Deep Learning-Based Multi-Class Segmentation and 3D Reconstruction for Modelling Transcranial Stimulation*

*Note: This paper is the submission of BIOM9021 Master Project B in the form of an IEEE journalist paper.

MingWei Hsieh
Graduate School of Biomedical Engineering
University of New South Walse
Sydney, Australia
m.hsieh@student.unsw.edu.au

***Abstract*—Medical image segmentation is a crucial requirement for the development of medical research, disease diagnosis and treatment planning. Moreover, the segmentation of medical imaging can be utilised in certain physics modelling such as studies of transcranial stimulation, and which is the purpose of this segmentation project. In this paper, we have chosen the 3D-UNet as the main architecture for accurately segmenting skin, skull, cerebrospinal fluid (CSF), grey matter (GM), and white matter (WM) regions using deep learning approach with given seven labelled MRIs for training. Then we utilised the trained network to make prediction label of each tissue and generate each mesh accordingly. Our results show the attempts and comparisons between the results we have achieved in this project. We have opted for the 4-layer-Unet, which has achieved 0.65 in MeanIoU and 0.74 in F.W.IoU.**

By reconstructing the predicted sub-volumes back into a whole 3D volume, this project was able to generate a mesh of the head, which has important applications in medical image segmentation and reduces the cost of time required to get a head model for the physics simulation study. Overall, our project demonstrates the effectiveness of 3D-UNET in multi-class segmentation and its potential for use in multiclass segmentation applications of the head MRI scans applications. At the end of this paper, we suggest some methods that can push this project further

In the introduction, we discussed the algorithms that have been applied in image segmentation tasks and related machine-learning based approaches. We analyzed the algorithms and chose the 3D-UNet based on its performance in multi-class segmentation. Through our experimentation, we have confirmed that the 3D-UNet architecture performs well in segmenting the different regions of the brain.

Future work could explore other variations of the U-Net architecture, such as the Attention U-Net or the Nested U-Net, and compare their performance with the 3D-UNet architecture. Additionally, this project's methodology provides the potential that could be applied to other medical imaging modalities and other body parts for segmentation.

Working with whole-head modelling with specific stimulation further study of brain science

I. Introduction

The demand for Magnetic Resonance Imaging (MRI) segmentation has increased rapidly in recent years due to improvements in image processing techniques. Nowadays, medical images are captured and stored digitally. With the increasing importance of medical imaging in various medical procedures, such as surgical planning, post-surgical assessment, and abnormality diagnosis, the demand for image segmentation has also increased.

Head, and brain MRI segmentation is another high-demand and challenging task, as it is not only used in the clinical diagnosis purpose but as well for research use when studying the physics properties of the tissues or segmentation accuracy can significantly affect the detection of lesions, tumours, and necrotic tissues. MRI has been implemented in clinical practice to assist in disease research and surgical diagnosis. The demand for MRI segmentation is also applicable in research areas, such as noninvasive brain stimulation techniques used in therapy for Major Depressive Disorder (MDD) [15], Alzheimer's diagnosis [1], and other related research have been done. In this case, a 3D mesh head model with given mechanical properties is required to study the simulation results. Unlike segmentation for diagnosis, segmentation for modelling requires labels for multiple layers. For example, in diagnosing brain tumours, the practitioner only needs to know the position, size, and area of the lesion tissue, while modelling requires at least five segmented layers: skin, skull, cerebrospinal fluid (CSF), grey matter (GM), and white matter (WM).

Segmentation of MRI images is challenging due to imperfections and image artifacts. A variety of techniques for image segmentation have been developed to address the diversity of image processing applications. However, no single method is suitable for every case, and different methods have various effects on different types of images. Some methods rely solely on grey-level histograms, while others integrate spatial image information to increase robustness in noisy environments. Additional approaches include probabilistic or fuzzy set-theoretic approaches and the integration of prior knowledge, such as an MRI brain atlas, to improve segmentation performance. Although developed for one class of images, many segmentation methods can be easily extended to other classes of

images. For example, the theory of graph cuts, which was initially developed for binary images, can be modified for MRI segmentation of the brain tissue. Unsupervised fuzzy clustering has been successfully applied in various areas, including remote sensing, geology, and medical, biological, and molecular imaging. Segmentation methods for brain MRI can be grouped into

i) manual segmentation
ii) intensity-based methods,
iii) atlas-based methods,
iv) surface-based methods, and
v) hybrid segmentation methods.

according to the sum-up review by Despotović, I. et al [3].

The objective of this thesis project is to find an algorithm for automated labelling of five tissues(Skin, Skull, cerebrospinal fluid, grey matter, and white matter) of the whole head MRI scans using a deep learning approach. This introduction will review prior used segmentation methods for whole-head segmentation and related deep-learning approaches for image segmentation tasks.

A. Manual Segmentation

Typically, Manual segmentation involves a human operator, such as a skilled physician, manually segmenting and labelling an image by hand. This process is typically performed slice-by-slice for 3D volumetric imagery and is considered the most accurate due to the difficulty in accurately and reliably delineating structures in medical images. However, with improvements in imaging technology, manual segmentation has become an intensive and time-consuming task. Operators may need to go through around eighty spatial images, slice by slice, to extract target structures' tissue.

Manual segmentation is not only time-consuming but also prone to errors. Additionally, manual segmentation results are often challenging to reproduce due to significant variability even among experienced operators. Despite the aid of software such as Materialise Mimics [35], manual segmentation is still a challenging task and remains extensively used for defining a surrogate for true delineation (called "ground truth") and quantitatively evaluating automated segmentation.

B. Intensity-based Method

Segmentation methods that are intensity-based classify pixels or voxels based on their intensity. In brain MRI, three primary tissue classes, namely cerebrospinal fluid (CSF), grey matter (GM), and white matter (WM), can be differentiated based on image intensity.

1) Thresholding: Thresholding is one of the simplest and most widely used techniques for image segmentation. It involves setting a threshold value that separates the image into foreground and background. Pixels with intensities above the threshold are considered foreground, while the pixels below the threshold are considered background. In the MRI scans, most of the tissue parts have their own intensity feature on the MRI. Therefore, thresholding is a good method to identify a certain type of tissue on MRIs.

The simplest thresholding method is the global thresholding technique. It assumes that the foreground and background regions have significantly different pixel intensities. A single threshold value is applied to the entire image to separate the foreground and background regions. This method works well for images with uniform lighting conditions and high contrast between foreground and background. However, it might be limited in the presence of noise or variations in different weighted images.

To overcome the limitations of global thresholding, several adaptive thresholding methods have been proposed. Adaptive thresholding methods apply different threshold values to different parts of the image based on local properties such as mean intensity, standard deviation, or entropy. One such method is the Otsu thresholding method, which calculates the threshold value that minimizes the intra-class variance between foreground and background pixels. This method is effective for images with bimodal intensity histograms but may fail for images with multi-modal histograms.

Another popular adaptive thresholding method is the local adaptive thresholding method. It partitions the image into small regions and calculates the threshold value for each region separately. This method is effective in the presence of non-uniform lighting conditions and uneven backgrounds. One example of a local adaptive thresholding method is the Sauvola thresholding method, which calculates the threshold value for each pixel based on the local mean and standard deviation of the neighboring pixels. This method works well for documents with text and background but may not work well for images with high-frequency patterns.

The mathematical formula for Otsu's method can be expressed as:

$$t^* = \underset{t}{\operatorname{argmax}} \left[\frac{w_0(t) w_1(t) (\mu_1(t) - \mu_0(t))^2}{\sigma_B^2(t)} \right]$$

Where t^* is the optimal threshold value, $w_0(t)$ and $w_1(t)$ are the probabilities of the background and foreground classes, respectively, at threshold t, $\mu_0(t)$ and $\mu_1(t)$ are the mean intensities of the background and foreground pixels at threshold t, and $\sigma_B^2(t)$ is the between-class variance.

Thresholding methods can also be combined with other segmentation techniques to improve their performance. For example, thresholding can be used as a pre-processing step before applying edge detection techniques. The Canny edge detection algorithm is a popular method for detecting edges in images. It uses a combination of Gaussian smoothing, gradient calculation, non-maximum suppression, and hysteresis thresholding to detect edges in images. Thresholding is used to set the high and low threshold values for hysteresis thresholding, which helps to remove weak edges and preserve strong edges.

In conclusion, thresholding methods are simple and effective techniques for image segmentation. Global thresholding is a basic method that works well for images with uniform lighting conditions and high contrast between foreground and background. Adaptive thresholding methods, such as Otsu

and Sauvola thresholding, are effective for images with non-uniform lighting conditions and uneven backgrounds. Thresholding can also be combined with other segmentation techniques, such as edge detection, to improve their performance. However, thresholding methods have their limitations, and their effectiveness depends on the characteristics of the image being segmented.

2) Clustering: Clustering is a popular technique used in image segmentation to group similar pixels or regions together based on their properties. Clustering methods are unsupervised learning techniques, which means that they do not require any prior knowledge about the data. In this literature review, we will discuss various clustering methods and their applications in image segmentation.

One of the most popular clustering methods is the k-means algorithm. It is a simple and efficient method for partitioning an image into k clusters based on pixel intensity values. The algorithm starts by randomly selecting k cluster centers and assigning each pixel to the closest center. The algorithm then recalculates the cluster centers based on the mean of the pixel values in each cluster and repeats the process until convergence. The k-means algorithm works well for images with low complexity, but it may fail for images with high dimensionality and noise.

The mathematical formula for k-means clustering can be expressed as:

$$J(c_1, c_2, ..., c_k) = \sum_{i=1}^{n} \min_{j=1}^{k} \|x_i - c_j\|^2$$

Where J is the objective function, $c_1, c_2, ..., c_k$ are the cluster centres, x_i is the i^{th} pixel in the image, and n is the total number of pixels. The objective function aims to minimize the distance between each pixel and its assigned cluster centre.

Another popular clustering method is the fuzzy c-means algorithm. It is a soft clustering method that assigns each pixel a probability of belonging to each cluster. The algorithm iteratively updates the cluster centers and the probability values until convergence. The fuzzy c-means algorithm works well for images with noise and overlapping regions. However, it may be computationally expensive for large datasets.

Spectral clustering is a graph-based clustering method that uses the eigenvalues and eigenvectors of the similarity matrix to partition the image into clusters. The similarity matrix is calculated based on the similarity between pixel intensities or feature vectors. Spectral clustering is effective for images with non-linear boundaries and complex structures. However, it may be sensitive to the choice of parameters and the size of the image.

Hierarchical clustering is a method that builds a hierarchical tree of clusters, where each node represents a cluster at a different level of granularity. The algorithm starts by treating each pixel as a separate cluster and iteratively merges the closest clusters until all pixels are in the same cluster. The tree of clusters can be cut at different levels to obtain different segmentations of the image. Hierarchical clustering is effective for images with varying levels of detail and can provide a hierarchy of regions at different scales.

Clustering methods can also be combined with other segmentation techniques to improve their performance. For example, clustering can be used as a pre-processing step before applying edge detection techniques. The watershed algorithm is a popular method for segmenting images based on the topography of the image. It starts by treating the image as a topographic map and flooding the valleys with water until they meet at the watershed lines. Clustering can be used to group pixels with similar topographic properties before applying the watershed algorithm. This helps to reduce over-segmentation and improve the quality of the segmentation.

The mathematical formula for fuzzy c-means clustering can be expressed as:

$$J_m(U, V) = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m \|x_i - v_j\|^2$$

Where J_m is the objective function, U is the membership matrix, V is the cluster centre matrix, u_{ij} is the membership value of pixel i to cluster j, m is a weighting exponent that controls the degree of fuzziness in the clustering, x_i is the ith pixel in the image, and n is the total number of pixels.

In conclusion, clustering methods are popular techniques for image segmentation that group similar pixels or regions together based on their properties. The k-means algorithm is a simple and efficient method for partitioning an image into k clusters. The fuzzy c-means algorithm is a soft clustering method that assigns each pixel a probability of belonging to each cluster. Spectral clustering is effective for images with non-linear boundaries and complex structures, while hierarchical clustering provides a hierarchy of regions at different scales. Clustering methods can also be combined with other segmentation techniques to improve their performance.

