



UNIVERSITY OF PADOVA

DEPARTMENT OF MATHEMATICS "TULLIO LEVI-CIVITA"

MASTER THESIS IN DATA SCIENCE

EVALUATION OF DEEP LEARNING TRANSFORMERS MODELS FOR BRAIN STROKE LESIONS AUTOMATIC SEGMENTATION

SUPERVISOR

PROF. MARCO ZORZI
UNIVERSITY OF PADOVA

CO-SUPERVISOR

MASTER CANDIDATE

ANDRES ARANGUREN

STUDENT ID

2009432

ACADEMIC YEAR

2021-2022

“I AWAIT YOUR SENTENCE WITH LESS FEAR THAN YOU PASS IT. THE TIME WILL COME
WHEN ALL WILL SEE WHAT I SEE”
— GIORDANO BRUNO

Abstract

The U-Net framework consisting in a U-shaped encoder decoder architecture based on the convolution operation has been used as the most common framework for automatic lesion segmentation in the medical field. However due to the convolution limitations to a local area, U-Net based models are not fully capable of capturing long range dependencies required to obtain precise masks when dealing with lesions such as brain stroke, for which it is necessary to consider the pixel-scale level to guarantee precise segmentation. Transformers, initially developed for natural language processing domain, can model long-range dependencies because of the self-attention mechanism. These models have been introduced in the last years to the computer vision field with promising results for lesion segmentation, due to their inherent capacity to process the inputs as sequences considering their dependencies regardless of the sequence length. In this work, we propose an analysis of U-Net based frameworks, and Transformers based models within U-net architectures, evaluating for each their performance metrics, running time and number of parameters. The models are tested on the anatomical tracing of lesions after stroke (ATLAS) dataset, the result demonstrate that the transformer models outperform all the previous algorithms however with a significant increase in the computational requirement during training. Transformer additionally manage to capture long-range dependencies as seen when segmenting bilateral subcortical lesions, it is however important to consider that the ideal sequence length should be limited to the pixel -scale in order to improve to consider the context for each pixel composing the lesion. Considering the hardware requirement to train Transformers, models such as the D-UNet that implements 3D operations in a 2D framework manage to increase the context information extraction with a considerably reduced number of trainable parameters.

Contents

ABSTRACT	v
LIST OF FIGURES	ix
LIST OF TABLES	xi
LISTING OF ACRONYMS	xiii
I INTRODUCTION	I
1.1 Preliminaries on Magnetic Resonance Imaging (MRI)	1
1.1.1 Magnetic Resonance Imaging (MRI)	1
1.1.2 Problems concerning MRI	3
1.1.3 MRI images preprocessing	3
1.2 Problem description	5
1.3 Early studies of lesion segmentation	6
1.3.1 Early studies	6
1.4 Deep Learning theory	7
1.4.1 Artificial Neural Networks ANN	8
1.4.2 Convolutional Neural Networks	8
1.4.3 Max pooling operation	10
1.5 Convolutional neural networks applications in medical image segmentation	11
1.5.1 U-Net	11
1.5.2 X-net model	13
1.5.3 Multipath Convolutional Neural Network	14
1.5.4 Post processing	15
1.6 Transformers models background	15
1.6.1 Attention mechanism	16
1.6.2 Transformer model architecture	19
1.7 Attention mechanism in computer vision	23
1.7.1 UTranseNet	23
1.7.2 Transformer mechanism for medical image segmentation	24
2 DATASET AND MODELS	25
2.1 ATLAS Dataset	25
2.2 Evaluation metrics	27
2.2.1 Dice loss	28
2.2.2 Loss function	28
2.2.3 Loss gradient	29
2.3 Implemented models	29
2.3.1 SegNet	29
2.3.2 X-Net model	30
2.3.3 D-UNet	31

2.3.4	Transformers within U-net	32
2.3.5	Transformer Upsampler	33
3	RESULTS	35
3.1	Experimental results of U-net and variants	35
3.1.1	X-net experimental results	35
3.1.2	D-UNet experimental results	37
3.2	Experimental results using Transformers based models	39
3.2.1	TransUNet transformer experimental results	39
3.2.2	Evaluation of lesion types	40
3.2.3	Ablation study	42
4	CONCLUSION	43
4.0.1	Transformers	43
4.0.2	U-Net based CNN models	44
4.1	Annex	45
	REFERENCES	47

Listing of figures

1.1	Intensities normalization for different tissue classes (Left): Cerebrospinal fluid, (Center): Gray matter, (Right) white matter	4
1.2	Artificial Neural Network neuron, All the element that form the input vector enter the node where the dot product with the input vector is performed. This operation is followed by an activation function, for this specific case a Sigmoid function which transforms the output values to a probability value from 0 to 1 with a 0.5 threshold	8
1.3	Convolutional neural network general architecture composed by three main layers. Convolutional layer, Max Pooling and fully connected layer	9
1.4	Overview of convolution operation using a 2D kernel over a MxN input matrix	10
1.5	U-Net encoder decoder architecture. Encoder performs a downsampling operation, whereas the Decoder converts the feature map into an image performing an upsampling process to restore input's original dimension	12
1.6	Multipath cnn architecture	14
1.7	Encoder decoder	15
1.8	Attention mechanism variants	17
1.9	Architecture of the unified attention model	18
1.10	Transformer architecture	19
1.11	Self attention mechanism	20
1.12	Multihead attention mechanism	22
1.13	Multi-dimensional mechanism	23
2.1	Example of lesion segmentation using MRICron	26
2.2	Overview of different slices forming the MRI volume	27
2.3	Visualization of MRI scans (Left: MRI scan, Right: manual lesion segmentation)	27
2.4	Dice metric visual representation	28
2.5	Structure of the X-net block used to reduce the number of trainable parameters and increase the strength of feature extraction	30
2.6	X-Net model pipeline consisting on a downsampling section, Feature similarity module and upsampling section to restore input resolution	31
3.1	Lesion segmentation results from ATLAS dataset. Col 1: MRI image, Col 2: Ground truth, Col 3: Res-UNet, Col 4: U-Net, Col 5: X-Net	36
3.2	Comparison of lesion segmentation performance of D-UNet, SegNet and U-net	38
3.3	Segmentation results for different lesion types. Row 1: left hemisphere cortical multiple lesion, Row 2: subcortical left single lesion, Row 3: bilateral cortical lesion.	41
3.4	Using pixel-scale patches to capture long-range dependencies in the Transformer module	42
4.1	U-Net baseline architecture	45

Listing of tables

1.1	Common normalization tools for MRI images preprocessing	4
2.1	Architecture of the model based on the D-Unet. Feature size corresponds to the size of the feature map received as input, the third dimension consists in the number of channels of the input which increase respectively in the downsampling portion. Upsampling blocks are additionally connected to the downsampling blocks, in this case up sampling block 1 is connected to the Convolution block 3, and the up sampling block 2 connected to convolution block 2 .	32
3.1	Evaluation of U-Net variants with the Feature similarity module FSM	35
3.2	Evaluation of U-Net variants without the Feature similarity module FSM	36
3.3	U-Net variants model performance	37
3.4	Evaluation of the proposed method with baseline U-Net and modified version using less trainable parameters	37
3.5	Evaluation of D-UNet performance against state of the art methods, SegNet	39
3.6	Transformer models performance	40
3.7	DSC score for lesion segmentation for three different types of lesions implementing X-net, D-UNet and TransUNet	41
3.8	Transformer models performance	42

Listing of acronyms

CNN	Convolutional neural networks
DNN	Deep neural networks
FCN	Fully connected network
ML	Machine learning
DL	Deep learning
RNN	Recursive neural network
LSTM	long short term memory
FSM	Feature similarity module
BN	Batch normalization
SGD	Stochastic gradient descent
ReLU	Rectified linear unit
Conv	Convolution

1

Introduction

Despite the intensive research, brain stroke represents a leading cause in long-term disability worldwide. Stroke rehabilitation research is focused on the understating of the relationship between brain, behavior, and recovery, using as a basis brain changes generated after a stroke, this allows for precise diagnostic and possible predictions in terms of functional outcomes. Neuroimaging represents the main resource for brain stroke research and therapies, it is of particular interest high-resolution T₁-weighted (T₁w) anatomical MRIs, which are used to evaluate structural brain changes after stroke episodes. Several techniques have been developed in order to accurately calculate or approximate the percentage between lesions and critical brain structures, this step constitutes a paramount step for precise lesion annotation. Despite the technological progress to date, manual lesion tracing by a team of experts in neuroimaging remains as the gold standard to draw valid clinical inferences for lesion segmentation[1].

Segmentation of brain lesions are a fundamental step that provides invaluable information for the analysis for neuropathology which are further necessary for planning therapy strategies, monitoring disease progression, and eventual prediction of patient outcome and location-behavior relationship. A deep understanding of pathophysiology of disease can provide clues about a disease characteristic and the direct effect on the involved anatomical structures. To date there is a lack of proper understanding between different lesion types, the spatial distribution and their relation with chronic sequelae after traumatic brain injuries[2].

1.1 PRELIMINARIES ON MAGNETIC RESONANCE IMAGING (MRI)

1.1.1 MAGNETIC RESONANCE IMAGING (MRI)

Magnetic resonance imaging (MRI) is as medical imaging technique used in radiology to form three-dimensional images of the anatomy and physiological processes of the body. The imagery techniques is based on the use of strong magnetic fields and magnetic gradients to generate images of the body structures. The technique is based

on the tendency of certain atom nuclei to respond or absorb radio frequency energy when placed in external magnetic fields, this results in an effect known as spin polarization which induces a radio frequency (RF) that can be detected. For clinical test and research, hydrogen atoms are mainly used for the spin polarization given their natural abundance in the human body, therefore MRI scans map the location of water and in the human body [3]. In the presence of strong magnetic fields hydrogen atoms absorb energy that is applied at a characteristic radio frequency (64 MHz for a standard clinical 1.5 Tesla MRI scanner), successively hydrogen atoms emit energy at same frequency while gradually returning to the equilibrium state. The MRI scanner captures the sum of the total emitted energy which decays over time, due to inhomogeneities in the magnetic field, this and several factors result in decreased image intensity. Different to PET scan MRI imaging does not involve ionizing radiation, it offers a better temporal resolution of about 1-4 seconds and an effective spatial resolution of 1mm, different to PET, MRI images does not require chemical tracers, however some brain regions are hard to image such as the near sinuses region [2].

T₁ AND T₂ SIGNALS

Tissues captured during an MRI scan can be characterized by two different relaxation times T₁ and T₂. Relaxation consists of the process in which nuclei spins release the energy received from the radio-frequency pulse. T₁ consists on the longitudinal relaxation time, it is the rate at which excited protons return to equilibrium, precisely the time for spinning protons to realign with the external magnetic field. T₂ transverse relaxation time is the time constant which determines the rate at which excited protons reach equilibrium or leave their phase with each other, hence it measures the time taken for spinning protons to lose phase coherence among the nuclei spinning perpendicularly to the magnetic field [4].

fMRI

In order to overcome difficulties in the imagery of certain brain regions further advances such as the functional magnetic resonance fMRI have been developed in order to achieve better image quality and functional analysis of the brain structures, this technique does not use radioactivity but directly measures the concentration of deoxyhemoglobin in the blood, this is called the BOLD response (Blood oxygen level contrast), the change in BOLD response over time is called the hemodynamic response function which has a number of distinct phases peaking at 6-8 seconds limiting the temporal resolution of the fMRI. fMRI techniques are designed to measure changes in the inhomogeneity in the magnetic field in small volumes of tissue, resulting from variation in the blood oxygenation, are considered both deoxy-and oxyhemoglobin which have different magnetic properties; deoxyhemoglobin is a paramagnetic molecule and induces inhomogeneity into the magnetic field, in contrast oxyhemoglobin is weakly diamagnetic having little effect in the magnetic field. Accordingly, a direct increase in the concentration levels of deoxyhemoglobin results in a decrease in the image intensity, in contrast a decrease of deoxyhemoglobin leads to an increase in the image intensity [5].

1.1.2 PROBLEMS CONCERNING MRI

MRI scanning has become a fundamental tool in diagnostics tests to detect, differentiate and characterize brain lesions of different natures. Despite MRI development and progress, visual image interpretation and manual segmentation of the brain area involved following a brain lesion, is still the most common and accepted standard to analyze clinical images, in order to conclude accepted diagnostics. Many cases involve lesion delineation, for example when dealing with brain tumors and structures at risk their respective boundaries have to be accurately localized, for radiation therapy planning, in order to avoid or minimize the risk of compromising unaffected tissue and optimize the treatment efficacy. The boundaries delineation process depends on a skilled rater or skilled committee which agrees based on different criteria on the affected region and the final delineation, despite its importance is often a time consuming process characterized by high variability between raters, and repeated delineations on the same image by the same rater, being a highly subjective process which depends on the experience and personal criteria of a professional [6]. The difficulty for accurate delineations has not progressed at the same rate of MRI developments, on one hand, recent image acquisition procedures resulted in high quality and complex images that capture a variety of structural and functional data, on the other hand increasing image complexity results in increasing analysis time, reducing the rate at which radiologist can analyze the images. The problems regarding clinical practices for image analysis can be summarized in three main issues: lesion area segmentation is a subjective and time consuming process, the analysis of stacked 2D images limits the accuracy of surrogate volume estimates, and the increasing information complexity leading to longer analysis procedures [6]. Considering this difficulties, automatic lesion segmentation has become a major research topic in clinical research, given their potential to alleviate image analysis issues through fast and consistent lesion delineation, it would increase diagnostic and prognostic accuracy of lesions segmentation, hence providing the medical experts explicit information of the lesion. Last decade, extensive research and literature has been published describing automatic segmentation algorithms, however there is still a lack of consensus regarding their viability or practical application in medical cases given the variability of brain lesions, additionally there is a lack of proper documentation in the several stages involved in the algorithm from input preprocessing to output generation. Furthermore, there is an increasing disconnection between literature developers and the users of the automatic lesion segmentation methods [7]. It is therefore mandatory to examine the clinical relevance of the current research conducted in the field and seek to understand their medical application considering possible limitations in order to bridge the gap between research and clinical practice. Automatic lesion segmentation research covers a wide range of algorithms with increasing complexity, among the most commonly used models are machine learning including decision trees, mixture models, fuzzy clustering, naïve bayes classifiers and in particular, deep learning algorithms, where recent research has been conducted implementing convolutional neural networks (CNN) based models and its variants such as the U-Net models, and more recently the application of algorithms developed initially for natural language processing such as Transformers to deep neural networks [7].

