

Automatic medical image segmentation with Unet and Unet++

Manuel Hernandez Zemljic

Student ID: 20213458

Submitted to The University of Nottingham
in September 2021

in partial fulfilment of the conditions for the award of the degree of
Master of Science, MSc Data Science

I declare that this dissertation is all my own work, except as indicated in the text

Contents

Contents.....	iii
List of tables	v
List of figures.....	vi
Abstract	viii
Acknowledgements	ix
Chapter 1: Introduction.....	1
1.1 Description of the problem.....	2
1.1.1 Placenta segmentation	2
1.1.2 Brain tumour segmentation.....	2
1.2 Aims and objectives	3
1.3 Scope.....	3
Chapter 2: Literature review	4
2.1 Literature review	4
2.1.1 FCN	4
2.1.2 Deconvolution Network	4
2.1.3 Unet.....	5
2.1.4 Resnet.....	6
2.1.5 Densely Connected Convolutional Networks (DenseNet)	7
2.1.6 Deep supervision	8
2.1.7 Unet ++	8
2.1.8 Medical imaging	8
2.1.9 CT vs MRI	9
2.1.10 Multimodal medical imaging.....	10
2.1.11 Placenta segmentation.....	10
2.1.12 Brain tumour segmentation	10
Chapter 3: Methodology	11
3.1 Model evaluation	11
3.1.1 Evaluation metrics	11
3.1.2 k-fold cross validation	12
3.2 Model implementation	12
3.2.1 Training and overfitting.....	12
3.2.2 Unet.....	12
3.2.3 Unet plus plus.....	13
Chapter 4: Data analysis and processing	14
4.1 Three-dimensional data segmentation	15
4.2 Placenta data set	15
4.3 BraTS data set.....	18
4.3.1 Multimodal learning.....	18
4.3.2 One hot encoding	21
Chapter 5: Experiments.....	22

5.1 Learning ratios.....	22
5.1.1 Placenta Unet learning ratios.....	22
5.1.2 Placenta Unet++ learning ratios	23
5.1.3 BraTS Unet learning ratios	24
5.1.4 BraTS Unet++ learning ratios.....	24
5.2 Model pruning.....	25
5.2.1 Model pruning placenta.....	25
5.2.2 Model pruning BraTS.....	26
Chapter 6: Segmentation results	27
6.1 Unet vs Unet ++	27
6.2 Comparison of results with previous studies.....	28
6.2.1 Previous studies placenta.....	28
6.2.2 Previous studies BraTS	28
Chapter 7: Conclusion	29
7.1 Further study.....	29
7.2 Areas of improvement	29
7.3 Limitations	30
References.....	31

List of tables

Table 5.1 bce, dice and loss obtained with a learning ratio of 1E-4. It includes the dice score, its standard deviation, IL and UL are the lower and the upper limits of the confidence interval with a 95% of confidence.....	31
Table 5.2 bce, dice and loss obtained with a learning ratio of 1E-4 and 1E-3 for placenta and unet++. It includes the dice score, its standard deviation, IL and UL are the lower and the upper limits of the confidence interval with a 95% of confidence.....	32
Table 5.3 bce, dice and loss obtained with a learning ratio of 1E-4 and 1E-3 for BraTS and Unet model. It includes the dice score, its standard deviation, IL and UL are the lower and the upper limits of the confidence interval with a 95% of confidence.....	33
Table 5.4 bce, dice and loss obtained with a learning ratio of 1E-4 and 1E-3 for BraTS and Unet++ model. It includes the dice score, its standard deviation, IL and UL are the lower and the upper limits of the confidence interval with a 95% of confidence. Scheduled learning ratio is from 1E-4 and from the epoch 20 it decreases to 1E-5.....	34
Table 5.5 Number of parameters, average dice and average time of inference with a standard CPU (placenta data).....	35
Table 5.6 Number of parameters, average dice and Average time of inference with a standard GPU (BraTS data).....	35
Table 6.1 Comparison of the results obtained with the different models and datasets, it includes the dice score, its standard deviation, IL and UL are the lower and the upper limits of the confidence interval with 95% of confidence.....	36
Table 6.2 Placenta results obtained in this work with unet and unet++ (Unet 2021, Unet++ 2021), results from UoN (Unet 2020) and three publically available placenta segmentation results. BraTS results obtained in this work (Unet 2021) and results from other sources, (Henry et al. 2020), (Fidon, Ourselin, and Vercauteren 2021), (*) the placenta results from (Alansary et al. 2016), (Looney et al. 2017) and (Twickler et al. 2020) are from different datasets to the datasets used in this work. (**) the BraTS results of (Henry et al. 2020) and (Fidon, Ourselin, and Vercauteren 2021) although they are from the same version of BraTS data set, these results have been obtained with the test data set which ground truth are not publicly available.....	37

List of figures

Figure 1.1: “Example of the utero-placental pump, left the relaxed placenta before the contraction, right the contracted placenta (the placenta is the area with the red line around it): it's clear how the placenta is smaller during a contraction, pushing out the maternal blood”.....	12
Figure 1.2 Sub-regions of glioma. In the multiple MRI modalities, image patches with tumour sub-regions annotated. From left to right, the picture patches display the whole tumour (WT - yellow) seen in T2-FLAIR (Fig. A), the tumour core (TC - orange) visible in T2 (Fig. B), the enhancing tumour (ET - light blue) evident in T1-Gd, and the cystic/necrotic components of the core (green) (Fig. C). The final labels for the tumour sub-regions (Fig. D) are edema/invasion (yellow), non-enhancing solid core (orange), necrotic/cystic core (green), and enhancing core (red) (blue).....	12
Figure 2.1 The fully convolutional network extracts image features using a CNN, then uses an 1x1 convolutional layer to convert the number of channels into the number of classes, and finally utilises transposed convolution to translate the height and width of the feature maps to those of the input picture.....	14
Figure 2.2 Architecture of the deconvolution network, the convolutional network comprises a total of 13 convolutional layers, with rectification and pooling operations conducted between convolutions on occasion, and two fully connected layers augmented at the end to impose class-specific projection. The deconvolution network has numerous series of unpooling, deconvolution, and rectification layers, and is a mirrored counterpart of the convolution network.....	15
Figure 2.3 Deconvolution and unpooling layers are depicted.....	15
Figure 2.4 An example of U-net architecture for 32x32 pixels in the lowest resolution is shown. A multi-channel feature map is represented by each blue box. On the top of the box, the number of channels is indicated. The x-y-size is indicated on the box's lower left border. Copies of feature maps are represented by white boxes. The arrows represent the various operations.....	16
Figure 2.5 A standard block on the left and a residual block on the right.....	17
Figure 2.6 Network structure of DenseNet.....	17
Figure 2.7 2D vs 3D acquisition mode.....	19
Figure 3.1 k-fold cross validation with a value of three for k. This is the structure of the splits applied to the datasets. The data is divided into three equal parts, two parts are used as training set and the last part is divided into two parts as validation and test sets. For the three splits - Split 1, Split 2 and Split 3 - validation and test are different. In this way three values for the evaluation metrics are obtained and averaged, the average of these three values can then be used to compare with another model.....	22
Figure 3.2 Structure of Unet. Both, down-convolution part (encoder) and the up-convolution part(decoder) are symmetrically composed by convolution 3x3 and Relu, however the encoder is composed by max pooling and the decoder by bilinear up-sample.	
Figure 3.3 Structure of Unet++, the main difference of this network with u-net is the redefined skip pathways, deep-supervision and model pruning.	
Figure 3.4 Unet++ pruned at three levels. The requisite for the network to work at pruned levels is the network to be trained with deep supervision.	
Figure 4.1 3D images of the utero placental pump (training set). Each of these 20 images corresponds to only one of the 50 slices which constitute each 3D images. To visualize the 3D images in 2D, here we are visualising just one slice of each 3D image.....	24
Figure 4.2.A Complete sequence of slices which make the first 3D image.....	24
Figure 4.2.B Complete sequence of slices of masks which correspond to the first 3D image.....	26
Figure 4.2.C Complete sequence of slices which make the first 3D image with mask.....	27
Figure 4.3 Images from the four modes –Image flair, Image t1, Image t1ce and Image t2- and respective ground truth.....	28

Figure 4.4 Full 3D image first mode (Image flair).....	29
Figure 4.5 Full 3D ground truth image.....	30
Figure 4.6 Displayed one hot encoding transformation of the BraTS dataset ground truth.....	31
Figure 5.1 Visualisation of the training process of unet model placenta data (Average values from 3-fold cross validation).....	32
Figure 5.2 Visualisation of the training process of unet++ model placenta data (Average values from 3-fold cross validation for the best model).....	32
Figure 5.3 Visualisation of the training process of u-net model BraTS data (Average values from 3-fold cross validation for the best model)	33
Figure 5.4 Visualisation of the training process of unet++ model BraTS data (Average values from 3-fold cross validation for the best model).....	34
Figure 5.5 Qualitative evaluation of the placenta Unet++ predictions obtained with the last segmentation output (pred_3), second segmentation output (pred_2) and the first segmentation output(pred_1).....	35
Figure 5.6 Qualitative evaluation of the BraTS Unet++ predictions obtained with the last segmentation output (pred_3), second segmentation output (pred_2) and the first segmentation output(pred_1).....	36

Abstract

Medical MRI image segmentation is a critical task performed by radiologist to quantify pathology, without deep learning this task has been handcrafted and time consuming. Last year in UoN a quite successful u-net deep-learning solution was presented, that is why in this work I present a Unet solution as a benchmark and a Unet++ network with the idea of improving previous Unet work. Additionally, to the Placenta work I present implementations of Unet and Unet++ applied to BraTS 2020 data set. With the placenta data set the results outperform previous year, but for BraTS dataset the results were less encouraging explained by limited computation capacity.

The training process was optimized with Adam Optimizer, the loss function was the dice-bce loss function. The unet++ network was trained with deep supervision. The unet++ structure was tested with model pruning, obtaining in this way a model with higher speed inference but with lower predictive capacity.

My performance cross-validated with unseen data for Placenta-UoN is a dice score of 0.9 [0.85, 0.95] for with Unet and 0.92 [0.89, 0.93] for u-net++, and for BraTS 2020 the dice score obtained is 0.71 [0.59, 0.83] for u-net and 0.71 for u-net++ [0.60, 0.81]. The models were trained and tested in a NVIDIA 2080TI GPU card running Red Hat Linux. The code for this implementation is available at: <https://github.com/manuelhz/dissertation>

Acknowledgements

I want to thank to my family for their invaluable and unconditional support during these tough pandemic times, also thank to my research project supervisor Andrew French for his flexibility, support and advice. Thank to Nottingham University medicine researchers Neele Dellschaft and Penny Gowland for providing the unique placenta data set for this project and their time given to explain the data to me.

Chapter 1: Introduction

Semantic segmentation is a computer vision task that is becoming every day more relevant as AI rises and smart devices need to understand their environment and learn how to interact with it. Medicine is not an exception. In the medical field manual segmentation is performed by a specialist. The importance of medical segmentation is to identify and quantify pathology, however, the problem of doing this manually is that it requires too much time, that is why automatic segmentation becomes so relevant.

Working with more than one data-set bring to the table the possibility to test how models perform in datasets with different characteristics.

One of the datasets I worked with is placenta data, this dataset is unique to Nottingham University. It is really important segmenting this placenta as from the volume of the placenta it is possible to know the flow of blood and from its value predicting foetal pathology, however this is something that is being on research. For this type of research the placenta segmentation is carried out manually by a specialist and this takes too much time, an automatic system of segmentation is worth time for analysts and at the same time could help to accelerate the research on detection and quantification of foetal illnesses and save time of research and diagnosis.

Automatic placenta segmentation has not been much researched in the field of computer vision and deep learning, this make it somehow challenging, but at the same time, as not much research exists about this, it should be an opportunity to obtain promising results that in other fields of application, such as brain which has been exhaustively researched, will take a lot more effort.

Brain tumour localisation and segmentation is a difficult task and it requires analysis of different modes as each one shows different parts of the tumour. MRI is a popular exam to detect and segment tumours that is performed by neuroradiologist and is highly difficult and time consuming. BraTS is a well-known competence in the industry which attracts many participants and as it is highly competitive, it is expected to see publications with highly performing results.

In this work the structure of u-net and u-net++ are described, implemented and evaluated their performance with these two different data sets. As u-net structure has proved its effectiveness segmenting placenta imaging in previous research, an improved version of u-net, unet++ is computationally implemented using the framework Pytorch, and different parameters are experimented to obtain the best performance of the network. To prove the effectivity of these network structures in a different context I implemented them with BraTS data as well.

1.1 Description of the problem

Segmentation is a version of the computer vision problem object recognition, the most fundamental approach to simultaneous recognition and segmentation is to think of this as one of classify each pixel in a picture in its class membership. One of the most promising usage of image segmentation is in the medical imaging field, as it may be used to segment anatomical structures for further quantitative study. (Szeliski and Szeliski 2011)

This project applies deep learning on the problem of medical image segmentation. Specifically the implementation of unet and unet++ - which are state of art network architectures – to the placenta segmentation and brain tumour segmentation problems.

1.1.1 Placenta segmentation

The utero-placental pump is a binary segmentation problem, it means that the subregions to classify the picture pixels are: 1) Placenta and 0) Everything else.

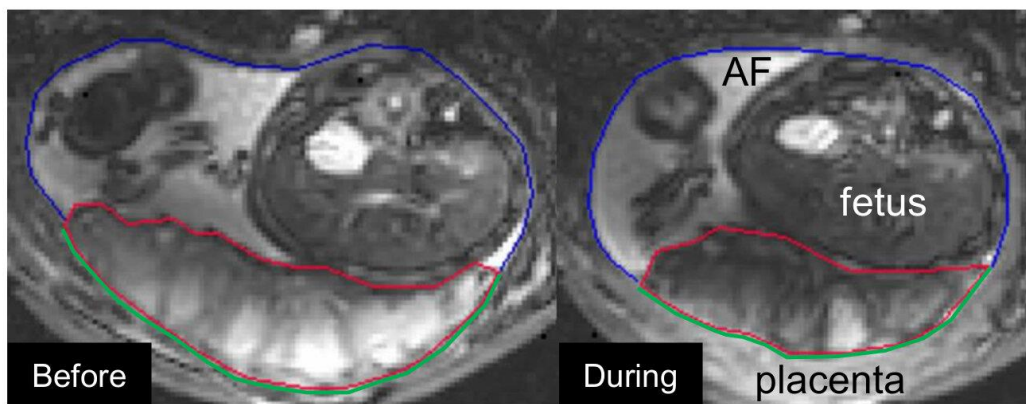


Figure 1.1: “Example of the utero-placental pump, left the relaxed placenta before the contraction, right the contracted placenta (the placenta is the area with the red line around it): it's clear how the placenta is smaller during a contraction, pushing out the maternal blood”. (Nottingham n.d.)

1.1.2 Brain tumour segmentation

The idea of this task is to generate segmentation labels of the different sub-regions of the glioma (tumour). The sub-regions studied for estimation are 1) Enhancing tumour (ET), 2) Tumour core (TC), and 3) Whole tumour (WT). (‘Multimodal Brain Tumor Segmentation Challenge 2020: Data | CBICA | Perelman School of Medicine at the University of Pennsylvania’ n.d.)

