promising starting point for further research.

Appendix A

Hyperparameter configuration

The following two tables document the hyperparameter configurations that provided the best segmentation results. Table A.1 shows the configuration of all models for the skull stripping task and Table A.2 for the brain tissue segmentation task.

	Hyperparameters		
Domain Model	Pixel Domain	K-Space Domain	K-Space-to-Pixel Domain
MLP	Hidden layer: 3 Hidden factor: 1 Learning rate: 0.01 Loss: BCE Loss Step size: 300 Optimizer: Lamb	Hidden layer: 3 Hidden factor: 1 Learning rate: 0.01 Loss: MSE Loss Step size: 300 Optimizer: Lamb	Hidden layer: 3 Hidden factor: 1 Learning rate: 0.01 Loss: BCE Loss Step size: 300 Optimizer: Lamb
ResMLP	Hidden layer: 6 Hidden factor: 1 Learning rate: 0.01 Loss: BCE Loss Step size: 300 Optimizer: Lamb	Hidden layer: 6 Hidden factor: 1 Learning rate: 0.01 Loss: MSE Loss Step size: 300 Optimizer: Lamb	Hidden layer: 6 Hidden factor: 1 Learning rate: 0.01 Loss: BCE Loss Step size: 300 Optimizer: Lamb
PerceiverIO	Num latents: 512 Num latent channels: 1024 Num cross attention layers: 8 Num self-attention layers per block: 6 Num self-attention blocks: 8 Dropout: 0.1 Learning rate: 0.01 Loss: BCE Loss Step size: 300 Optimizer: Lamb	Num latents: 512 Num latent channels: 1024 Num cross attention layers: 8 Num self-attention layers per block: 6 Num self-attention blocks: 8 Dropout: 0.1 Learning rate: 0.01 Loss: MSE Loss Step size: 300 Optimizer: Lamb	Num latents: 512 Num latent channels: 1024 Num cross attention layers: 8 Num self-attention layers per block: 6 Num self-attention blocks: 8 Dropout: 0.1 Learning rate: 0.01 Loss: BCE Loss Step size: 300 Optimizer: Lamb
Transformer encoder	Hidden layer: 2 Hidden factor: 1 Learning rate: 0.01 Loss: BCE Loss Step size: 300 Optimizer: Lamb	Hidden layer: 2 Hidden factor: 1 Learning rate: 0.01 Loss: MSE Loss Step size: 300 Optimizer: Lamb	Hidden layer: 2 Hidden factor: 1 Learning rate: 0.01 Loss: BCE Loss Step size: 300 Optimizer: Lamb

Table A.1: Hyperparameters used for the skull stripping task.

	Hyperparameters		
Domain Model	Pixel Domain	K-Space Domain	K-Space-to-Pixel Domain
MLP	Hidden layer: 3 Hidden factor: 1 Learning rate: 0.01 Loss: Weighted CE Loss Step size: 300 Optimizer: Lamb	Hidden layer: 3 Hidden factor: 1 Learning rate: 0.01 Loss: Weighted MSE Loss Step size: 300 Optimizer: Lamb	Hidden layer: 3 Hidden factor: 1 Learning rate: 0.01 Loss: Weighted CE Loss Step size: 300 Optimizer: Lamb
ResMLP	Hidden layer: 6 Hidden factor: 1 Learning rate: 0.01 Loss: CE Loss Step size: 300 Optimizer: Lamb	Hidden layer: 6 Hidden factor: 1 Learning rate: 0.01 Loss: MSE Loss Step size: 300 Optimizer: Lamb	Hidden layer: 6 Hidden factor: 1 Learning rate: 0.01 Loss: CE Loss Step size: 300 Optimizer: Lamb
PerceiverIO	Num latents: 512 Num latent channels: 1024 Num cross attention layers: 8 Num self-attention layers per block: 6 Num self-attention blocks: 8 Dropout: 0.1 Learning rate: 0.01 Loss: CE Loss Step size: 300 Optimizer: Lamb	Num latents: 512 Num latent channels: 1024 Num cross attention layers: 8 Num self-attention layers per block: 6 Num self-attention blocks: 8 Dropout: 0.1 Learning rate: 0.01 Loss: Weighted MSE Loss Step size: 300 Optimizer: Lamb	Num latents: 512 Num latent channels: 1024 Num cross attention layers: 8 Num self-attention layers per block: 6 Num self-attention blocks: 8 Dropout: 0.1 Learning rate: 0.01 Loss: CE Loss Step size: 300 Optimizer: Lamb
Transformer encoder	Hidden layer: 2 Hidden factor: 1 Learning rate: 0.01 Loss: CE Loss Step size: 300 Optimizer: Lamb	Hidden layer: 2 Hidden factor: 1 Learning rate: 0.01 Loss: MSE Loss Step size: 300 Optimizer: Lamb	Hidden layer: 2 Hidden factor: 1 Learning rate: 0.01 Loss: CE Loss Step size: 300 Optimizer: Lamb