3) Region-Growing: Region-growing is a method for image segmentation that starts with a set of seed points and iteratively grows the region by adding adjacent pixels or regions that meet certain criteria. In this literature review, we will discuss some region-growing methods and their applications in image segmentation.

a) Seeded Region Growing: One of the simplest region-growing methods is the threshold-based region-growing algorithm. This algorithm starts by selecting a seed pixel and checking whether its neighbouring pixels have intensity values within a certain threshold. If a neighbouring pixel meets the threshold criteria, it is added to the region, and its neighbours are checked in turn. The process is repeated until all neighbouring pixels have been checked. The threshold-based region growing algorithm is effective for images with clear intensity boundaries but may fail for images with complex structures and noise.

The mathematical formula for threshold-based(seeded) region growing can be expressed as:

$$R = \{p \in \Omega | d(p, s) \leqslant \delta\}$$

Where R is the segmented region, Ω is the set of all pixels in the image, p is a pixel in R, s is the seed point, and δ is the threshold that controls the similarity criterion.

b) Region merging: Region merging is another region-growing method that involves starting with an over-segmented image and iteratively merging neighbouring regions based on their similarity. The similarity criterion can be based on a variety of properties, such as colour or texture.

The mathematical formula for region merging can be expressed as:

$$(R_i, R_j) = \max_{p_i \in space; R_i, p_j \in space; R_j} \{d(p_i, p_j)\}$$

Where S is the similarity measure between regions R_i and R_j, d is the distance measure between pixels, p_i is a pixel in R_i, and p_j is a pixel in R_j.

c) Watershed: Watershed segmentation is a region-growing method that is based on the idea of simulating the flow of water over the image. The algorithm starts by identifying local minima in the image and treating them as seeds. The image is then flooded from these seeds, and the resulting regions are merged based on their similarity.

d) Fuzzy Region growing: Region-growing methods can also be used with more advanced techniques such as fuzzy logic and Markov random fields. Fuzzy region growing is a method that uses fuzzy logic to determine the membership of a pixel to a region. It considers both the similarity of the pixel intensity values and the spatial proximity to neighbouring pixels. Markov random fields (MRF) are a probabilistic modelling framework that uses local dependencies between pixels to segment the image into regions. MRF-based region-growing algorithms use a probabilistic model to determine whether a pixel belongs to a region and iteratively adjust the model parameters until convergence.

The mathematical formula for fuzzy region growing can be expressed as:

$$w(p_i) = \frac{1}{1 + \left(\frac{d(p_i, s)}{\theta}\right)^2}$$

Where $w(p_i)$ is the membership value of pixel p_i, $d(p_i, s)$ is the distance measure between pixel p_i and the seed point s, and θ is a parameter that controls the degree of fuzziness.

The algorithm starts with a seed point and assigns a membership value to it of 1. It then iteratively adds neighboring pixels with the highest membership values to the region until no more pixels can be added. The algorithm can be stopped based on a variety of criteria, such as a maximum number of iterations or a threshold for the membership values.

Fuzzy region growing can produce more robust segmentations than traditional region growing methods because it allows for pixels that are partially similar to the seed to be included in the region. However, it can also be computationally intensive because it requires the calculation of membership values for every pixel in the image.

In conclusion, Region-growing methods can also be applied to multi-modal images, where each pixel has multiple intensity values corresponding to different imaging modalities. Multi-modal region growing algorithms use similarity measures that take into account the intensity values of all modalities. These algorithms can produce more accurate segmentations than single-modality methods and are useful in medical imaging applications where different modalities such as CT and MRI are used.

C. Altas-based Method

1) Surfaced Based Method: Surface-based methods have also been used for medical image segmentation, where the goal is to partition an image into regions of interest that correspond to different anatomical structures. In this literature review, we will discuss the principles of surface-based methods in medical image segmentation and their applications.

Surface-based methods for medical image segmentation typically involve the following steps: first, the image is pre-processed to remove noise and artifacts. Next, the image is segmented using thresholding or other image processing method to create a rough segmentation of the image. Then, a surface mesh is created from the segmented image using surface extraction techniques. Finally, the surface mesh is refined and segmented into regions of interest using algorithms such as region growing, graph cuts, or deformable models.

One application of surface-based methods in medical image segmentation is in the segmentation of the cortical surface of the brain from MRI images. Surface-based methods have been used to segment the cortical surface and subcortical structures of the brain, allowing for the identification of different brain regions and the study of changes in brain structure and function.

Surface-based methods have also been used for the segmentation of bones from CT scans. Surface-based methods have been used to segment the surface of bones, allowing for the measurement of bone density and the identification of regions of interest for surgical planning.

In addition to their applications in brain and bone segmentation, surface-based methods have been used for the segmentation of other anatomical structures, such as the heart and blood vessels. Surface-based methods have been used to segment the surface of the heart from MRI images, allowing for the study of cardiac function and the identification of regions of interest for surgical planning.

Overall, surface-based methods have proven to be a powerful tool for medical image segmentation. These methods allow for the creation of accurate surface meshes from medical images, which can be used for the segmentation of different anatomical structures. Surface-based methods have found ap-

plications in the segmentation of the brain, bones, heart, and blood vessels, among other structures.

D. Deep learning Approach

In this session, we will introduce some deep learning network structures used for the image semantic segmentation task and some similar segmentation challenges had been made bofore. However, most of the segmentation task was implemented under the 2-dimension task, that is, to identify the objects in the 2D image. But in this project, the given MRI scans are 3D image and to finely segment the tissue from 3D volume, the network shall be trained and make prediction on 3D volumes.

1) Brain Tumour Segmentation: Brain tumour segmentation is an important problem in medical image analysis, as accurate segmentation can help in the diagnosis, treatment planning and monitoring of brain tumours. Over the past few decades, several research efforts have been made towards the development of brain tumour segmentation methods. In this literature review, we will discuss the Brain Tumour Segmentation (BraTS) project, one of the most comprehensive and widely used benchmarks for brain tumour segmentation.

The BraTS project was initiated in 2012 and has since been held annually as a challenge for researchers in the field of medical image analysis. The challenge provides a standardized dataset of brain magnetic resonance imaging (MRI) scans with ground truth segmentation labels for the four major types of brain tumours: glioblastoma, astrocytoma, oligodendroglioma and meningioma. The challenge is divided into two parts: a training dataset with 285 MRI scans and a validation dataset with 66 MRI scans. The challenge participants are required to develop algorithms for brain tumour segmentation and submit their results for evaluation.

Several approaches have been proposed for brain tumour segmentation using the BraTS dataset. The most successful methods employ deep learning techniques such as convolutional neural networks (CNNs), which have been shown to produce state-of-the-art results. CNNs are capable of learning the complex relationships between image features and tumour labels, thereby producing accurate segmentations. Several variants of CNNs have been proposed, including 3D CNNs, which can model the spatial information of the MRI scans.

Other approaches include multi-modal segmentation methods, which combine information from different imaging modalities such as T1-weighted, T2-weighted and FLAIR MRI scans. These methods have been shown to produce more accurate segmentations than single-modal methods, as they can account for the variability in tumour appearance across different imaging modalities.

Some researchers have also proposed the use of graph-based methods for brain tumour segmentation. These methods model the image as a graph, where each pixel is a node, and the edges between the nodes represent the spatial relationships between them. The segmentation problem is then formulated as a graph-cut optimization problem, where the goal is to find the minimum cut that separates the tumour from the normal brain tissue.

The BraTS challenge has contributed significantly to the development of brain tumour segmentation methods, and the best-performing methods have achieved high accuracy in segmenting brain tumours. However, the challenge also highlights the limitations of current segmentation methods, as there is still a significant gap between the performance of automated methods and human experts.

In conclusion, the BraTS project is a comprehensive benchmark for brain tumour segmentation that provides a standardized dataset for evaluating segmentation methods. Deep learning techniques such as CNNs have shown promising results for brain tumour segmentation, while multi-modal segmentation methods and graph-based methods can improve accuracy by incorporating additional information. The challenge has facilitated the development of accurate and robust brain tumour segmentation methods, but there is still room for improvement in automated segmentation methods.

2) Brain Tissue Segmentation: The iSeg challenge [10] is a community-driven initiative that aims to evaluate and compare different methods for segmenting the brain tissues and substructures of 6-month-old infants based on MRI scans. The challenge was first introduced in 2012 and has since been held every year as part of the MICCAI conference.

The iSeg challenge provides a standardized dataset of MRI scans of 6-month-old infants, which includes T1-weighted, T2-weighted, and diffusion-weighted images. Participants in the challenge are tasked with developing and implementing algorithms that can accurately segment the different brain tissues and substructures in these images.

The iSeg challenge has been instrumental in advancing the field of medical image segmentation, particularly in the context of neonatal and infant brain imaging. It has also helped to establish benchmarks for evaluating the performance of different segmentation algorithms and has facilitated the sharing of knowledge and techniques among researchers and practitioners in the field.

E. Network Structure for Image Segmentation

in this session, this paper will introduce some typical deep learning network architecture for image segmentation and briefly introduce the general evaluation metrics of the image segmentation result. Prior

1) FCN: Fully Convolutional Networks (FCNs) have revolutionized the field of semantic segmentation in recent years, and have been successfully applied to a wide range of image segmentation tasks. In this literature review, we will discuss the development of FCNs and their applications in image segmentation.

Before the introduction of FCNs, traditional segmentation methods involved the use of hand-crafted features and pixel-wise classification methods. However, these methods suffered from limitations such as poor generalization to new images and computational inefficiency. FCNs, on the other hand, are end-to-end trainable neural networks that can perform pixel-wise

classification in a single forward pass, making them highly efficient and accurate.

FCNs were first introduced by Long et al [12]. in their seminal paper in 2015. The authors proposed a fully convolutional architecture for semantic segmentation that used only convolutional layers and pooling layers, without any fully connected layers. This allowed the network to accept images of any size and output segmentation maps of the same size as the input image. The network was trained using a pixel-wise cross-entropy loss function, which minimized the difference between the predicted segmentation map and the ground truth segmentation map.

Since the introduction of FCNs, several improvements have been made to the original architecture to increase its accuracy and efficiency. For example, Chen et al. proposed the use of skip connections in the network, which allowed it to retain high-resolution features from the earlier layers of the network. This helped to overcome the problem of loss of spatial information that occurs during pooling operations in the original FCN architecture.

Another improvement is the use of dilated convolutions, which allows the network to increase the receptive field of the convolutional layers without increasing the number of parameters. This enables the network to capture both local and global context information, which is important for accurate segmentation.

FCNs have been applied to a wide range of image segmentation tasks, including object segmentation, scene parsing, medical image segmentation, and even video segmentation. In the field of medical image analysis, FCNs have shown promising results in segmenting organs, tumours, and other structures from medical images such as MRI and CT scans.

2) Mask-RNN: Mask R-CNN is a popular image segmentation framework that builds upon the success of region-based convolutional neural networks (R-CNN) and extends them to instance segmentation tasks. Mask R-CNN has shown impressive performance in a wide range of applications such as object detection, instance segmentation, and pose estimation.

The Mask R-CNN framework was first introduced by He et al [14]. in their 2017 paper, "Mask R-CNN." The authors extended the Faster R-CNN architecture, which was originally developed for object detection, by adding a branch for predicting object masks. The authors incorporated a simple, fully convolutional network called the "mask branch" to predict masks for each object proposal generated by the RPN network. The mask branch consisted of a small convolutional network followed by a set of deconvolutional layers to upsample the predicted mask to the same resolution as the input image. The mask branch is trained in parallel with the Faster R-CNN network, optimizing both the detection and segmentation tasks jointly.

Since the introduction of Mask R-CNN, several improvements have been made to further enhance its performance. For instance, the authors introduced the concept of RoIAlign, which overcomes the misalignment problem that arises when using RoIPooling. RoIAlign involves bilinear interpolation to compute the exact values of the features at four regularly sampled locations within each RoI, thus enabling the prediction of more accurate object masks.

Another notable improvement to Mask R-CNN is the introduction of Mask R-CNN with ResNeXt. This architecture replaces the original ResNet backbone network with the ResNeXt architecture, which has shown superior performance in image classification tasks. The authors further improved the performance by introducing a decoupled training scheme, where the Mask R-CNN and ResNeXt networks are trained separately and then fine-tuned together.

Mask R-CNN has been applied to a wide range of computer vision tasks, including object detection, instance segmentation, and pose estimation. It has been used in applications such as autonomous driving, face detection, and surgical tool tracking. Mask R-CNN has also been applied to medical image analysis tasks such as brain tumor segmentation, where it has shown promising results in accurately segmenting tumors from MRI scans.