1.1.3 MRI IMAGES PREPROCESSING

A fundamental step when evaluating the previously mentioned models is the preprocessing steps which is used to remove noise sources, enhance the image contrast and in the best scenario better highlight the lesion area in order to be easier its detection in terms of the pixel values with respect to the image at the voxel level. The most used

preprocessing methods for automatic lesion segmentation are presented in Table 1.1. For literature review, even though the methods are clearly documented and described thoroughly in individual steps, there is no elucidation as to whether the methods are directly implemented inside the segmentation tool workflow, hence the difficulty on establishing a common standard for imaging preprocessing given the uncertainty of the stage at which algorithms are being applied.

Procedure	Tool/Approach
Intensity normalization	Histogram matching, intensity scaling
Bias Field	N_3, N_4
Brain extraction	FSL BET
Denoising	Anisotropic diffusion filtering
Image (co-)registration	—

Table 1.1: Common normalization tools for MRI images preprocessing

INTENSITY NORMALIZATION

Intensity normalization is a fundamental preprocessing step for MRI images analysis, it is used specifically to alleviate the large intensity variations that result from the acquisition of images using different scanners and parameters, which may compromise performance of further analysis such as segmentation and volume estimation [8] hence the utility of intensity normalization bringing the intensities to a common scale across patients. Figure 1.1, illustrates the result of intensity normalization applied to three different types of tissues.

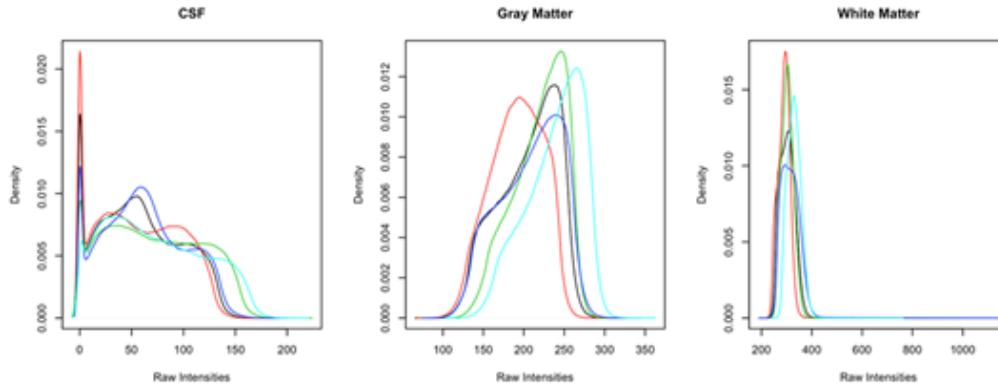


Figure 1.1: Intensities normalization for different tissue classes (Left): Cerebrospinal fluid, (Center): Gray matter, (Right) white matter

A first approach for intensity normalization consists on whole brain normalization, by Z-scoring each voxel using the mean and standard deviation computed from all voxels in the brain mask. The intensities from differ-

ent tissues become comparable with each other as seen in 1.1, however this method is sensitive to outliers, for example brain lesions may have very high intensities, which leads to bad estimates of mean/variance. More robust methods may be used such as using the median to the center or the IQR to scale. One alternative is White stripe methodology which was developed based on parameters obtained from a samples of normal appearing white matter (NAWM), which is a strapping for outliers, the main goal consists on making NAWM comparable across different patients. The main steps for white stripe are, finding the white matter area on the MRI histogram, then estimating the mean and variance of voxel intensities in that area, and subsequently normalize the voxel areas with the variance, after normalization the NAVM has a standard normal distribution with units in standard deviations, however following these steps gray matter and cerebrospinal fluid CSF may not be comparable [9].

BIAS FIELD

The bias field consists of a low frequency and smooth signal that compromises MRI images, consequently algorithms like segmentation that use gray level values of pixel intensities will produce inadequate results, hence the need for a pre-processing algorithm to the bias field signal. Several approaches have been developed. One alternative consists on dividing the corrupted MRI image by an estimated bias field using a surface fitting approach, another consists on implementing the fuzzy c-means algorithm to segment a corrupted MRI image [10].

IMAGE CO-REGISTRATION

Image coregistration consists on the process of aligning two or more images with the purpose that corresponding pixels representing the same objects may be integrated or fused, so they can be analyzed as one single unit. This is prerequisite for any further image processing step, it allows different matrices containing voxel values to be indexed and accessed identically, avoiding possible misalignment [11].

1.2 PROBLEM DESCRIPTION

Regardless of the current advances in medical imaging in terms of image acquisition, resolution and preprocessing techniques, a major impediment to progress in brain lesion studies remains, the intensive and mainly subjective process of requiring a group of experts to manually segment brain lesions from MRI images scans, according to literature a main concerns is the inter-rater reliability, reporting 67% agreement in overlapping voxels between two expert raters across ten subjects, apart from the inter-rater variance manual segmentation is mainly a time consuming task often in a range between 4.8 to 9.6 hours, hence the need for fast and automated brain lesion segmentation algorithms that can outperform available methods [12]. Previous studies have examined the use of standard machine learning algorithms, methods such as random forest which have demonstrated to perform competitively, however far below convolutional neural networks methods [13].

1.3 EARLY STUDIES OF LESION SEGMENTATION

1.3.1 EARLY STUDIES

Early studies have centered in methods combining probabilistic tissue segmentation with outlier based fuzzy clustering used for automated and semi-automated lesion delineation. The method developed by [14] performs tissue segmentation by using healthy control subjects to estimate gray matter (GM) and white matter (WM) probabilities for each voxel subsequently to extract the lesions in MRI scans, the previous estimates are used to calculate a metric that quantifies each voxel membership probability to a fuzzy set within each tissue class, this method is known as automated lesion identification (ALI). While ALI is an automated segmentation algorithm it suffers a major drawback of requiring scans from control group of healthy subjects to identify voxels that can be identified as outliers and assigned in the lesion tissue class, resulting incompatible for the cases when no control group data is available [1].

Supervised machine learning models such as random forests [15] and extra-tree forests [16] have been proposed for automatically identifying stroke lesions in MRI scans. These methods provide alternatives to solve the fuzzy clustering limitation since they do not require healthy control data, user interaction or arbitrary threshold selection, nonetheless these approaches typically require significant data volumes from MRI scans to achieve optimal performance.

The following sections provide a general overview of the theory behind some of the machine learning and deep learning based models that have been studied for automatic lesion segmentation

GAUSSIAN NAIVE BAYES CLASSIFICATION

Gaussian naïve bayes (GNB) classification consists on a supervised learning algorithm based on Bayes' probability theorem for classifying observations in a specific class based on the information given by the predictive variables, therefore it estimates the conditional probability that a data point belongs to a given class under the assumption that the predictive variable are independent, hence excluding the covariance among variables. The posterior probability of an observation Y with a given predictor variables' values belongs to a class k is defined by:

$$\hat{P}(Y = k | X_1, \dots, X_p) = \frac{\pi(Y = k) \prod_{j=1}^p P(X_j | Y = k)}{\sum_{k=1}^K \pi(Y = k) \prod_{j=1}^p P(X_j | Y = k)} \quad (1.1)$$

For each variable predictor the model estimates a different Gaussian distribution for each class, observations are then assigned to the class with the maximum posterior probability [13].

RANDOM FOREST

Further development of Bayes classification has been studied, proposing mixed algorithms for automatic segmentation combining two main methods:

- Multilevel splitting and merging Gaussian intensity classes using Bayesian – Markov random field (MRF) classification to extract probable lesion areas

- Probabilistic classification using random forest of the probable lesion

The proposed model was evaluated using the dataset available from the MS segmentation lesion challenge 2008.

The proposed model consists on the following levels :

- Hierarchical segmentation of probable lesion class using FLAIR MRI
- Random forest training and probabilistic classification on context-aware features of multimodal MRI

The main difference of the proposed method compared to other models implementing random-forest [17] for lesion segmentation is the use of multilevel Bayesian-MRF to estimate the initial lesion class before the random forest. The use of MRF, leads to informative sampling of the lesion class during training and testing stages, hence voxels with similar intensities are localized before proceeding to the random forest algorithm [15].

These models consist of supervised binary classification of voxel samples in either background or lesions using decision trees, however decision trees are known to suffer from overfitting [18], hence the implementation of discriminative random decision forest which achieves a better generalization by growing an ensemble of independent decision trees on a random subset of the training data, randomizing also the features available to each node in the training step. For these models the training data consists on a set of labeled voxels where the label is given by a random rater which manually segments the lesion boundaries. The classifier aims to assign every voxel to the labels lesion and background 1,10 respectively [15]:

1.4 DEEP LEARNING THEORY

In the last decade deep convolutional neural networks have outperformed machine learning models in visual recognition tasks. While CNN models have been researched for a long time their performance was limited due to the training size volumes for proper parameters optimization, an important breakthrough was achieved thanks to the supervised training on the imageNet dataset containing 1 million images, which allowed to reach the potential of more robust architectures which can approximate target functions at a relatively low computational expense [19]. Convolutional neural networks (CNN) such as the U-Net and other encoder-decoder networks have rapidly become state of the art methods for automatic image segmentation. U-Net networks consist of a down sampling encoders section which reduces the input image to semantic features and an up sampling section which expands the features for the segmentation prediction, this architecture is widely used in deep learning solutions for lesion detection [20]. Nonetheless, these models are based on 2D operations applied to inputs containing 3D MRI scans. A major drawback when implementing 2D operations is considering only a slice of the 3D scan, when in fact slices are spatially connected in the volume, hence losing important spatial information required to fully characterize the compromised region. In order to alleviate the spatial dependency models such as long short term memory models (LSTM) have been proposed, however there are simpler approaches such as implementing 3D kernels, which at the same time increases significantly the number of trainable parameters and additionally are prone to overfitting hence requiring proper regularization techniques and more training samples. Given the difficulty of collecting training data, one solution consists on dividing the original 3D image into smaller 3D cubes to increase the training dataset size, however these cubes provide less spatial information, resulting in a poor performance on smaller lesions.

In order to better comprehend the state of the art models being currently implement, it is worth providing a general introduction of deep learning starting with the concept of artificial neural network and then proceeding to convolution neural networks (CNN) which became the standard algorithm in computer vision.

1.4.1 ARTIFICIAL NEURAL NETWORKS ANN

Artificial neural networks consist of computation processing systems, inspired by the biological nervous systems operations across the neurons connections' network, as such, ANN consist on a high number of interconnected computational nodes (neurons), which through a series of mathematical operations collectively learn from a given input an objective function by optimizing the final output. The basic structure of an ANN is illustrated in 1.2. In this kind of model, the input usually consists on a multidimensional vector, which enters the network and is then distributed to the hidden layers. Further layers operate according to previous layers adjusting the weight values and evaluating if it results in a detriment or improvement of the final output.

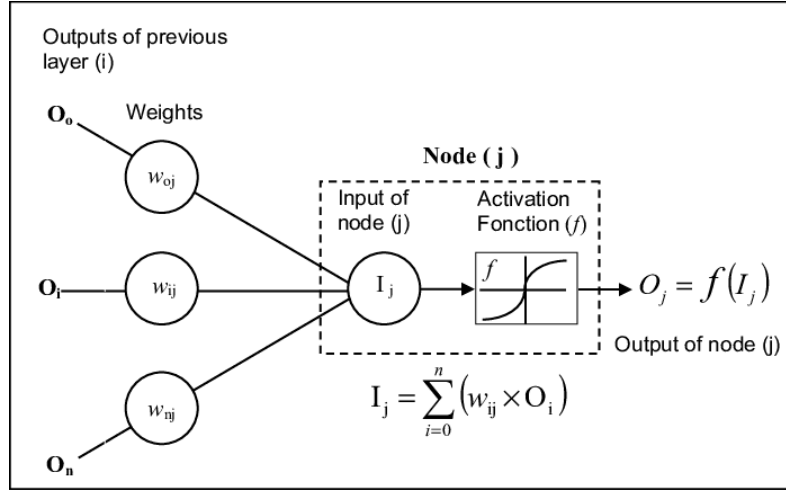


Figure 1.2: Artificial Neural Network neuron, All the element that form the input vector enter the node where the dot product with the input vector is performed. This operation is followed by an activation function, for this specific case a Sigmoid function which transforms the output values to a probability value from 0 to 1 with a 0.5 threshold

1.4.2 CONVOLUTIONAL NEURAL NETWORKS

CNN are a type of ANN, comprised of neurons that optimize their result based on the learning process. Each neuron receives an input and performs an operation (generally a scalar product followed by a nonlinear function). The main difference lies in the application, CNN have been implemented mainly in the field of pattern recognition in images, since they allow to encode image-specific feature maps, making it suitable for image processing. CNN also alleviate a main limitation of traditional ANN which is their computational limitation to compute image data. A common benchmark suitable for most ANN consists on the MNIST dataset due to the relative small dimensionality of 28 x 28 resolution images. Using this data each neuron in the first hidden layer will be

computing 784 weights ($28 \times 28 \times 1$) where 1 represents the third dimension of the input vector which is 1 given that images are normalized to just black and white values, this case is manageable for most forms of ANN. However when considering a colored image input of 64×64 , the number of weights computed by a single neuron in the first hidden layer increases substantially to 12,288. In order to overcome this problem, a possible solution might be to just increase the number of neurons and number of layers, however there are two main problems that emerge when increasing the number of neurons, first concerns computational resources. Training this model would imply immense computational costs, the second problem concerns overfitting, by increasing the model complexity, model might be approximating a more complex function than the actual objective function which leads to model noise and outliers, in more general perspective the model would start modeling the variance of the dataset. This over specified model leads to a decreased generalization when predicting unseen data. In general terms CNNs architecture is comprised by three types of layers which are convolutional layers, pooling layers and fully connected layers, as illustrated in 1.3

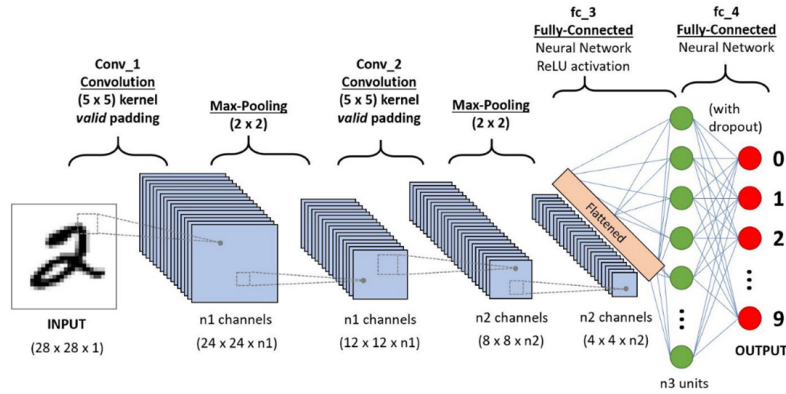


Figure 1.3: Convolutional neural network general architecture composed by three main layers. Convolutional layer, Max Pooling and fully connected layer

The basic functionalities of the different CNN layers can be described as :

1. The input layer contains the pixel values of the image.
2. The convolutional layer determines the output of the neurons which are connected to local regions of the input through calculation of the scalar product between their weights and the connected region to the input volume. The convolution operation is then followed by the activation function, widely applied rectified linear unit ReLu.
3. The pooling layer, performs down sampling along the spatial dimensionality of the input. It reduces the number of parameters.
4. The fully connected layers, perform the same operation as ANN, it calculates a prediction class score from the activation, used as classification.