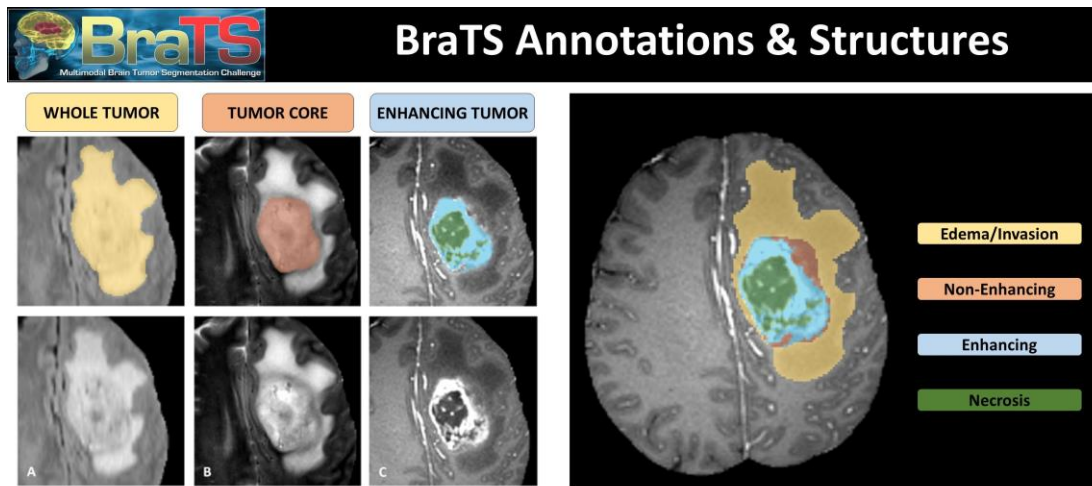


Figure 1.2 Sub-regions of glioma. In the multiple MRI modalities, image patches with tumour sub-regions annotated. From left to right, the picture patches display the whole tumour (WT - yellow) seen in T2-FLAIR (Fig. A), the tumour core (TC - orange) visible in T2 (Fig. B), the enhancing tumour (ET - light blue) evident in T1-Gd, and the cystic/necrotic components of the core (green) (Fig. C). The final labels for the tumour sub-regions (Fig. D) are edema/invasion (yellow), non-enhancing solid core (orange), necrotic/cystic core (green), and enhancing core (red) (blue). (‘Multimodal Brain Tumor Segmentation Challenge 2020: Data | CBICA | Perelman School of Medicine at the University of Pennsylvania’ n.d.)

1.2 Aims and objectives

Develop software python/pytorch for the implementation of model segmentation with Unet and Unet++ to each dataset.

Segmenting the placenta from the utero-placental pump MRI images using Unet and Unet++

Segmenting tumour in its specified regions whole tumour (WT), tumour core (TC), and enhancing tumour.

For the placenta segmentation task one aim is to evaluate performance of the implementation obtained and compare them with previous work in the university. For the BraTS challenge the aim is to compare the results with previous ones in the field

Bibliographical review in other to have an understanding to the state of art in the field.

1.3 Scope

Even though this study involves the use of MRI (Magnetic Resonance Imaging) images as inputs. In this project I disregard medical and medical imaging concepts, as my focus is on the application of machine learning and deep learning algorithms on segmentation of medical images and masks provided by MRI specialist. Out of the scope of this project are any other computer vision of machine learning tasks different to segmentation of the data sets provided and the models developed. Part of this work is the evaluation of the results obtained, but nevertheless, ignored are further applications of the results of the segmentation to the medical field.

Chapter 2: Literature review

Most of the computer vision and segmentation work currently relies on deep learning, this review involves the main network structures that have been used the last years in an evolutive manner since the FCN to Unet++. I also spend chapters reviewing some basic concepts on medical imaging, CT, MRI, multimodal medical imaging, placenta segmentation, Brain tumour segmentation.

2.1 Literature review

2.1.1 FCN

Most current state-of-the-art algorithms for picture segmentation use a fully convolutional network in some form (Long, Shelhamer, and Darrell 2015). This type of network uses a series of convolution layers with the same resolution. The problem of this approach is that it requires to train a high number of parameters which implies that this also is computationally costly. One way of solving this problem is using down sampling with pooling layers, but this would mean loose of resolution, this problem was overcome using a Deconvolution network (Noh, Hong, and Han 2015).

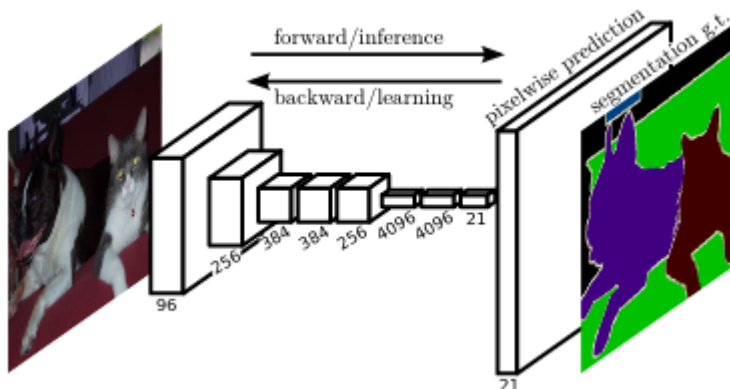


Figure 2.1 The fully convolutional network extracts image features using a CNN, then uses an 1x1 convolutional layer to convert the number of channels into the number of classes, and finally utilises transposed convolution to translate the height and width of the feature maps to those of the input picture (Zhang et al. 2021), image from (Long, Shelhamer, and Darrell 2015).

2.1.2 Deconvolution Network

The deconvolution network proposed by (Noh, Hong, and Han 2015) is divided in two stages: the convolution network and the deconvolution network. The convolution network has a set of convolutions followed by max pooling; The convolution network is a feature extractor that converts an image into a multidimensional feature representation. The deconvolution network, on the other hand, is a shape generator that uses the feature derived from the convolution network to create object segmentation. The network's final output is a probability map with the same dimensions as the input image, showing the likelihood of each pixel belonging to one of the predefined classes.

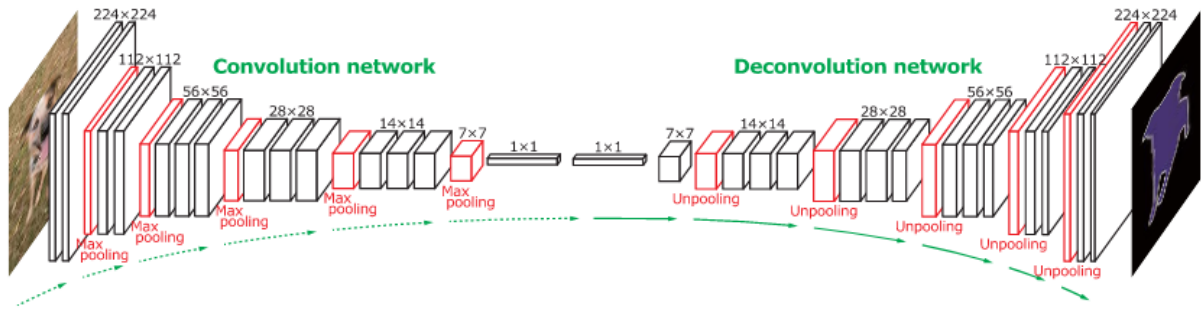


Figure 2.2 Architecture of the deconvolution network, the convolutional network comprises a total of 13 convolutional layers, with rectification and pooling operations conducted between convolutions on occasion, and two fully connected layers augmented at the end to impose class-specific projection. The deconvolution network has numerous series of unpooling, deconvolution, and rectification layers, and is a mirrored counterpart of the convolution network (Noh, Hong, and Han 2015).

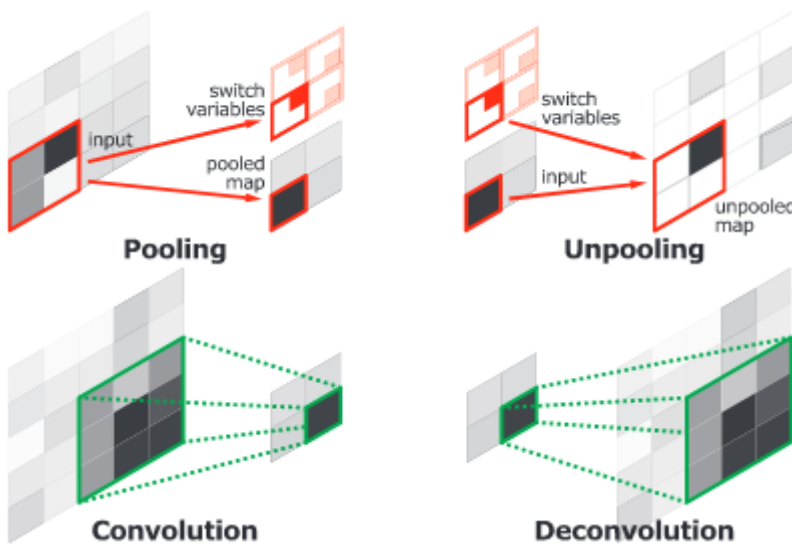


Figure 2.3 Deconvolution and unpooling layers are depicted (Noh, Hong, and Han 2015).

2.1.3 Unet

(Ronneberger, Fischer, and Brox 2015) proposed an improvement to the fully convolutional architecture, Unet, a simetric network with skip pathways in which a contracting path is used to collect context, while a symmetric expanding path is used to ensure exact localisation. Figure 2.4 illustrates the network design.

It is made up of a contracting path (on the left) and an expansive path (on the right) (right side).

The encoder follows the standard architecture of a convolutional network. It comprises of two 3x3 convolutions (unpadded convolutions) that are applied repeatedly, each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. They quadruples the number of feature channels with each downsampling step. An upsampling of the feature map is followed by a 2x2 convolution (“up-convolution”) that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU in the expansive path. Due to the loss of boundary pixels in every convolution, cropping is required. A 1x1 convolution is employed at the final layer to convert each 64-component feature vector to the desired number of classes. The network comprises a total of 23 convolutional layers.

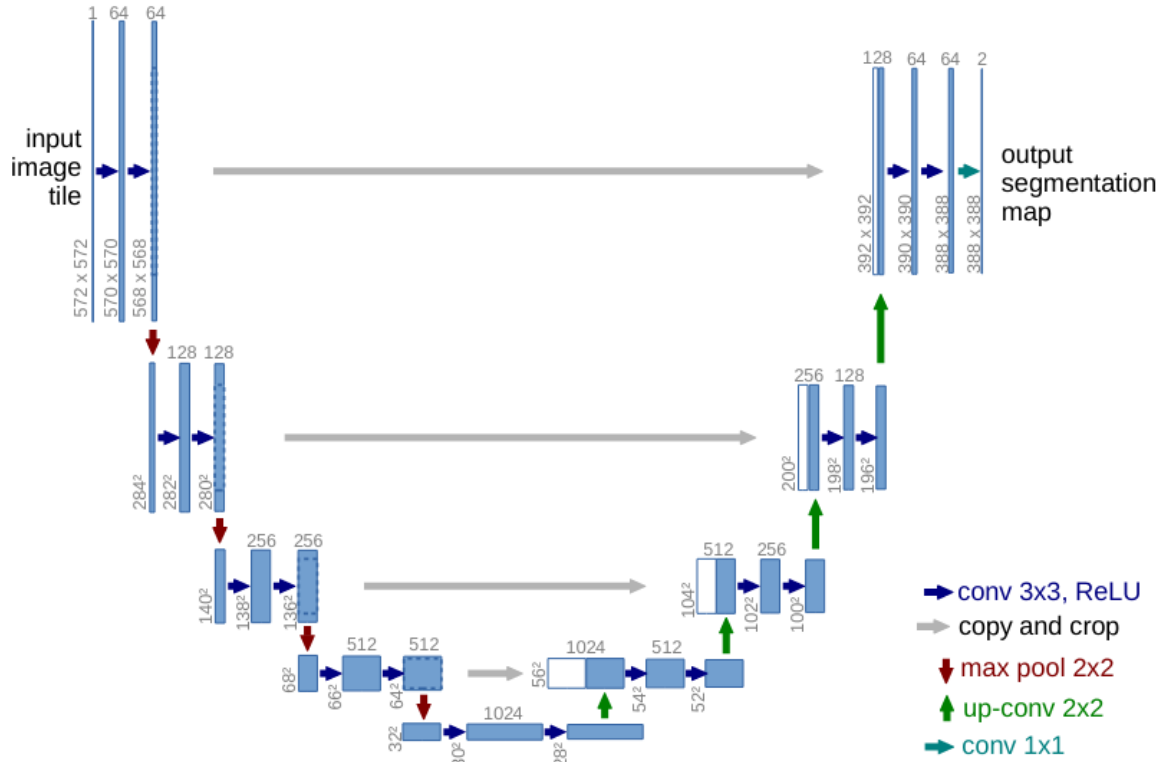


Figure 2.4 An example of U-net architecture for 32x32 pixels in the lowest resolution is shown. A multi-channel feature map is represented by each blue box. On the top of the box, the number of channels is indicated. The x-y-size is indicated on the box's lower left border. Copies of feature maps are represented by white boxes. The arrows represent the various operations. (Ronneberger, Fischer, and Brox 2015)

Even though Unet was proposed in 2015 still in 2020 shows as a focus of research (Song et al. 2020) (Zyuzin and Chumarnaya 2019) (Delibasoglu and Cetin 2020) and several other networks have been developed from this. For example, LadderNet is a network of numerous U-Nets that can be regarded as a chain. A LadderNet includes numerous pairs of encoder-decoder branches with skip connections between every pair of adjacent decoder and decoder branches in each level, whereas a U-Net has only one pair of encoder and decoder branches (Zhuang 2019).

2.1.4 Resnet

Skip connections, or shortcuts, are used by residual neural networks to jump some layers. The residual block of ResNet is depicted on the right in Fig. 2.5, where the solid line transporting the layer input x to the addition operator is referred to as a residual link (or shortcut connection). Inputs can propagate faster through the residual connections across layers using residual blocks. (Zhang et al. 2021)

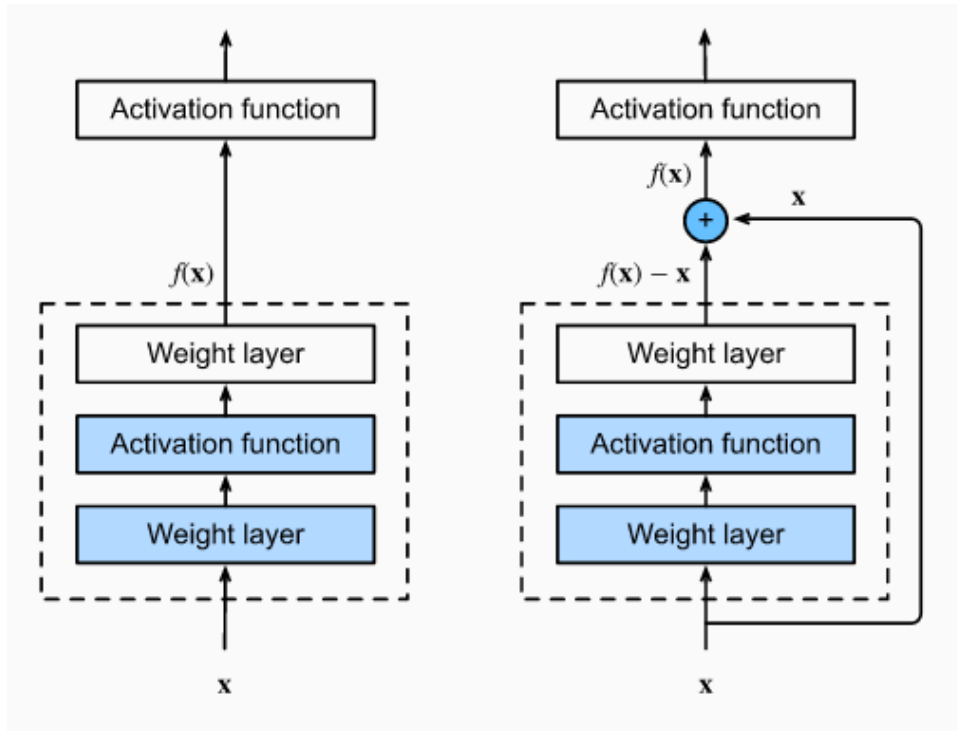


Figure 2.5 A standard block on the left and a residual block on the right. (Zhang et al. 2021)

2.1.5 Densely Connected Convolutional Networks (DenseNet)

Dense Convolutional Network (DenseNet) is a feed-forward network that connects each layer to every other layer. All previous layers' feature-maps are utilised as inputs into each immediately posterior layer, and its own feature-maps are used as inputs into all subsequent layers. DenseNets have a number of compelling advantages, including the elimination of the vanishing-gradient problem, improved feature propagation, feature reuse, and a significant reduction in the number of parameters.