In conclusion, Mask R-CNN is a powerful and flexible image segmentation framework that has shown impressive performance in various computer vision tasks. The incorporation of the mask branch enables the accurate segmentation of object instances, and the use of RoIAlign further improves the quality of the predicted masks. The use of ResNeXt has further improved the performance of Mask R-CNN, making it a popular choice for many applications. With ongoing research and development, Mask R-CNN is expected to continue playing a significant role in the field of image segmentation and other computer vision tasks.

3) UNET: U-Net is a popular convolutional neural network (CNN) architecture that has been widely used in biomedical image segmentation. It was first introduced by Ronneberger et al [**?**]. in their 2015 paper, "U-Net: Convolutional Networks for Biomedical Image Segmentation."

The U-Net architecture is designed for semantic segmentation, which involves classifying each pixel in an image into different categories. It is a fully convolutional network, which means that it can take inputs of different sizes and produce outputs of the same size. The U-Net architecture is composed of two parts: the contracting path and the expansive path.

The contracting path is similar to the traditional CNN architecture, where a series of convolutional layers are used to extract high-level features from the input image. However, in U-Net, the convolutional layers are arranged in a way that reduces the spatial dimension of the input image while increasing the number of feature channels.

The expansive path is designed to produce a segmentation map that has the same spatial dimensions as the original input image. The expansive path consists of a series of up-convolutional layers, which are used to increase the spatial resolution of the feature map. Additionally, the expansive path uses skip connections to concatenate the feature maps from the corresponding contracting path layers, enabling the network to retain more spatial information.

In this literature review, we will discuss the different layers that are used in the UNET architecture.

a) Convolutional Layer: Convolutional layers are the primary building blocks of convolutional neural networks. They perform a convolution operation on the input data and apply a set of filters to the input data to extract features. In the UNET architecture, the convolutional layers are used to extract the features from the input images. Max Pooling Layer: Max pooling layers are used to downsample the input data. They reduce the spatial dimension of the data by taking the maximum value of each non-overlapping subregion of the input data. In the UNET architecture, the max pooling layers are used in the contracting path to reduce the spatial dimensions of the feature maps.

b) Upconvolutional Layer: Upconvolutional layers, also known as transposed convolutional layers, are used to increase the spatial dimensions of the data. They perform a reverse convolution operation and apply a set of filters to the input data to increase the spatial dimensions of the data. In the UNET architecture, the upconvolutional layers are used in the expansive path to increase the spatial dimensions of the feature maps.

c) Concatenation Layer: Concatenation layers are used to combine the feature maps from the contracting path with the feature maps from the expansive path. They concatenate the feature maps along the depth axis to create a merged feature map. In the UNET architecture, the concatenation layers are used to combine the feature maps from the contracting path with the corresponding feature maps from the expansive path.

d) Dropout Layer: Dropout layers are used to prevent overfitting in neural networks. They randomly drop out a fraction of the neurons during training to prevent the network from relying too heavily on any single neuron. In the UNET architecture, dropout layers are often used to prevent overfitting. In conclusion, the UNET architecture is a powerful and popular deep learning architecture that is widely used in image segmentation tasks. The architecture consists of a contracting path, followed by an expansive path, and uses a variety of layers, including convolutional layers, max pooling layers, upconvolutional layers, concatenation layers, and dropout layers. Each layer has its unique function in the architecture, and the choice of layers depends on the specific task being performed.

The U-Net architecture has been shown to achieve state-of-the-art performance in various biomedical image segmentation tasks. For example, it has been used to segment nuclei, cells, and tumors in microscopy images, as well as organs and lesions in medical images such as CT and MRI scans. One of the main advantages of U-Net is that it can achieve accurate segmentation results even when the training data is limited.

Several variations of the U-Net architecture have been proposed to improve its performance. For instance, the authors of the original U-Net paper introduced a modified U-Net architecture that uses batch normalization and residual connections to further improve its performance. Other modifications include introducing attention mechanisms, adding dense connections, and using dilated convolutions.

In conclusion, U-Net is a widely used CNN architecture that has shown impressive performance in biomedical image segmentation. Its fully convolutional nature allows it to handle inputs of different sizes, and the use of skip connections enables it to retain more spatial information. With ongoing research and development, U-Net is expected to continue playing a significant role in the field of image segmentation and other computer vision tasks.

4) UNET Variants: U-Net is a popular architecture for biomedical image segmentation, but several variants have been proposed to improve its performance further. In this literature review, we will discuss some of the most widely used variants of the U-Net architecture.

a) ResUNet: ResUNet is a variant of U-Net that incorporates residual connections to improve the training and performance of the network. Residual connections allow the network to skip over unnecessary layers, making it easier to train deeper neural networks. ResUNet has been shown to improve the performance of U-Net in various medical image segmentation tasks.

b) Attention U-Net: Attention U-Net is another variant of U-Net that incorporates attention mechanisms to improve the performance of the network. Attention mechanisms enable the network to focus on the most relevant parts of the image, making it more efficient and accurate. Attention U-Net has been shown to achieve state-of-the-art performance in various medical image segmentation tasks.

c) Dense U-Net: Dense U-Net is a variant of U-Net that uses dense connections between layers. Dense connections allow the network to reuse features learned in previous layers, enabling the network to learn more complex representations of the input image. Dense U-Net has been shown to improve the performance of U-Net in various medical image segmentation tasks.

d) TransUNet: TransUNet is a variant of U-Net that uses transformers to process the feature maps produced by the network. Transformers have been widely used in natural language processing, but TransUNet is one of the first attempts to apply transformers to image segmentation. TransUNet has been shown to achieve state-of-the-art performance in various medical image segmentation tasks.

e) U-Net++: U-Net++ is a variant of U-Net that uses a nested architecture to improve the performance of the network. U-Net++ uses multiple levels of nested U-Net architectures, each with a different level of feature abstraction. U-Net++ has been shown to improve the performance of U-Net in various medical image segmentation tasks.

In conclusion, U-Net is a popular architecture for biomedical image segmentation, and several variants have been proposed to improve its performance further. ResUNet, Attention U-Net, Dense U-Net, TransUNet, and U-Net++ are some of the most widely used variants of the U-Net architecture, each with its unique strengths and advantages. With ongoing research and development, these variants are expected to continue playing a significant role in the field of image segmentation and other computer vision tasks.

5) Evaluation Metrics: In the field of image segmentation, the evaluation of segmentation results is an essential aspect of assessing the performance of the segmentation algorithms. In this literature review, we will discuss four popular evaluation metrics used in image segmentation, including Pixel Accuracy, Intersection Over Union (IoU), Dice Coefficient Similarity (DCS), and F1 Score.

a) Pixel Accuracy: Pixel Accuracy is a simple evaluation metric that measures the percentage of correctly classified pixels in the segmentation result. It is calculated as the number of correctly classified pixels divided by the total number of pixels in the image. However, Pixel Accuracy can be misleading when the object of interest is much smaller than the image's background. Therefore, it is often used in combination with other metrics to provide a more comprehensive evaluation of the segmentation results.

b) Intersection Over Union (IoU): Intersection Over Union (IoU) is a widely used evaluation metric that measures the overlap between the ground truth segmentation and the predicted segmentation. It is calculated as the intersection of the two segmentation masks divided by their union. IoU values range from 0 to 1, where 1 indicates perfect overlap between the two masks. IoU is a robust metric that is suitable for evaluating segmentation results, even when the objects of interest are small and have irregular shapes.

c) Dice Coefficient Similarity (DCS): Dice Coefficient Similarity (DCS), is a commonly used evaluation metric in image segmentation tasks. It measures the similarity between the ground truth segmentation and the predicted segmentation. It is calculated as twice the intersection of the two segmentation masks divided by the sum of the pixels in both masks. DCS values range from 0 to 1, where 1 indicates perfect overlap between the two masks. Like IoU, DCS is a robust metric that is suitable for evaluating segmentation results, even when the objects of interest have irregular shapes.

d) F1 Score: F1 Score is a composite evaluation metric that combines both precision and recall. Precision measures the fraction of correctly classified pixels in the predicted segmentation, while recall measures the fraction of correctly classified pixels in the ground truth segmentation. F1 Score is calculated as the harmonic mean of precision and recall. F1 Score values range from 0 to 1, where 1 indicates perfect segmentation performance.

Pixel Accuracy, Intersection Over Union (IoU), Dice Coefficient Similarity (DCS), and F1 Score are four widely used evaluation metrics used in image segmentation. Each metric has its feature in evaluation, and the choice of metric depends on the requirements of the segmentation task. It is recommended to use multiple metrics in combination to provide a more comprehensive evaluation of the segmentation results.

F. Generating Mesh

Generating a mesh from segmented MRI data is an important step in many biomedical applications, such as patient-specific modeling and simulation. In this literature review, we will discuss some of the most widely used methods for generating a mesh from segmented MRI data.

1) Marching Cubes Algorithm: The marching cubes algorithm is a widely used method for generating a mesh from segmented MRI data. The algorithm takes a voxel-based segmentation as input and produces a triangulated mesh as output. The algorithm works by dividing the volume into small cubes and then applying a set of rules to determine the configuration of each cube. The rules are based on the classification of the voxel values within each cube. The marching cubes algorithm has been widely used in various biomedical applications, including modelling the brain, lungs, and cardiovascular system.

2) Surface Reconstruction Techniques: Surface reconstruction techniques are another widely used method for generating a mesh from segmented MRI data. These techniques work by first extracting a set of surface points from the segmented data and then constructing a mesh that interpolates these points. Some commonly used surface reconstruction techniques include the Poisson surface reconstruction method, which uses a partial differential equation to reconstruct a surface from a set of points, and the Ball-Pivoting Algorithm, which constructs a surface by iteratively adding triangles to a set of surface points.

3) Graph-based Techniques: Graph-based techniques have also been proposed for generating a mesh from segmented MRI data. These techniques work by representing the segmented data as a graph, where the nodes correspond to the surface points and the edges correspond to the connections between the surface points. The graph is then partitioned into regions using clustering algorithms, and a mesh is generated for each region. Some commonly used graph-based techniques include the Watershed Algorithm and the Minimum Spanning Tree Algorithm.

4) Deep Learning Techniques: Recently, deep learning techniques have been proposed for generating a mesh from segmented MRI data. These techniques use neural networks to learn a mapping between the segmented data and the corresponding mesh. Some commonly used deep learning techniques include the Variational Autoencoder and the Generative Adversarial Network.

In conclusion, generating a mesh from segmented MRI data is an important step in many biomedical applications. The marching cubes algorithm, surface reconstruction techniques, graph-based techniques, and deep learning techniques are some of the most widely used methods for generating a mesh from segmented MRI data. Each method has its unique strengths and weaknesses, and the choice of method depends on the specific application and the characteristics of the data being analyzed.

G. Thesis Objective

The objective of the thesis is to find an algorithm for whole head MRI segmentation from a deep learning approach. The segmentation scope is a loadable weighted learning outcome that can make prediction result of a labelled whole head MRI.

Considering the computation and complexity of the model, the estimated goal in the project will only segment five tissues in the head modelling, which are skin, skull, CSF, GM, and WM. The prediction result of a head MRI model is also expected to be built into a three-dimensional mesh model that can be used for further modelling works.

1) Unique: Put this in project we will use only seven headset with Unet Why Unet is the best candidate

H. Proposed Method

1) Choose of Activation Function: Activation functions are a crucial component of artificial neural networks. They introduce non-linearity into the neural network and help it to model complex relationships between input and output. In this literature review, we will discuss some of the most widely used activation functions in neural networks.

a) Sigmoid Activation Function: The sigmoid activation function is one of the earliest and most widely used activation functions. It maps the input to a range between 0 and 1, making it suitable for binary classification problems. However, it suffers from the problem of vanishing gradients, where the gradient of the function becomes very small, making it difficult for the neural network to learn.

b) Rectified Linear Unit (ReLU) Activation Function: The ReLU activation function is currently one of the most widely used activation functions. It is a piecewise linear function that sets all negative inputs to zero and leaves positive inputs unchanged. This activation function is simple, fast, and has been found to be very effective in deep neural networks. However, it suffers from the problem of dead neurons, where neurons get stuck in the negative side and no longer contribute to the network's output.

c) Softmax Activation Function: The softmax activation function is commonly used in the output layer of neural networks for multiclass classification problems. It maps the inputs to a probability distribution over the classes. This activation function is widely used in natural language processing and computer vision applications.

The sigmoid activation function, ReLU activation function, and softmax activation function are some of the most widely used activation functions in neural networks. Each activation function has its unique strengths and weaknesses, and the choice of activation function depends on the specific problem being solved and the characteristics of the data being analyzed.