Table A.2: Hyperparameters used for the brain tissue segmentation task.

List of Abbreviations

AI Artificial Intelligence

CT computed tomography

CNN convolutional neural network

DL Deep Learning

FFT fast Fourier transform

iFFT inverse fast Fourier transform

LSTM long short-term memory

ML Machine Learning

MRI magnetic resonance imaging

MRF Markov random field

MSA multi-head self-attention

MLP multi-layer perceptron

NLP natural language processing

PET positron emission tomography

PET-CT positron emission tomography–computed tomography

PET-MRI positron emission tomography–magnetic resonance imaging

RNN recurrent neural network

SPECT single-photon emission computed tomography

List of Figures

List of Tables

Bibliography

[Ada+20] N. Adaloglou and S. Karagiannakos. How attention works in deep learning: understanding the attention mechanism in sequence models. en, 2020. URL: https://theaisummer.com/attention/ (visited on 10/25/2023) (cited on p. 20).

[Ant+22] L. Antonelli, V. De Simone, and D. di Serafino. A view of computational models for image segmentation. en. *ANNALI DELL'UNIVERSITA' DI FERRARA*, 68(2):277–294, November 2022. ISSN: 1827-1510. DOI: 10.1007/s11565-022-00417-6. URL: https://doi.org/10.1007/s11565-022-00417-6 (visited on 10/23/2023) (cited on p. 7).

[Ara+19] A. Araujo, W. Norris, and J. Sim. Computing Receptive Fields of Convolutional Neural Networks. en. *Distill*, 4(11):e21, November 2019. ISSN: 2476-0757. DOI: 10.23915/distill.00021. URL: https://distill.pub/2019/computing-receptive-fields (visited on 10/20/2023) (cited on p. 20).

[Bak+22] S. Bakas, C. Sako, H. Akbari, M. Bilello, A. Sotiras, G. Shukla, J. D. Rudie, N. F. Santamaría, A. F. Kazerooni, S. Pati, S. Rathore, E. Mamourian, S. M. Ha, W. Parker, J. Doshi, U. Baid, M. Bergman, Z. A. Binder, R. Verma, R. A. Lustig, A. S. Desai, S. J. Bagley, Z. Mourelatos, J. Morrissette, C. D. Watt, S. Brem, R. L. Wolf, E. R. Melhem, M. P. Nasrallah, S. Mohan, D. M. O'Rourke, and C. Davatzikos. The University of Pennsylvania glioblastoma (UPenn-GBM) cohort: advanced MRI, clinical, genomics, & radiomics. en. *Scientific Data*, 9(1):453, July 2022. ISSN: 2052-4463. DOI: 10.1038/s41597-022-01560-7. URL: https://www.nature.com/articles/s41597-022-01560-7 (visited on 10/11/2023). Number: 1 Publisher: Nature Publishing Group (cited on p. 23).

[Bal22] M. K. Balwant. A Review on Convolutional Neural Networks for Brain Tumor Segmentation: Methods, Datasets, Libraries, and Future Directions. *IRBM*, 43(6):521–537, December 2022. ISSN: 1959-0318. DOI: 10.1016/j.irbm.2022.