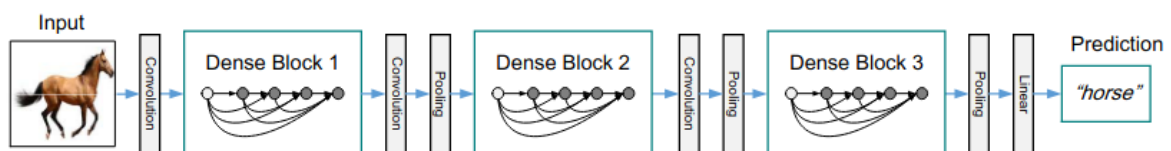


Figure 2.6 Network structure of DenseNet. Image from (Huang et al. 2018).

DenseNets take advantage of the network's potential by reusing features, resulting in condensed models that are simple to train and parameter efficient. Concatenating feature-maps learned by distinct layers improves efficiency and enhances variety in the input of following levels. (Huang et al. 2018)

Even though DenseNet - According to my bibliographical research - hasn't been used much for image segmentation, the idea of densely connected layer has been used a little bit more. (Krešo, Krapac, and Šegvić 2019) propose a unique DenseNet-based ladder-style architecture with great modelling power and a relatively slim datapath for upsampling. They also propose that by utilising the DenseNet feature extractor's inherent spatial efficiency, they can significantly minimise the amount of feature map caching. The generated models outperform competitor approaches by requiring fewer parameters and allowing training at megapixel resolution on inexpensive hardware.

(Wu et al. 2019) propose a cascaded FC-DenseNet model of coarse-to-fine segmentation for autonomous kidney segmentation. The prediction probability that a pixel will become the proper class label is determined using the coarse segmentation model, and the proposal region of the kidney in the ultrasound picture is derived. The proposal region of the kidney is finely segmented and restored to its original size

after training the fine segmentation model. Cascaded FC-DenseNet outperforms the original FC-DenseNet when it comes to kidney segmentation.

(Yuan et al. 2019) To partition the prostate region automatically, they presented a novel Encoder-Decoder Densely Connected Convolutional Network (ED-DenseNet). Their model is made up of two interconnected pathways: a dense encoder pathway that learns discriminative high-level image features, and a dense decoder pathway that predicts pixel-level segmentation. Instead of employing a convolutional network as the fundamental unit in the encoder-decoder architecture, they used a Densely Connected Convolutional Network (DenseNet) that uses a densely-connected method to maintain the maximum information flow across layers.

DenseNet and ResNet from a high-level point of view look that they are quite similar in the way that both connect preceding features to posterior layers, but they vary in that ResNet uses summation to connect all prior feature-maps, whereas DenseNet concatenates them all. (Wang et al. 2018) taking advantage of this fact introduced a modularized Mixed Link Network (MixNet) with modular inner (addition) and outer (concatenation) link modules that is highly efficient.

2.1.6 Deep supervision

Deeply-Supervised Nets was introduced by (Lee et al. 2014), It is a technique that minimises classification error while also making the hidden layer learning process direct and transparent. The effect of intermediary layers on total classification, robustness of learned features, and avoiding "exploding" and "vanishing" gradients are all considered in this architecture. At all intermediate layers, auxiliary classifiers are added, and their losses are summed to the final layer's loss. (Wang et al. 2015) have used the same approach of DSN adding supporting classifiers in halfway convolutional layers "motivated by studying vanishing gradients in deep networks", but this time training deeper networks.

Even though – according to my bibliographical research- deep supervision or DSN has not been a popular approach –such as Unet or ResNet for example- it took my attention because it has been used in the development of Unet++ and it still has a respectable number of citations.

2.1.7 Unet ++

Unet ++ is a CNN structure which comes from Unet. The novel of this structure is that mixes the last one with elements of DenseNet and DSN (Deeply-Supervised Nets). These characteristics make this structure efficient and modular. Densely connected skip connections have been proved by (Huang et al. 2018) as an efficient structure in terms of number of parameters, and on the other hand the use of deep supervision allows the use of the model with different accuracy depending on the need of speed of the application (Zhou et al. 2018), having off course a trade-off speed/accuracy.

2.1.8 Medical imaging

The purpose of medical imaging is to provide information about the interior anatomy of the human body in a non-invasive manner. The central idea is that the patient is scanned using some form of radiation, and the interaction between the radiation and the body is measured. This is the data that must be traced back to its source. As a result, we have an inverse problem. There are a variety of imaging techniques available, as well as diverse approaches to define them. Whether the source is inside or outside the body, i.e., whether we use emission or transmission tomography, makes a significant impact for the patient. (Louis 1992)

The generated information can be used to discriminate between different approaches from a diagnostic standpoint. X-ray computer tomography, ultrasonic computer tomography, and diffuse tomography are examples of technologies that provide information regarding tissue density. With magnetic resonance imaging and impedance computer tomography, it is feasible to distinguish between tissue qualities. Finally, biomagnetism (electrical activities) and emission computer tomography can be used to identify activity locations (nuclear activities of injected pharmaceuticals). (Louis 1992)

Magnetic Resonance Imaging (MRI) is a non-invasive imaging technique that creates comprehensive three-dimensional anatomy images. It's frequently used to detect diseases, diagnose them, and track their progress. It is based on cutting-edge technology that excites and detects changes in the rotational axis of protons in the water that makes up biological tissues. ('Magnetic Resonance Imaging (MRI)' n.d.)

Despite to the fact that MRI is a 3D imaging technique, according to the way in which the image is been collected, it is possible to classify them in two types: 2D acquisition and 3D acquisition. In a 2D acquisition, individual cross-sectional images representing slices of the anatomy are captured, despite showing a very thin 3D volume of tissue, each image is two-dimensional. A three-dimensional acquisition encodes a whole volume of tissue, allowing each volume element to be found retroactively. This means that images can be rebuilt in any plane and at any angle, for example. In addition, 3D volumes can be reassembled into CGI anatomy models and used in surgery planning. ('What Is the Difference between MRI Data Collected in 2D versus 3D?' n.d.)

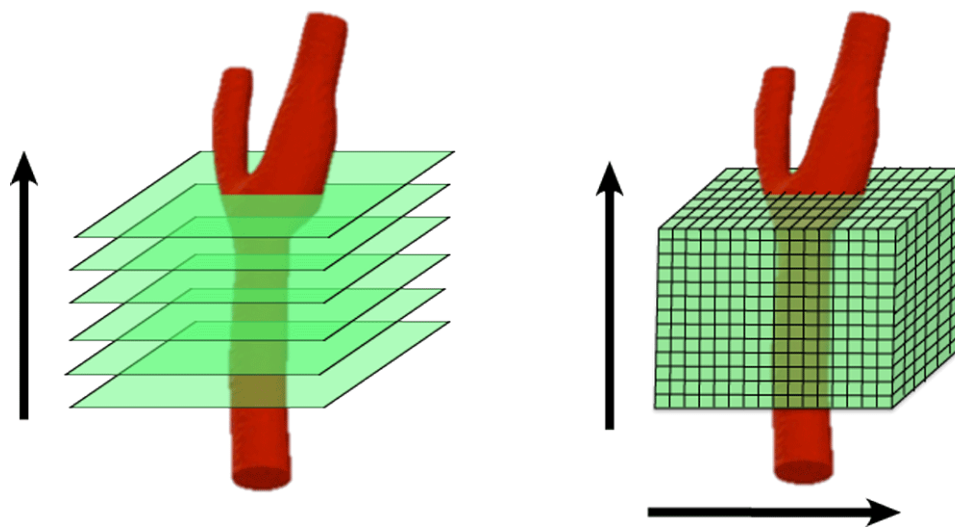


Figure 2.7 2D vs 3D acquisition mode, image from ('2D vs 3D MRA' n.d.).

2.1.9 CT vs MRI

A CT scan (computed tomography) is a type of imaging procedure that employs X-rays to make detailed images of organs, bones, and other tissues. The person is positioned on a table that rotates around a scanning ring that resembles a huge doughnut. The information gathered can be combined to create three-dimensional images. The pictures indicate anomalies in both bone and soft tissue, such as lung pneumonia, tumours in various organs, and bone fractures. (Stallard n.d.)

MRI generates comprehensive images of locations inside the body as well, but it does so using radio waves and a powerful magnet. The person is similarly lying on a table that transforms into a doughnut-shaped gadget, but the doughnut is far thicker. Similarly, these images can distinguish between healthy and sick tissue. (Stallard n.d.)

In a few seconds, a CT scan may produce an image of practically the complete body, from the neck to the thighs. CT scans are extremely important for diagnosing and staging cancer, determining whether it has returned, and evaluating the effectiveness of a treatment. It's particularly useful for scanning the entire body for areas where cancer has progressed, such as the lungs, liver, or bones. These are referred to as metastases. CT scans are the most common way to stage cancer. (Stallard n.d.)

The MRI particularly shines when it comes to detecting disorders that a CT scan can't. On a CT scan, some tumours, such as prostate cancer, uterine cancer, and some liver cancers, are virtually undetectable. On an MRI, metastatic bone and brain metastases are also more visible. (Stallard n.d.)

CT scans employ ionising radiation, which can damage DNA and increase the risk of cancer by a small amount. According to the Food and Drug Administration, a conventional CT procedure increases the risk of a person having a deadly malignancy by around 1 in 2,000. Because MRIs do not employ ionising radiation, there is no risk of cancer. They do, however, take much longer to complete than CT scans. For an MRI, the patient must lie still for 20 to 40 minutes in a closed environment. Some persons with claustrophobia may be affected, and the process is noisy, so ear protection is given. (Stallard n.d.)

2.1.10 Multimodal medical imaging

Medical imaging modalities are utilised in medicine and are based on a variety of energy sources, such as light, X-rays, electrons, radionuclides, lasers, ultrasound, and nuclear magnetic resonance (NMR). Mammography, radiation treatment, CAT SCANS, and radiofrequency (RF) X-ray machines are all imaging modalities used in radiology technologies. Image fusion allows medical imaging modalities to be integrated to create a full picture of the human body. The combination of positron emission tomography (PET), a nuclear medicine approach that scans metabolic processes in 3-D, and computed tomography (CT) to develop PET/CT imaging to find cancer spots is an example of image fusion. ('Medical Imaging Modalities Used in Medicine | Open Medscience' n.d.)

Multimodality imaging, which combines multiple imaging modalities, may be a better way to get around the limits of individual techniques. For each pre-clinical investigation, this strategy could yield a lot of data. BLI, for example, produces only a 2D image with minimal spatial information for cell tracking. MRI and PET, on the other hand, have a higher spatial resolution and can be utilised in clinical imaging, although they have a much lower detection threshold. The extent of information supplied will be improved and expanded by combining imaging modalities. (van der Horst, Buijs, and van der Pluijm 2015)

2.1.11 Placenta segmentation

To obtain reliable placenta segmentations from MRI, (Wang et al. 2015) suggest Slic-Seg, a slightly interactive online learning-based method. An online random forest is first trained using data from the user's handwriting in an individual chosen start slice. This then serves as the foundation for a slice-by-slice structure that segments successive slices before inserting them into the training set as they come.

(Stevenson et al. 2015) present a semi-automated method based on the Random Walker algorithm, however method presented requires initialization.

To find placental potential pixels automatically, (Alansary et al. 2016) used a 3D multiscale convolutional neural network. A 3D dense conditional random field refines the categorization, allowing a high resolution placental volume to be recreated from many overlapped layers of slices.

(Looney et al. 2017) used the output of the semi-automated Random Walker method as ground truth to train a deep convolutional neural network (cNN), DeepMedic. The placenta was segmented from 3D ultrasound data using a previously released open-source convolutional neural network, DeepMedic.

(Han et al. 2019) present an automated human placenta segmentation approach based on u-net, which decreases manual intervention and substantially speeds up segmentation, allowing for large-scale segmentation.

For automated segmentation of the placenta, (Twickler et al. 2020) created a modified version of U-Net.

2.1.12 Brain tumour segmentation

On the Multimodal Brain Tumor Segmentation Challenge (BraTS) 2020 dataset, (Henry et al. 2020) trained numerous U-net-like neural networks, mostly with deep supervision and stochastic weight

averaging. Two separate training pipelines were used to train two different combinations of models, each of which produced a brain tumour segmentation mapping. The performance of each combinations for specific tumour segments was subsequently taken into account while merging these two label-mapping per patient. They are ranked together with the top ten teams.

(Fidon, Ourselin, and Vercauteren 2021) use a non-standard loss function, the generalized Wasserstein Dice loss, and a non-standard optimizer, Ranger, to implement a generic and state-of-the-art 3D U-Net architecture. Those variants were chosen particularly for the challenge of segmenting multi-class brain tumours. The generalised Wasserstein Dice loss is a per-sample loss function that takes use of the tumour areas' hierarchical nature in BraTS. Ranger is a version of the popular Adam optimizer that performs better with small batches and noisy labels. They discovered that each of these versions of deep neural network optimization for brain tumour segmentation results in better Dice scores and Hausdorff distances.

Chapter 3: Methodology

In this chapter I present the network structures implemented, the perspective used throughout the project to train the models and evaluate and compare which model performs better than others.

3.1 Model evaluation

3.1.1 Evaluation metrics

In order to obtain an appreciation of the performance of the models and an understanding about which models performs better than others in each data set I considered a set of evaluation metrics to do so. There so are many metrics that can be used that is why it is not practical to use all of them in this project. Also, metrics must be adapted to the type of project. For this implementation, the decision was that Dice loss is the main evaluation metric to use, the reasons are first than all because previous work in UoN in the placental segmentation challenge have been evaluated using this metric and I also will need it to compare with previous results. Another measure widely used for segmentation is BCE or Binary Cross Entropy. During training a combination of binary cross-entropy and dice coefficient is used to evaluate the performance of the model, according to the paper, this loss function is the same as used in the unet++ original implementation (Zhou et al. 2018).

3.1.1.1 Binary Cross Entropy

Binary cross entropy is the same loss function used with logistic regression (Bishop 2006) and it widely used for classification and classification problems.

$$L(Y, \hat{Y}) = -\frac{1}{N} \sum_{n=1}^N [y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n)]$$

3.1.1.2 Dice coefficient

Dice coefficient is a similarity statistic and expresses the relation between pixel overlapping (between the predicted classes and the ground truth) and the total number of pixels of the prediction plus the ground truth. Its value goes from zero to one, zero means no overlap and one mean full overlap. One advantage of the dice coefficient is that it can be used as a dissimilarity measure and indeed it is possible to use it as a loss function (Dice loss) ('Sørensen–Dice Coefficient' 2021).