2) Loss Function: In deep learning, loss functions are used to measure the difference between the predicted output and the ground truth label. The choice of loss function depends on the task at hand, such as classification, regression, or segmentation. In this literature review, we will discuss the popular loss functions used in deep learning and their characteristics.

a) Mean Squared Error (MSE): MSE is a common loss function used in regression tasks, where the goal is to predict a continuous value. It measures the average squared difference between the predicted output and the ground truth label. MSE is sensitive to outliers and can result in large gradients during training.

b) Binary Cross-Entropy (BCE): BCE is a common loss function used in binary classification tasks, where the output is either 0 or 1. It measures the difference between the predicted output and the ground truth label using the binary cross-entropy formula. BCE is widely used in training neural networks for binary classification tasks.

c) Categorical Cross-Entropy (CCE): CCE is a regular loss function used in multi-class classification tasks, where the output is one of several possible classes. It measures the difference between the predicted output and the ground truth label using the categorical cross-entropy formula. CCE is widely used in training neural networks for multi-class classification tasks.

d) Dice Loss: Dice Loss is a loss function commonly used in segmentation tasks, where the goal is to predict a binary mask indicating the presence or absence of an object in an image. It measures the similarity between the predicted and ground truth masks using the Dice coefficient. Dice Loss is robust to class imbalance and is widely used in medical image segmentation tasks.

e) Focal Loss: Focal Loss is a loss function that addresses the class imbalance problem in classification tasks. It assigns higher weights to misclassified samples, which are harder to classify. Focal Loss is widely used in object detection tasks, where the number of background samples is much higher than the number of object samples.

3) Hyper parameters: Hyperparameters are parameters that are set before the training of a deep learning model and are not learned during the training process. Hyperparameters affect the behavior of the model, such as the learning rate, batch size, and regularization strength. In this literature review, we will discuss some of the popular hyperparameters used in deep learning and their impact on the performance of the model.

a) Learning Rate: The learning rate is a hyperparameter that controls the step size during the gradient descent optimization process. A large learning rate can result in overshooting the minimum, while a small learning rate can result in slow convergence. A common approach to setting the learning rate is to use a learning rate schedule that gradually reduces the learning rate over time.

b) Batch Size: Batch size is a hyperparameter that controls the number of samples processed in each training iteration. A larger batch size can result in faster training times but can also lead to overfitting. A smaller batch size can lead to slower training times but can generalize better.

c) Dropout Rate: Dropout is a regularization technique that randomly drops out neurons during training to prevent overfitting. The dropout rate is a hyperparameter that controls the probability of dropping out each neuron. A higher dropout rate can result in better regularization but can also reduce the model's capacity to learn.

d) Number of Layers: The number of layers is a hyperparameter that controls the depth of the neural network. A deeper network can learn more complex features but can also be prone to overfitting. The number of layers also affects the computational cost of training the model.

e) Activation Functions: Activation functions are functions applied to the output of each neuron in the network. The choice of activation function is a hyperparameter that affects the non-linearity of the network. Popular activation functions include ReLU, sigmoid, and tanh.

Hyperparameters play a crucial role in training deep neural networks. The choice of hyperparameters affects the performance of the model and the speed of convergence. The learning rate, batch size, dropout rate, number of layers, and activation functions are popular hyperparameters used in deep learning. The selection of hyperparameters depends on the specific task, the size of the dataset, and the network architecture. A systematic search of the hyperparameter space, such as grid search or random search, can help identify the optimal hyperparameters for a given task.

II. Methods

A. Dataset

This project utilized a dataset comprising seven MRI head scans, which were generously provided by Associate Professor Socrates Dokos. The head MRI scans were sourced from the participants involved in the projects cited as references [15] and

These datasets were instrumental in training and evaluating the performance of the 3D-UNet architecture for multi-class segmentation of the five tissues. The dataset served as a valuable resource in conducting the experiments and assessing the effectiveness of the proposed approach.

B. 2D U-Net Network for verification

According to the uncertainty surrounding the trainability of networks with limited datasets in U-Net, we initially opted to use 2D U-Net in our segmentation task for verification. This approach allowed us to prepare the dataset more easily by slicing the volume into images along the sagittal plane of the head MRI, thereby maximising the amount of training data available.

However, we chose to implement 3D U-Net for this project instead of 2D U-Net. While the accuracy (measured by the Dice Coefficient) reached 0.55 using 2D U-Net, the overall accuracy was unsatisfactory, and the volume consistency in prediction was lost. This issue can be attributed to the fact that one dimension of feature extraction was lost when learning in the network. Therefore, despite the initial ease of data preparation using 2D U-Net, the use of 3D U-Net ultimately proved to be more effective in achieving the desired segmentation results.

C. Data-Preprocessing

Before putting the MR image into our training network, there are still works to do in cleaning the given dataset

1) Scale volume and label pre-processing:

a) Scale volume: Among the given headsets, they all have non-labelled background areas with random noise intensity. By removing the unnecessary area, the afterwards sliced sub-volumes would contain more labelled volumes for training and reduce the partition of meaningless information.

b) Label Fix: Missing labels is an issue we faced during data pre-processing. In the case of MRI scans, the label is a volume with voxel values ranging from 0 to 5, where each value corresponds to a specific tissue class (0: Background, 1: Skin, 2: Skull, 3: CSF, 4: GM, 5: WM). However, since the labelling of the dataset was done manually using thresholding and clustering methods, some pixels that correspond to the same tissue may not be labelled. This can lead to inaccuracies in the network's feature extraction process.

To mitigate this issue, we opted a two-step process. First, we sliced the volume along an axis into a series of images. Then, we utilized a 3x3 window to scan over the pixels of each image and assigned the maximum value of the surrounding eight pixels to any mislabelled pixel.

Furthermore, we set certain conditions in the algorithm to ensure continuity among the tissues or to specify actions based on the location of the mislabelled pixels. These measures helped us to effectively address the issue of missing labels and improve the accuracy of our network's feature extraction process.

c) One-Hot encoding: The original label is a volume containing values corresponding to different classes. We convert a categorical label (e.g., a label indicating which class a pixel belongs to) into a one-hot encoded vector representation. The label with shape $(Length, Width, Height)$ with values(labels) from 0 to 5, is converted into shape $(Num\_Classes, Length, Width, Height)$ with value 0,1 in each class.

One-hot encoding is a way of representing categorical data in a format that is suitable for input to a neural network. One-hot encoding is useful because it allows the neural network to treat the classes as separate entities rather than as a single continuous variable. This can help the network to learn more easily and accurately distinguish between different classes during training.

2) Sub-volume Generated and Data Augmentation: To reduce the network size and ensure the feature extraction in the network encoder is detailed enough. We generate the sub-volumes size(in pixel) $(80, 80, 64)$ from the scaled volume size(in pixel) $(240, 240, 192)$. The sub-volumes are generated in two ways, equally sliced along each axis and selected at random. Equally sliced along the axis ensures the edges and all the information of the whole volume are trained by the network at least once, while randomly selected sub-volumes ensures other parts of tissues' continuity.

To augment the training data in this study, typical methods such as expansion and rotation were employed. The headset in the MRI scans was expanded or shrunk by 7% to generate new sub-volumes. These sub-volumes were used as a new headset for obtaining additional data. Additionally, a rotation method was applied by slightly rotating the headset 10 degrees clockwise along the sagittal plane. Randomly rotating the image with a large range of angles was not considered as it may interrupt the direction of the images and the consistency of the features during training. Similarly, flipping the images

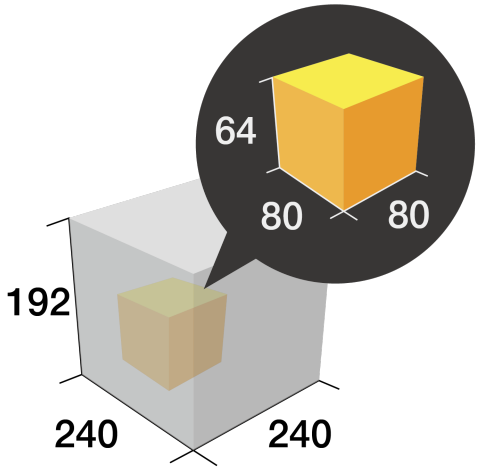


Fig. 1: Demostration of Generating a Subvolume from MRI Scan Volume, unit in pixel

vertically or horizontally was not applicable in this case due to the same reason.

Randomly selected sub-volume can be a method of data augmentation, thereby reducing the imbalance between the classes in data. Take this project as an example, CSF and GM have lower partitions compared to other tissues. We set up a threshold defined by the ratio of CSF and GM to the whole sub-volume, to ensure the captured sub-volume has enough information on CSF and GM to enhance the network's prediction capability of these two classes.

3) Data Distribution: With the given seven headsets, we distributed the headsets into training, validation and test set.

a) Training Set: The training set is utilized to train the convolutional neural network model. During the training process, the model learns from the data within the training set by adjusting its parameters according to the backpropagation of the loss value. This allows the model to continually improve its performance as it learns to better understand the underlying patterns within the data.

b) Validation Set: The purpose of the validation set is to evaluate the performance during training and to tune the hyperparameters of the model. By evaluating the model's performance on the validation set, we can identify any potential issues, such as overfitting, and adjust the hyperparameters accordingly. The model's performance on the validation set is not used to update the model's parameters. This is because the validation set is typically seen by the model during training, and using its performance to update the model's parameters would introduce bias into the model. Therefore, the validation set is used solely for evaluation and tuning purposes.

c) Test Set: The test set is used to evaluate the final performance of the model after training. It is an independent portion of the dataset, separate from the training and validation sets. The test set is used to simulate the model's performance on new, unseen data, and it allows us to determine the model's prediction ability to new/external data.

The purpose of splitting the dataset into these three sets is to ensure that the model adapts well to new data and is not just memorizing the training data. The training set is used to train the model, the validation set is used to fine-tune the model's hyperparameters, and the test set is used to evaluate the final performance of the model.

For our study, a total of seven headsets were used, with five being assigned to the training set, one to the validation set, and one to the test set. Sub-volumes were generated after the distribution, ensuring that the sub-volumes from each headset were not mixed with others. Note that the sub-volumes from the test set were not randomly generated. Further details regarding the reconstruction method are provided in the dedicated session.

D. 3D U-Net

The fully convolutional network(CNN) architecture we applied in this project is UNET proposed by Ronneberger in 2015, for the task of medical image segmentation. [**?**]. The original structure of UNET proposed is like in Figure 2, it has a U-shape and that's what it was named after. The structure of UNET consists of two main parts: encoder with downsampling and decoder with upsampling. The encoder extracts the features of the input images through convolutional layers and max-pooling layers; while the decoder up-samples and recovers the spatial information from the extracted feature maps.

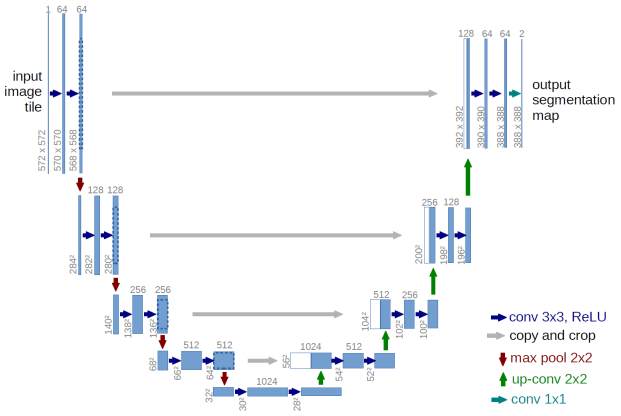


Fig. 2: The U-NET Architecture, Origin Figure Adapted from [**?**]

The encoder part of the U-Net structure is similar to that of typical convolutional neural networks(CNN) like VGG, ResNet., etc. The encoder par is composed of a series of convolutional blocks and max-pooling layers. The convolutional

layers extract the low- and high-level features from the input images, while max-pooling layers reduce the spatial resolution of the feature maps.

The decoder part consists of a series of up-convolutional layers, followed by concatenation operations with corresponding feature maps from the encoder part. The up-convolutional layers are responsible for up-sampling the feature maps and returning the matrix to the input original size. In addition, the concatenation operations help the up-sampling blocks preserve the spatial resolution to rebuild the information that was lost during the down-sampling process of the encoder. The segmentation results, therefore, can be compared with the ground truth in every pixel.