The general workflow can be described as a transformation of the original input layer by layer using convolutional and down sampling operations to produce class scores used for classification or regression purposes.

CONVOLUTIONAL LAYER

The convolutional layer is the main building block of a CNN. It contains a set of filters known as kernels, which are to be learned throughout the training. The size of the kernels is usually smaller than the actual image. The filter convolves with the image creating an activation map. During the convolution the filter slides across the height and width of the image, for each displacement, the dot product between every element of the filter and the input is calculated. Figure 1.4 illustrates the convolution process. To form the activation map, the first entry is created by convolving the filter with the portion marked blue in the input image, this process is repeated for every element of the input image. The output volume of convolutional layer is formed by stacking the activation maps of every filter along the depth of the input dimension [19].

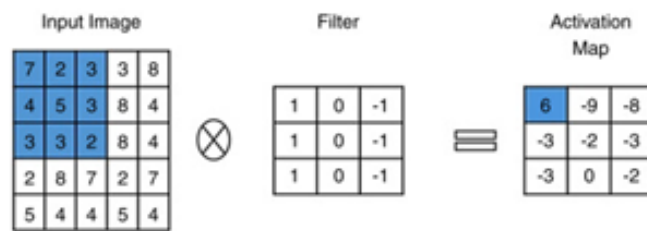


Figure 1.4: Overview of convolution operation using a 2D kernel over a MxN input matrix

POOLING LAYER

The pooling layers aim to gradually reduce the dimensionality of the representation, thus reducing the number of the parameters and computational complexity. The pooling layer operates over the activation map and scales its dimensionality using the MAX function. Most CNNs use max pooling layers with kernels of 2 x 2 dimension applying a stride of 2 along the input dimension. This process scales the activation map down to 25% of the original size. (insert citation)

1.4.3 MAX POOLING OPERATION

One major limitation of the feature map obtained after the convolutional operation is their position based dependence, meaning that the smallest alteration or movement of the input would result in a different feature map for each sample. Hence the need to down-sample the map, which is a commonly used technique in signal processing, in order to extract the most relevant features of the signal without compromising the process by added noise or small details that do not characterize the input. Thus, the pooling layer consists on a filter applied to the feature map, it generally consists on a 2 X 2 window with stride 2, where for each considered portion looks for the maximum element in the overlapping section between the feature map and the filter. The result is then, a low resolution feature map with a dimension reduced by a factor of 2.[21]

1.5 CONVOLUTIONAL NEURAL NETWORKS APPLICATIONS IN MEDICAL IMAGE SEGMENTATION

The models presented in this section have been developed and tested for lesion segmentation, their corresponding architecture is based on the U-shape structure of the U-Net and have been tested on the ATLAS dataset.

1.5.1 U-NET

Given its demonstrated performance in image analysis for lesion segmentation the U-Net architecture will be used as standard baseline for this study, further variants will be evaluated considering computational cost and performance metrics.

U-Net algorithm was designed to tackle classic CNN dimensionality problem where the final layer output dimension is much smaller than the input image, a feature map with the prominent characteristics, however losing much of detail and spatial information [22]. The U-Net is based on a fully convolutional network with a modified architecture so it works with few training images and provides a more precise segmentation. The main idea, of the U-Net consists on supplementing the usual contracting section of a CNN with additional layers, where pooling operators are replaced by up sampling operators. Therefore, the role of this layers is, to increase the resolution of the image, hence yielding a more precise feature map output. The up sampling sections contains additionally a large number of feature channels which allow the network to propagate context information to higher resolution layers [23]. The models do not have fully connected layers instead ,it uses the valid part of each convolution operation, consequently the segmentation output is composed only by the pixels for which the full context is available in the input image. Given the reduced training data volumes this kind of model has frequently been trained using data augmentation, applying deformation, dilation and contraction operation, which are especially useful in biomedical segmentation, since tissue deformation is the most common variation in different patients.

U-NET APPLICATION IN MEDICAL IMAGE ANALYSIS

Convolutional neural networks canonical task is to classify a whole image into a single label, however this type of models fail when dealing with pixel-level classification which is fundamental in medical image analysis for diagnostics and lesion evaluation. It is the U-Net by Ronneberger [20] that marked a breakthrough leading to a significant improvement in medical segmentation performance. The U-net is specially useful since it provides highly detailed segmentation maps using limited training samples, which is significant in the medical images analysis field as labeled images are often limited [24]. This obstacle is overcome by using data augmentation more specifically random elastic deformation on the training data learning from the variations without requiring new input data feed to the network. Another challenge that was resolved in the development of the U-Net consists on separating touching objects by applying a weighted loss function that penalizes the model when it fails separating two adjacent objects. Since its development in 2015 many advancements have been made in the U-net architecture by incorporating new image methods, which range from 2D standard base U-net, 3D U-net, residual U-net and attention U-Net [24].

BASE U-NET ARCHITECTURE

The U-Net architecture proposed by [20] is depicted in 4.1, it consists on a contracting and expansive path creating the U-shape symmetric model. The contracting path also known as the encoder consists of repeated two 3×3 convolutions followed by a ReLu activation function, 2×2 max pooling operation, this section is similar to a regular convolution network and provides classification information. For each downsampling step the number of features channels is doubled. The expansive path instead upsamples the feature map by applying 2×2 convolutions that instead halves the number of feature channels, and two 3×3 convolutions followed by a ReLu function. Expansion operations allow the network to learn localized classification information. The final layer consists on a 1×1 convolution layer which serves to map the 64 components feature vector into the desired number of classes.

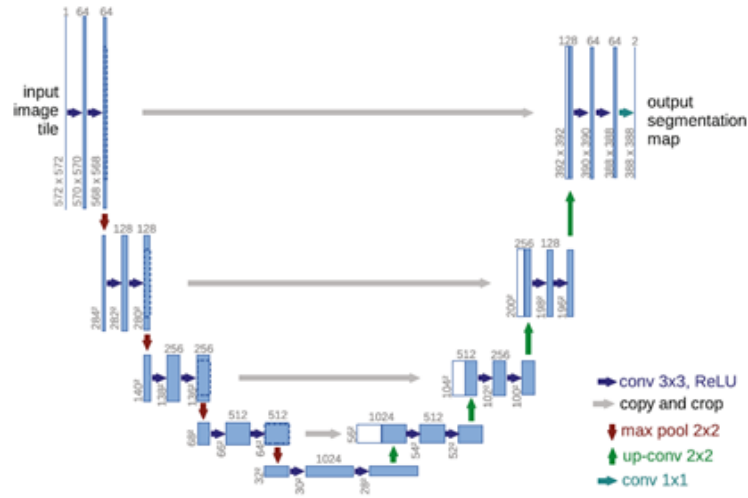


Figure 1.5: U-Net encoder decoder architecture. Encoder performs a downsampling operation, whereas the Decoder converts the feature map into an image performing an upsampling process to restore input's original dimension

TRAINING U-NET

The input image and their corresponding masks are used to train the network using stochastic gradient descent. Since the model does not include any padding the output image size is reduced, for computing reasons the model favors high resolution images over large batch size, hence each mini batch is reduced to one single image. As first regularization technique the model implements momentum up to 0.99, hence a large number of previously training samples will determine the update in the upcoming optimization steps, making the models to become highly dependent on the previous knowledge collected from each image [20].

3D U-NET

A major modification of the base 2D-U-Net, consists on the implementation of 3D convolution operations to process volumetric data as input instead of diving it in 2D slices, which has proven to reduce model capacity to capture the spatial context within the region of interest. The 3D U-Net allows image volumetric segmentation,

it has the same structure as the base U-Net with a contracting and expansive path, where 2D operations are replaced by 3D operations, hence 3D convolutions and 3D max pooling. This architecture is extensively used in volumetric CT and MRI images including bone structure evaluation, brain tumors, post-stroke lesion and multi organ segmentation[25].

In 2D models MRI volumes are divided in several independent planar slices, predicting for each one the lesion segmentation, hence ignoring spatial dependency between slices. 3D CNNs on the other hand, spatial information is extracted by inference however given their computational requirements, are usually avoided. In order to solve spatial dependencies both 2D and 3D U-net architectures have been extensively studied, one example is the Dimension fusion-UNet (D-UNet), that uses the 3D spatial information in the MRI under the 2D framework. The proposed model has two main technical improvements [26].

- Dimension fusion network: this section is placed in the downsampling portion of the network and is used to extract information from consecutive slices 2D slices of the MRI data. The section performs both 3D and 2D operation to extract the information and not lose the spatial dependencies within the slices. The respective feature maps are then fused together to reduce the number of trainable parameters.
- Enhanced mixing loss: to improve the networks convergence speed recent methods propose a new loss function known as enhanced mixing loss combining the advantages of dice loss and focal loss functions.

The proposed method was tested on the ATLAS dataset and compared to three state of the art models.

1.5.2 X-NET MODEL

Most models are not able to fully capture the contextual information among the pixels that compose the image. Models such as long short term memory network (LSTM) have been proposed to capture spatial contextual information however its efficiency heavily depend on long term memorization, unfortunately even this methods are limited to the surrounding pixels. To address this problem, the model known as X-Net was proposed [27], in order to address the problem of increasing number of trainable parameters, and the reduced performance when capturing long range dependencies. The model implements the effectiveness of depthwise separable convolution in reducing convolution kernel parameters [28]. Additionally it includes the Feature similarity module (FSM) which is designed to capture long range spatial dependence, this module is added as an extension of the convolutional neural network.

X-net method can be summarized by two main contributions:

1. Design of a FSM module for lesion segmentation extracting long range dependencies
2. A convolutional neural network integrated with the proposed FSM, with a reduced number of trainable parameters

FEATURE SIMILARITY MODULE FSM

The FSM module is implemented to capture long range dependencies, this module demonstrates its effectiveness based on the increase of the following metrics Dice, IoU and recall (see Section (2,2)) compared to the base model,

the increase in recall for this study case is even more important than precision since it implies that most of the strokes can be detected. Additionally the FSM as presented by [26] is more effective in the baseline 2D U-Net and ResUnet than in the X-Net model suggesting that some dependencies are already captured in the Xnet architecture.

1.5.3 MULTIPATH CONVOLUTIONAL NEURAL NETWORK

Multipath systems were introduced in order to overcome regular CNN limitations when capturing images context and capturing dependencies between spatial elements. To address this limitation, CNN systems have introduced that can process multiple 2D slice orientations. DeepMedic for example [2] is a simple multipath dual CNN using two pathways, one for high and the other low-resolution slices. Another example is [22] which uses a three path network, for the canonical axial, sagittal and coronal views respectively, nonetheless, adding multiple paths incurs in increasing the number of parameters to be learned, increasing the computational complexity leading to risk of over-fitting. Another aspect that has to be carefully considered in multipath architectures is the final output calculation. One approach consists on combining the path predictions using a majority vote system, however this methods risks ignoring important maybe less common features or information from the input, since the outputs from the different paths are coming into a final voxel prediction. One of the most recent multipath architectures consists on a multipath U-Net based network with a 3D convolutional kernel for post processing. The architecture can be summarized as a multipath model consisting in three different paths for the axial, sagittal and coronal plane. see Figure ??

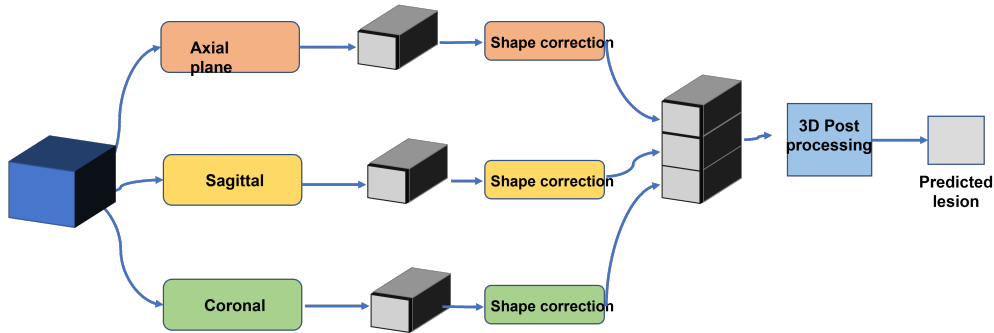


Figure 1.6: Multipath cnn architecture

Each path corresponds to a basic U-Net. Since the atlas dataset contains unimodal T1 images, one way to model a multipath system consists of considering the flipped version of the original image as an augmented synthetic path. The U-Net has the same architecture presented by (et [20]). In this module each convolution is succeeded by a 2×2 average pooling with stride 2 in order to halve the image initial dimensions.