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

3.1.1.3 BCE-Dice loss

BCE-Dice loss is a combination between binary cross entropy and dice loss. This is represented as:

$$L(Y, \hat{Y}) = -\frac{1}{N} \sum_{n=1}^N \left[\frac{1}{2} Y_b \log \hat{Y}_n + \frac{2Y_n \hat{Y}_b}{Y_b + \hat{Y}_b} \right]$$

Where \hat{Y}_n and \hat{Y} indicate the flatten predicted probabilities and the flatten ground truths of n^{th} image respectively and N the batch size (Zhou et al. 2018).

3.1.2 k-fold cross validation

In this project K-fold cross validation is used to carry out more representative evaluation of the inference capacity of the model. Cross-validation is one of the most extensively used data resampling strategies for assessing a predictive model's generalisation capacity and preventing over-fitting. The learning function (or learning algorithm) f is usually used to the complete learning set to develop the final model for the prediction of real future cases. Cross-validation is not possible with this final model. Cross-validation is used in the model development phase to offer an estimate of the final model's performance on new data. For example, in a k-nearest neighbour classifier, cross-validation is widely used to optimise model parameters, such as the ideal number of nearest neighbours. The tuning parameter is cross-validated numerous times for different values, and the parameter that minimises the cross-validated error is then applied to generate the final model. Cross-validation thereby overcomes the issue of overfitting. (Berrar 2019)

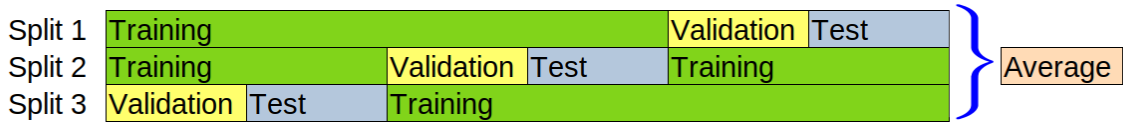


Figure 3.1 k-fold cross validation with a value of three for k . This is the structure of the splits applied to the datasets. The data is divided into three equal parts, two parts are used as training set and the last part is divided into two parts as validation and test sets. For the three splits - Split 1, Split 2 and Split 3 - validation and test are different. In this way three values for the evaluation metrics are obtained and averaged, the average of these three values can then be used to compare with another model.

3.2 Model implementation

3.2.1 Training and overfitting

Overfitting occurs when the model memorises the data instead of generalization. To avoid overfitting, I split the data in training, validation, and test. The training set is used to carry out the forward and back-propagation to learn the internal parameters. After each epoch, the loss is calculated with the validation set. If the model does not improve the validation loss, the model with the best validation loss is kept for the next epoch.

3.2.2 Unet

The network is divided into the down-convolution part and the up-convolution part, sometimes the down-convolution part is called encoder and the up-convolution-part is called decoder.

The down convolution begins with the image 256x256 and one channel, this is followed by four double convolutions that increase the number of channels to 64, 128, 256 and 512, respectively. The convolution 1, 2 and three are followed by a max pool 2x2 which decreases the resolution to 128x128, 64x64 and 32x32.

The up-convolution part consists of three up-sample, copy and double convolution. The resolution is increased twice in each up-sample, and the number of channels is increased by adding a copy from the down-convolution side. Each double convolution decreases the number of channels to its equivalent up-convolution part.

Finally, the last convolution 1x1 decreases the number of channels to the number of masks necessary to predict, in this case just one.

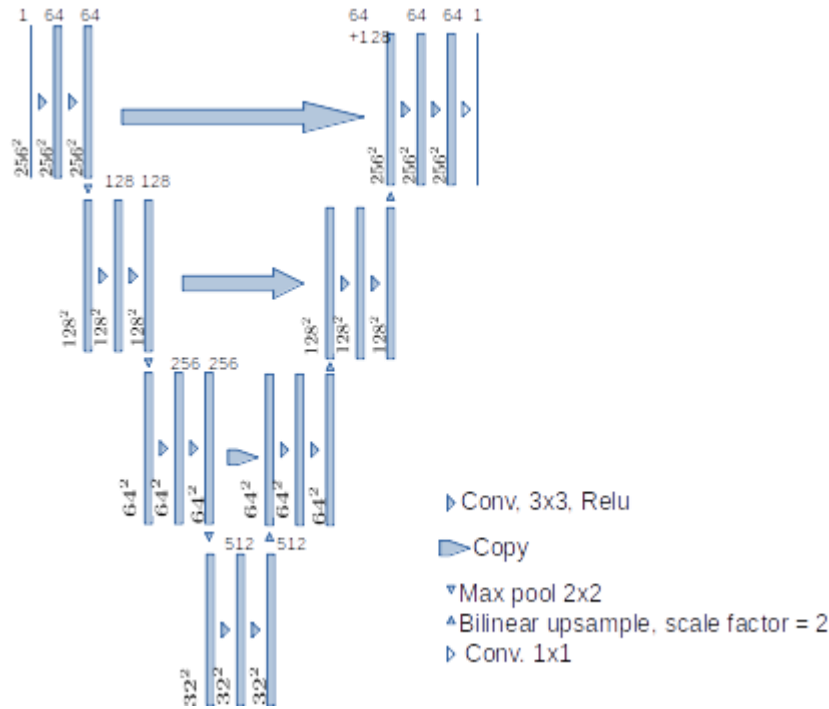


Figure 3.2 Structure of Unet. Both, down-convolution part (encoder) and the up-convolution part(decoder) are symmetrically composed by convolution 3x3 and Relu, however the encoder is composed by max pooling and the decoder by bilinear up-sample.

3.2.3 Unet plus plus

As Unet network Unet++ has a down convolution and an up-convolution part. The down convolution receive an image of 256x256 pixels and one channel as input, after this, four double convolutions increase the number of channels to 64, 128, 256 and 512, respectively. This Unet++ implementation differs from the Unet in that the last one has one extra max pool layer decreasing the resolution to 128x128, 64x64 32x32 and 16x16 in the end. Each convolution has also a batch normalization function.

The decoder of this implementation has four convolutions, but as in the Unet, three of them receive input from its same semantic level encoder.

One special feature of Unet++ is its pass forward with a dense convolutional block from the encoder to the decoder. Another feature of this structure is deep supervision and model pruning. Deep supervision means to obtain outputs –that is predictions- of the model before the last layer. Those early predictions pass through fewer layers. The loss function is then calculated for each early layer, after that, the final loss function is the sum of all the early loss and the last layer. For example as we see of figure 3.3 Y_2 corresponds to the output of the final layer and this is using the full model. During deep supervision, the loss of the outputs Y_0 and Y_1 are also calculated. The final loss corresponds to the sum of the loss from Y_0 to Y_2.

According to (Zhou et al. 2018) if this network structure is trained with deep supervision, then it is possible to prune the resulting model. Model pruning is carried out using Y_0 or Y_1 instead of Y_2 as a model prediction. Y_2 will have the highest accuracy and Y_0 the lowest one.

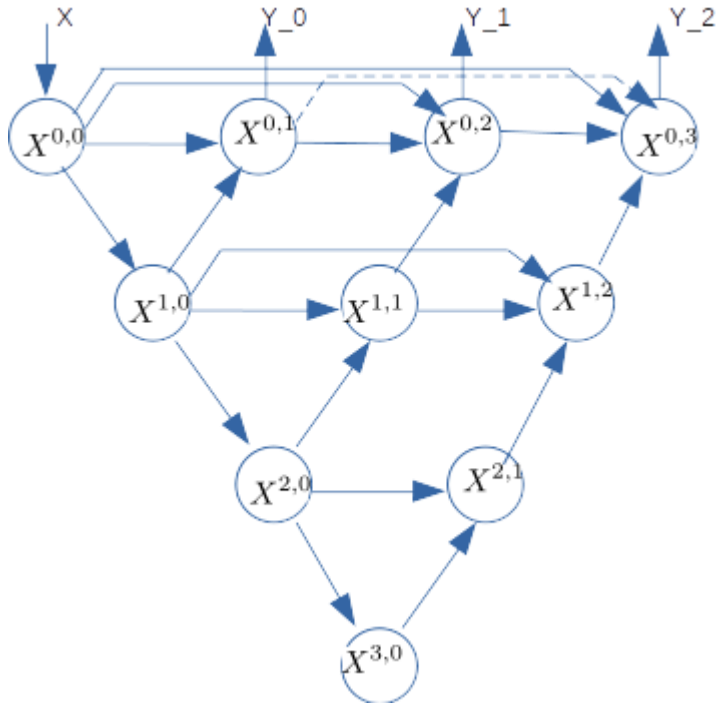


Figure 3.3 Structure of Unet++, the main difference of this network with u-net is the redefined skip pathways, deep-supervision and model pruning.

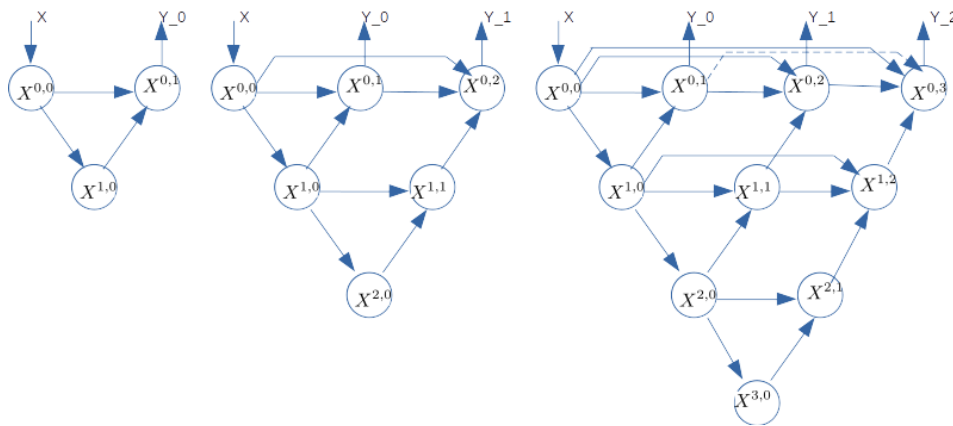


Figure 3.4 Unet++ pruned at three levels. The requisite for the network to work at pruned levels is the network to be trained with deep supervision.

Chapter 4: Data analysis and processing

As 3D and multimodal images are not the same as tabular information, one of the challenges of this project was to represent it in 2D. This, and its processing are described in this chapter.

4.1 Three-dimensional data segmentation

Both placenta and BraTS data are 3D images, based on (IEEE Staff 2019) the approach I used to segment 3D medical images is processing them as a set of 2D slices, it means applying 2D segmentation to 3D images.

4.2 Placenta data set

To understand the 21 3D images which compose the dataset, I take just one slice of each image. I take a slice from the center of the 3D image in which I can see the placenta as big as possible. I did so because as we see on the Fig. 4.2.B on the first three and on the last six slices placenta is not visible, but on the slice 24, placenta is quite big. In this way I can see how much placenta changes over the data. The placenta data set is a sequence of 21 3D images of $256 \times 256 \times 50$ pixels. On the Fig. 4.1 we can see the first 20 images sequence for the slice 24.



Figure 4.1 3D images of the utero placental pump (training set). Each of these 20 images corresponds to only one of the 50 slices which constitute each 3D images. To visualize the 3D images in 2D, here we are visualising just one slice of each 3D image.

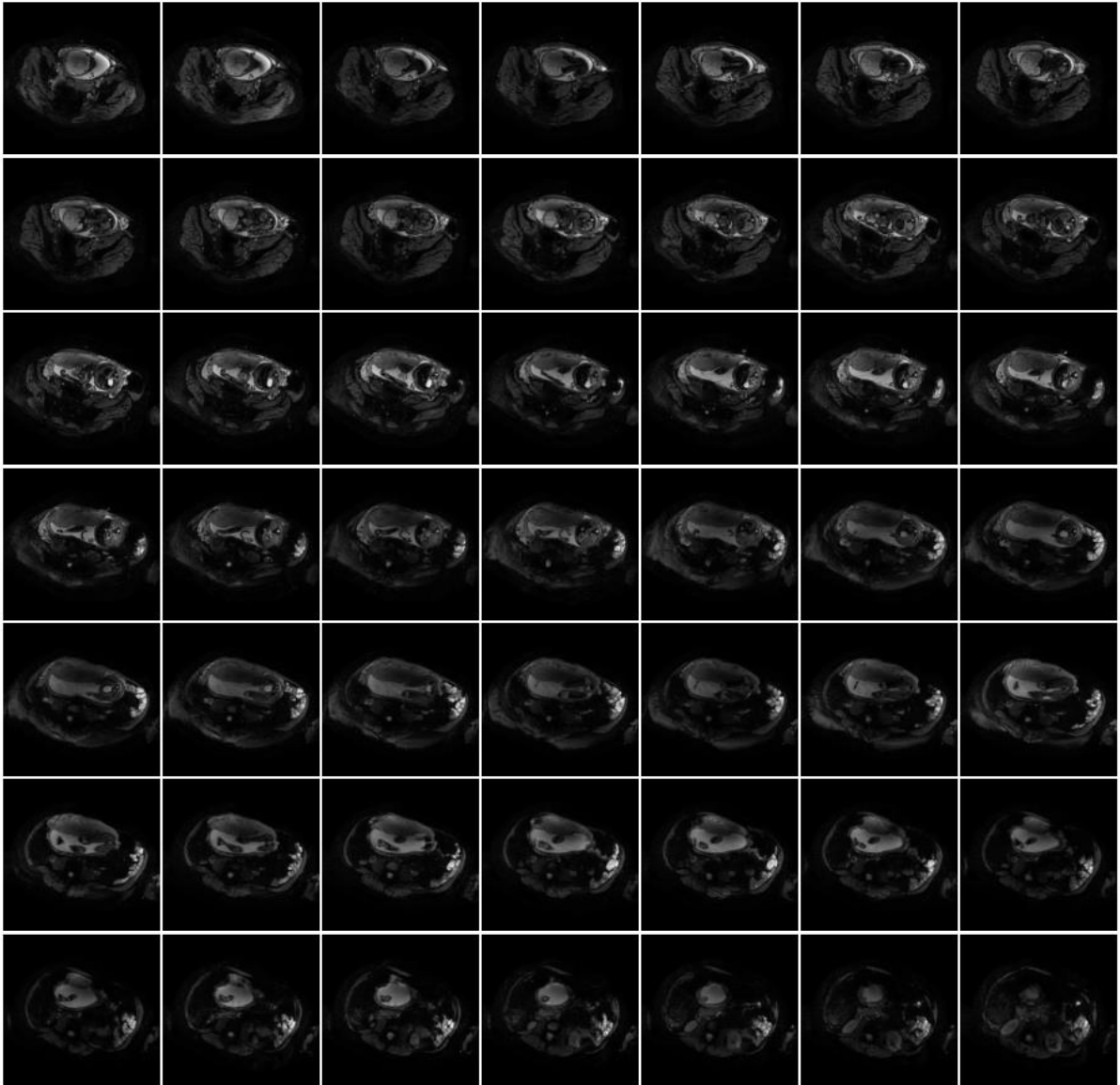


Figure 4.2.A Complete sequence of slices which make the first 3D image.