After the proposal of UNET, numerous expansion forms of U-Net have been published. These include U-Nets with different architectures as the backbone, such as VGG, ResNet, and EfficientNet, which perform encoding and downsampling independently. Additionally, some expansions incorporate self-attention gates, such as Attention U-Net [22] and TransUNet [23]. These more complex structures have demonstrated improved performance over typical U-Nets in large datasets. While they have stronger feature extraction and adaptability to dig deeper into images' features, however, they also require more data in training in order to meet better accuracy.

In our particular case, due to the scarcity of data available, we opted to use a typical U-Net with 3D convolutional blocks. Although the more complex U-Net expansions may have greater potential for performance improvement, the limited amount of data made it challenging to train such models effectively. Therefore, using a typical U-Net was a more suitable choice for our project. The 3D Unet architecture applied in this project as shown in Figure 3 with input size $(80, 80, 64)$

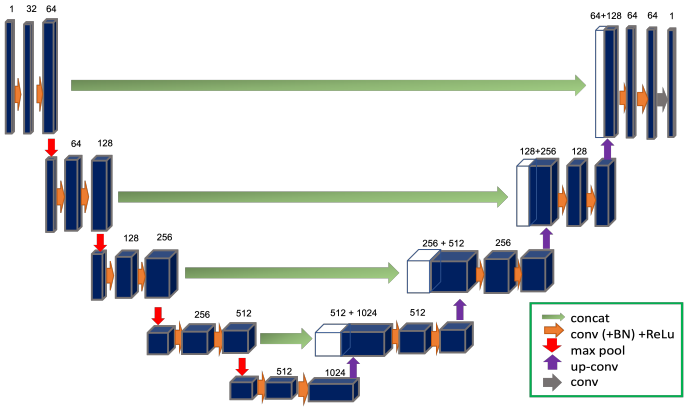


Fig. 3: The 3D U-NET Architecture applied in this project

E. Training Parameters

1) Optimiser: The optimisation used in this project is Adaptive Moment Estimator(Adam). Adam optimizer is an adaptive learning rate optimization algorithm that combines the benefits of both the Adagrad and RMSProp algorithms. It maintains a running average of the first and second moments of the gradients, and then uses this information to update the parameters of the model.

The Adam optimizer computes adaptive learning rates for each parameter in the neural network. It does this by maintaining two moving averages of the gradient: the first moment (the mean) and the second moment (the uncentered variance). These moving averages are initialized as zero vectors.

At each iteration t, the Adam optimizer computes the gradient of the loss function with respect to the model parameters, and then updates the moving averages:

$$m_t = \beta_1 * m_{t-1} + (1 - \beta_1) * g_t$$

$$v_t = \beta_2 * v_{t-1} + (1 - \beta_2) * g_t^2$$

Here, g_t is the gradient at iteration t, and β_1 and β_2 are hyperparameters that control the decay rates of the moving averages. Typically, β_1 is set to 0.9 and β_2 is set to 0.999.

The Adam optimizer then calculates the bias-corrected estimates of the first and second moments:

$$\hat{m}_t = \frac{m_t}{(1 - \beta_1^t)}$$

$$\hat{v}_t = \frac{v_t}{(1 - \beta_2^t)}$$

Finally, the optimizer updates the parameters using the following formula:

$$\theta_{t+1} = \theta_t - \frac{\alpha * \hat{m}_t}{(\sqrt{\hat{v}_t} + epsilon)}$$

Here, α is the learning rate, and ϵ is a small constant added for numerical stability. The denominator $\sqrt{\hat{v}_t} + \epsilon$ is added to prevent division by zero.

This update rule effectively scales the learning rate for each parameter based on the magnitude and direction of the gradients. In practice, this adaptive learning rate strategy has been shown to be very effective for a wide range of deep learning tasks.

2) Total loss: In this project, a sum of *Soft Dice Loss* and *Focal Loss* is used as a total loss function.

a) Soft Dice Loss: Soft Dice Loss [**?**] is a loss function commonly used in image segmentation tasks to optimize the Dice coefficient.

$$SDL = 1 - \frac{2\sum_{i=1}^{N} p_i g_i w_i + \epsilon}{\sum_{i=1}^{N} p_i^2 w_i + \sum_{i=1}^{N} g_i^2 w_i + \epsilon} \quad (1)$$

where N is the number of pixels, p_i is the predicted probability of pixel i belonging to the foreground class, g_i is the ground truth label of pixel i (1 for foreground, 0 for background), and w_i is a per-pixel weight. The numerator represents the weighted sum of the intersection between the predicted and ground truth segmentation, and the denominator

represents the sum of the predicted and ground truth segmentations, excluding the intersection. The parameter $\epsilon = 0.0001$ is added to avoid division by zero

Soft Dice Loss is commonly used in medical image segmentation tasks, where it is important to accurately identify the foreground (e.g., tumor) while minimizing false positives and false negatives. The per-pixel weight w_i can be used to balance the contribution of each pixel to the loss function based on its importance or difficulty.

b) Focal Loss: The Focal Loss [**?**] is a loss function used in deep learning for imbalanced classification problems, which down-weights the contribution of easy examples and focuses more on hard examples. It can be expressed as:

$$FL(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t) \tag{2}$$

where p_t is the predicted probability of the true class, α_t is the weight assigned to the true class (usually inversely proportional to its frequency), and γ is a tunable parameter that controls the degree of down-weighting. The term $(1-p_t)^\gamma$ down-weights easy examples (where p_t is close to 1) more than hard examples (where p_t is close to 0).

Soft Dice loss optimises the network to preserve image details while focal loss optimises the network to focus on the class with less partition in volume. In the results session, a comparison between focal loss as a total loss and soft dice loss only shows that focal loss promotes the prediction accuracy of the CSF, GM and WM.

3) Evaluation Metrics: There are many evaluation metrics for segmentation tasks in deep learning. The Dice Coefficient Similarity is chosen to be used in the network during the training, while Intersect over Union, Mean Intersect over Union and Frequently-weighted Intersection over Union are used as additional metrics to evaluate the result

a) Dice Coefficient Similarity: Dice Coefficient Similarity (DSC) is used as the evaluation metric in the network. The Dice Coefficient Similarity (DSC) is a widely used evaluation metric in the field of image segmentation, and is often used to assess the performance of deep learning models.

To calculate the dice coefficient similarity, we first need to obtain the predicted segmentation and the ground truth segmentation. These two sets of data are then compared using the following formula:

$$DSC = \frac{2\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{N} w_i + \sum_{i=1}^{N}\sum_{j=1}^{N} w_j} \tag{3}$$

The numerator in this formula represents the intersection between the predicted segmentation and the ground truth segmentation, while the denominator represents the total number of pixels in both sets. The resulting value ranges from 0 to 1, with 1 indicating a perfect overlap between the two sets of data.

To evaluate the performance of a segmentation model using the Dice coefficient, we calculate the coefficient for each class separately, as well as the average coefficient across all classes. This allows us to assess the model's performance in each individual class, as well as its overall performance.

b) Intersect over Union: Intersect over Union(IoU) is defined as the ratio of the intersection between two sets to their union. In the context of segmentation, the intersection refers to the pixels that are correctly classified by both the model and the ground truth, while the union refers to all the pixels that are either correctly classified or misclassified by the model and/or the ground truth. The IoU score for each class is computed by dividing the number of intersecting pixels by the number of pixels in the union of the two masks.

c) Mean Intersect over Union: Mean Intersection over Union (Mean IoU) is a popular performance metric used to evaluate the accuracy of segmentation models in computer vision. It measures the overlap between the predicted segmentation mask and the ground truth mask by computing the intersection over union (IoU) of each class.

Mean IoU is the average of the IoU scores across all classes in the dataset. It provides a measure of how well a segmentation model is able to separate different classes in an image. Mean IoU is a popular evaluation metric because it takes into account both the precision and the recall of the segmentation model, making it a more comprehensive metric than accuracy or F1-score.

In practice, Mean IoU is often used as a benchmark metric for segmentation models, and it is commonly reported in research papers and competitions. While Mean IoU is a useful metric for evaluating the performance of segmentation models, it should be used in conjunction with other metrics and visual inspection of the segmentation results to fully assess the model's performance.

$$Mean\ IoU = \frac{1}{N}\sum_{i=1}^{N}\frac{w_{ii}}{\sum_{j=1}^{N} w_{ij} + \sum_{j=1}^{N} w_{ji} - w_{ii}} \tag{4}$$

where N is the number of classes, and w_{ij} is the number of pixels that are classified as i and j. The numerator represents the number of pixels that are classified as both i and j, and the denominator represents the total number of pixels that are classified as i or j, excluding the ones that are classified as both.

The formula assumes that the classes are mutually exclusive and collectively exhaustive, which means that each pixel belongs to exactly one class, and all possible classes are represented in the segmentation.

d) Frequently Weighted Intersection over Union: Frequently-weighted Intersection over Union (F.W.IoU) is another performance metric used to evaluate the accuracy of segmentation models. It is an extension of the Intersection over Union (IoU) metric that assigns weights to different classes based on their importance, and is particularly useful for imbalanced datasets.

Different from MeanIoU, where all classes are given equal weight, which can be problematic in datasets where some classes are more important than others. Take our case for

example, in the MRI scans, the Class Skin has the highest proportion of the head volume while Class CSF contains the least partition. F.W.IoU addresses this issue by assigning weights to each class based on their partition. These weights are typically defined as the inverse of the frequency of each class in the dataset.

F.W.IoU is calculated by multiplying the IoU score of each class by its weight, and then averaging the weighted IoU scores across all classes. This gives a more accurate representation of the model's performance, particularly in imbalanced datasets where some classes may be significantly underrepresented.

F.W.IoU is a useful metric for evaluating the performance of segmentation models in real-world scenarios, where the importance of different classes may vary depending on the application. It provides a more nuanced evaluation of the model's performance by taking into account the relative importance of each class. However, like with Mean IoU, it is important to use F.W.IoU in conjunction with other metrics and visual inspection of the segmentation results to fully assess the model's performance.

$$F.W.IoU = \frac{\sum_{i=1}^{N} n_i w_{ii}}{\sum_{i=1}^{N} n_i \sum_{j=1}^{N} w_{ij} + \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} - n_i w_{ii}} \tag{5}$$

where N is the number of classes, n_i is the number of pixels in class i, and w_{ij} is the number of pixels that are classified as both i and j. The numerator represents the weighted sum of intersection over union for each class, where the weight is the number of pixels in the class. The denominator represents the total number of pixels that are classified as i or j, excluding the ones that are classified as both, with the weight of each class.

This formula assumes that the classes are mutually exclusive and collectively exhaustive, which means that each pixel belongs to exactly one class, and all possible classes are represented in the segmentation.

In addition to the Dice coefficient, other metrics such as the Mean Intersection over Union (IoU) may also be used to evaluate the performance of segmentation models. By using multiple metrics, we can obtain a more comprehensive evaluation of the model's performance and better understand its strengths and weaknesses.

F. Post 3D Volume Reconstruction

In the data-preprocessing session, we discussed how the sub-volumes were generated from the original image by slicing along each axis and randomly selecting sub-volumes from the volume. However, in the test set, we only sliced the volume along the axis for the purpose of rebuilding the original volume.

After obtaining the prediction labels from the trained network, we reconstructed the original volume by placing the prediction labels into the corresponding coordinates. However, to avoid edge defects, which will be discussed in the results session, we removed some voxels near the edges of the sub-volumes before reconstructing the original volume. This approach helps to ensure the final reconstructed volume maintains the continuity of the label. Figure 4 provides a visual demonstration of how this method was implemented.

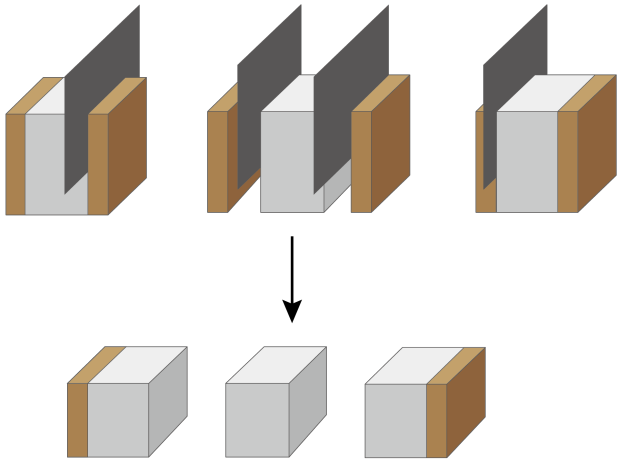


Fig. 4: Remove edges on sub-volumes to get rid of the defects, and combine with other sub-volumes. Brown areas in figure indicate defects near the edge

G. Generate Mesh from software

After performing the prediction label reconstruction, the resulting file was outputted in NIFTI (.nii) format. To generate a mesh from the labeled volume, we utilized the software 3D-Slicer [30]. Mesh volumes were created for each tissue and subsequently exported in STL format. To refine the meshes and resolve any intersecting areas within the volumes, we employed Materialise Mimic [35] software.