Since two paths are using for both original and flipped version, the corresponding features are then combined using a fuse block. In the fuse block we consider the two feature maps that result from the encoders, which have dimension $32 \times X \times Y$, being 32 the number of convolutional filters and x and y the encoder output dimension. Both features are merged starting by the stacking process where an extra dimension is added resulting in an input dimension of $32 \times 2 \times x \times y$ which is then passed to a $2 \times 1 \times 1$ 3D kernel, resulting in an output dimension of

$32 \times 1 \times X \times Y$, this result is finally squeezed to reduce the extra dimension to a dimension of $32 \times X \times Y$. The fused output is then given to the decoder which is added to the output of the deconvolutional layers. Like the encoder the decoder's convolution blocks consist on 3×3 blocks with a stride of 1 in order to preserve the input dimensions. The decoder then applies the deconvolution operation which leads to the up sampling process that allows in the encoder decoders architectures to restore the input original dimension in order to not lose spatial information required to obtain a comprehensive overview of the image. The deconvolution is also known as transposed convolution, where the feature maps gets converted into an image. Convolution adds up the information contained in various pixels to one single pixel, whereas deconvolution spreads the information contained in one pixel to several pixels. In order to increase image dimension, the deconvolution operation uses padding before applying the convolution operation in order to avoid dimension reduction when operating with the convolutional kernel. The deconvolutional blocks allow to increase the image dimensionality, the operation consists of the product of the convolution weight matrix with the output when stride equals 1, if not zeros are inserted as outer padding in order to match the dimensions.

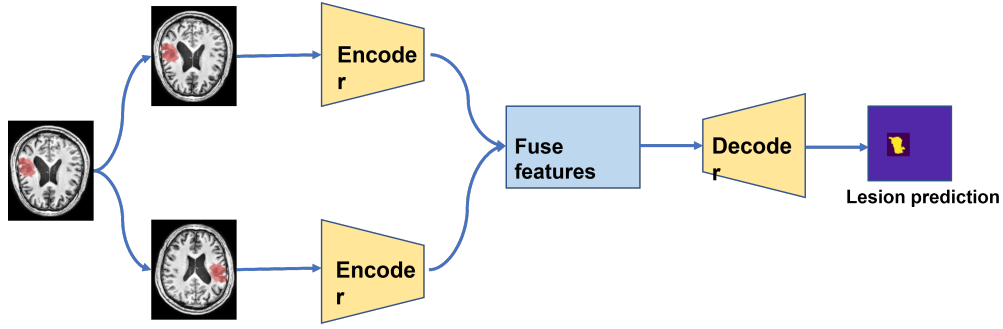


Figure 1.7: Encoder decoder

1.5.4 POST PROCESSING

The output of each of the three paths in the model consists of a 2D mask, which is binarized rounding to 0 if values are below 0.5, otherwise rounded to 1. Prediction masks are stacked together alongside the original image creating a $2 \times 192 \times 224 \times 192$ volume. Since the model has three paths when stacking all the masks the volume becomes $18 \times 192 \times 224 \times 192$ which is passed to the CNN post processor. The post processor consists on a main path containing 18 3D $3 \times 3 \times 3$ kernels each with 18 channels, followed by a second path having 9 3D $3 \times 3 \times 3$ kernels with 36 channels.

1.6 TRANSFORMERS MODELS BACKGROUND

Dominant sequence transduction models are based on complex recurrent convolutional neural networks that include encoding and decoding sections, which are used to improve their performance usually connected through an attention mechanism. The proposed architecture known as the transformer, is based solely in the attention

mechanism disregarding recurrent gates and convolutions operations. Recurrent neural networks, long short-term memory and gated recurrent neural networks have become the state-of-the-art approach when dealing with sequence modeling transduction problems that include language modelling tasks such as machine translation, sentiment analysis etc. Usually, this kind of models partition the computation along the symbol position of the input and output sequences, generating a sequence of hidden states h_t , as a function of the previous state h_{t-1} and their respecting index position t , this operation prevents parallelization in training example, i.e carrying out multiple input index processing simultaneously. This impediment becomes critical when dealing with long sequence input due to memory constraints. The attention mechanism has been widely used in sequence modeling, allowing to capture dependencies regardless of their distance in the input sequence, these mechanism is however in most of the cases used in conjunction with a RNN model [29]. The transformer on the other hand relies entirely on an attention mechanism to extract global dependencies between input and output.

The goal of reducing the computation comes from the foundation of the Extended Neural GPU, ByteNet and ConvS2S, which use convolutional neural networks as building blocks. In these type of models, the number of operations required to relate signals from two arbitrary input or output positions grows in the distance between positions, linearly from ConvS2S and logarithmically for ByteNet. In the transformers models this is reduced to a constant number of operations, consequently reducing the effective resolution due to averaging attention-weighted positions, which can be counteracted with multi-head attention. Summarized, attention also know as self-attention or intra-attention is a mechanism relating different positions of a single sequence in order to compute a representation of a sequence.

1.6.1 ATTENTION MECHANISM

Attention is a complex cognitive function indispensable for human beings, an important property of perception is the fact that humans do not process whole information, instead tend to selectively concentrate on a part of the information where is needed according to the cognitive task, ignoring other perceivable information, this means for humans to have the capacity of quickly selecting high value information from massive information inputs using limited processing resources. Attention mechanism can be divided in two categories. The first category consists on the unconscious attention, known as saliency-based attention, driven by external stimuli, for example hearing loud voices or noises will probably caught the attention of an individual immediately regardless the activity they are carrying on. Similar to the max-pooling and gating mechanism which passes the most relevant values (i.e larger values). The second category is the conscious attention, called focused attention. Refers to the attention that has a predetermined purpose relying on a specific task. Attention has become a common building block of neural architectures applied in various tasks such as image caption generation, text classification, machine translation image-based analysis, speech recognition. In addition to providing performance improvements the attention mechanism has proved to be a useful tool to explain incomprehensible neural architectures. Many models that have demonstrated great success share a common problem which is the lack of interpretability facing both practical and ethical issues. Thus attention, can provided an intuitive explanation to a certain extent, using the visualization of attention weights.

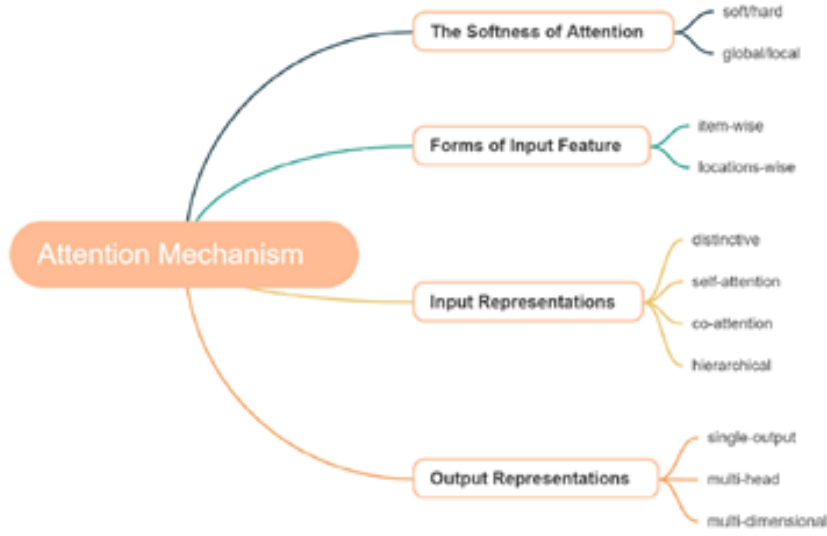


Figure 1.8: Attention mechanism variants

ATTENTION MECHANISM ARCHITECTURE

Machine translation models using attention mechanism are one of the most common architectures, and can be used as an example to define a general attention model. The following model is called the RNNsearch, it consists of a bidirectional recurrent neural network (BiRNN). The encoder calculates annotations $(h_1, \dots, h_t)$ that are the hidden state of the BiRNN based on input sequence $(x_1, \dots, x_t)$

BiRNN is used to solve one shortcoming of regular RNN which is that they only use previous context, bidirectional networks instead use all available input information in the past and future, hence hidden states are extracted forward and backward RNN respectively. The encoder then obtains an annotation for one word x_i concatenating the forward hidden state h_1 and the backward one h_i . The decoder consists of an attention block and RNN. The function of the attention is to compute the context vector c that represents the relationship between the current output symbol and each term of the entire input sequence. At each time step the context vector is computed as a weighted sum of the annotations h_j :

$$c_t = \sum_{n=1}^{\infty} 2^{-n} = 1 \quad (1.2)$$

The attention weight alpha of each annotation h_j is given by

$$c_{ij} = a(S_{t-1}, h_j) \quad (1.3)$$

$$\alpha = \frac{a}{\sum_{k=1}^T 2^{-n}} \quad (1.4)$$

Where a is a learnable function and reflects the importance of the annotation h_j to the next hidden state s_t according to state s_{t-1} . Successively, the RNN outputs the most probable symbol y_t at the current step. The

information of the sequence is distributed in the entire sequence instead of encoding the information limited to a fixed-length vector, while the decoder can selectively retrieve information at each step. This mechanism allows the neural network to focus on relevant elements of the input, thus ignoring the irrelevant ones.

1. Compute the attention distribution on the input information
2. Compute the context vector according to the attention distribution

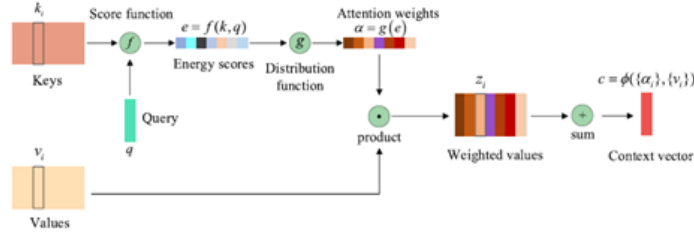


Figure 1.9: Architecture of the unified attention model

To compute the attention distribution the neural network encodes the source data features as k called a key. K can be expressed in several representations according to the specific, it may be features of a certain area of an image, word embedding when processing language, or the hidden states of RNNs. It is necessary to add a task-related representation vector q , the query, which in the case of RNN consists on the previous hidden state of the output $st-1$. The neural network computes the correlation between the queries and keys through the score function f (called energy function to obtain the energy score e). The energy score reflects the importance of queries with respect to keys when deciding the next output.

$$e = f(q, K) \quad (1.5)$$

The two most common score function used in attention mechanisms are additive attention and multiplicative (dot-product) attention. Conversely self-attention is computed only based on K without the need of q . After calculation the energy scores e , these are mapped to attention weights through the attention distribution function g

$$\alpha = g(e) \quad (1.6)$$

In the RNNsearch the distribution function consists on the softmax, which normalizes all the energy scores to a probability distribution, however this functions results in a full supported distribution ie $\text{softmax}(z) > 0$ for every term z , which is a disadvantage when requiring sparse probability distributions. In this case sparsemax function is proposed that may assign exactly zero probability given a certain input. Sigmoid function can also be used scaling the energy scores between 0 and 1. To compute the context vector, a new data representation V is introduced, called value. Each element of V corresponds to one element of K , usually they are both the same representation of the input data. V and K are different representations of the same data in their key value pair attention mechanism.

Once attention weights and values are computed, the context vector c is computed

1.6.2 TRANSFORMER MODEL ARCHITECTURE

Transformer models architecture is based on transduction models which usually consist on an encoder-decoder structure. The encoder maps an input sequence of symbol representations $(x_1, \dots, x_n)$ to a sequence of continuous representations $z = (z_1, \dots, z_n)$. Given z , the decoder generates an output sequence $(y_1, \dots, y_n)$ of symbols. Transformers adopt this architecture using self-attention and fully connected layers for both encoder and decoder. Figure illustrates the transformers general architecture

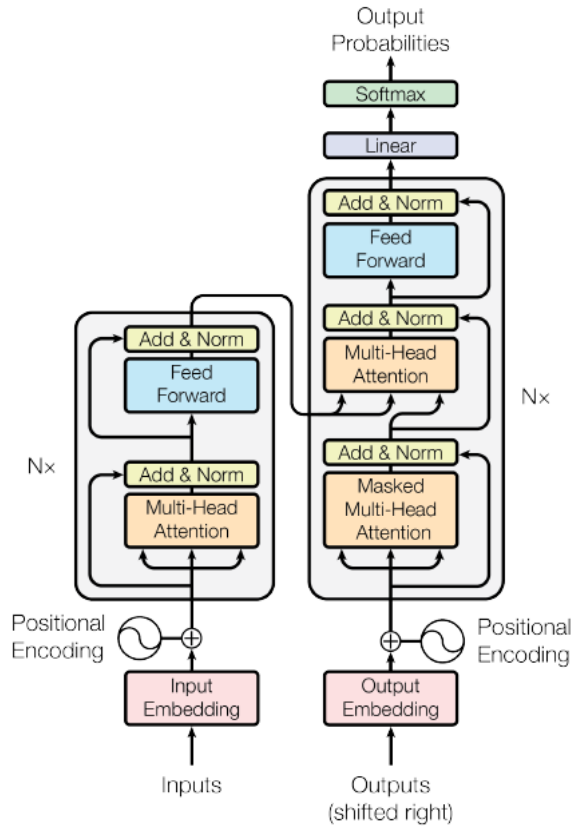


Figure 1.10: Transformer architecture

The encoder basically maps the input sequence into an abstract continuous vector representation, that holds all the learned information of the input. The decoder takes the representation and step by step generates a single output while being fed by the previous output.

In terms of architecture:

- Encoder: the encoder consists on 6 identical layers stacked together. Each layer is composed by two main units, first a multihead-self attention mechanism followed by fully connected forward network
- Decoder : composed by 6 identical layers stacked together, additionally to the two sublayers, attention mechanism and fully connected network the decoder includes a third layer that performs multi head attention over the output of the encoder stack.