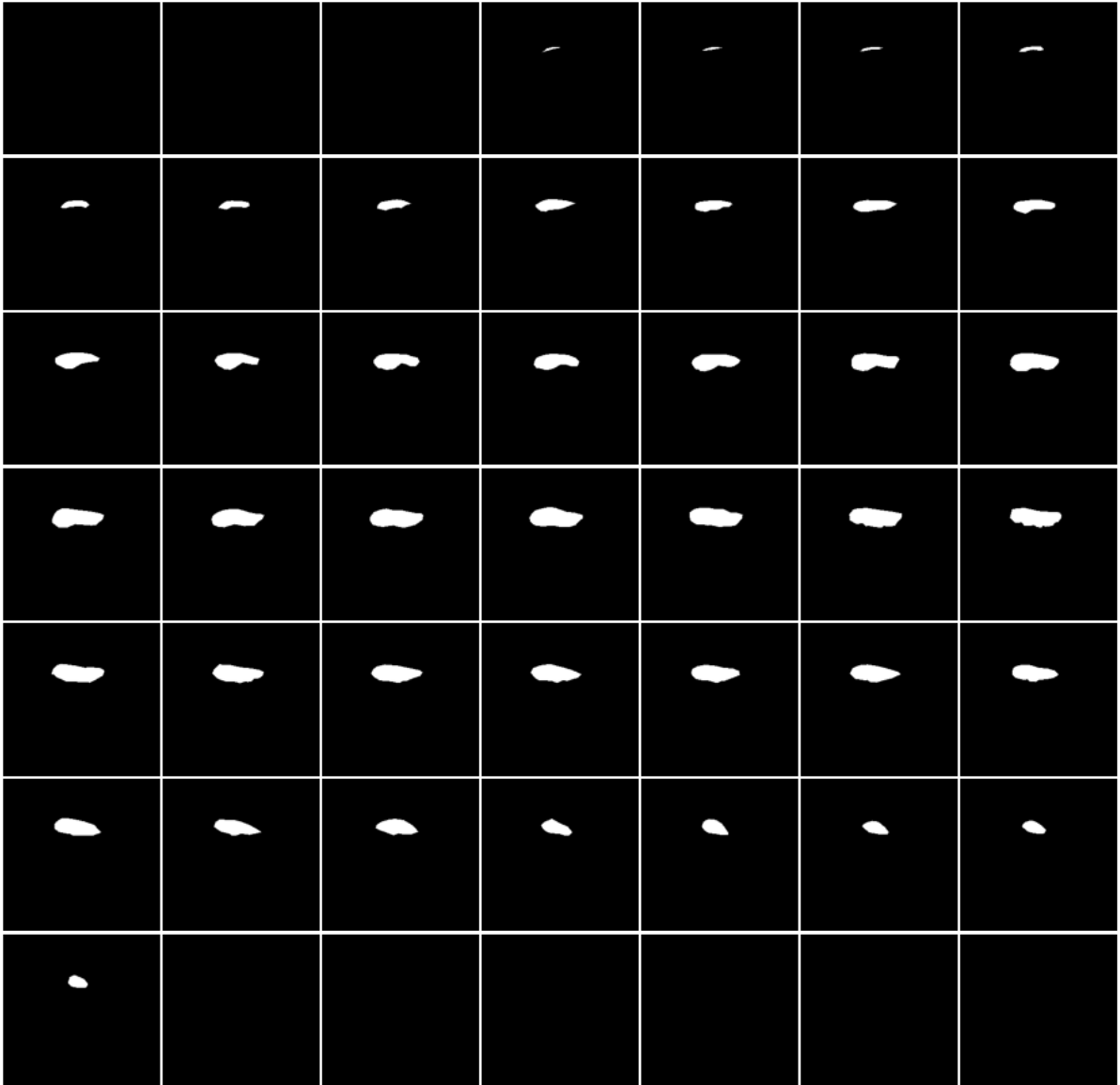


Figure 4.2.B Complete sequence of slices of masks which correspond to the first 3D image.

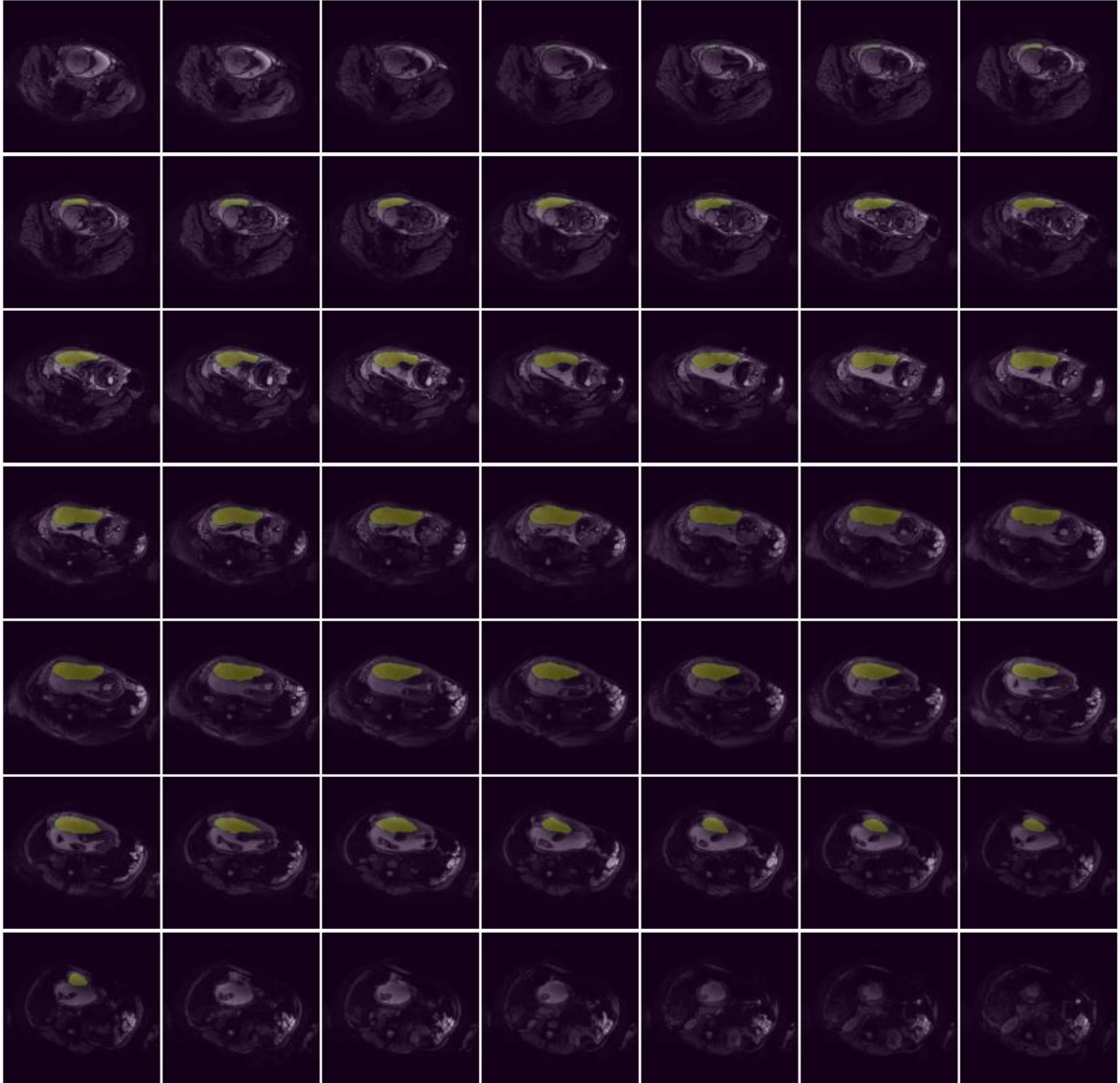


Figure 4.2.C Complete sequence of slices which make the first 3D image with mask.

4.3 BraTS data set

4.3.1 Multimodal learning

This data set is multi modal MRI 3D image, it means that more than one 3D MRI image mode have been used at the same time, in order to predict the labels, and it is expected a better performance of inference using all the modes provided than using just one of them. The modes provided on this dataset are four as follows: a) native (T1), b) post-contrast T1-weighted (T1Gd), c) T2-weighted (T2), and d) T2 Fluid Attenuated Inversion Recovery (T2-Flair) and were obtained with a variety of clinical protocols and scanners from various (n=19) institutions.

And to feed the deep learning model with the four modes I am feeding the network with one mode per channel. After initiated the training of the model with this data I realised that the time consuming during the training process was not affordable (may be more than a week per each epoch). To make the training process quicker and carry out the experiments in an affordable timespan, it was necessary to reduce the resolution of the 3D images from (240 x 240 x 155) pixels to (120 x 120 x 78) pixels and work with a

fraction of the data. Another reduction of size applied to this data was working with just one of the modes, based on ('MRI Basics' n.d.) I worked with the mode Image flair as it is expected to be more important than the other modes.

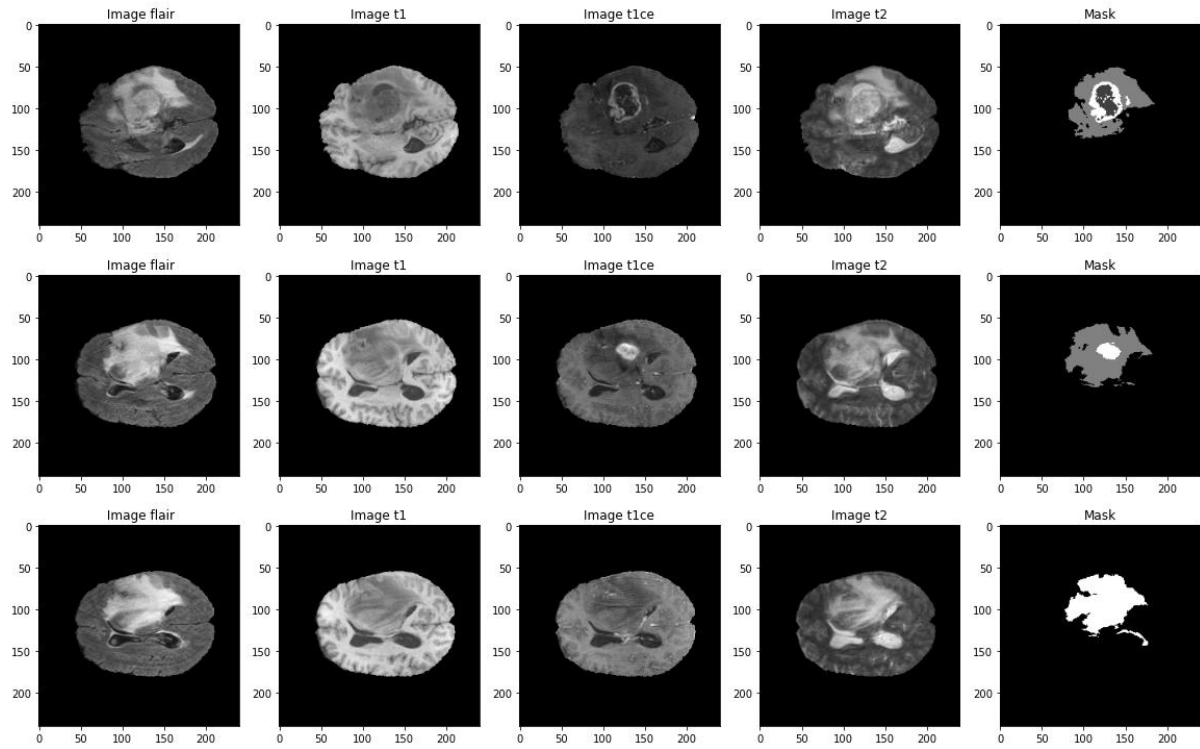


Figure 4.3 Images from the four modes –Image flair, Image t1, Image t1ce and Image t2- and respective ground truth.

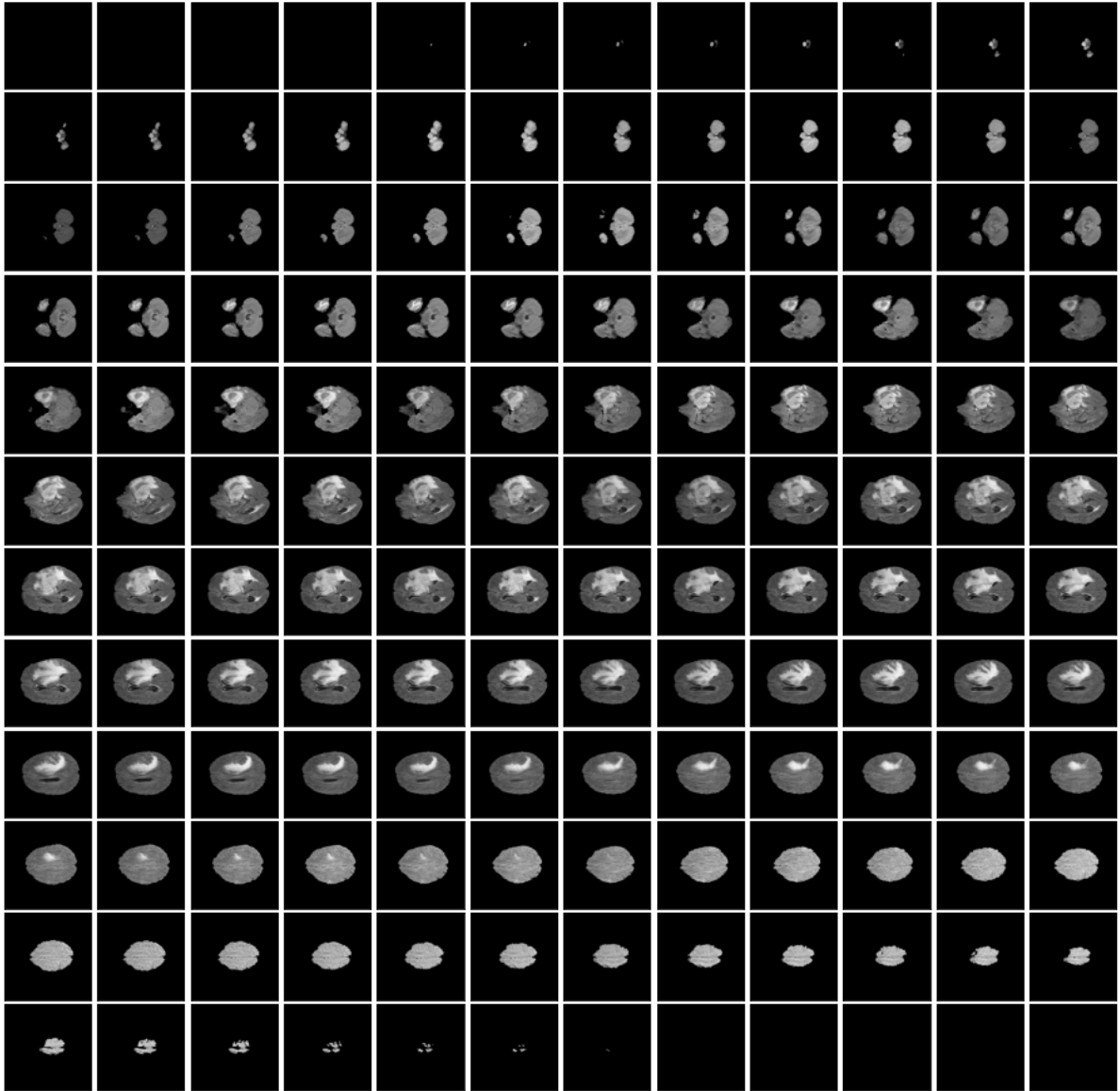


Figure 4.4 Full 3D image first mode (Image flair).