III. Results

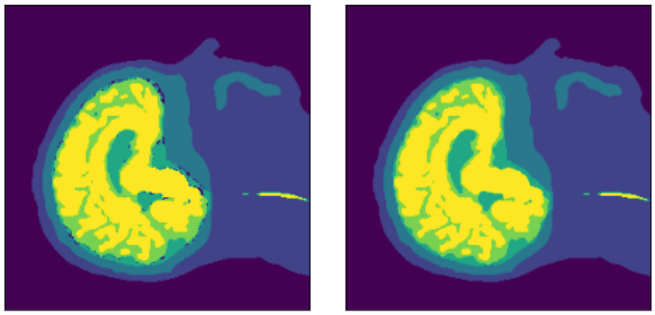
A. 2D Model results

The results of the 2D segmentation task were only for reference. In the 2D network, the input images are the slices along the sagittal plane of the MRI head scans. After removing the background and non-label images, the distribution of the dataset is 896 images in the training set and 185 images in the test set. Because of the lack of training data, the network overfits the training set quickly. The comparison between ground truth and prediction is shown in Fig4.

According to the result, the network ability is limited by the scale of trained data. That's another reason we did not opt for 2D Unet as the learning network for this task.

B. Label Fixing

Figure 5 illustrates the comparison between the original labels and the corrected labels. Note that the missing labels are not corrected after careful examination of the MRI scans. The determination of the missing label possible types was only based on their location, which carries the risk of introducing incorrect labels and negatively affecting the network during the learning process.



(a) Original with missing lables on CSF, GM, and WM (b) Corrected Label using Scanning Window

Fig. 5: Comparison between Before and After Label fixing(index 90 on Sagittal xy-Plane)

C. Training Results

The complexity of the 3D U-Net(in the proposed 4-layer UNET has 65,863,621 parameters)

In this thesis, we have tried multiple attempts on this segmentation task with optimisations applied on each attempt. Four major attempt are chosen and recorded as in TABLE I.

a) 4-layer-UNET: Initially, we trained the learning network using 135 subvolumes as the training set and 27 subvolumes as the validation set. The loss function used in this training was the soft dice loss, and the evaluation metric used was the Dice Coefficient Similarity. This initial training helped to establish a baseline performance for the later improvements.

b) 4-layer-UNET with Data Augmentation: To improve the prediction ability on CSF, GM, and WM. In the second attempt, we have augmented the training dataset from 135 subvolumes to 750 subvolumes, with validation set from 27 to 150. The training data were augmented by adding more subvolumes with greater brain proportion. The result showed the improvement on the increase of IoU on each tissue.

c) 4-layer-UNET with Data Augmentation, Total Loss: In our third attempt, we have implemented additional data augmentation techniques by extracting subvolumes from rotated and expanded MRI scans, as mentioned in the previous Method section. To address the issue of class imbalance, we have used a combined loss function consisting of the soft dice loss and focal loss. Our results show an improvement of approximately 0.02 in the Intersection over Union (IoU) metric for the CSF, GM, and WM classes, with only a minor decrease of 0.007 in the IoU for the Skull class. Although there was a slight drop in the IoU for the Skin class, this is acceptable as it is of relatively less importance compared to the other classes.

d) 5-layer-UNET with Data Augmentation, Total Loss: For our fourth attempt, we aimed to improve the performance of the learning network by exploring a deeper architecture. We added an additional series of convolutional blocks after another maxpooling layer to extract smaller features and evaluate if this would lead to better performance. We used the same dataset and hyperparameters as in our previous attempts.

The results of our experiment indicate that the 5-layer-UNET architecture has similar prediction ability as the previous architecture, with a slightly reduced IoU for the CSF, GM, and WM classes. While this reduction in IoU is not ideal, it suggests that increasing the depth of the learning network may not always lead to significant improvements in performance.

D. Choosing the best model

From the prediction result on the test set shown in TABLE I, we opted the model: the 4-layer-UNET with DataAugmentation and total loss, as the one for later analysis and description. Although the 5-layer-UNET has similar prediction capability and has a slightly better score on F.W.IoU, the prediction accuracy on the minor classes is worse than the 4-layer-UNET, which did not meet the expectation of the more precise of network purpose. The reason that the 5-layer-UNET did not meet the expectation might be the network generalisation ability is constrained by the scarcity of datasets, which will be discussed more in the Discussion session.

As the Figure 6 histogram plot, after the completion of 100 epochs of training, the final loss of the training set reached 0.0327, and the dice coefficient for the training set reached 0.9347; while the validation loss value reached 0.154, and the dice coefficient for the validation set reached 0.8195. It appeared that the model overfit the training set during the training process, with only a minor decrease in loss observed for the validation set after 50 epochs.

The result of the evalutaion on the test set is 0.7042 MeanIoU, the IoU for each tissue is: Skin 0.9399, Skull 0.8296, CSF 0.7428, GM 0.7457, WM 0.7995. The Figure 7 compares the ground truth label with prediction results by tissues

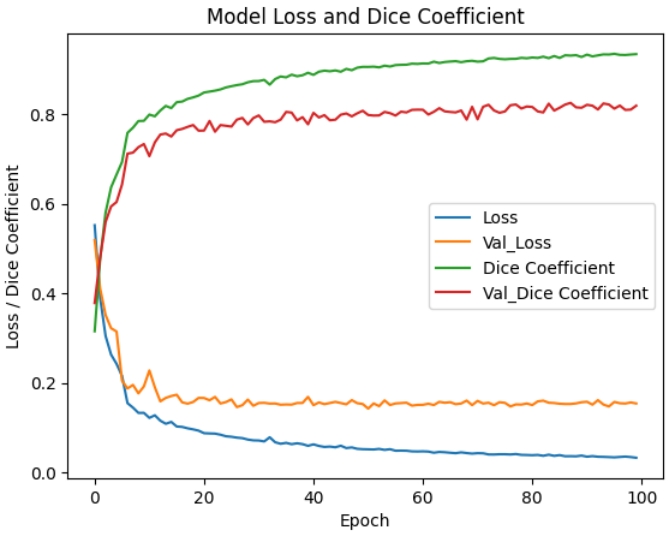


Fig. 6: Network Training Histogram

E. Post 3D image reconstruction

To predict the entire head volume, the first step involved generating a predicted label by using the trained model weights

	1^{st}	2^{nd}	3^{rd}	4^{th}
	4-layer-UNET	*4-layer-UNET with D.A.*	*4-layer-UNET with D.A. use Total Loss*	*5-layer-UNET with D.A. use Total Loss*
Skin(IoU)	0.9316	0.9407	0.9399	0.943
Skull(IoU)	0.7990	0.8222	0.8296	0.831
CSF(IoU)	0.6808	0.7210	0.7428	0.7394
GM(IoU)	0.6936	0.7293	0.7457	0.7426
WM(IoU)	0.7135	0.7778	0.7995	0.7964
MeanIoU	0.6290	0.6830	0.7042	0.7025
F.W.IoU	0.6725	0.7393	0.7442	0.7450

TABLE I: Results Comparison of UNET with Optimisations
*D.A. denotes Data Augmentation

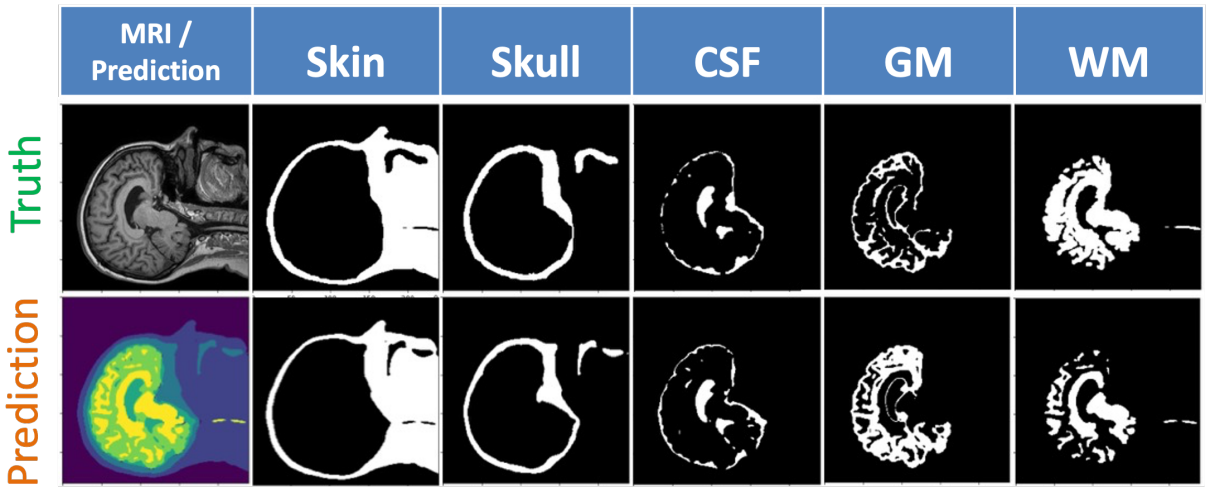


Fig. 7: Comparison of Truth and Prediction Label by Tissue (index 90 on Sagittal xy-Plane)

to predict the sub-volumes of the test set. The predicted label was then relocated according to the original coordinates of the volume to reconstruct the prediction label for the entire head volume.

However, we observed edge defects on the borders of the sub-volumes, where there were not enough pixels or lack of continuity, which led to false predictions. To overcome this issue, we opted to use more subvolumes (125 subvolumes) to replace some pixels to ensure the prediction of each area was predicted under continuity.

Our initial MRI scan size(in pixel) was $(240, 240, 192)$, and the subvolume size(in pixel) was $(80, 80, 64)$, which is perfectly one-third of each side. Ideally, we would only need 27 subvolumes to reconstruct the original scans. However, as shown in Figure 8a and Figure 8b, there was a loss of continuity of tissue at the edges of the subvolumes, and mislabelled areas occurred due to a lack of information.

The results of our experiment show that by using more subvolumes and replacing some pixels, we were able to overcome these challenges and improve the accuracy of our predictions. As shown in Figure 8b, this method helped to solve mis-predictions of background volume outside the skin that were previously mis-predicted as skull. These findings suggest that further exploration of optimization methods for reconstructing the entire head volume may lead to even better results

2^{nd} Result IoU	*Reconstruct using 27 subvolumes*	*Reconstruct using 125 subvolumes*
Skin	0.9407	0.9447
Skull	0.8222	0.8346
CSF	0.7210	0.7316
GM	0.7293	0.7332
WM	0.7778	0.7817
MeanIoU	0.6830	0.6945

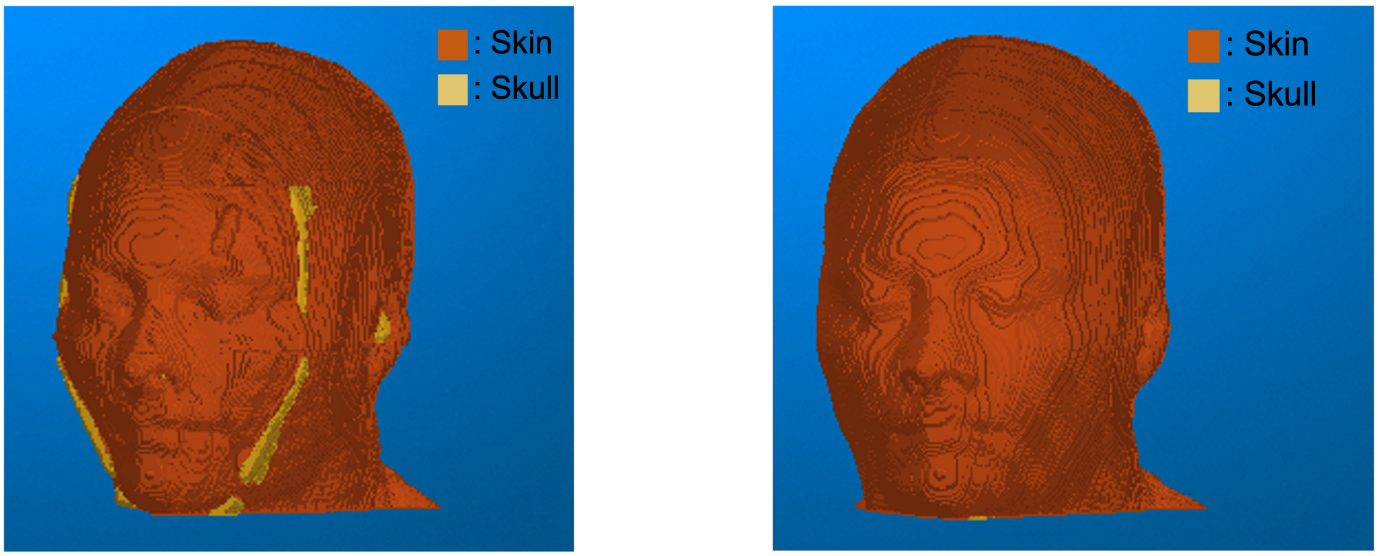
TABLE II: Comparison of Reconstruct Method

With this method applied, we improved the IoU of each tissue as shown in TABLE II. The improvements seems minor on number but it prevented the major mistake of different tissue appear in the wrong place.