The first operations consist of the input embedding followed by the positional encoding

1. Input embedding: perform vector representation of each word, hence each word maps to a single numerical vector with continuous values
2. Positional encoding: since the model contains no recurrence nor convolutions, to consider the sequence order it is necessary to add information about the relative or absolute positions of the words (tokens) in the sequence. The positional encoding is performed using sine and cosine functions for odd and even time steps, given the following functions

SELF-ATTENTION

As described in Figure the model implements a self-attention mechanism in order to extract the relevant features of the input sequence. This algorithm consists on mapping a given set key – values pairs to an output, being queries, keys, and output all vector representations. The general method is described as scaled-dot product attention, where is computed the dot product between queries and keys, a scaling process, followed by a softmax function to obtain the attention weights or scores. The self-attention mechanism is illustrated in figure

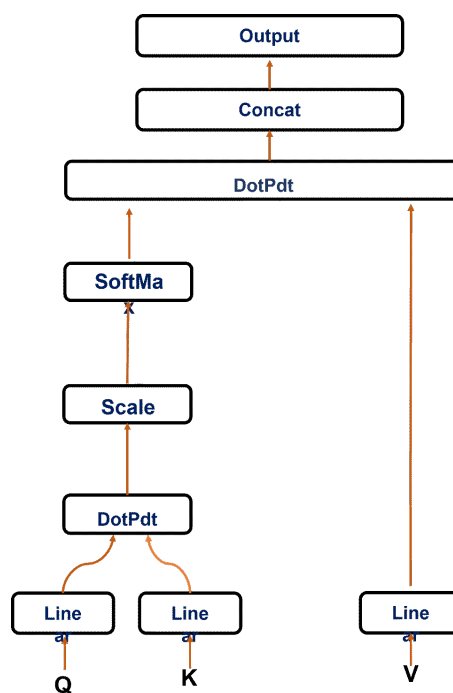


Figure 1.11: Self attention mechanism

In order to better understand the attention mechanism, it can be visualized using an example for natural language transduction example, for example a transformer applied to build a chatbot. The operation of self attention mechanism can be divided in the following steps

1. The input is fed into three distinct fully connected layers to create the query key and value vectors. These concepts come from the retrieval systems . Queries and keys undergoes a dot product matrix to produce a score matrix .

$$s = QK^T \quad (1.7)$$

Score matrix: determines how much focus should a word be put on other words, hence each word will have a correspond that corresponds to other words in the time step. The higher the score the more focus the model will apply, or the more relevant the word is in the sequence to capture the input context.

2. The scores matrix is then scaled down dividing by the square root of the dimension of the queries and the keys, this method is used to stabilize gradients since dot product operation may lead to exploding gradients problems

$$S_{scaled} = \frac{QK^T}{\sqrt{d_k}} \quad (1.8)$$

3. Softmax function is applied to the scaled score matrix in order to obtain the attention weights which outputs probability values between 0 and 1. Through this function the higher scores get heightened and lower scores are depressed, allowing the model to be more confident on which elements to attend.

$$softmax(x_i) = \frac{\exp^{x_i}}{\sum x_j} \quad (1.9)$$

$$attention_{weights} = softmax \frac{QK^T}{\sqrt{d_k}} \quad (1.10)$$

4. Having the attention weights multiply by the value vector to obtain an output vector

$$Attention(Q, K, V) = softmax \left(\frac{QK^T}{\sqrt{d_k}} V \right) \quad (1.11)$$

5. The output layer is the used to feed the linear layer
6. To obtain a multi-head attention model the query key and value are split into adding vectors before applying self-attention. Each vector then undergoes the same self-attention process, each process composes a head in the system, each head produces an output which are then concatenated to obtain a single vector before going into the final linear layer. Through this process each head will learn differently hence giving the model more representation power.

MULTI-HEAD ATTENTION PROCESS MORE IN DETAIL

The multi-head attention process comes originally from the study of CNN limitations, where in many cases using a single feature representation may not be sufficient to capture the desired context or dependencies related to a single input feature. To overcome this, several multioutput representations have been developed the most prominent ones are: multi-head and multidimensional representation. For many CNN models depending on the input complexity and task, it has been proven the efficiency of using multiple input channels instead of single channel, same with attention models, since using single attention distribution of the input ay not suffice the model complexity. Multi-head attention linearly projects the input sequence given my (Q,K,V) to multiple subspaces based on learnable parameters, it then applies scaled dot-product attention to the representation in each subspace finally concatenating the output. This mechanism allows the model to capture information from different representation subspaces at different positions. This mechanism has proven to be especially useful in computer vision problems when dealing with multichannel or multimodal input images. Figure illustrates the multi-head attention mechanism architecture

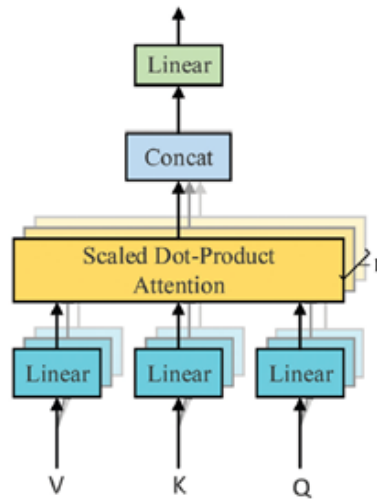


Figure 1.12: Multihead attention mechanism

Another mechanism consists on the multi-dimensional mechanism, which computes a feature-wise score vectors for keys, replacing weights cores vector with a matrix. Thus the neural network can calculate multiple attention distributions for the same data, this mechanism is widely used in natural language processing where word embedding suffer from polysemy problem.

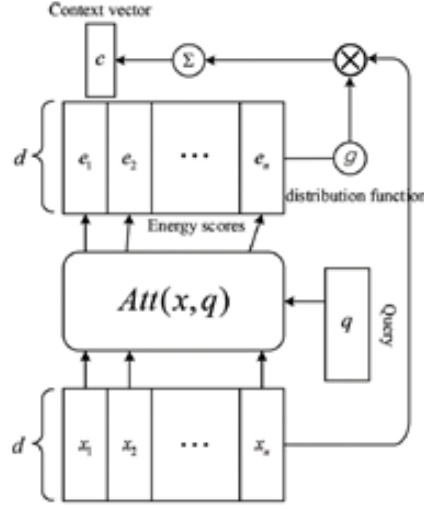


Figure 1.13: Multi-dimensional mechanism

1.7 ATTENTION MECHANISM IN COMPUTER VISION

Attention mechanisms have been widely used in computer vision, depending on the cognitive task this mechanism offer different advantages. Spatial attention (cite), which allows NN to learn the positions that should be focused on. The spatial information is transformed into another space while the key information is retained. Example using the MNIST dataset, spatial transformer that can find out the areas that need to be paid attention to in the feature map through different data augmentation techniques such as cropping, translation, rotation and scale. Channel attention (cite). Allows NN to learn what should be focused on. Ex. Squeeze and excitation network which is used to model interdependencies between channels. Another example is the CCNet which uses positional-wise attention, that generates an attention map to capture the relationship between each pixel pair in the feature map. The crisscross attention module collects contextual information in horizontal and vertical directions to enhance pixel wise representative capability.

1.7.1 UTRANSENET

Although U-shape CNN based methods have a very well performance, they are unable to capture the relationship between long-distance elements given the inherent limitation of the convolution operation. To overcome the limitation of CNN the convolution is used to increase the receptive field while the attention mechanism enhances the connection of relevant features thus suppressing irrelevant features. Nevertheless, this methods still rely on the convolution operation. Transformers models, offer an alternative to overcome this problems, they have the ability of modeling long-distance dependencies due to the self-attention mechanism, this models were developed initially in the field of natural language processing, however current research focuses on their application in the computer vision field, for example a transformer model used for classification tasks has obtained comparable or

even surpass convolution based models such as SOTA network [30], demonstrating a great potential in image processing.

The proposed model is based on [31] which explores the way of adding transformers after the convolution layers in U-Net, thus taking advantage of the characteristics of both convolutions and transformers to improve automatic lesions segmentation accuracy. The contribution of the cited work [31] can be divided in two main points

1. Inserting transformer mechanism into a downsampling, upsampling U shape architecture
2. Evaluation the model performance on the ATLAS dataset

1.7.2 TRANSFORMER MECHANISM FOR MEDICAL IMAGE SEGMENTATION

CNN models have been the de-fact standard model for image analysis tasks such as recognition and segmentation. U-Net specially became the standard model for medical image analysis, however when facing three-dimensional biomedical images, 2D models lose much of the contextual information, specially spatial information and its dependencies given that the image is divided in a 2D stacked images, it cannot extract the relationship between elements in different planes or slices. Hence the introduction of 3D convolution, which even though improve the context information and feature map, lack the ability of modeling long distance dependencies.

As previously stated, transformers models implementing the self-attention mechanism were developed initially for NLP task in order to capture long distance relationships between the elements of the input sequence, given their success in NLP many scholars are now researching their potential in computer vision to alleviate standard CNN limitations. A first approach was performed by Vit which implemented a pure transformer model in a classification task attaining a comparable or even better result the SOTA network, however when dealing with small datasets Vit has a decreased performance

Different approaches have been investigated when combining the U-Net with transformers, the TransUNet implements the ViT transformer in the deepest encoder to extract more deeper semantic information enhancing segmentation performance. [32] adopts squeeze and expansion transformer to regularize the attention matrix and learn multiple representations to increase segmentation accuracy and improve the model generalization. Another variant consists on the TransAttUNet which employs a self-aware transformer and global spatial attention to learn the non-local interactions between encoder features.

2

Dataset and Models

2.1 ATLAS DATASET

The ATLAS dataset (Anatomical tracing of lesions after stroke) Dataset - Release 2.0 is an open source data set of 955 T₁-weighted MRIs with manual segmented brain lesions. This dataset has been widely used for training and testing automatic lesion segmentation algorithms and provides a standardized dataset for comparing the performance of different models [33].

The dataset includes:

- 955 T₁-weighted MRI scans divided into training dataset (n = 655 T₁w with manually segmented lesion masks) and a test dataset (n=300 T₁w MRI)
- Two csv files containing lesion metadata

The MRI images were collected from 11 cohorts by the research groups in the ENIGMA consortium. The images are unimodal consisting of T₁-weighted anatomical MRIs of patients after stroke. For each MRI scan, brain lesions were identified and segmented by professional raters using MRICron, which is an open source tool for brain imaging visualization. Each image has at least one corresponding lesion mask, if separate lesions were identified, they were identified by separate masks [34].



Figure 2.1: Example of lesion segmentation using MRICron

Each subject considered in the dataset is only represented once, all of which have undergone a lesion tracing and preprocessing pipeline, the file is stored according to the Brain Imaging Data Structure. The dataset contains also metadata which provides information on the lesion properties in order to give the user qualitative information besides the lesion mask. The information can be used to spatially filter the dataset according to a specific location, (e.g left hemisphere lesion, subcortical lesion). It is important to remark that the presented models were evaluated using previous versions of the ATLAS dataset which had less training samples hence limiting the model performance.

DATA VISUALIZATION

Neuroimaging data are stored as .nii format file which is associated with NIfTI-1 Data format by neuroimaging informatics technology initiative. Using Python, the nilearn library is a python module for easy statistical learning on NeuroImaging data. Three main components can be found in the NifTi data structure :

- Data: raw scans in form of a numpy array
- Affine; contains the transformation matrix that map from voxel values of the numpy array to real-world locations of the brain
- Header: lower level information about the data (slice,duration)

Figure 2.2 illustrates an overview of several slices obtained from an MRI sample, contained in the ATLAS dataset. Figure 2.3 instead illustrates a sample slice and its respective manual lesion segmentation performed by a professional rater.

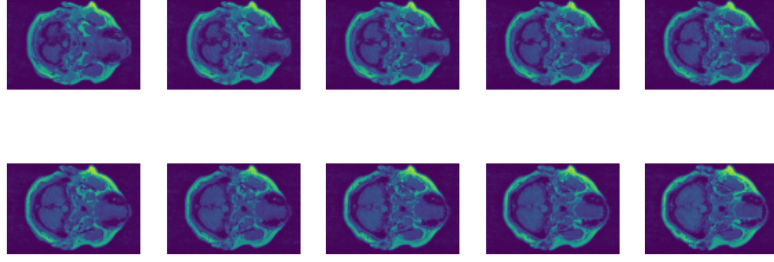


Figure 2.2: Overview of different slices forming the MRI volume

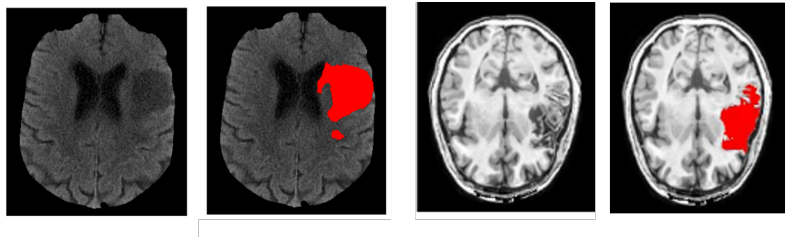


Figure 2.3: Visualization of MRI scans (Left: MRI scan, Right: manual lesion segmentation)

2.2 EVALUATION METRICS

To measure the model performance in terms of the predicted mask and the ground truth, the following metrics were measured Dice-similarity coefficient (DSC), F1-score, Intersection over union (IOU) The metrics are described as follows:

$$Dice = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (2.1)$$

$$IoU = \frac{TP}{TP + FP + FN} \quad (2.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.3)$$

$$Precision = \frac{TP}{TP + FP} \quad (2.4)$$

2.2.1 DICE LOSS

The dice coefficient has been used as the default metric to measure the accuracy or performance of automatic lesion segmentation tools using MRI images. The output of the implemented models consist on binary mask of the dimensions as the input image, having 1 for each voxel containing a lesion and 0 otherwise. The comparison is then calculated between the expert manually annotated mask and the automated mask from the algorithm. Using the expert binary mask as ground truth each predicted voxel is classified as true position (TP, also in true or ground truth mask), false positive (FP, predicted 1 but 0 in ground truth), or false negative (predicted 0 but 1 in ground truth). The Dice coefficient, provides a spatial overlap index, ranging from 0 indicating no overlap between predicted and automatic mask and 1 indicating complete overlap.