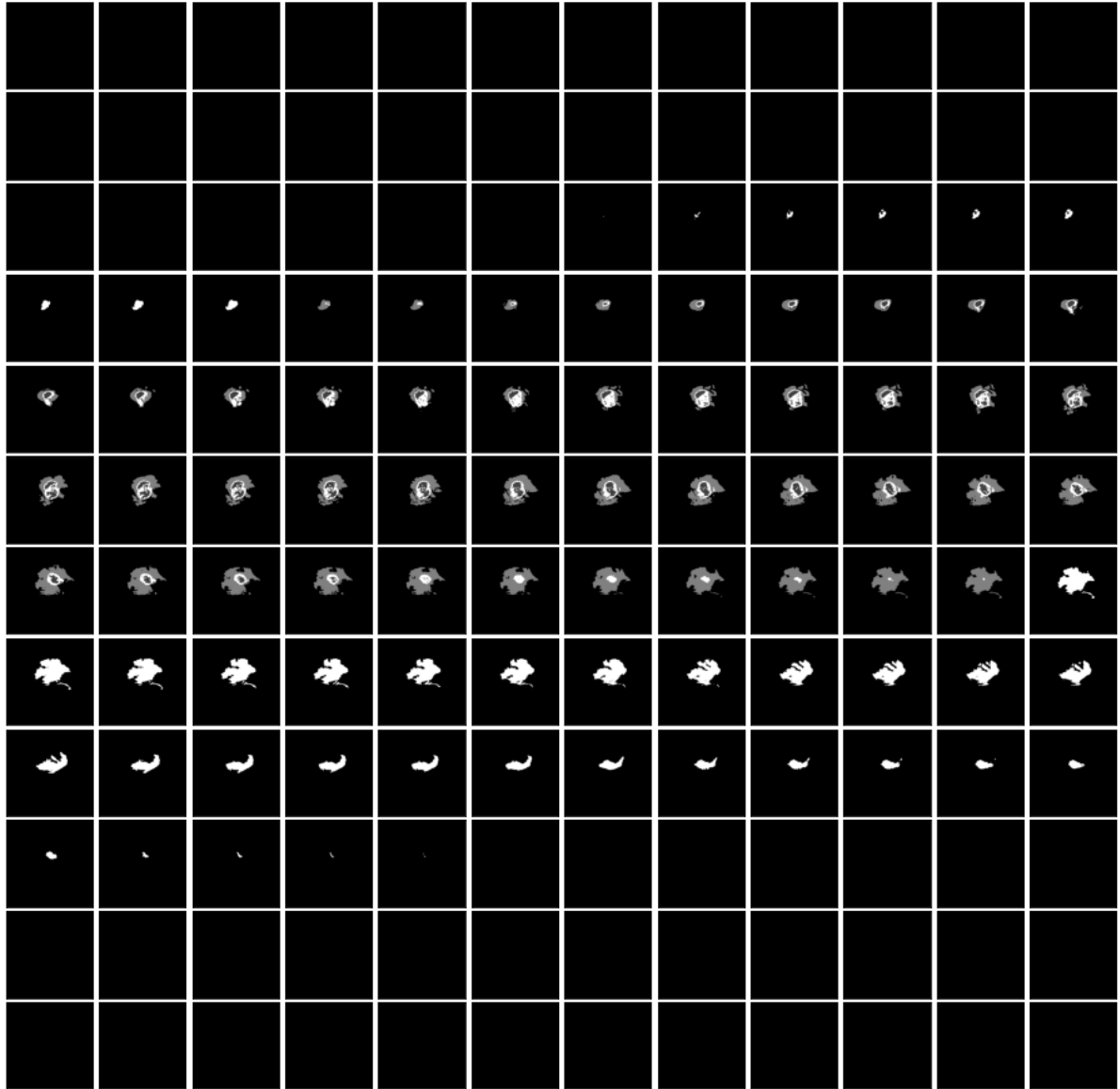
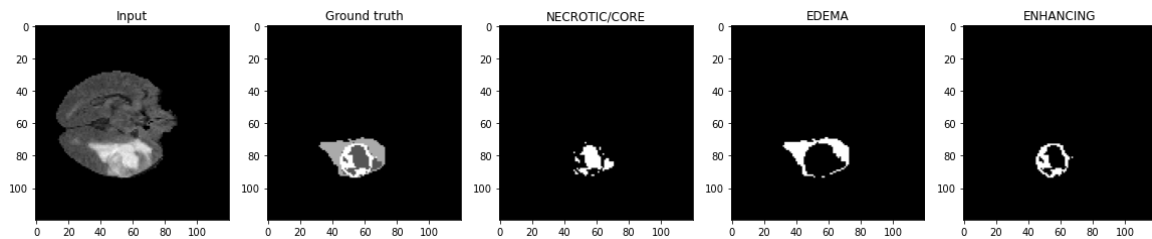


Figure 4.5 Full 3D ground truth image.

4.3.2 One hot encoding

The original data-set contains three classes stored in a categorical variable, for that reason, transforming was necessary to obtain one hot encoded images.



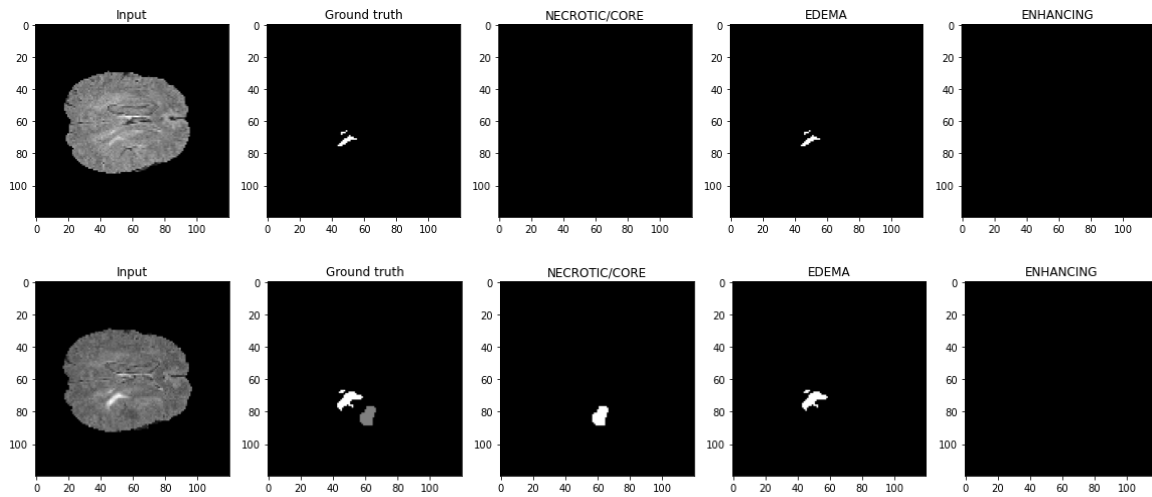


Figure 4.6 Displayed one hot encoding transformation of the BraTS dataset ground truth.

Chapter 5: Experiments

All the experiments have been cross validated and performed on GPUs 2080TI.

5.1 Learning ratios

It was observed that the learning ratio is highly influential in the performance of the models, for that reason I carried out experiments with different learning ratios to tune this parameter.

5.1.1 Placenta Unet learning ratios

As I show in the table 5.1, with a learning ratio of 1E-4 a dice coefficient of 0.9 has been obtained.

Table 5.1 bce, dice and loss obtained with a learning ratio of 1E-4. It includes the dice score, its standard deviation, LI and UL are the lower and the upper limits of the confidence interval with a 95% of confidence.

LR	bce	dice	loss	SD	LI	LU
1.00E-04	0.02508	0.90043	0.06232	0.02143	0.84720	0.95366

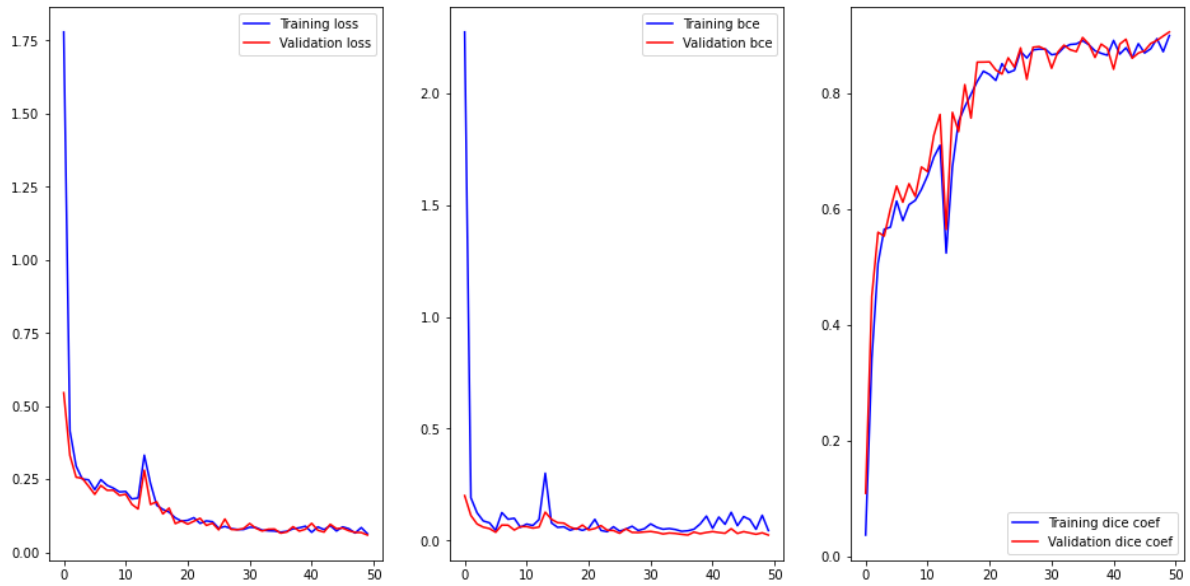


Figure 5.1 Visualisation of the training process of unet model placenta data (Average values from 3-fold cross validation)

5.1.2 Placenta Unet++ learning ratios

For the placenta data set as applying u-net++ model the performance was improved using a lower learning ratio $1E-3$ than with u-net.

Table 5.2 bce, dice and loss obtained with a learning ratio of $1E-4$ and $1E-3$ for placenta and unet++. It includes the dice score, its standard deviation, LI and UL are the lower and the upper limits of the confidence interval with a 95% of confidence.

LR	bce	dice	loss	SD	LI	LU
1.00E-04	0.08452	0.62585	0.35518	0.11381	0.34312	0.90858
1.00E-03	0.00693	0.91505	0.04594	0.00663	0.89859	0.93151

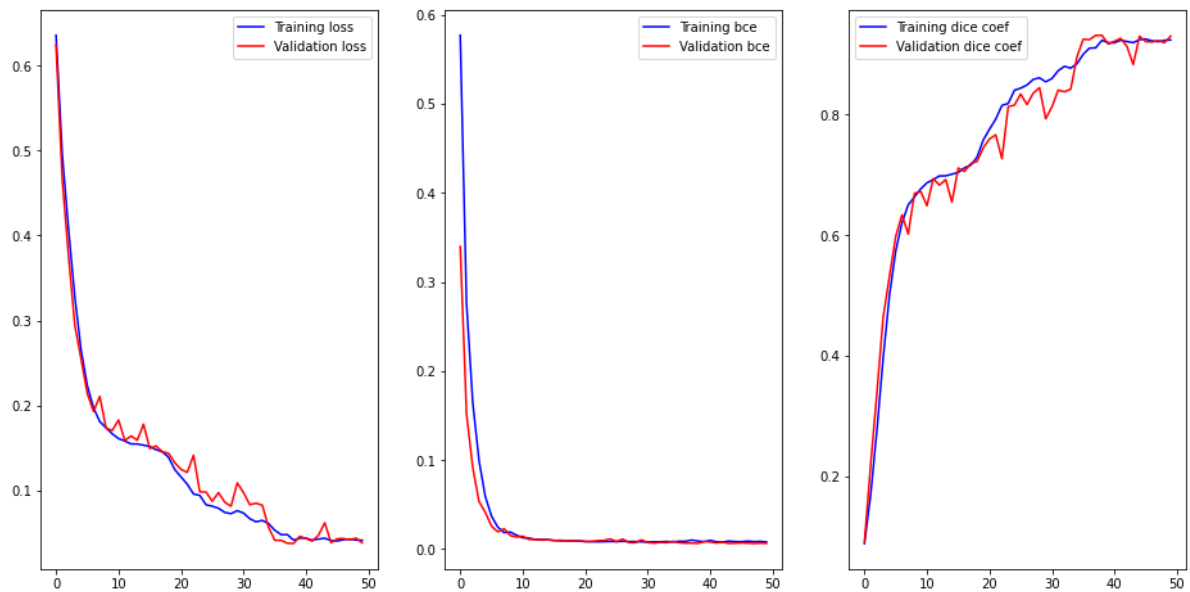


Figure 5.2 Visualisation of the training process of unet++ model placenta data (Average values from 3-fold cross validation for the best model)

As we see in the figure 5.2 after the epoch 10 the bce loss doesn't continue diminishing, however the loss function continues decreasing as the dice coefficient increases until after the 40th epoch when the dice coefficient and loss plateau and the model doesn't learn more.

5.1.3 BraTS Unet learning ratios

Table 5.3 bce, dice and loss obtained with a learning ratio of 1E-4 and 1E-3 for BraTS and Unet model. It includes the dice score, its standard deviation, IL and UL are the lower and the upper limits of the confidence interval with a 95% of confidence.

LR	bce	loss	dice	SD	LI	LU
1.00E-04	0.131	0.230	0.672	0.029	0.599	0.745
scheduled	0.105	0.196	0.714	0.046	0.599	0.828
1.00E-03	0.048	0.171	0.706	0.043	0.600	0.812

The scheduled learning ratio was scheduled to begin with 1E-4 and then decrease the learning ratio to 1E-5 in the epoch 12 and 1E-6 in the epoch 24.

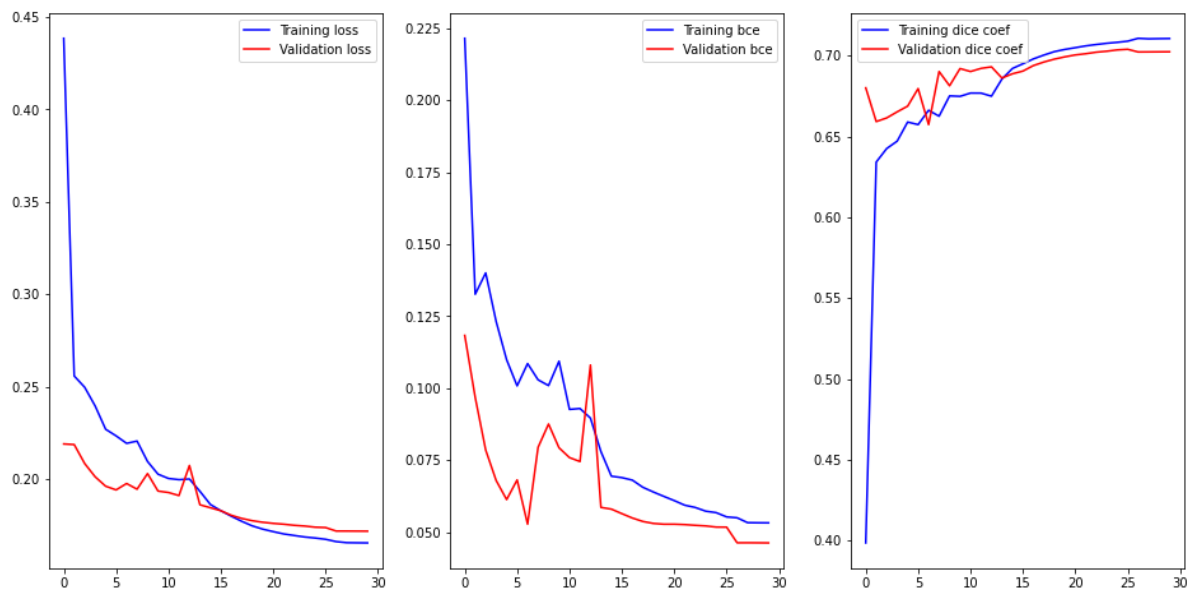


Figure 5.3 Visualisation of the training process of u-net model BraTS data (Average values from 3-fold cross validation for the best model)

For the u-net configuration applied to BraTS data the use of a scheduled learning ratio resulted in the model with the best performance. Despite the fact that loss seems plateau after 25 it may have been possible to train it for more epochs.