05.002. URL: https://www.sciencedirect.com/science/article/pii/S1959031822000550 (visited on 10/24/2023) (cited on p. 10).

[Bau+10] S. Bauer, C. Seiler, T. Bardyn, P. Buechler, and M. Reyes. Atlas-based segmentation of brain tumor images using a Markov Random Field-based tumor growth model and non-rigid registration. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 4080–4083, August 2010. DOI: 10.1109/IEMBS.2010.5627302. URL: https://ieeexplore.ieee.org/document/5627302 (visited on 10/24/2023). ISSN: 1558-4615 (cited on p. 9).

[Ber+18] O. Bernard, A. Lalande, C. Zotti, F. Cervenansky, X. Yang, P.-A. Heng, I. Cetin, K. Lekadir, O. Camara, M. A. Gonzalez Ballester, G. Sanroma, S. Napel, S. Petersen, G. Tziritas, E. Grinias, M. Khened, V. A. Kollerathu, G. Krishnamurthi, M.-M. Rohé, X. Pennec, M. Sermesant, F. Isensee, P. Jäger, K. H. Maier-Hein, P. M. Full, I. Wolf, S. Engelhardt, C. F. Baumgartner, L. M. Koch, J. M. Wolterink, I. Išgum, Y. Jang, Y. Hong, J. Patravali, S. Jain, O. Humbert, and P.-M. Jodoin. Deep Learning Techniques for Automatic MRI Cardiac Multi-Structures Segmentation and Diagnosis: Is the Problem Solved? *IEEE Transactions on Medical Imaging*, 37(11):2514–2525, November 2018. ISSN: 1558-254X. DOI: 10.1109/TMI.2018.2837502. URL: https://ieeexplore.ieee.org/document/8360453 (visited on 10/24/2023). Conference Name: IEEE Transactions on Medical Imaging (cited on p. 14).

[Bra+23] G. Brauwers and F. Frasincar. A General Survey on Attention Mechanisms in Deep Learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3279–3298, April 2023. ISSN: 1558-2191. DOI: 10.1109/TKDE.2021.3126456. URL: https://ieeexplore.ieee.org/document/9609539 (visited on 10/31/2023). Conference Name: IEEE Transactions on Knowledge and Data Engineering (cited on p. 18).

[Bra00] R. Bracewell. *The Fourier Transform And Its Applications*. eng. McGraw-Hill, 3rd edition, 2000. ISBN: 0-07-303938-1. URL: http://archive.org/details/TheFourierTransformAndItsApplicationsBracewell (visited on 10/20/2023) (cited on p. 20).

[Cao+23] 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. en. In L. Karlinsky, T. Michaeli, and K. Nishino, editors, volume 13803, pages 205–

218, Cham. Springer Nature Switzerland, 2023. ISBN: 978-3-031-25065-1 978-3-031-25066-8. DOI: 10.1007/978-3-031-25066-8_9. URL: https://link.springer.com/10.1007/978-3-031-25066-8_9 (visited on 10/10/2023). Book Title: Computer Vision – ECCV 2022 Workshops Series Title: Lecture Notes in Computer Science (cited on pp. 2, 13).

[Che+21] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou. TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. en, February 2021. URL: https://arxiv.org/abs/2102.04306v1 (visited on 10/10/2023) (cited on pp. 2, 13, 14).

[Col+95] D. L. Collins, C. J. Holmes, T. M. Peters, and A. C. Evans. Automatic 3-D model-based neuroanatomical segmentation. en. *Human Brain Mapping*, 3(3):190–208, 1995. ISSN: 1097-0193. DOI: 10.1002/hbm.460030304. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.460030304 (visited on 10/24/2023) (cited on p. 9).

[Dev+19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, May 2019. DOI: 10.48550/arXiv.1810.04805. URL: http://arxiv.org/abs/1810.04805 (visited on 10/11/2023). arXiv:1810.04805 [cs] (cited on pp. 3, 19).