The Dice coefficient is defined as, it can also be visualized in Figure 2.4:

$$DICE = \frac{2TP}{2TP + FP + FN} \quad (2.5)$$

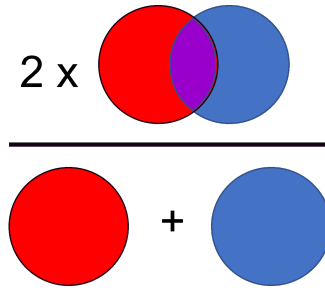


Figure 2.4: Dice metric visual representation

2.2.2 LOSS FUNCTION

Cross entropy loss poses a major disadvantage for unbalanced segmentation tasks, since the statistical distributions of labels have major effect in the training accuracy, since the more unbalanced the distributions the more difficult the training will. In cross entropy loss is calculated as the average of loss per pixel, which is calculated discretely without knowing whether its adjacent pixels are boundaries or not. Hence cross entropy considers loss in a micro sense or reduced spatial dimensions rather than considering the global composition of the image [35]. As a result Dice loss can alleviate cross entropy limitations.

$$D = \frac{2 \sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2} \quad (2.6)$$

Dice coefficient p_i and g_i represent the pairs of corresponding pixel values of prediction and ground truth. Values of p_i and g_i are either 0 or 1. Denominator Is the sum of total area of both prediction and ground truth

and numerator sum of corrected predicted pixels, it is the product of p_i and g_i so that the sum increments only when both p_i and g_i match having both value of 1. In boundary detection tasks the ground truth boundary pixels and predicted pixels or mask are seen as two sets. By using Dice loss the two sets are trained to overlap step by step, the denominator considers the total number of boundary pixels at global scale, while the numerator considers the overlap between the two sets. In this way, Dice loss considers the loss information both locally and globally which is necessary in order to increase segmentation accuracy.

2.2.3 LOSS GRADIENT

Once selected the loss function the gradient of the loss implemented to compute during the backpropagation can be described as follows: Given two matrixes p and g of size $M \times N$, where p is the predicted mask and g is ground truth.

$$DL(p, g) = \frac{2 \sum_{i=1}^N \sum_{j=1}^N p_{ij} g_{ij}}{\sum_{i=1}^N \sum_{j=1}^N p_{ij} + \sum_{i=1}^N \sum_{j=1}^N g_{ij}} \quad (2.7)$$

We first apply the derivative quotient rule for a value of p

$$dp_{ij} = \frac{\partial dice}{\partial p_{ij}} = \frac{2g_{ij} \left[\sum_{i=1}^N \sum_{j=1}^M p_{ij} + \sum_{i=1}^N \sum_{j=1}^M g_{ij} \right] - 2 \sum_{i=1}^N \sum_{j=1}^M p_{ij} g_{ij}}{\left(\sum_{i=1}^N \sum_{j=1}^M p_{ij} + \sum_{i=1}^N \sum_{j=1}^M g_{ij} \right)^2} \quad (2.8)$$

$$dp_{ij} = \frac{2g_{ij}(|p| + |g|) - 2|p \cap g|}{|p| + |g|} \quad (2.9)$$

Dividing by $|X| + |Y|$ dice loss gradient results in:

$$dp_{ij} = \frac{2g_{ij} - dice}{|p| + |g|} \quad (2.10)$$

2.3 IMPLEMENTED MODELS

2.3.1 SEGNET

SegNet is a fully convolutional neural network architecture used for semantic image segmentation. The model consists on an encoder for downsampling followed by an encoder for upsampling to the initial resolution of the input. The architecture of the encoder network is identical as the one proposed for the VGG16 network [36]. The decoder maps the low resolution encoded feature maps to the original input resolution for pixel classification. The novelty of SegNet is the way in which the decoder upsamples the low-resolution feature map. It uses pooling indices computed in the max-pooling step of the corresponding encoder to perform non-linear upsampling which eliminates the need for learning to upsample [36].

2.3.2 X-NET MODEL

The U-Net model is based on the original convolution for feature extraction, it also does not include residual connections which makes the model prone to overfitting. Considering this, the X-net model proposes a new block where depthwise separable convolution layer is employed reducing the number of parameters, this operation consists on a type of convolution applied to each channel of the input independently [27]. The proposed block is described in Figure 2.5, it consists on 3 cascaded depthwise convolutions each with kernel size of 3×3 . The original block described in literature however also includes a residual connection with 1×1 convolution layer.

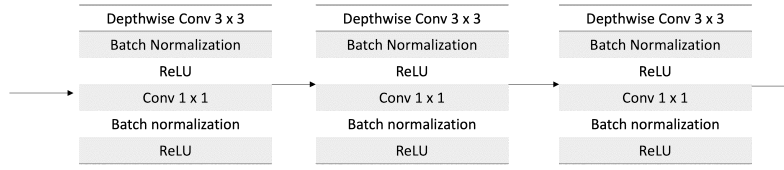


Figure 2.5: Structure of the X-net block used to reduce the number of trainable parameters and increase the strength of feature extraction

The pipeline of the model is presented in Figure 2.6. The method is based on the encoder-decoder architecture proposed originally by the U-Net, including skip connections which have demonstrated to improve segmentation accuracy. High dimension features are extracted by the encoder filters connected in cascade followed by the FSM module which calculates long range dependencies by capturing the relation between two any given points contained in the feature map. The decoder is then used to restore the spatial resolution of the image, finally after each convolution layer, the model implements batch normalization and Rectified linear unit (ReLU)

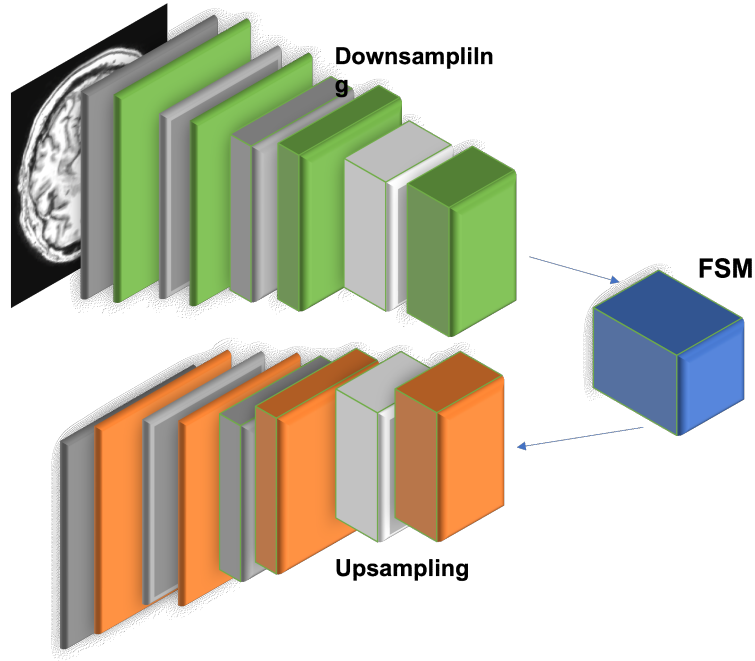


Figure 2.6: X-Net model pipeline consisting on a downsampling section, Feature similarity module and upsampling section to restore input resolution

2.3.3 D-UNET

The D-Unet consists on an improved U net architecture using both symmetric encoder and decoders structure. In the downsampling section 2D and 3D convolutions perform their operations in their respective dimensions. The result is later combined through the fusion block that enables to integrate subsequent 2D networks into the 3D information, improve the edges of the target region and increase the potential of the network to identify small lesion areas.

Considering that the 3D information is extracted in the early stage and the number of trainable parameters increase substantially as the network deepens, the 3D convolution and fuse block are added in the early coding stage [26].

ENHANCED MIXING LOSS FUNCTION

In 3D medical images such as MRI the volume occupied by the lesion region occupies a very small portion of the whole volume, which is instead characterized by background regions that tend to dominate the loss function, leading the learning process to fall in local optimum solutions that may compromise the models performance. In order to address the foreground-background voxel imbalance a new loss function by fusing both Focal loss (FL)

Table 2.1: Architecture of the model based on the D-Unet. Feature size corresponds to the size of the feature map received as input, the third dimension consists in the number of channels of the input which increase respectively in the downsampling portion. Upsampling blocks are additionally connected to the downsampling blocks, in this case up sampling block 1 is connected to the Convolution block 3, and the up sampling block 2 connected to convolution block 2

	Feature size	2D opeartion	3D operation
Input	192x192x4	-	
Conv block 1	192x192x32	2x(3x3 Bn)	2x(3x3x3 Conv + Bn)
Pooling	96x96x32	2x2 Max pooling	2 x 2 x 2 Max pooling
Conv block 2	96x96x64	2x(3x3 Bn)	2 x (3 x 3 x 3 Conv + Bn)
Dimension fusion block 2	96x96x64	-	-
Pooling	48x48x64	2x2 Max pooling	2 x 2 x 2 Max pooling
Conv block 3	48x48x128	2x(3x3 Bn)	2 x (3 x 3 x 3 Conv + Bn)
Dimension fusion block 3	48x48x128	-	
Pooling	24x24x128	2x2 Max pooling	
Conv block 4	24x24x256	2x(3x3 Bn)	
Dropout	24x24x256		
Up-sampling	192x192x32	2x2 Up-sampling – 2x(3x3 Bn)	

and Dice coefficient loss. The formula for the enhanced mixing loss is:

$$EML(p, g) = \frac{1}{N} FL(p, g) - \log(DL(p, g)) \quad (2.11)$$

2.3.4 TRANSFORMERS WITHIN U-NET

The baseline model for transformers within neural networks was the TransUNet proposed in [37]. TransUNet is considered the first model to establish the self-attention mechanism for sequence-to-sequence prediction in medical imaging. The model uses a hybrid CNN-transformer architecture to compensate for the loss of feature resolution inherent to transformers models., this operation is also useful to obtain high-resolution spatial information from the CNN features considering the global context encoded by the transformer. Inspired by the U-shape U-Net architecture, the self-attention mechanism encoded by the transformer is up-sampled to be combined with the high resolution feature maps skipped in the encoding portion to enable precise localization, preserving the advantages of the context provided by transformer and the high resolution features obtained through convolution in order to perform accurate segmentation without losing long range dependencies.

CNN - TRANSFORMER AS ENCODER

In order to apply the transformer model to computer vision, images have to be seen as sequence part of the process called image sequentialization. The first step consists on performing tokenization of the image hence converting the input into a 2D flattened sequence. The whole sequence is then divided in N patches of length l. The vectorized patches are then mapped into a latent D-dimensional embedding space using a linear projection, in order to save the spatial information of the projection, the position embedding are added to the patch vector. To leverage the potential of convolution, rather than using a Transformer as the encoder, the proposed model uses a CNN-

Transformer encoder where CNN is initially applied to the raw image, generating a high resolution feature map, patch embedding is then applied directly to the feature map.

2.3.5 TRANSFORMER UPSAMPLER

In order to output the segmentation mask, cascade upsampler (CUP) consists on a series of upsampling steps to restore the resolution of the input. Each upsampling block consists on a 2 X upsampling operator a 3 x 3 convolution followed by the ReLU activation function

3

Results

3.1 EXPERIMENTAL RESULTS OF U-NET AND VARIANTS

3.1.1 X-NET EXPERIMENTAL RESULTS

To evaluate the performance of the Xnet the following metrics were selected, including Dice score, intersection of union (IoU), precision and recall. In order to increase the learning process and achieve better metrics the model includes a reduced learning strategy where the value of the learning rate is reduced by a constant factor when the performance metric plateaus. The loss function was selected as the sum of the crossentropy loss and the dice loss, in order to better capture the spatial dependencies in the images. Batch size was set to 4, and a maximum number of epochs set to 20 due to computational resources limitations.

Evaluation indicators					
Models	FSM	DSC	IoU	Precision	recall
ResUNet (2021)	x	0.4109	0.3238	0.4489	0.408
U-Net	x	0.3921	0.3015	0.4276	0.3856
X-Net	x	0.525	0.485	0.477	0.4938

Table 3.1: Evaluation of U-Net variants with the Feature similarity module FSM

Evaluation indicators					
Models	FSM	DSC	IoU	Precision	recall
ResUNet (2021)	-	0.3910	0.3046	0.4309	0.3925
U-Net	-	0.3706	0.2877	0.4081	0.3678
X-Net	-	0.5216	0.4911	0.4673	0.4921

Table 3.2: Evaluation of U-Net variants without the Feature similarity module FSM

Considering the results of Table 3.1 and Table 3.2, it is possible to infer the increased performance obtained by implementing the FSM module when being included in both ResUNet and base U-Net models. As for the X-Net the selected metrics did not improve which suggests that the architecture of the proposed model already captures long range dependencies without the actual need of the FSM module.

The result of the lesion segmentation process for three randomly selected patients are presented in Figure 3.1.

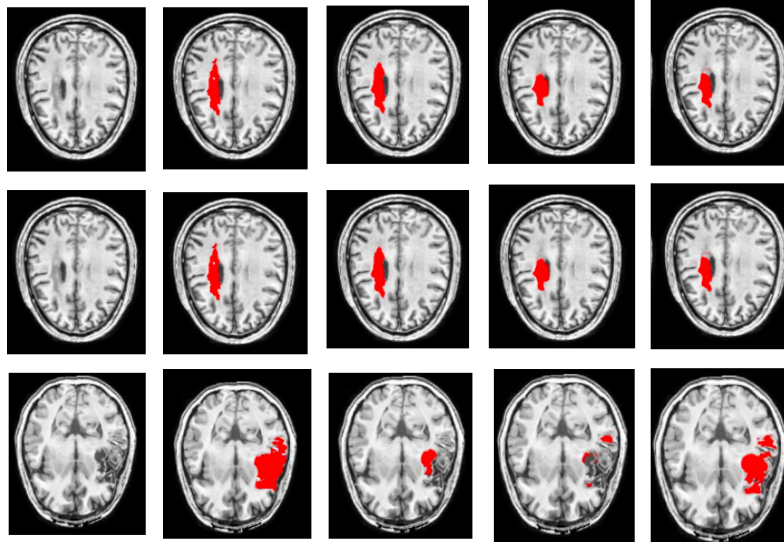


Figure 3.1: Lesion segmentation results from ATLAS dataset. Col 1: MRI image, Col 2: Ground truth, Col 3: Res-UNet, Col 4: U-Net, Col 5: X-Net

To validate the models performance the proposed method was compared with two state of the art methods Res-UNet and standard U-Net. According to the results reported in Table 3.3 the model demonstrated a better performance than the other models by 0.1051, 0.1772, 0.1391 and 0.127 improvement in terms of Dice, IoU, precision and recall respectively. This results suggest an acceptable generalization capability of the model to capture lesions from different samples with increased precision. The model has also less number of trainable parameters compared to the ResUNet which leads to faster training and requirement of less computational resources.