5.1.4 BraTS Unet++ learning ratios

Table 5.4 bce, dice and loss obtained with a learning ratio of 1E-4 and 1E-3 for BraTS and Unet++ model. It includes the dice score, its standard deviation, IL and UL are the lower and the upper limits of the confidence interval with a 95% of confidence. Scheduled learning ratio is from 1E-4 and from the epoch 20 it decreases to 1E-5

LR	bce	loss	dice	SD	LI	LU
----	-----	------	------	----	----	----

1.00E-03	0.049	0.172	0.706	0.041	0.603	0.808
1.00E-04	0.039	0.170	0.301	0.054	0.167	0.435
scheduled	0.033	0.253	0.528	0.093	0.298	0.758
1.00E-05	0.065	0.528	0.009	0.041	-0.094	0.111

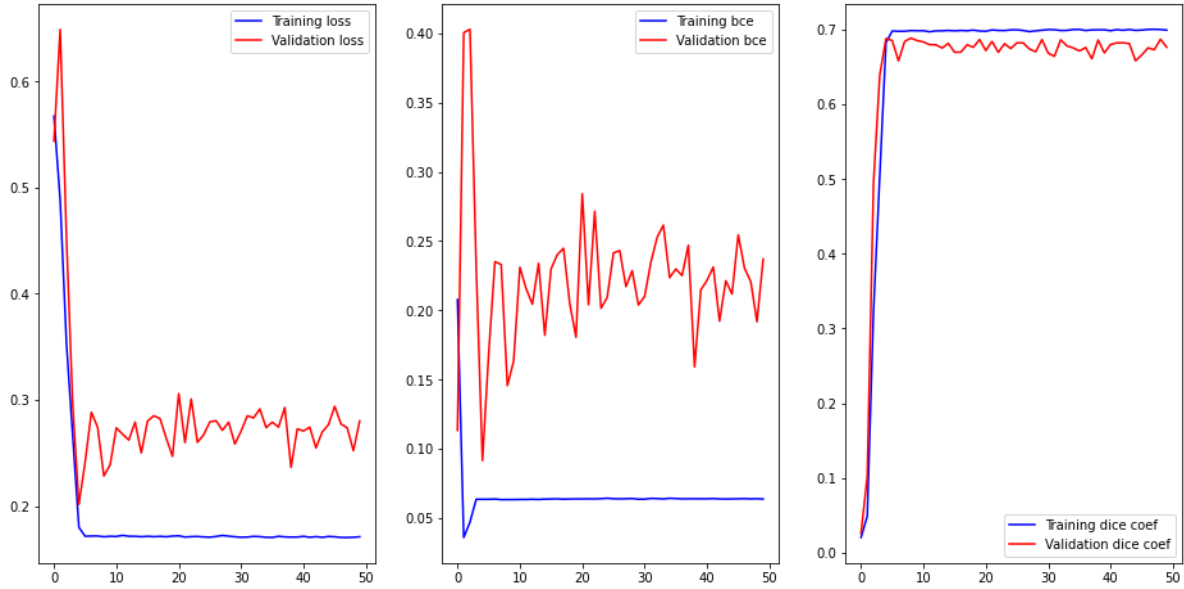


Figure 5.4 Visualisation of the training process of unet++ model BraTS data (Average values from 3-fold cross validation for the best model)

For the BraTS u-net data set it seems to be learning pretty fast as I am using a learning ratio of 1E-3 the first seven epochs (Fig. 5.4) and after that point the validation loss doesn't decrease anymore, because of that, I tried applying a lowest learning ratio – to make the learning process slower and increase the dice score, but the performance was not better decreasing the learning ratio to 1E-4 (Table 5.4).

5.2 Model pruning.

A nice characteristic of the u-net++ model proposed in the original paper is model pruning, it means that the model can work with fewer parameters obtaining a prediction from an intermediate layer. It impacts negatively the performance of the model but decreases the inference time. My implementation has three outputs, the first output is obtained from the second layer (using the fewest number of parameters), the second output from the third layer and the third output is obtained from the fourth (last) layer and uses all the parameters. One important note about the model pruning is that to obtain this characteristic on the model this must be trained with deep supervision. I applied deep supervision in the same way as explained in the original paper of Unet++ (Zhou et al. 2018) it means that during the training process I obtain predictions from each of my three outputs, I calculate the respective loss for each prediction, and then I average these three loss values. This average value corresponds to the training loss. Although, the validation and test loss come from the last output layer (using all the parameters).

5.2.1 Model pruning placenta.

The evaluation of the pruned model in the table 5.5 indicates that even though the reduction on the inference time to 28.936 seconds and 12.424 seconds from 51.523 seconds with pruning 1 and pruning 0 respectively are huge, the average dice score has not dropped that much. The reduction on the number of parameters is also huge from 6,571,039 to 1,478,750 and 259,869 for pruning 2, pruning 1 and pruning 0 respectively, that explains the reduction on the inference time.

Table 5.5 Number of parameters, average dice and average time of inference with a standard CPU (placenta data).

pruning	number of parameters	Average - dice	Average – time (seconds)
0 (Pred_1)	259,869	0.891	12.424
1 (Pred_2)	1,478,750	0.916	28.936
2 (Pred_3)	6,571,039	0.915	51.523

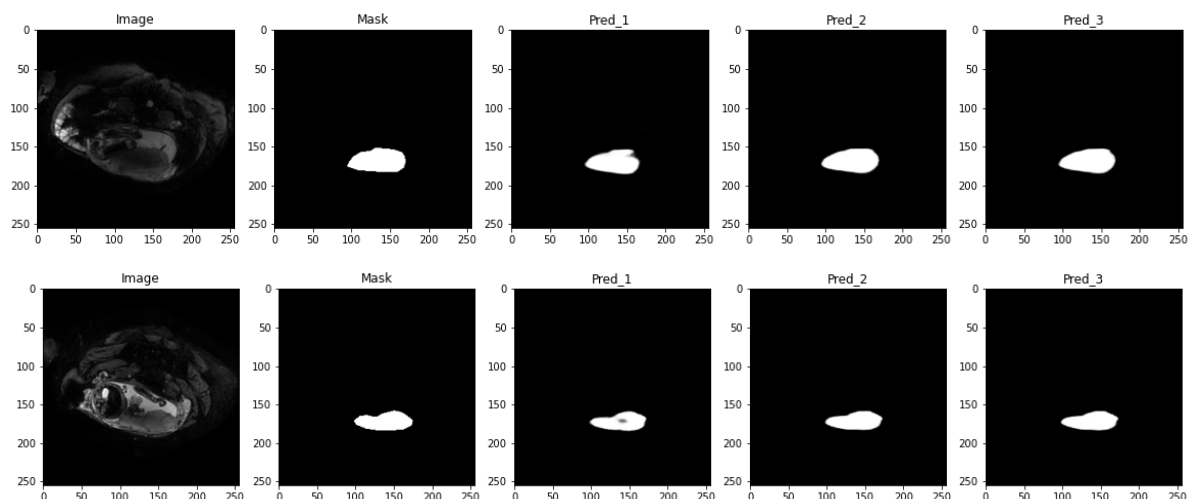


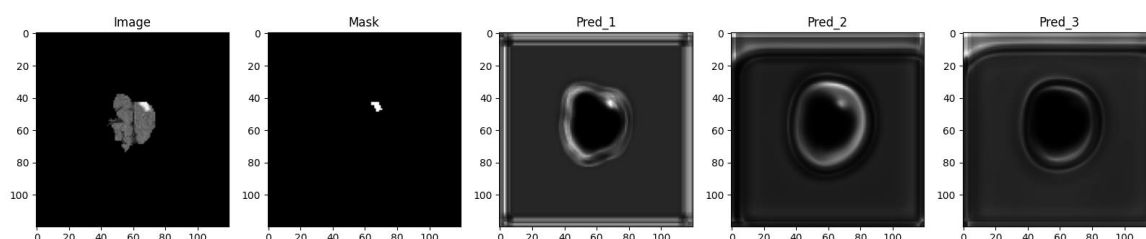
Figure 5.5 Qualitative evaluation of the placenta Unet++ predictions obtained with the last segmentation output (pred_3), second segmentation output (pred_2) and the first segmentation output(pred_1).

5.2.2 Model pruning BraTS.

The quantitative results of pruning the BraTS Unet++ coincident with the results for placenta. As the number of parameters increases the dice score and the inference time also do.

Table 5.6 Number of parameters, average dice and Average time of inference with a standard GPU (BraTS data).

pruning	number of parameters	Average – dice	Average – time (seconds)
0 (Pred_1)	259,983	0.699	65.962
1 (Pred_2)	1,478,994	0.703	186.209
2 (Pred_3)	6,571,413	0.706	364.806



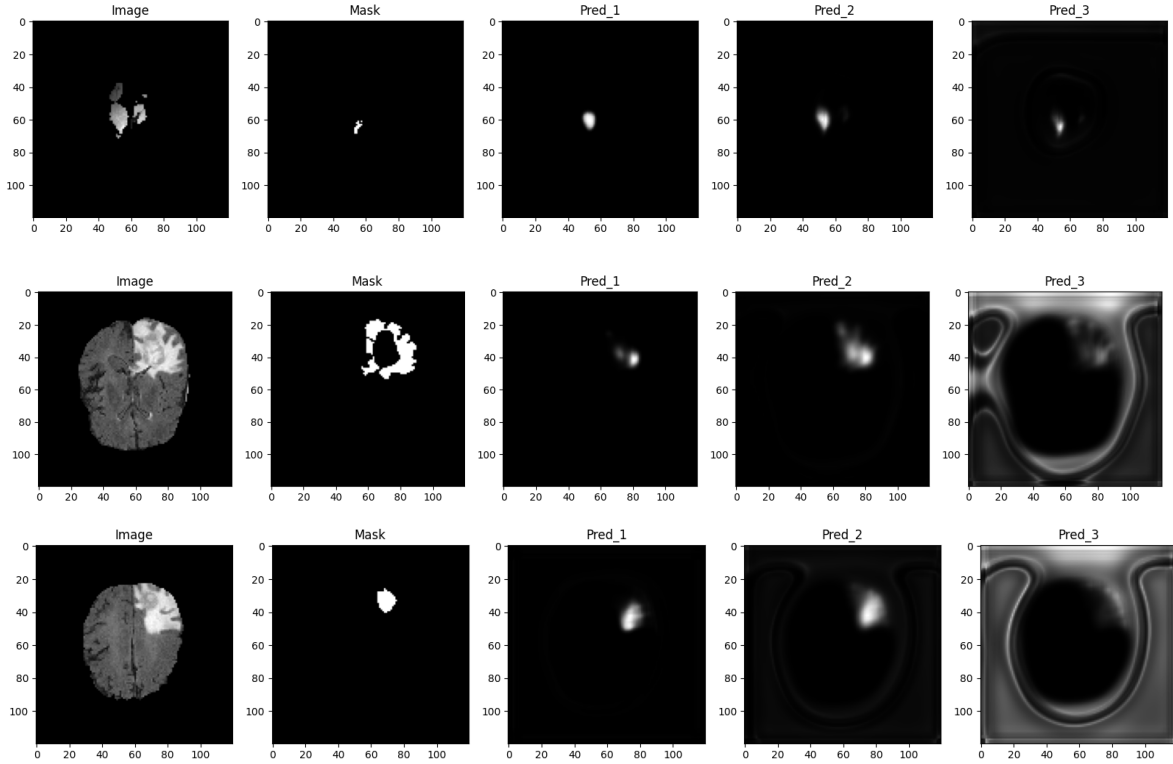


Figure 5.6 Qualitative evaluation of the BraTS Unet++ predictions obtained with the last segmentation output (pred_3), second segmentation output (pred_2) and the first segmentation output(pred_1).

Chapter 6: Segmentation results

6.1 Unet vs Unet ++

As we can see in the table 6.1 for the placenta data, Unet++ performs better than Unet, even having fewer number of parameters. Unet++ reached a Dice score of 91.5% compared with a Dice of 90% of Unet. Different if the case with BraTS data where the Dice score is quite similar for Unet and Unet++.

Table 6.1 Comparison of the results obtained with the different models and datasets, it includes the dice score, its standard deviation, IL and UL are the lower and the upper limits of the confidence interval with 95% of confidence.

dataset	model	parameters	dice	sd	IL	UL
placenta	Unet	7,781,891	0.900	0.021	0.847	0.954
placenta	Unet++	6,571,039	0.915	0.007	0.899	0.932
BraTS	Unet	7,781,891	0.714	0.046	0.599	0.828
BraTS	Unet++	6,571,413	0.706	0.041	0.603	0.808

dataset	model	training time	sd	validation time	sd	testing time
placenta	Unet	16.284	0.089	2.808	0.058	1.718
placenta	Unet++	13.814	0.214	2.228	0.110	0.890

BraTS	Unet	274.230	30.752	88.068	9.685	39.004
BraTS	Unet++	276.419	22.494	85.931	7.187	43.325

6.2 Comparison of results with previous studies

For placenta-UoN data I include results from the previous year in the university and other results collected from some paper published. For BraTS 2020 data set I compare these results with others published with the same BraTS data set.

6.2.1 Previous studies placenta

In table 6.2 (Alansary et al. 2016), (Looney et al. 2017) and (Twickler et al. 2020) present results of 0.72, 0.73 and 0.72 respectively, and those are the lowest for the placenta dataset. The result of the Unet 2020 dice score is 0.87, that result was obtained in the university of nottingham in the summer of 2020, even though this result is a lot higher than the first three in the table are lower than the results obtained in this work (unet 2021) working with a different u-net implementation. The figures for the model u-net++ are the highest achieved for the placenta data set. Although, the data obtained with unet++ overcome previous results documented, these are not comparable, as come from different dataset.

6.2.2 Previous studies BraTS

(Fidon, Ourselin, and Vercauteren 2021) obtained a dice score of 84.8%, and they were in the fourth position in the BraTS 2020 competition, they did so apply a Unet 3D model, a customised loss function and a no standard optimizer, Ranger. The solution of (Henry et al. 2020) was among the top ten participants in the competition with a dice of 84%, and as we see in the table 6.2 the difference of performance with the other implementation (Fidon, Ourselin, and Vercauteren 2021) is quite narrow. The dice coefficient obtained in this implementation (0.71) is quite low despite to the fact that the data was highly reduced.

Table 6.2 Placenta results obtained in this work with Unet and Unet++ (Unet 2021, Unet++ 2021), results from UoN (Unet 2020) and three publically available placenta segmentation results. BraTS results obtained in this work (Unet 2021) and results from other sources, (Henry et al. 2020), (Fidon, Ourselin, and Vercauteren 2021), (*) the placenta results from (Alansary et al. 2016), (Looney et al. 2017) and (Twickler et al. 2020) are from different datasets to the datasets used in this work. (**) the BraTS results of (Henry et al. 2020) and (Fidon, Ourselin, and Vercauteren 2021) although they are from the same version of BraTS data set, these results have been obtained with the test data set which ground truth are not publicly available.