[Dos+20] 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. en. In October 2020. URL: https://openreview.net/forum?id=YicbFdNTTy (visited on 10/11/2023) (cited on p. 2).

[Faw+21] A. Fawzi, A. Achuthan, and B. Belaton. Brain Image Segmentation in Recent Years: A Narrative Review. *Brain Sciences*, 11(8):1055, August 2021. ISSN: 2076-3425. DOI: 10.3390/brainsci11081055. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8392552/ (visited on 10/11/2023) (cited on pp. 8–10).

[Gol+20] K. Goldberg, S. Shapiro, E. Richardson, and S. Avidan. Rethinking FUN: Frequency-Domain Utilization Networks, December 2020. DOI: 10.48550/arXiv.2012.03357. URL: http://arxiv.org/abs/2012.03357 (visited on 10/24/2023). arXiv:2012.03357 [cs] (cited on p. 14).

[Gut+23] R. Gutsche, C. Lowis, K. Ziemons, M. Kocher, G. Ceccon, C. Régio Brambilla, N. J. Shah, K.-J. Langen, N. Galldiks, F. Isensee, and P. Lohmann. Automated Brain Tumor Detection and Segmentation for Treatment Response Assessment Using Amino Acid PET. eng. *Journal of Nuclear Medicine: Official Publication, Society of Nuclear Medicine*, 64(10):1594–1602, October 2023. ISSN: 1535-5667. DOI: 10.2967/jnumed.123.265725 (cited on p. 1).

[Har+85] R. M. Haralick and L. G. Shapiro. Image Segmentation Techniques. In J. F. Gilmore, editor, page 2, Arlington, April 1985. DOI: 10.1117/12.948400. URL: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.948400 (visited on 10/23/2023). ISSN: 0277-786X (cited on p. 7).

[Hat+22] 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. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*:1748–1758, January 2022. DOI: 10.1109/WACV51458.2022.00181. URL: https://ieeexplore.ieee.org/document/9706678/ (visited on 10/10/2023). Conference Name: 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) ISBN: 9781665409155 Place: Waikoloa, HI, USA Publisher: IEEE (cited on pp. 2, 13–15).

[Ise+21] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. en. *Nature Methods*, 18(2):203–211, February 2021. ISSN: 1548-7105. DOI: 10.1038/s41592-020-01008-z. URL: https://www.nature.com/articles/s41592-020-01008-z (visited on 10/12/2023). Number: 2 Publisher: Nature Publishing Group (cited on pp. 3, 12).

[Jae+22] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, O. Hénaff, M. M. Botvinick, A. Zisserman, O. Vinyals, and J. Carreira. Perceiver IO: A General Architecture for Structured Inputs & Outputs, March 2022. DOI: 10.48550/arXiv.2107.14795. URL: http://arxiv.org/abs/2107.14795 (visited on 10/11/2023). arXiv:2107.14795 [cs, eess] (cited on pp. 3, 27).

[Kap+14] T. Kapur, J. Egger, J. Jayender, M. Toews, and W. M. Wells. Registration and Segmentation for Image-Guided Therapy. en. In F. A. Jolesz, editor, *Intraoperative Imaging and Image-Guided Therapy*, pages 79–91. Springer, New

York, NY, 2014. ISBN: 978-1-4614-7657-3. DOI: 10.1007/978-1-4614-7657-3_5. URL: https://doi.org/10.1007/978-1-4614-7657-3_5 (visited on 10/11/2023) (cited on p. 1).

[Kra+23] M. Krasser and C. Stumpf. A PyTorch implementation of Perceiver, Perceiver IO and Perceiver AR with PyTorch Lightning scripts for distributed training. May 2023. URL: https://github.com/krasserm/perceiver-io (visited on 10/26/2023) (cited on p. 27).

[Lan+15] B. Landman, Z. Xu, J. Igelsias, M. Styner, T. Langerak, and A. Klein. Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge. en-US. In *MICCAI Multi-Atlas Labeling Beyond the Cranial Vault - Workshop and Challenge*, 2015. URL: https://www.synapse.org/#!Synapse:syn3193805/wiki/217789 (visited on 10/24/2023) (cited on p. 14).