Evaluation indicators				
Models	DSC	IoU	precision	recall
ResUNet (2021)	0.4109	0.3238	0.4489	0.408
U-Net	0.3921	0.301	0.4276	0.3856
X-net	0.5168	0.501	0.588	0.535

Table 3.3: U-Net variants model performance

3.1.2 D-UNET EXPERIMENTAL RESULTS

The model was initially compared to the baseline U-Net model. Two versions of the U-Net were trained in order to verify the proposed method performance. Initially the original version using the architecture described in [20] and a modified version using less convolutional kernels and adding batch normalization after each convolution operation. DSC, precision and recall were selected as performance metrics. The batch size of the three networks was set to 32. The results are reported in Table 3.4 for U-Net, U-Net(improved), D-UNet.

Table 3.4: Evaluation of the proposed method with baseline U-Net and modified version using less trainable parameters

Method	DSC	Precision	Recall	Number of parameters
2D UNet	0.4721±0.1956	0.4871 ± 0.7214	0.4621 ± 0.6489	29,030,225
2D UNet improved	0.48021±0.7652	0.4961 ± 0.9828	0.4935 ± 0.2789	7,558,694
D-UNet	0.5074±0.2896	0.59835 ± 0.1864	0.53771 ± 0.8214	7,256,232

Based on the reported results on Table 3.4, both U-net and U-Net (improved) have similar performance in terms of the considered metrics, nonetheless when looking at the number of trainable parameters improved version implements four times less the number of parameters reported for the original U-net suggesting that for the considered dataset there is a large number of convolution kernels are being unnecessary in the training complexity. The decrease in the number of parameter translates additionally in decreased computational requirements. For all the implemented models batch normalization was added after each convolution layer to increase model convergence stability. Additionally it is also worth considering the training times for each model, the time required for the baseline U-net was 20.5 hours while the proposed model D-UNet 39 hours; considering the increase in the model performance and acceptable training time increase the results show that with the addition of the 3D structure the overall performance of the network is improved being able to capture 3D information.

The model should have been compared to a full 3D U-Net however convergence was not achieved this can be explained by two main elements:

1. The complexity of 3D models require an increased amount of time for training, making difficult to tune the hyper-parameters while obtaining a faster convergence
2. Overfitting is a common problem in large networks especially when available fewer training samples.
3. 3D structures require large amounts of computing resources which limit the width and depth of the 3D structure leading to a decrease in performance.

COMPARISON WITH OTHER STATE OF THE ART METHODS

The proposed model was compared with some of the frameworks proposed in [1] for this case U-Net [20] and SegNet [38]. The prediction results are illustrated in Figure 3.2. As shown in Figure 3.2, it is possible to observe that the lesions with small foreground area is a little fuzzy, which suggests that is more difficult for the network to be identified. However D-Unet correctly detects the lesion area, proving that the proposed approach has the ability to identify difficult samples. For the other sample, the considered models correctly predicted the lesion area. Another observation is the fact that the proposed method has a good feature expression ability in the cases with blurry boundaries, this is due to the combination of the 3D features and the spatial dimensions to express in a more effective way the characteristics of the edge blurred lesion.

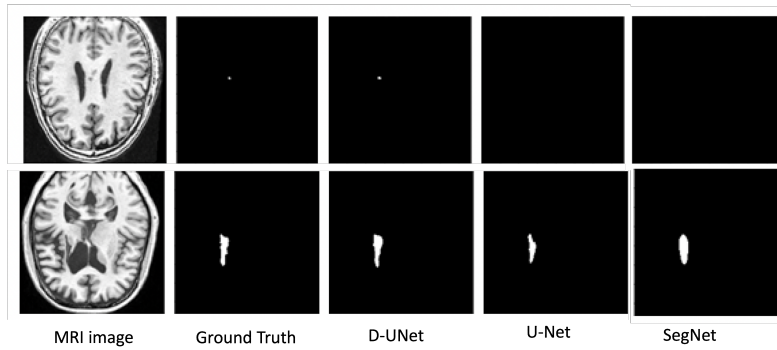


Figure 3.2: Comparison of lesion segmentation performance of D-UNet, SegNet and U-net

To illustrate the model performance against state of the art method, results are reported in Table 3.5, using the same metrics present in Table 3.4. The proposed method ranked first with a score of 0.522 demonstrating its superiority in terms of lesion segmentation. All the models were trained with a maximum of 50 epochs, and batch size of 36. It is important to remark that the proposed method D-U-Net was trained for every experiment using the enhanced mixing loss function described in section 2.2.2

Table 3.5: Evaluation of D-UNet performance against state of the art methods, SegNet

Model	DSC	Precision	Recall
SegNet	0.3187 $\pm$ 0.1851	0.3445 $\pm$ 0.7674	0.3245 $\pm$ 0.0984
U-Net	0.4356 $\pm$ 0	0.4371 $\pm$ 0.2005	0.4453 $\pm$ 0.0987
D-UNet	0.5226 $\pm$ 0.2902	0.5226 $\pm$ 0.2902	0.5226 $\pm$ 0.2902

3.2 EXPERIMENTAL RESULTS USING TRANSFORMERS BASED MODELS

The TransUNet model was designed using PyTorch as deep learning framework, in order to increase the model performance and avoid issues such as over-fitting data augmentation was applied using rotation and image flipping. The model was trained using a batch size of 8, maximum number of epochs fixed at 50 and stochastic gradient descent (SGD) optimizer with initial learning rate 1×10^{-3} . For training and validation each slice was cropped to 192×192 dimension and stacked alongside four adjacent slices creating an initial volume of $192 \times 192 \times 4$ [37].

3.2.1 TRANSUNET TRANSFORMER EXPERIMENTAL RESULTS

SwinUnet [39] and UCTransNet [40] were used to compare the TransUNet metrics performance

SWINUNET AND UCTRANSNET

The Swin-UNet consists on a U-Net transformer used for medical image segmentation. As described in Section (2.3.3) the image patches are fed into the Transformer as input sequence, the transformer is based on U-shaped Encoder-Decoder architecture with skip-connections for local- global semantic feature learning. The novelty of the model is the implementation of hierarchical Swin Transformer with shifted windows as the encoder to extract context features [39].

The UCTransNet presents a similar architecture to the Swin-UNet however in the Transformer encoder it operates at a single pixel scale in order to better capture long range dependencies within the input feature map, it additionally proposes he proposed a new connection which is able to replace the original skip connection of the original U-Net to solve the semantic gaps for an accurate automatic medical image segmentation [40].

COMPARISON WITH TRANSFORMER BASED MODELS

The TransUNet model [37] was compared with two state of the art models implementing transformers within U-Net architectures 1. SwinUnet [39] and 2. UCTransNet [40], both based on the implementation of transformers within deep neural networks based on the U-net encoder decoder architecture. Both SwinUnet and UCTransNet were loaded as pretrained models and tested on the ATLAS dataset release version 2.0 given that these models were

previously trained using a previous and reduced version of the ATLAS dataset results in terms of performance metrics are inferior compared to the results reported in [31]

Table 3.6: Transformer models performance

Model	DSC	F1-score	IoU	Training time (hr)
TransUNet	0.751	0.725	0.735	71
UCTransNet	0.739	0.727	0.733	60
SwinUNet	0.734	0.722	0.709	55

Considering the results in Table 3.6, the three frameworks present a similar performance in terms of the selected metrics, with an increase of 0.06 for TransUNet in terms of the DSC score which is an increase of 1.62 % with respect of the UCTransNet. With respect of the training time it was the TransUNet the model who took the longest to train, it is noticeable the increase in computational expenses when implementing Transformers module within CNNs, nonetheless the result immediately suggest a significant increase in the model performance and its ability to capture long range dependencies.

3.2.2 EVALUATION OF LESION TYPES

Additionally to the MRI scans the ATLAS dataset provides a metadata file containing a qualitative description of the lesion at the group level. The description consists on the number of lesions and the hemisphere location per subject, it is also provided lesion location being cortical or subcortical lesions. In order to evaluate the model performance in lesion segmentation different lesion types were tested varying the number of lesions and their locations. The tested model was the TransUNet with patch length 16, X-Net and D-UNet.

The first test consisted on evaluating the model performance when dealing with multiple lesions, which is a major challenge given the difficulty to capture long range dependencies, especially when dealing with adjacent lesions which represents the most difficult case to correctly separate the lesion areas [41]. The second type of lesion was a subcortical left hemisphere lesion which is one of the most common lesion types in the dataset. The third lesion consisted on a multiple bilateral lesion which represents the least frequent lesion type in the dataset

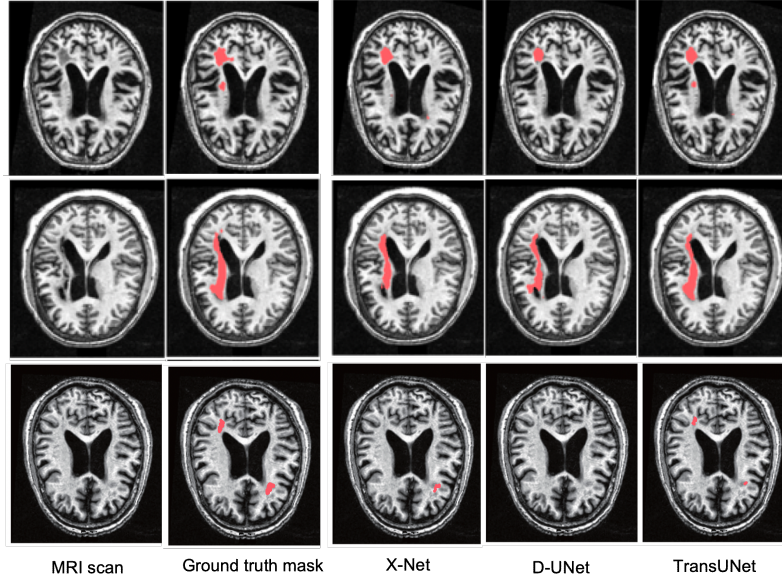


Figure 3.3: Segmentation results for different lesion types. Row 1: left hemisphere cortical multiple lesion, Row 2: subcortical left single lesion, Row 3: bilateral cortical lesion.

Table 3.7: DSC score for lesion segmentation for three different types of lesions implementing X-net, D-UNet and TransUNet

DSC score			
Lesion	X-Net	D-UNet	TransUNet
Cortical/left/unilateral	0.654	0.617	0.786
Cortical/left	0.621	0.648	0.792
Sub-cortical/bilateral	0.236	-	0.535

According to the results reported in Figure 3.3 standard CNN based models such as X-net and D-UNet struggle when segmenting multiple lesion in one sample, especially in the case for multiple lesions in different hemispheres as seen in the third row Figure 3.3. For this cases the mentioned models are not able to capture the long range dependencies between the different lesions. Transformer model TransUNet demonstrates a higher performance when segmenting multiple lesions at unilateral and bilateral level obtaining the highest DSC scores of 0.786 and 0.535 respectively; this can be attributed to the use of transformers in the encoding section using the attention mechanism to capture dependencies between the input sequences. As for the cortical lesion reported in Row:2 Figure 3.3, the three proposed models suggest an acceptable performance with a noticeable decrease in the mask accuracy of the X-Net model.

3.2.3 ABLATION STUDY

In order to validate the models performance under different configuration an ablation study was carried out evaluating: 1) Patch size in the encoding section and 2) Number of skip-connections.

In order to evaluate the effect of patch length in the encoding section of the model. The original convolution block of the TransUNet was modified according to the proposed architecture of the UTransNet [31]. Accordingly instead of splitting the feature map into patches which are then flattened and tokenized with a fixed l length, the cited model implements a patch sequence length of 1 therefore it limits to the single pixel level of the feature map. This sequence is then fed to the transformer in order to capture the relationship at the pixel-scale which should led to more accurate results when especially when dealing with small lesion areas. The process of capturing dependencies at the pixel-scale can visualized in Figure 3.4

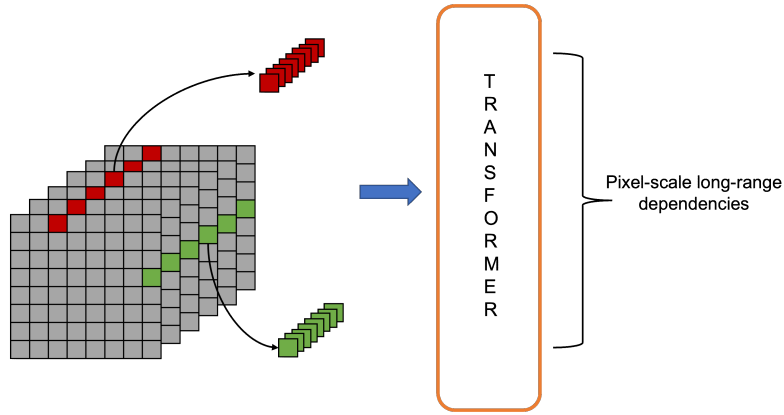


Figure 3.4: Using pixel-scale patches to capture long-range dependencies in the Transformer module

Accordingly the TransUNet patch size evaluation is reported in Table 3.8 where lengths of 32, 16 and 8 were tested, however last one required the highest computational requirements hence the performance metric could not be verified. It is important to remark that the Transformers sequence length is inversely proportional to the patch size, hence the smaller the patch length the longest will be the sequence length entering the transformer hence increasing its computational complexity. As reported in Table 3.8, decreasing patch size leads to improvements in the model performance as the Transformer is able to encode more complex dependencies between elements for longer input sequences.

Table 3.8: Transformer models performance

Patch size	Sequence length	DSC
32	49	0.751
16	196	0.776

4

Conclusion

4.0.1 TRANSFORMERS

Based on the obtained results in this project, Transformers models implemented within convolutional neural networks have demonstrated to be the algorithm with best performance in terms of DSC, FI-core and IoU metrics with values of 0.751, 0.725 and 0.735 respectively which compared to the best results obtained using the U-shaped models, suggest an average of 43,86% improvement in models performance. To leverage the power of transformers it was also demonstrated the advantages of using the encoding of the feature map obtained through the convolution operation, hence implementing an hybrid encoder. The models hence used low-level CNN features using the U-shaped architecture.