Coverage the edge defects by uasdfsing more sub-volumes to intersect the defects pixels. From using 27 to 125 sub-volumes to reconstruct the original size volume. Removing Edge defects promotes the IoU on each tissue

F. 3D Slicer reulst, and MeshLab result, show Mesh (3D), COMSOL

After the reconstruction of the whole head volume, we used the software 3D Slicer [30] to generate the mesh of each tissue then we opted MeshLab to render the finite elemented



(a) Whole Head Label Reconstruction with 27 subvolumes

(b) Whole Head Label Reconstruction with 125 subvolumes

Fig. 8: Comparison of Before and After Removing Edge Defects. View in MATLAB VolumeViewer

mesh surface using software MeshLab [31] in Figure 10 is the transparent view of the whole head mesh stacked by the generated tissue mesh of the predicted label and viewed in Rhino [32] Software. The generated mesh is importable to COMSOL [34] software(as shown in Figure 11) however, the mesh requires further finalise by using mesh fixing software like materialise Mimics [35] to deal with the intersection on the mesh surfaces.

IV. Discussion

A. Prediction Imbalance Between Classes

Class Skin, as the major partition among the classes, is also the class that becomes the primary target for prediction by learning networks. Despite recognizing the inherent disadvantage of this approach, we have addressed it by incorporating focal loss as a combined loss function. This helps to mitigate the imbalanced predictions between the various classes.

Nevertheless, it is inevitable that the learning network will tend to prioritize predicting the major partition class over the minors. This is because the marginal benefits of decreasing the loss value are greater when predicting the major partition, as compared to predicting a class with a smaller partition.

In the later Future Work session, a possible method is proposed that could further reduce the impact of the class imbalance issue.

B. Edge Defects

In the Method section's reconstruction description, we noted false predictions occurring at the edges of sub-volumes. These false predictions may be attributed to the discontinuity of the image and truncated tissue at the edges, which provide insufficient information for the network to accurately predict the output. We have termed these types of false prediction defects as "edge defects."

These edge defects persist regardless of the trained network structure employed for prediction and result in inconsistencies when relocating the predicted label to its original coordinate. Given the inevitability of this characteristic, we opted to use additional sub-volumes to ensure that each voxel within the headset is predicted with continuity.

C. Limitation

1) DataSet: The generalisation of the network was restricted by the limited amount of training data available. Only seven headset MRI scans were provided for this project. Although we attempted to augment the training dataset by generating more subvolumes and implementing data augmentation methods (such as rotation and expansion), the network's performance remained constrained by the size of the dataset and could not finely perform inadequately when predicting on a new or foreign test dataset. This limitation can be attributed to the high consistency among the subvolumes, as they were all generated from the same headset.

Additionally, the network's performance is susceptible to degradation due to the lack of variety between the headset. When new data from a different headset is added to the training set, the network may be easily affected, resulting in decreased accuracy in prediction. For instance, in our project's final stages, we attempted to train the network with six headsets, while reserving the seventh as the test set. However, the accuracy of the prediction was not satisfactory when compared to the network trained with five headsets only.

If additional headsets were provided for this project, it would also yield several benefits. Firstly, as mentioned in previous sessions, a larger dataset would allow for the use of advanced Unet-based learning architecture such as the Attention-Unet, a Unet based network structure with self-attention mechanism, or TransUnet, a combined structure of

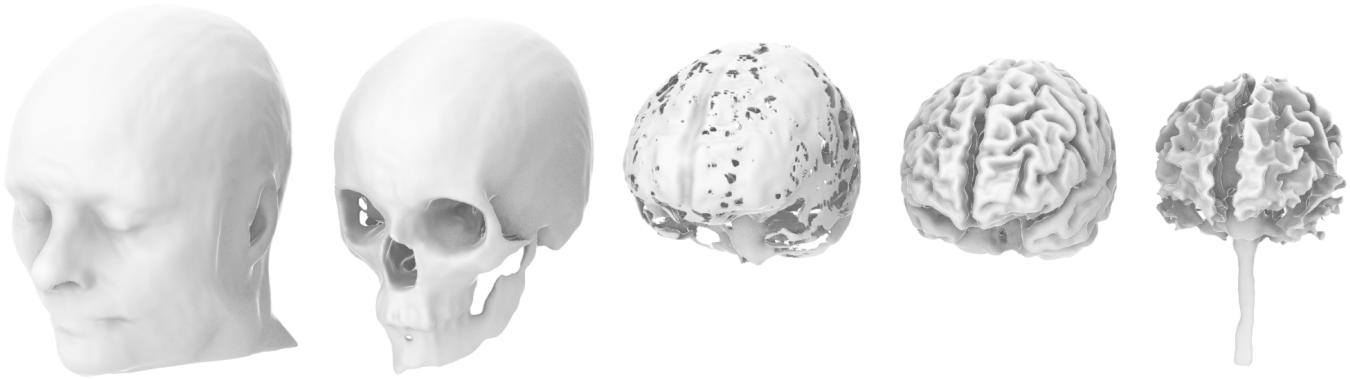


Fig. 9: Tissues Mesh Generated according to Prediction.
From left to right: Skin, Skull, CSF, GM, WM

Fig. 10: Unified Mesh transparent view in *Rhino*

	Tissues				
	Skin	Skull	CSF	GM	WM
Precision	0.963	0.795	0.694	0.577	0.954
Recall	0.933	0.827	0.644	0.855	0.644

TABLE III: Precision and Recall by Tissues

the Transformer [**?**] and Unet, both have demonstrated superior performance compared to standard Unet networks in previous studies [23].

In conclusion, obtaining more headsets for training would be beneficial in achieving further improvements in this project's performance.

2) Learning Network: For this project, the U-Net learning network was chosen due to its simplicity, widespread usage, and adaptability to small datasets. However, it is important to note that Unet has its limitations. For example, the paper on TransUnet highlights Unet's over-predicting and under-predicting issues in liver segmentation. Similarly, our project encountered similar issues. In Figure 7, it is evident that the network is over-predicting the grey matter and under-predicting the white matter when compared to the ground truth.

This is obviously or can be calculated using the indicator called $precision$ and $recall$.

$Precision$ is defined as

$$Precision = \frac{TP}{TP + FP}$$

,and $Recall$ is defined as

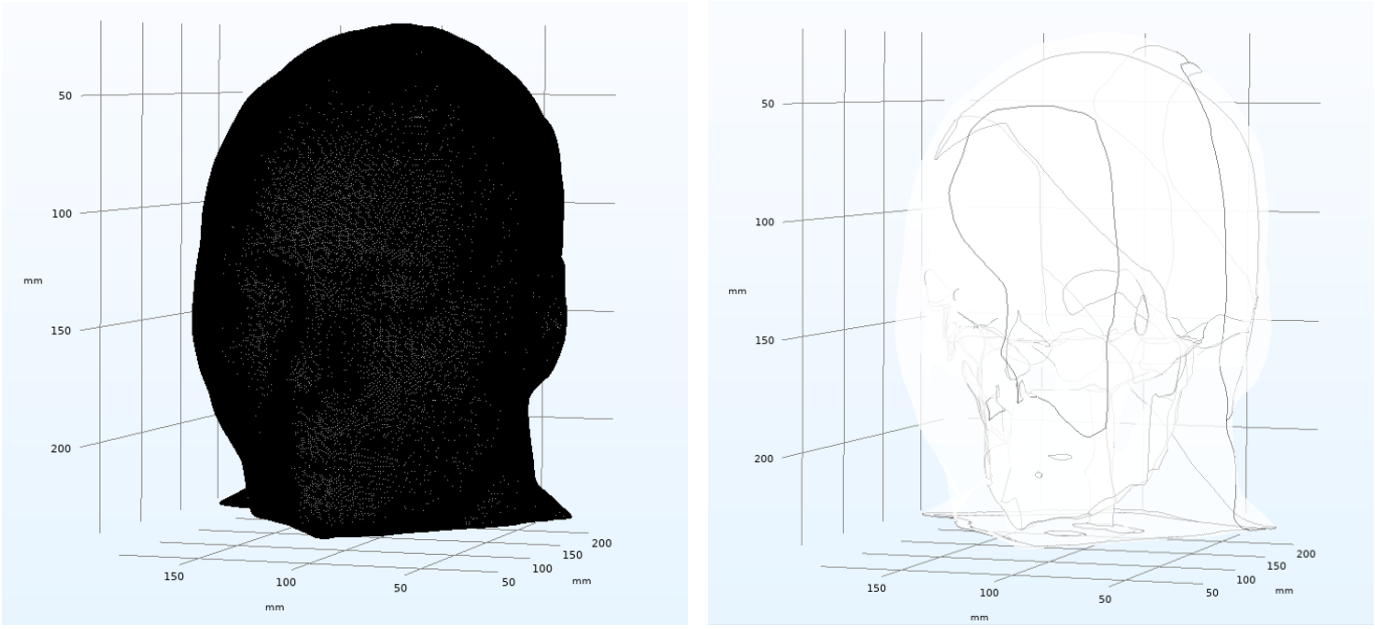
$$Recall = \frac{TP}{TP + FN}$$

,where TP is true positive, FP is false positive, TN is false negative.

Ideally, a finely classified prediction should have a balance between precision and recall. However, Table III shows that for the grey matter class, recall is higher than precision, while for the white matter class, precision is higher than recall.

From the TABEL III, the false positive rate of 73.3% for the true positive in grey matter while the false negative rate is 55.2% for the true positive in white matter. These errors can significantly impact the model's accuracy in physics stimulation experiments as grey matter and white matter have different conductivities. If these prediction results were used as the model for transcranial stimulation, the results could be inaccurate.

Additionally, the over-predicting and the under-predicting issue may not always occur in the same classes. For instance, in the validation set (6th headset), the over-predicting and



(a) Imported Mesh viewed in COMSOL

(b) Transparent View of the Mesh in COMSOL Geometry

Fig. 11: Generated Mesh viewed in COMSOL

under-predicting issues were observed in the Skull and CSF classes. Further investigation is required to determine the exact cause of these issues, but it is currently speculated that they may be due to limitations in the network structure.

D. Suggest Improvements

1) Include External Data for Training: One of the most challenging and critical aspects of this project is the complexity of the CSF, grey matter, and white matter. These three classes not only differ in intensity and volume, but also in the proportion of their volume relative to the entire head scan.

In this project, we subtracted more subvolumes containing mainly CSF, grey matter, and white matter from the headset to improve the learning ability on these three classes. This approach led to a direct improvement in IoU for these three classes, as shown in TABEL I. However, another available approach to improve the model's performance is to use external data to augment the training set.

For instance, the 3DBrainTissueSegmentation dataset on Kaggle [37] contains 726 brain MRI scans labeled with CSF, grey matter, and white matter. These brain MRI scans have the same voxel size of $(1mm \times 1mm \times 1mm)$ as the headset used in this project, making it feasible to add external data to the training process. By incorporating external data, we can improve the network's generalisation to a broader range, ultimately improving its accuracy and ability.

2) Proper Initial Weights: Setting appropriate initial weights can facilitate faster convergence and shorten the learning time for the network. However, in this project, due to the limited amount of data, there is a risk of encountering the gradient vanishing issue during training if the initial weights are not set properly.

If this project were to be expanded to a larger network, it would be important to carefully consider the selection of initial weights to improve training efficiency and potentially yield better results. By fine-tuning the initial weights, the network can learn more effectively from the limited amount of available data and improve its generalisation to unseen headsets.