[Lee+22] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontanon. FNet: Mixing Tokens with Fourier Transforms, May 2022. DOI: 10.48550/arXiv.2105.03824. URL: http://arxiv.org/abs/2105.03824 (visited on 10/11/2023). arXiv:2105.03824 [cs] (cited on pp. 2, 15).

[Li+20] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan. Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting, January 2020. URL: http://arxiv.org/abs/1907.00235 (visited on 10/25/2023). arXiv:1907.00235 [cs, stat] (cited on p. 15).

[Li+21] J. Li, M. Erdt, F. Janoos, T.-c. Chang, and J. Egger. Medical image segmentation in oral-maxillofacial surgery. In J. Egger and X. Chen, editors, *Computer-Aided Oral and Maxillofacial Surgery*, pages 1–27. Academic Press, January 2021. ISBN: 978-0-12-823299-6. DOI: 10.1016/B978-0-12-823299-6.00001-8. URL: https://www.sciencedirect.com/science/article/pii/B9780128232996000018 (visited on 10/11/2023) (cited on p. 1).

[Li+22] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer. MViTv2: Improved Multiscale Vision Transformers for Classification and Detection, March 2022. DOI: 10.48550/arXiv.2112.01526. URL: http://arxiv.org/abs/2112.01526 (visited on 10/22/2023). arXiv:2112.01526 [cs] (cited on p. 2).

[Lin+22] Y. Lin, L. Liu, K. Ma, and Y. Zheng. Seg4Reg+: Consistency Learning between Spine Segmentation and Cobb Angle Regression, August 2022. DOI: 10.48550/arXiv.2208.12462. URL: http://arxiv.org/abs/2208.12462 (visited on 10/22/2023). arXiv:2208.12462 [cs] (cited on p. 2).

[Mar+07] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner. Open Access Series of Imaging Studies (OASIS): Cross-sectional MRI Data in Young, Middle Aged, Nondemented, and Demented Older Adults. *Journal of Cognitive Neuroscience*, 19(9):1498–1507, September 2007. ISSN: 0898-929X. DOI: 10.1162/jocn.2007.19.9.1498. URL: https://doi.org/10.1162/jocn.2007.19.9.1498 (visited on 10/11/2023) (cited on p. 23).

[Men+15] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M.-A. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, Ç. Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H.-C. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. Van Leemput. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, October 2015. ISSN: 1558-254X. DOI: 10.1109/TMI.2014.2377694. URL: https://ieeexplore.ieee.org/abstract/document/6975210 (visited on 10/28/2023). Conference Name: IEEE Transactions on Medical Imaging (cited on p. 10).

[Nie+23] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers, March 2023. URL: http://arxiv.org/abs/2211.14730 (visited on 10/25/2023). arXiv:2211.14730 [cs] (cited on p. 15).

[Ots79] N. Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, January 1979. ISSN: 2168-2909. DOI: 10.1109/TSMC.1979.4310076. URL: https://ieeexplore.

ieee.org/document/4310076 (visited on 10/11/2023). Conference Name: IEEE Transactions on Systems, Man, and Cybernetics (cited on p. 8).

[Pal+93] N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26:1277–1294, January 1993. DOI: 10.1016/0031-3203(93)90135-J. URL: https://ui.adsabs.harvard.edu/abs/1993PatRe..26.1277P (visited on 10/23/2023). ADS Bibcode: 1993PatRe..26.1277P (cited on p. 7).

[Rad+18] A. Radford and K. Narasimhan. Improving Language Understanding by Generative Pre-Training, 2018. URL: https://www.semanticscholar.org/paper/Improving-Language-Understanding-by-Generative-Radford-Narasimhan/cd18800a0fe0b668a1cc19f2ec95b5003d0a5035 (cited on p. 19).