Source	DATA	DICE
(Alansary et al. 2016)	*Placenta MRI	0.72
(Looney et al. 2017)	*Placenta ultrasound	0.73
(Twickler et al. 2020)	*Placenta MRI	0.82
Unet 2020	Placenta MRI	0.87
Unet 2021	Placenta MRI	0.90
Unet++ 2021	Placenta MRI	0.92
(Henry et al. 2020)	**BraTS	0.840
(Fidon, Ourselin, and Vercauteren 2021)	**BraTS	0.848

Unet 2021	BraTS	0.71
-----------	-------	------

Chapter 7: Conclusion

The u-net and u-net++ models have been implemented with Python and Pytorch framework. Both models have tested their effectivity to work with different datasets. Implementing u-net and u-net++ with placenta data set the performance obtained has been higher than the performance obtained with u-net on the year 2020. However, when I implemented u-net for BraTS I could not make u-net++ to perform better than u-net.

The outcomes obtained with placenta were promising compared with others publicly available. However not much work has been done in placenta segmentation and there still are a lot of room for future improvement.

Both data sets are composed by 3D images, and to use these models that for 2D image with these data, it was necessary to use the data by 2D slices. Then a 3D problem becomes a 2D one. Not only 3D but also multimode is BraTS data set. So besides of using the same strategy as for placenta to feed the model with 3D data- working with each 2D slice to transform the problem in a two dimensional one, additionally to that, each mode was tried to pass through a different channel, however because BraTS 2020 is quite heavy It was necessary the compression of the data and working with only one channel.

The u-net++ implementation was trained with deep supervision; it means that the loss in distinct levels of the network is averaged. Another consequence of deep supervision is model pruning, and one of the conclusions of the experiments with u-net++ pruning is that as the depth of the network and the number of parameters increase, the time of inference and the performance also increase. The potential of time reduction with deep supervision is for example for tracking.

7.1 Further study

The Dice BCE loss was calculated using a ponderation of 50% Dice and 50% BCE loss, prove different weights of dice BCE loss would be an interesting test to do.

As in the original papers I used Sigmoid activation function, for the output layer, it would be interesting to use a Softmax for the multiclass Unet and Unet++.

Even though - For the placenta dataset - results obtained are like state of art, the level of accuracy of the predictions is not good enough to fulfil the requirements of accuracy for its use, as 100% dice score is needed for quantification of pathology. Having said that, a good approach would be using a semi-automated deep learning approach.

7.2 Areas of improvement

As I proved experimentally (Table 5.5 and table 5.6) while the depth of the structure u-net++ is incremented, it increases the performance of the net. Also, previous studies have shown that Unet (Ronneberger, Fischer, and Brox 2015) and Unet++ (Zhou et al. 2018) perform better Increasing the depth of the network. On the other hand, it is important to notice that the original papers of Unet and Unet++ implemented one level more of depth of the networks (what they call Deep Unet) as well.

The pre-processing of the data was quite simple in this implementation, I just applied standardization of the data and one hot encoding, It would be interesting to apply data normalization, some regularization technique such as drop-out, some data augmentation technique to know how the models behave.

7.3 Limitations

Placenta data comes from just one clinical institution and as in different institutions the data could be different, this fact can produce overfitting. This is not the case of BraTS which comes from 19 different institutions and so it is more representative.

A huge limitation for this project was the computing capacity for the BraTS dataset, a further study could include working with two modes or with all of them. It would be interesting how the model would perform with the full data set uncompressed and with all the examples.

The results obtained for the BraTS data set are quite limited, I explain that because of the huge reduction of the data that was necessary to run the experiments in brief time.

References

- ‘2D vs 3D MRA’. n.d. Questions and Answers in MRI. Accessed 19 September 2021. <http://mriquestions.com/2d-vs-3d-mra.html>.
- Alansary, Amir, Konstantinos Kamnitsas, Alice Davidson, Rostislav Khlebnikov, Martin Rajchl, Christina Malamateniou, Mary Rutherford, et al. 2016. ‘Fast Fully Automatic Segmentation of the Human Placenta from Motion Corrupted MRI’. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, edited by Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells, 9901:589–97. Lecture Notes in Computer Science. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-46723-8_68.
- Bakas, Spyridon, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin Kirby, John Freymann, Keyvan Farahani, and Christos Davatzikos. 2017a. ‘Segmentation Labels for the Pre-Operative Scans of the TCGA-GBM Collection’. The Cancer Imaging Archive. <https://doi.org/10.7937/K9/TCIA.2017.KLXWJJ1Q>.
- . 2017b. ‘Segmentation Labels for the Pre-Operative Scans of the TCGA-LGG Collection’. The Cancer Imaging Archive. <https://doi.org/10.7937/K9/TCIA.2017.GJQ7R0EF>.
- Bakas, Spyridon, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S. Kirby, John B. Freymann, Keyvan Farahani, and Christos Davatzikos. 2017c. ‘Advancing The Cancer Genome Atlas Glioma MRI Collections with Expert Segmentation Labels and Radiomic Features’. *Scientific Data* 4 (1): 170117. <https://doi.org/10.1038/sdata.2017.117>.
- Berrar, Daniel. 2019. ‘Cross-Validation’. In *Encyclopedia of Bioinformatics and Computational Biology*, 542–45. Elsevier. <https://doi.org/10.1016/B978-0-12-809633-8.20349-X>.
- Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*. Information Science and Statistics. New York: Springer.
- Delibasoglu, Ibrahim, and Mufit Cetin. 2020. ‘Building Segmentation with Inception-Unet and Classical Methods’. In *2020 28th Signal Processing and Communications Applications Conference (SIU)*, 1–4. Gaziantep, Turkey: IEEE. <https://doi.org/10.1109/SIU49456.2020.9302155>.
- Fidon, Lucas, Sébastien Ourselin, and Tom Vercauteren. 2021. ‘Generalized Wasserstein Dice Score, Distributionally Robust Deep Learning, and Ranger for Brain Tumor Segmentation: BraTS 2020 Challenge’. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, edited by Alessandro Crimi and Spyridon Bakas, 12659:200–214. Lecture Notes in Computer Science. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-72087-2_18.
- Han, Mo, Yuwei Bao, Ziyang Sun, Shiping Wen, Liming Xia, Jingyang Zhao, Junfeng Du, and Zheng Yan. 2019. ‘Automatic Segmentation of Human Placenta Images With U-Net’. *IEEE Access* 7: 180083–92. <https://doi.org/10.1109/ACCESS.2019.2958133>.
- Henry, Theophraste, Alexandre Carre, Marvin Lerousseau, Theo Estienne, Charlotte Robert, Nikos Paragios, and Eric Deutsch. 2020. ‘Brain Tumor Segmentation with Self-Ensembled, Deeply-Supervised 3D U-Net Neural Networks: A BraTS 2020 Challenge Solution’. *ArXiv:2011.01045 [Cs, Eess]*, November. <http://arxiv.org/abs/2011.01045>.
- Horst, Geertje van der, Jeroen T. Buijs, and Gabri van der Pluijm. 2015. ‘Pre-Clinical Molecular Imaging of “the Seed and the Soil” in Bone Metastasis’. In *Bone Cancer*, 557–70. Elsevier. <https://doi.org/10.1016/B978-0-12-416721-6.00046-7>.
- Huang, Gao, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2018. ‘Densely Connected Convolutional Networks’. *ArXiv:1608.06993 [Cs]*, January. <http://arxiv.org/abs/1608.06993>.

- IEEE Staff. 2019. *2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC)*. Piscataway: IEEE.
<http://ezproxy.canterbury.ac.nz/login?url=https://ieeexplore.ieee.org/servlet/opac?punumber=8966539>.
- Krešo, Ivan, Josip Krapac, and Siniša Šegvić. 2019. 'Efficient Ladder-Style DenseNets for Semantic Segmentation of Large Images'. *ArXiv:1905.05661 [Cs]*, May.
<http://arxiv.org/abs/1905.05661>.
- Lee, Chen-Yu, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. 2014. 'Deeply-Supervised Nets'. *ArXiv:1409.5185 [Cs, Stat]*, September. <http://arxiv.org/abs/1409.5185>.
- Long, Jonathan, Evan Shelhamer, and Trevor Darrell. 2015. 'Fully Convolutional Networks for Semantic Segmentation'. *ArXiv:1411.4038 [Cs]*, March. <http://arxiv.org/abs/1411.4038>.
- Looney, Padraig, Gordon N. Stevenson, Kypros H. Nicolaides, Walter Plasencia, Malid Molloholli, Stavros Natsis, and Sally L. Collins. 2017. 'Automatic 3D Ultrasound Segmentation of the First Trimester Placenta Using Deep Learning'. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, 279–82. Melbourne, Australia: IEEE.
<https://doi.org/10.1109/ISBI.2017.7950519>.
- Louis, A K. 1992. 'Medical Imaging: State of the Art and Future Development'. *Inverse Problems* 8 (5): 709–38. <https://doi.org/10.1088/0266-5611/8/5/003>.
- 'Magnetic Resonance Imaging (MRI)'. n.d. Accessed 19 September 2021.
<https://www.nibib.nih.gov/science-education/science-topics/magnetic-resonance-imaging-mri>.
- 'Medical Imaging Modalities Used in Medicine | Open Medscience'. n.d. Accessed 17 September 2021. <https://openmedscience.com/medical-imaging-modalities/>.
- Menze, Bjoern H., Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, et al. 2015. 'The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)'. *IEEE Transactions on Medical Imaging* 34 (10): 1993–2024.
<https://doi.org/10.1109/TMI.2014.2377694>.
- 'MRI Basics'. n.d. Accessed 17 September 2021.
<https://case.edu/med/neurology/NR/MRI%20Basics.htm>.
- 'Multimodal Brain Tumor Segmentation Challenge 2020: Data | CBICA | Perelman School of Medicine at the University of Pennsylvania'. n.d. Accessed 17 September 2021.
<https://www.med.upenn.edu/cbica/brats2020/data.html>.
- Noh, Hyeonwoo, Seunghoon Hong, and Bohyung Han. 2015. 'Learning Deconvolution Network for Semantic Segmentation'. *ArXiv:1505.04366 [Cs]*, May. <http://arxiv.org/abs/1505.04366>.
- Nottingham, University of. n.d. 'MRI Pregnancy Study Gives New Insights into the All-Important Placenta'. Accessed 10 September 2021. <https://medicalxpress.com/news/2020-05-mri-pregnancy-insights-all-important-placenta.html>.
- Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. 2015. 'U-Net: Convolutional Networks for Biomedical Image Segmentation'. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, edited by Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, 9351:234–41. Lecture Notes in Computer Science. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28.
- Song, Guanghui, Yan Nie, Jiajian Zhang, and Genlang Chen. 2020. 'Research on the Fusion Method of 2D and 3D UNet in Pulmonary Nodules Segmentation Task'. In *2020 International Conference on Computer Science and Management Technology (ICCSMT)*, 44–47. Shanghai, China: IEEE. <https://doi.org/10.1109/ICCSMT51754.2020.00016>.

- 'Sørensen–Dice Coefficient'. 2021. In *Wikipedia*.
https://en.wikipedia.org/w/index.php?title=S%C3%B8rensen%E2%80%93Dice_coefficient&oldid=1038621606.
- Stallard, Jim. n.d. 'CT vs MRI: What's the Difference? And How Do Doctors Choose Which Imaging Method to Use? | Memorial Sloan Kettering Cancer Center'. Accessed 19 September 2021. <https://www.mskcc.org/news/ct-vs-mri-what-s-difference-and-how-do-doctors-choose-which-imaging-method-use>.
- Stevenson, Gordon N., Sally L. Collins, Jane Ding, Lawrence Impey, and J. Alison Noble. 2015. '3-D Ultrasound Segmentation of the Placenta Using the Random Walker Algorithm: Reliability and Agreement'. *Ultrasound in Medicine & Biology* 41 (12): 3182–93.
<https://doi.org/10.1016/j.ultrasmedbio.2015.07.021>.
- Szeliski, Richard, and Richard Szeliski. 2011. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. New York London: Springer.
- Twickler, Diane M., Quyen N. Do, Yin Xi, Maysam Shahedi, James Dormer, Anusha Devi TT, Matthew A. Lewis, et al. 2020. '228: Automated Segmentation of the Human Placenta and Uterus with MR Imaging Using Artificial Intelligence (AI)'. *American Journal of Obstetrics and Gynecology* 222 (1): S158–59. <https://doi.org/10.1016/j.ajog.2019.11.244>.
- Wang, Guotai, Maria A. Zuluaga, Rosalind Pratt, Michael Aertsen, Anna L. David, Jan Deprest, Tom Vercauteren, and Sebastien Ourselin. 2015. 'Slic-Seg: Slice-by-Slice Segmentation Propagation of the Placenta in Fetal MRI Using One-Plane Scribbles and Online Learning'. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, edited by Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, 9351:29–37. Lecture Notes in Computer Science. Cham: Springer International Publishing.
https://doi.org/10.1007/978-3-319-24574-4_4.
- Wang, Liwei, Chen-Yu Lee, Zhuowen Tu, and Svetlana Lazebnik. 2015. 'Training Deeper Convolutional Networks with Deep Supervision'. *ArXiv:1505.02496 [Cs]*, May.
<http://arxiv.org/abs/1505.02496>.
- Wang, Wenhai, Xiang Li, Jian Yang, and Tong Lu. 2018. 'Mixed Link Networks'. *ArXiv:1802.01808 [Cs]*, February. <http://arxiv.org/abs/1802.01808>.
- 'What Is the Difference between MRI Data Collected in 2D versus 3D?' n.d. Quora. Accessed 19 September 2021. <https://www.quora.com/What-is-the-difference-between-MRI-data-collected-in-2D-versus-3D>.
- Wu, Zhiwei, Jinjin Hai, Lijie Zhang, Jian Chen, Genyang Cheng, and Bin Yan. 2019. 'Cascaded Fully Convolutional DenseNet for Automatic Kidney Segmentation in Ultrasound Images'. In *2019 2nd International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 384–88. Chengdu, China: IEEE. <https://doi.org/10.1109/ICAIBD.2019.8836994>.
- Yuan, Yixuan, Wenjian Qin, Xiaoqing Guo, Mark Buyyounouski, Steve Hancock, Bin Han, and Lei Xing. 2019. 'Prostate Segmentation with Encoder-Decoder Densely Connected Convolutional Network (Ed-Densenet)'. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, 434–37. Venice, Italy: IEEE. <https://doi.org/10.1109/ISBI.2019.8759498>.
- Zhang, Aston, Zachary C. Lipton, Mu Li, and Alexander J. Smola. 2021. 'Dive into Deep Learning'. *ArXiv:2106.11342 [Cs]*, July. <http://arxiv.org/abs/2106.11342>.
- Zhou, Zongwei, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. 2018. 'UNet++: A Nested U-Net Architecture for Medical Image Segmentation'. *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, S...* 11045 (September): 3–11. https://doi.org/10.1007/978-3-030-00889-5_1.

Zhuang, Juntang. 2019. 'LadderNet: Multi-Path Networks Based on U-Net for Medical Image Segmentation'. *ArXiv:1810.07810 [Cs, Eess]*, August. <http://arxiv.org/abs/1810.07810>.

Zyuzin, Vasily, and Tatiana Chumarnaya. 2019. 'Comparison of Unet Architectures for Segmentation of the Left Ventricle Endocardial Border on Two-Dimensional Ultrasound Images'. In *2019 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBREIT)*, 110–13. Yekaterinburg, Russia: IEEE. <https://doi.org/10.1109/USBREIT.2019.8736616>.