E. Future works

In this session, more future works of this project that can be pushed further are proposed, and some possible methods can be applied to improve the algorithm.

1) Mesh Finialisaing: Due to the time constraints for this project, the generated mesh was not finalised enough for further simulation in COMSOL. As shown in Figure 11b, COMSOL software could not define the proper domains and boundaries of the mesh, because they were not properly fixed and defined before import.

2) Physics Stimulation of the Prediction Model: Based on the proposed algorithm in this paper, the tissue mesh of the headset represents the stage of the result. However, due to time constraints, the mesh was not finalised to eliminate the inside intersections and overlaps. Further refinement of the mesh is necessary to allow for stimulation in modelling software such as COMSOL or ANSYS. This will enable a comparison of simulation results between the ground truth and predictions to be conducted.

3) More Data, More Classes: One of the project's limitations, as noted earlier, is the scale of the dataset available for training. The expectation was to have a more extensive dataset to improve the network's generalisation ability. Another approach to advancing this project is to segment more classes. Currently, the project only considers the five main tissues:

Skin, Skull, CSF, GM, and WM. However, for a more accurate simulation study, there are additional detailed tissues that need to be segmented in the head model, such as fat (included in the skin class in this project), compact bone, and spongy bone in the Skull tissue, and Vertebrae, which have different electrical conductivity (S/m). Incorporating more classes and considering the minority volume of some classes would make this project more challenging.

4) Advanced Network Structure: Given the dataset limitations, another suggested algorithm to consider is one-shot learning [27] or few-shot learning [28] in segmentation work. However, despite these algorithms being useful for limited data for training, the generalizability of the network with limited data for training may still be questionable.

Although U-Net itself can handle small datasets with decent accuracy, some variant U-Net architectures with transfer learning using other networks as a backbone (encoder) are worth exploring. Using a pre-trained Inception, VGG, ResNet, EfficientNet, etc. [29] as the backbone of the U-Net is an encouraging direction for further study in this project.

As mentioned earlier, with the addition of more headsets, a more complex network such as TransUnet can be considered as the main network architecture for this project. However, TransUnet architecture requires additional data from the training set to use in training the meta-learner.

5) Combinational Segmentation Methods as the algorithm: As discussed in the Combined Loss section, the network tends to make predictions on the larger partition of the MRI scans, which rewards the network with a smaller loss value and promotes gradient descent. To address this issue, a combinational segmentation method could be considered.

If we only consider the head volume, we could consider the Class Skin as the related background of the remaining four classes (Skull, CSF, GM, WM). This is because, in the MRI scan, the Class Skin actually contains more than one tissue type and organ, each with its own image intensity on the MRI. This complexity can lead to false predictions on the other classes. Additionally, the Class Skin is the largest partition of the segmentation task but is relatively the least important among all. Finding another method to segment the Skin class would benefit the network's learning efficiency and accuracy since the network would only have to learn to segment the remaining four classes.

A proposed method of using edge detection algorithm to get the outer shape of the head and then segmenting the remaining four classes (Skull, CSF, GM, WM) only, and labeling up the remaining undefined volume between the shape of the head and segmented volume as Skin.

Another possible segmentation method that can be considered is to implement a two-step segmentation on the MRI. For example, the purpose of the first network is to learn to segment the brain partition (including CSF, GM, WM) from the Skin and Skull, and the second network only focuses on segmenting the CSF, GM, WM from the brain partition. This could be a better method because in multi-class segmentation tasks, the learning network has less capability to segment the smaller class, as can be seen from the result of CSF IoU in this project.

V. Conclusion

Based on the multi-class segmentation task using 3D U-NET, the network can successfully predicted the five different head tissues in MRI scans with 6 headsets trained only, reached 0.65 in MeanIoU and 0.74 in F.W.IoU. By reconstructing the predicted sub-volumes back into a whole 3D volume, the project was able to generate a mesh of the head. This work has important applications in medical image segmentation and reduce the cost of time to get head model for physics stimulation study. Overall, the project demonstrates the effectiveness of 3D-UNET in multi-class segmentation and its potential for use in medical imaging applications.

References

[1] Yamanakkanavar, N., Choi, J. Y., & Lee, B. (2020). MRI segmentation and classification of human brain using deep learning for diagnosis of Alzheimer's disease: a survey. Sensors, 20(11), 3243.

[2] Yang, M. S., Hu, Y. J., Lin, K. C. R., & Lin, C. C. L. (2002). Segmentation techniques for tissue differentiation in MRI of ophthalmology using fuzzy clustering algorithms. Magnetic Resonance Imaging, 20(2), 173-179.

[3] Despotović, I., Goossens, B., Philips, W. (2015). MRI segmentation of the human brain: challenges, methods, and applications. Computational and mathematical methods in medicine, 2015.

[4] Balafar, M. A., Ramli, A. R., Saripan, M. I., & Mashohor, S. (2010). Review of brain MRI image segmentation methods. Artificial Intelligence Review, 33, 261-274.

[5] Zotti, C., Luo, Z., Lalande, A., & Jodoin, P. M. (2018). Convolutional neural network with shape prior applied to cardiac MRI segmentation. IEEE journal of biomedical and health informatics, 23(3), 1119-1128.

[6] F. Xu, H. Ma, J. Sun, R. Wu, X. Liu and Y. Kong, "LSTM Multi-modal U-Net for Brain Tumor Segmentation," 2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC), 2019, pp. 236-240, doi: 10.1109/ICIVC47709.2019.8981027.

[7] Wong, K. K., Zhang, A., Yang, K., Wu, S., & Ghista, D. N. (2022). GCW-U-Net segmentation of cardiac magnetic resonance images for evaluation of left atrial enlargement. Computer Methods and Programs in Biomedicine, 106915.

[8] Jiang, Z., Ding, C., Liu, M., Tao, D. (2020). Two-Stage Cascaded U-Net: 1st Place Solution to BraTS Challenge 2019 Segmentation Task. In: Crimi, A., Bakas, S. (eds) Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. BrainLes 2019. Lecture Notes in Computer Science(), vol 11992. Springer, Cham. https://doi.org/10.1007/978-3-030-46640-4-22

[9] Edalati-rad, A., & Mosleh, M. (2019). Improving brain tumor diagnosis using MRI segmentation based on collaboration of beta mixture model and learning automata. Arabian Journal for Science and Engineering, 44(4), 2945-2957.

[10] Li Wang, Dong Nie, Guannan Li, Élodie Puybareau, Jose Dolz, Qian Zhang, Fan Wang, Jing Xia, Zhengwang Wu, Jiawei Chen, Kim-Han Thung, Toan Duc Bui, Jitae Shin, Guodong Zeng, Guoyan Zheng, Vladimir S. Fonov, Andrew Doyle, Yongchao Xu, Pim Moeskops, Josien P.W. Pluim, Christian Desrosiers, Ismail Ben Ayed, Gerard Sanroma, Oualid M. Benkarim, Adrià Casamitjana, Verónica Vilaplana, Weili Lin, Gang Li, and Dinggang Shen. "Benchmark on Automatic 6-month-old Infant Brain Segmentation Algorithms: The iSeg-2017 Challenge." IEEE Transactions on Medical Imaging, 38 (9), 2219-2230, 2019

[11] Dolz, J., Desrosiers, C., Wang, L., Yuan, J., Shen, D., & Ayed, I. B. (2020). Deep CNN ensembles and suggestive annotations for infant brain MRI segmentation. Computerized Medical Imaging and Graphics, 79, 101660.

[12] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

[13] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham. 35463660; PMCID: PMC9033381.

[14] He, Kaiming, et al. "Mask r-cnn." Proceedings of the IEEE international conference on computer vision. 2017.

[15] Bai, S., Gálvez, V., Dokos, S., Martin, D., Bikson, M., & Loo, C. (2017). Computational models of Bitemporal, Bifrontal and Right Unilateral ECT predict differential stimulation of brain regions associated with efficacy and cognitive side effects. European Psychiatry, 41(1), 21-29.

[16] Bai, S., Martin, D., Guo, T., Dokos, S., & Loo, C. (2019). Computational comparison of conventional and novel electroconvulsive therapy electrode placements for the treatment of depression. European Psychiatry, 60, 71-78.

[17] Yin XX, Sun L, Fu Y, Lu R, Zhang Y. U-Net-Based Medical Image Segmentation. J Healthc Eng. 2022 Apr 15;2022:4189781. doi: 10.1155/2022/4189781. PMID:

[18] Dolz, J., Desrosiers, C., & Ben Ayed, I. (2018, September). IVD-Net: Intervertebral disc localization and segmentation in MRI with a multi-modal U-Net. In International workshop and challenge on computational methods and clinical applications for spine imaging (pp. 130-143). Springer, Cham.

[19] GZhu, Y., Wei, R., Gao, G., Ding, L., Zhang, X., Wang, X., & Zhang, J. (2019). Fully automatic segmentation on prostate MR images based on cascaded fully convolution network. Journal of Magnetic Resonance Imaging, 49(4), 1149-1156.

[20] Siddique, N., Paheding, S., Elkin, C. P., & Devabhaktuni, V. (2021). U-net and its variants for medical image segmentation: A review of theory and applications. Ieee Access, 9, 82031-82057.

[21] Diakogiannis, Foivos I., et al. "ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data." ISPRS Journal of Photogrammetry and Remote Sensing 162 (2020): 94-114.

[22] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., ... & Rueckert, D. (2018). Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999.

[23] Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., ... & Zhou, Y. (2021). Transunet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306.

[24] Li, S., Chen, Y., Yang, S., Luo, W. (2019). Cascade Dense-U-Net for Prostate Segmentation in MR Images. In: Huang, DS., Bevilacqua, V., Premaratne, P. (eds) Intelligent Computing Theories and Application. ICIC 2019. Lecture Notes in Computer Science(), vol 11643. Springer, Cham. doi.org:10.1007/978-3-030-26763-6

[25] X. Wang et al., "SK-U-Net: An Improved U-Net Model With Selective Kernel for the Segmentation of LGE Cardiac MR Images," in IEEE Sensors Journal, vol. 21, no. 10, pp. 11643-11653, 15 May15, 2021, doi: 10.1109/JSEN.2021.3056131.

[26] Li, W., Wang, L., Qin, S. (2020). CMS-U-Net: Cardiac Multi-task Segmentation in MRI with a U-Shaped Network. In: Zhuang, X., Li, L. (eds) Myocardial Pathology Segmentation Combining Multi-Sequence Cardiac Magnetic Resonance Images. MyoPS 2020. Lecture Notes in Computer Science(), vol 12554. Springer, Cham. https://doi.org/10.1007/978-3-030-65651-5

[27] Vinyals, O., Blundell, C., Lillicrap, T., & Wierstra, D. (2016). Matching networks for one shot learning. Advances in neural information processing systems, 29.

[28] Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. Advances in neural information processing systems, 30.

[29] Shi, F., Yap, P. T., Fan, Y., Gilmore, J. H., Lin, W., & Shen, D. (2010). Construction of multi-region-multi-reference atlases for neonatal brain MRI segmentation. Neuroimage, 51(2), 684-693.

[30] Kikinis R, Pieper SD, Vosburgh K (2014) 3D Slicer: a platform for subject-specific image analysis, visualization, and clinical support. Intraoperative Imaging Image-Guided Therapy, Ferenc A. Jolesz, Editor 3(19):277–289 ISBN: 978-1-4614-7656-6 (Print) 978-1-4614-7657-3 (Online)

[31] Cignoni, P., Corsini, M. Ranzuglia, G. (2008). MeshLab: an Open-Source 3D Mesh Processing System.. ERCIM News, 2008.

[32] McNeel, R., others. (2010). Rhinoceros 3D, Version 6.0. Robert McNeel amp; Associates, Seattle, WA.

[33] MATLAB Version: 9.13.0 (R2022b)

[34] COMSOL Multiphysics® v. 6.1. www.comsol.com. COMSOL AB, Stockholm, Sweden.

[35] Materialise 3-matic® Medical Version 17.0

[36] Pravitasari, A. A., Iriawan, N., Almuhayar, M., Azmi, T., Irhamah, I., Fithriasari, K., ... & Ferriastuti, W. (2020). UNet-VGG16 with transfer learning for MRI-based brain tumor segmentation. TELKOMNIKA (Telecommunication Computing Electronics and Control), 18(3), 1310-1318.

[37] 3DBrainSegmentation. www.kaggle.com.

[38] Luo, S., Li, Y., Gao, P., Wang, Y., Serikawa, S. (2022). Meta-seg: A survey of meta-learning for image segmentation. Pattern Recognition, 108586.