[Raj+00] J. Rajapakse, J. Giedd, and J. Rapoport. Statistical Approach to Segmentation of Single-Channel Cerebral MR Images. *IEEE Transactions on Medical Imaging*, 16, July 2000. DOI: 10.1109/42.563663 (cited on p. 7).

[Ran+17] R. S. Randhawa, A. Modi, P. Jain, and P. Warier. Improving Boundary Classification for Brain Tumor Segmentation and Longitudinal Disease Progression. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, Cham. Springer, 2017. DOI: https://doi.org/10.1007/978-3-319-55524-9_7. URL: https://link.springer.com/chapter/10.1007/978-3-319-55524-9_7 (visited on 10/28/2023) (cited on pp. 10, 11).

[Rog+09] J. Rogowska, M. A. Sutton, A. Wismueller, T. McInerney, D. Terzopoulos, L. H. Staib, C. Xu, M. S. Atkins, D. L. Pham, P.-L. Bazin, D. H. Laidlaw, and M. Petrou. Part II: Segmentation. English. In I. N. Bankman, editor, *Handbook of Medical Image Processing and Analysis*, pages 71–257. Academic Press, 2nd edition, 2009. ISBN: 978-0-12-373904-9. URL: https://www.sciencedirect.com/book/9780123739049/handbook-of-medical-image-processing-and-analysis?via=ihub= (visited on 10/11/2023) (cited on pp. 1, 8, 9).

[Ron+15] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation, May 2015. DOI: 10.48550/arXiv.1505.04597. URL: http://arxiv.org/abs/1505.04597 (visited on 10/24/2023). arXiv:1505.04597 [cs] (cited on p. 12).

[Rya+23] C. Ryali, Y.-T. Hu, D. Bolya, C. Wei, H. Fan, P.-Y. Huang, V. Aggarwal, A. Chowdhury, O. Poursaeed, J. Hoffman, J. Malik, Y. Li, and C. Feichtenhofer. Hiera: A Hierarchical Vision Transformer without the Bells-and-Whistles, June

2023. DOI: 10.48550/arXiv.2306.00989. URL: http://arxiv.org/abs/2306.00989 (visited on 10/11/2023). arXiv:2306.00989 [cs] (cited on p. 2).

[Sch+19] J. Schlemper, O. Oktay, M. Schaap, M. Heinrich, B. Kainz, B. Glocker, and D. Rueckert. Attention Gated Networks: Learning to Leverage Salient Regions in Medical Images, January 2019. URL: http://arxiv.org/abs/1808.08114 (visited on 10/24/2023). arXiv:1808.08114 [cs] (cited on pp. 13, 14).

[Seg+20] A. Segato, A. Marzullo, F. Calimeri, and E. De Momi. Artificial intelligence for brain diseases: A systematic review. *APL Bioengineering*, 4(4):041503, October 2020. ISSN: 2473-2877. DOI: 10.1063/5.0011697. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7556883/ (visited on 10/22/2023) (cited on pp. 1, 10).

[Sim+19] A. L. Simpson, M. Antonelli, S. Bakas, M. Bilello, K. Farahani, B. van Ginneken, A. Kopp-Schneider, B. A. Landman, G. Litjens, B. Menze, O. Ronneberger, R. M. Summers, P. Bilic, P. F. Christ, R. K. G. Do, M. Gollub, J. Golia-Pernicka, S. H. Heckers, W. R. Jarnagin, M. K. McHugo, S. Napel, E. Vorontsov, L. Maier-Hein, and M. J. Cardoso. A large annotated medical image dataset for the development and evaluation of segmentation algorithms, February 2019. DOI: 10.48550/arXiv.1902.09063. URL: http://arxiv.org/abs/1902.09063 (visited on 10/24/2023). arXiv:1902.09063 [cs, eess] (cited on p. 14).

[Str+21] R. Strudel, R. Garcia, I. Laptev, and C. Schmid. Segmenter: Transformer for Semantic Segmentation, September 2021. DOI: 10.48550/arXiv.2105.05633. URL: http://arxiv.org/abs/2105.05633 (visited on 10/22/2023). arXiv:2105.05633 [cs] (cited on p. 2).