Transformer models pose an important limitation in terms of their high computational requirements. The TransUNet model which was trained using the same system processor took about 3.2 times more to be trained than the D-UNet implementing 3D convolution operations. Nonetheless considering a fair trade-off between model complexity and performance it is worth mentioning that a significant increase of about 43,86% had not been achieved using standard CNN models, additionally taking into account the recent advances in terms of computational processing, and the need of accurate and reliable algorithms for lesion segmentation, the increase resources demand would not be classified as a detrimental factor in terms of model performance.

As it was demonstrated reducing the patch size results in models performance improvement given the inverse relation between sequence length and the patch size. This process allows to capture long range dependencies in a more efficient way, ideally the model should have been tested at the single pixel scale patch size as proposed in [31], however using a single pixel level results in a computational requirement that could not be satisfied.

In terms of lesion type, Transformer based models demonstrated a higher performance for multiple lesion segmentation, compared to the CNN based models such as the X-Net and D-UNet. Hence the attention mechanism suggests to be a better approach to capture long range dependencies, note that the evaluated transformer model TransUNet was trained using a patch size of 16, however the segmentation quality may possibly increase by

reducing the patch size up to the pixel-scale value as proposed in the UTransNet. [31]. The Transformer model outperformed the CNN based models in the three considered cases especially when dealing with multiple bilateral lesion which was the worst case for both X-Net and D-UNet.

4.0.2 U-NET BASED CNN MODELS

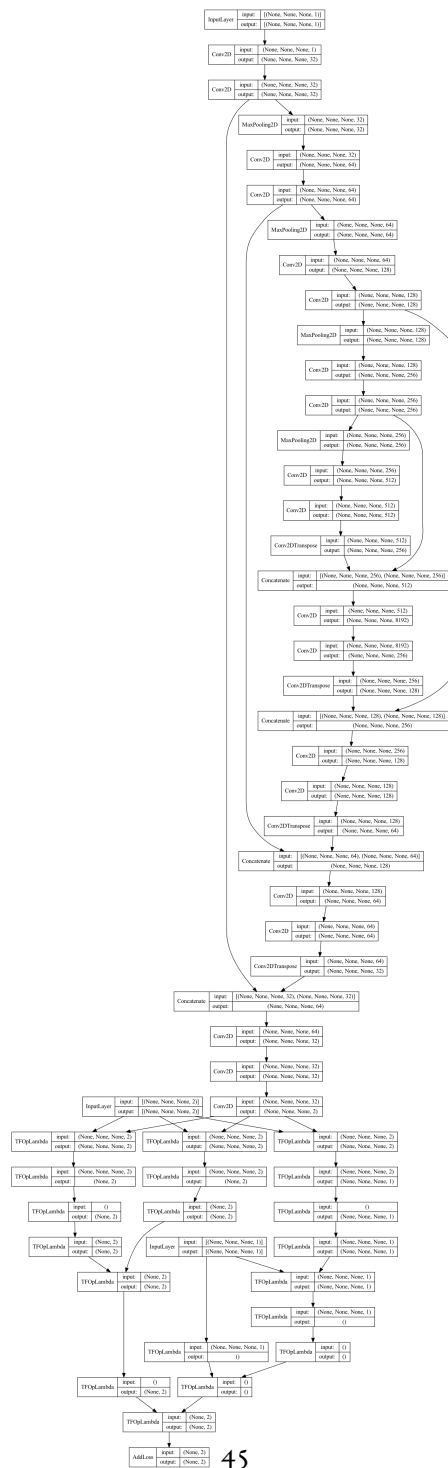
For the CNN models based on the U-net algorithm, the D-UNet model demonstrated the potential of extracting 3D context information in the early stage of the models architecture without high computational requirements as it has been verified for fully 3D CNN models [2]. Additionally the model surpassed the standard 2D U-Net obtaining a final DSC score of 0.507 with a considerable decrease in the number of parameters decreasing the model complexity and hardware requirements. In terms of training time the D-UNet by implementing the 3D operations in the early section of the model, reduces the required time for it to be trained, the model took approximately 40 hr which is six times less than the reported results for a full 3D U-Net [22].

The X-Net model demonstrated its effectiveness when extracting informative features including the long range dependencies with fewer trainable parameters. Dense contextual information was also captured by using the FSM module, hence the model tackles two of the main problems in image segmentation, high number of trainable parameters and the lack of proper methods to capture the context or inter-dependencies within the image. The model additionally outperformed the standard U-Net and ResUNet for the considered metrics, note that the evaluated model differs from the original X-net model [27], since the residual connection consisting on a $1 \times 1 \times 1$ convolution was not included in the evaluated model, this connection should improve the mask quality in the up-sampling portion when restoring the input original dimensions. Additionally the usefulness of the FSM module was demonstrated though its direct implementation in the standard U-Net increasing its performance, except for the X-Net which suggests to be capturing the long range dependencies even before the addition of the module. To improve the ablation study it would be ideal to perform a channel capacity study, by increasing the capacity the number of parameters increase as well as its capacity to extract context information improving ideally the segmentation performance, hence would be useful to double the channel capacity and evaluate the result on the metrics since as described in [31] by doubling channel capacity the DSC score while F1-score and IoU decrease which may be caused by the input of useless information added through the new channel.

Accurate automatic lesion segmentation is a process that can help professionals to conduct clinical diagnostics and formulate in a faster way treatment and procedures to tackle a specific lesion. Using the output of the convolution operation (feature map) as the input sequence of a transformer algorithms has proven to be currently the best method to capture long-range dependencies, the explored method exhibited competitive performance and currently outperforms the standard CNN based models. The presented project evaluated state of the art models using the latest version of the ATLAS dataset release version 2.0 which at the publication date is the most recent open-source database for lesion segmentation training, even though the designed models do not fully replicate the proposed architecture described in literature because of computational limitations, the improved dataset provides a more robust source of training samples that may alleviate the gap between model complexity and performance.

4.1 ANNEX

U-NET BASELINE MODEL ARCHITECTURE



References

- [1] K. L. Ito, H. Kim, and S.-L. Liew, “A comparison of automated lesion segmentation approaches for chronic stroke t1-weighted mri data,” *Human brain mapping*, vol. 40, no. 16, pp. 4669–4685, 2019.
- [2] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, “Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation,” *Medical image analysis*, vol. 36, pp. 61–78, 2017.
- [3] M. T. Vlaardingerbroek and J. A. Boer, *Magnetic resonance imaging: theory and practice*. Springer Science & Business Media, 2013.
- [4] M. Neema, J. Stankiewicz, A. Arora, V. S. Dandamudi, C. E. Batt, Z. D. Guss, A. Al-Sabbagh, and R. Bakshi, “T1-and t2-based mri measures of diffuse gray matter and white matter damage in patients with multiple sclerosis,” *Journal of Neuroimaging*, vol. 17, pp. 16S–21S, 2007.
- [5] D. J. Heeger and D. Ress, “What does fmri tell us about neuronal activity?” *Nature reviews neuroscience*, vol. 3, no. 2, pp. 142–151, 2002.
- [6] E. Gryska, J. Schneiderman, I. Björkman-Burtscher, and R. A. Heckemann, “Automatic brain lesion segmentation on standard magnetic resonance images: a scoping review,” *BMJ open*, vol. 11, no. 1, p. e042660, 2021.
- [7] G. Du, X. Cao, J. Liang, X. Chen, and Y. Zhan, “Medical image segmentation based on u-net: A review,” *Journal of Imaging Science and Technology*, vol. 64, pp. 1–12, 2020.
- [8] X. Sun, L. Shi, Y. Luo, W. Yang, H. Li, P. Liang, K. Li, V. C. Mok, W. C. Chu, and D. Wang, “Histogram-based normalization technique on human brain magnetic resonance images from different acquisitions,” *Biomedical engineering online*, vol. 14, no. 1, pp. 1–17, 2015.
- [9] R. T. Shinohara, E. M. Sweeney, J. Goldsmith, N. Shiee, F. J. Mateen, P. A. Calabresi, S. Jarso, D. L. Pham, D. S. Reich, C. M. Crainiceanu *et al.*, “Statistical normalization techniques for magnetic resonance imaging,” *NeuroImage: Clinical*, vol. 6, pp. 9–19, 2014.
- [10] J. Juntu, J. Sijbers, D. V. Dyck, and J. Gielen, “Bias field correction for mri images,” in *Computer recognition systems*. Springer, 2005, pp. 543–551.
- [11] H. Huhdanpaa, D. H. Hwang, G. G. Gasparian, M. T. Booker, Y. Cen, A. Lerner, O. B. Boyko, J. L. Go, P. E. Kim, A. Rajamohan *et al.*, “Image coregistration: quantitative processing framework for the assessment of brain lesions,” *Journal of digital imaging*, vol. 27, no. 3, pp. 369–379, 2014.
- [12] M. Wilke, B. de Haan, H. Juenger, and H.-O. Karnath, “Manual, semi-automated, and automated delineation of chronic brain lesions: a comparison of methods,” *NeuroImage*, vol. 56, no. 4, pp. 2038–2046, 2011.

- [13] J. C. Griffis, J. B. Allendorfer, and J. P. Szaflarski, "Voxel-based gaussian naïve bayes classification of ischemic stroke lesions in individual t1-weighted mri scans," *Journal of neuroscience methods*, vol. 257, pp. 97–108, 2016.
- [14] M. Sucharitha and K. P. Geetha, "Brain tissue segmentation using fuzzy clustering techniques," *Technology and Health Care*, vol. 23, no. 5, pp. 571–580, 2015.
- [15] J. Mitra, P. Bourgeat, J. Frupp, S. Ghose, S. Rose, O. Salvado, A. Connelly, B. Campbell, S. Palmer, G. Sharma *et al.*, "Lesion segmentation from multimodal mri using random forest following ischemic stroke," *NeuroImage*, vol. 98, pp. 324–335, 2014.
- [16] O. Maier, M. Wilms, J. von der Gablentz, U. M. Krämer, T. F. Münte, and H. Handels, "Extra tree forests for sub-acute ischemic stroke lesion segmentation in mr sequences," *Journal of neuroscience methods*, vol. 240, pp. 89–100, 2015.
- [17] E. Geremia, O. Clatz, B. H. Menze, E. Konukoglu, A. Criminisi, and N. Ayache, "Spatial decision forests for ms lesion segmentation in multi-channel magnetic resonance images," *NeuroImage*, vol. 57, no. 2, pp. 378–390, 2011.
- [18] M. Bramer, "Avoiding overfitting of decision trees," *Principles of data mining*, pp. 119–134, 2007.
- [19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [21] J. Nagi, F. Ducatelle, G. A. Di Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. M. Gambardella, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *2011 IEEE international conference on signal and image processing applications (ICSIPA)*. IEEE, 2011, pp. 342–347.
- [22] Y. Xue, F. G. Farhat, O. Boukrina, A. Barrett, J. R. Binder, U. W. Roshan, and W. W. Graves, "A multi-path 2.5 dimensional convolutional neural network system for segmenting stroke lesions in brain mri images," *NeuroImage: Clinical*, vol. 25, p. 102118, 2020.
- [23] N. Navab, J. Hornegger, W. M. Wells, and A. Frangi, *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III*. Springer, 2015, vol. 9351.
- [24] S. Nahian, S. Paheding, E. Colin, and D. Vijay, "U-net and its variants for medical image segmentation: Theory and applications," *Preprint at arxiv*, vol. 1118, p. VI, 2011.
- [25] R. Mehta and T. Arbel, "3d u-net for brain tumour segmentation," in *International MICCAI Brainlesion Workshop*. Springer, 2018, pp. 254–266.
- [26] Y. Zhou, W. Huang, P. Dong, Y. Xia, and S. Wang, "D-unet: a dimension-fusion u shape network for chronic stroke lesion segmentation," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 18, no. 3, pp. 940–950, 2019.

- [27] K. Qi, H. Yang, C. Li, Z. Liu, M. Wang, Q. Liu, and S. Wang, "X-net: Brain stroke lesion segmentation based on depthwise separable convolution and long-range dependencies," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2019, pp. 247–255.
- [28] F. Mamalet and C. Garcia, "Simplifying convnets for fast learning," in *International Conference on Artificial Neural Networks*. Springer, 2012, pp. 58–65.
- [29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [30] G. Yang and D. Ramanan, "Volumetric correspondence networks for optical flow," *Advances in neural information processing systems*, vol. 32, 2019.
- [31] P. Feng, B. Ni, X. Cai, and Y. Xie, "Utransnet: Transformer within u-net for stroke lesion segmentation," in *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 2022, pp. 359–364.
- [32] S. Li, X. Sui, X. Luo, X. Xu, Y. Liu, and R. Goh, "Medical image segmentation using squeeze-and-expansion transformers," *arXiv preprint arXiv:2105.09511*, 2021.
- [33] S.-L. Liew, J. M. Anglin, N. W. Banks, M. Sondag, K. L. Ito, H. Kim, J. Chan, J. Ito, C. Jung, S. Lefebvre *et al.*, "The anatomical tracings of lesions after stroke (atlas) dataset-release 1.1," *bioRxiv*, p. 179614, 2017.
- [34] S.-L. Liew, J. M. Anglin, N. W. Banks, M. Sondag, K. L. Ito, H. Kim, J. Chan, J. Ito, C. Jung, N. Khoshab *et al.*, "A large, open source dataset of stroke anatomical brain images and manual lesion segmentations," *Scientific data*, vol. 5, no. 1, pp. 1–11, 2018.
- [35] H. Yessou, G. Sumbul, and B. Demir, "A comparative study of deep learning loss functions for multi-label remote sensing image classification," in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2020, pp. 1349–1352.
- [36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [37] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "Transunet: Transformers make strong encoders for medical image segmentation," *arXiv preprint arXiv:2102.04306*, 2021.
- [38] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [39] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-unet: Unet-like pure transformer for medical image segmentation," *arXiv preprint arXiv:2105.05537*, 2021.
- [40] H. Wang, P. Cao, J. Wang, and O. R. Zaiane, "Utransnet: rethinking the skip connections in u-net from a channel-wise perspective with transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, 2022, pp. 2441–2449.
- [41] Y. Ou, Y. Yuan, X. Huang, K. Wong, J. Volpi, J. Z. Wang, and S. T. Wong, "Lambdaunet: 2.5 d stroke lesion segmentation of diffusion-weighted mr images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2021, pp. 731–741.