[Tou+21] H. Touvron, P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, G. Izacard, A. Joulin, G. Synnaeve, J. Verbeek, and H. Jégou. ResMLP: Feedforward networks for image classification with data-efficient training, June 2021. DOI: 10.48550/arXiv.2105.03404. URL: http://arxiv.org/abs/2105.03404 (visited on 10/11/2023). arXiv:2105.03404 [cs] (cited on pp. 3, 27, 28).

[Tri15] A. Trindade. ElectricityLoadDiagrams20112014, 2015. DOI: https://doi.org/10.24432/C58C86 (cited on p. 16).

[Vas+17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. en, June 2017. URL:

https://arxiv.org/abs/1706.03762v7 (visited on 10/10/2023) (cited on pp. 2, 17–19, 27).

[Wan+18] G. Wang, W. Li, S. Ourselin, and T. Vercauteren. Automatic Brain Tumor Segmentation Using Cascaded Anisotropic Convolutional Neural Networks. en. In A. Crimi, S. Bakas, H. Kuijf, B. Menze, and M. Reyes, editors, *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, Lecture Notes in Computer Science, pages 178–190, Cham. Springer International Publishing, 2018. ISBN: 978-3-319-75238-9. DOI: 10.1007/978-3-319-75238-9_16 (cited on pp. 11, 12).

[Wan+19] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding, February 2019. DOI: 10.48550/arXiv.1804.07461. URL: http://arxiv.org/abs/1804.07461 (visited on 10/24/2023). arXiv:1804.07461 [cs] (cited on p. 15).

[Wu+21] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang. CvT: Introducing Convolutions to Vision Transformers, March 2021. DOI: 10.48550/arXiv.2103.15808. URL: http://arxiv.org/abs/2103.15808 (visited on 10/22/2023). arXiv:2103.15808 [cs] (cited on p. 2).

[Wu+22] H. Wu, J. Xu, J. Wang, and M. Long. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting, January 2022. DOI: 10.48550/arXiv.2106.13008. URL: http://arxiv.org/abs/2106.13008 (visited on 10/10/2023). arXiv:2106.13008 [cs] (cited on pp. 15, 16).

[Xu+20] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, and F. Ren. Learning in the Frequency Domain, March 2020. URL: http://arxiv.org/abs/2002.12416 (visited on 10/24/2023). arXiv:2002.12416 [cs] (cited on p. 14).

[Yad+22] R. Yadav and M. Pandey. Image Segmentation Techniques: A Survey. en. In D. Gupta, Z. Polkowski, A. Khanna, S. Bhattacharyya, and O. Castillo, editors, *Proceedings of Data Analytics and Management*, Lecture Notes on Data Engineering and Communications Technologies, pages 231–239, Singapore. Springer Nature, 2022. ISBN: 9789811662898. DOI: 10.1007/978-981-16-6289-8_20 (cited on p. 7).

[Zad94] L. A. Zadeh. Fuzzy logic, neural networks, and soft computing. *Communications of the ACM*, 37(3):77–84, 1994. ISSN: 0001-0782. DOI: 10.1145/175247.175255. URL: https://dl.acm.org/doi/10.1145/175247.175255 (visited on 10/23/2023) (cited on p. 7).

[Zha+22] D. Zhang, J. Tang, and K.-T. Cheng. Graph Reasoning Transformer for Image Parsing, September 2022. URL: http://arxiv.org/abs/2209.09545 (visited on 10/22/2023). arXiv:2209.09545 [cs] (cited on p. 2).

[Zho+21] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting, March 2021. DOI: 10.48550/arXiv.2012.07436. URL: http://arxiv.org/abs/2012.07436 (visited on 10/11/2023). arXiv:2012.07436 [cs] (cited on p. 16).

[Zho+22] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting, June 2022. DOI: 10.48550/arXiv.2201.12740. URL: http://arxiv.org/abs/2201.12740 (visited on 10/10/2023). arXiv:2201.12740 [cs, stat] (cited on pp. 15, 16).