APSS Challenge – Distorted Brain Tissue Segmentation

Team 9

Reference people:

- Paolo Avesani
  NeuroInformatics Laboratory (NILab), Bruno Kessler Foundation, Trento, Italy principle investigator
  avesani@fbk.eu
- Umberto Rozzanigo
  Medical director
  umberto.rozzanigo@apss.tn.it

Mentors:

- Gabriele Amorosino
  PhD student at NeuroInformatics Laboratory (NILab), Bruno Kessler Foundation, Trento, Italy
  gabriele.amorosino@unitn.it

Solvers:

- Giorgio Letti
  Quantitative and computational biology master's course student at the University of Trento
  giorgio.letti@studenti.unitn.it
- Aleksa Krsmanović
  Quantitative and computational biology master's course student at the University of Trento
  aleksa.krsmanovic@studenti.unint.it
- Andrea Ferigo
  PhD student at the University of Trento
  andrea.ferigo@unitn.it
- Leonardo Lucio Custode
  PhD student at the University of Trento
  leonardo.custode@unitn.it

Contents

1 Introduction

In the medical field, rapidly analyzing the results of medical analysis and producing a diagnosis is a key factor, both for the medical team and the patient. On one hand, the medical team, receiving the results earlier, can detect malformations and diseases earlier, starting the medical treatment in time, and reducing the risk for the patient. On the other hand, the patient can also receive faster results, reducing the *waiting stress* Bolvin and Lancastle [2010], Portnoy [2010] for him and for his relatives.

However, some medical tests require manual work of an expert to produce the final result. For example, in the case of magnetic resonance imaging (MRI) of the brain, a doctor has to manually segment the image, dividing the different tissue based on his experience. This results in an tedious procedure that requires a lot of time for each MRI and is prone to error.

In Amorosino et al. [2022], the authors explored the capability of a Convolution Neural Network (CNN) to segment the 3D volume of the brain produced by the MRI. The results showed that the CNN proposed, a U-Net Çiçek et al. [2016], Ronneberger et al. [2015], can achieve state-of-the-art performance in the segmentation of distorted brain images.

In this work, we start from the baseline indicated in Amorosino et al. [2022], addressing one of the main criticality: the small size of the dataset. To improve the reliability of the baseline model, we proceed with two different approaches, the first, data augmentation, aims to improve the training data by generating fake, but realistic, pathological brain images for the training. The second approach instead is model-driven. Apart from utilizing the current state-of-the-art architecture previously unused on malformed brain segmentation, we also propose using a 2.5D approach, dividing the input volume into slices, then recomposing the 3D image after the inference.

The rest of the work is divided as follows. In Section 2, we present the challenge, describing the goals and the dataset. In Section 3, we describe the methods for the data augmentation, the 2.5D approach and present the network used. Then, in Section 4, we present the results achieved. Finally, in Section 5, we summarize how work and draw the conclusion.

2 The challenge

In this section, we will describe the challenge. Firstly, we will describe the context and the directions on which we decided to work. Then we will analyze the dataset. In particular, we will explain how it is composed and how the images are obtained. Finally, we will summarize our working procedure.

2.1 Description and goals

The goal of the challenge is to implement an AI model capable of producing more reliable results than state-of-the-art methods on the task of recognizing different tissues in the brain even in individuals harbouring great anatomical distortions (e.g. in cases of genetic diseases). This task of tissue recognition is defined as segmentation, i.e. the parcellation of the image into non-overlapping brain regions. The challenge requires to focus on the recognition of six labels, one for each brain tissue of interest. The parts of the image encoding the same type of tissue are assigned to the corresponding label. The output (the segmented image) has the same dimension as the original image, where the value of a pixel/voxel is the label code, instead of the signal intensity.

The reason for developing an AI model for this purpose is the fact that the traditional model-based methods used for tissue segmentation strongly rely on the assumption that the anatomical structures are located in similar anatomical regions as in healthy subjects. If this assumption is not met they often yield unsatisfactory results Amorosino et al. [2020]. In case of strong alterations of the brain anatomy, the correct labeling of the brain is generally obtained through a labour-intensive manual process.

Amorosino et al. [2022] have compared the performance of model-driven approaches with a 3D convolutional neural network model they have developed. This neural network was trained solely on volumetric

brain images of healthy individuals and by evaluating it on distorted brains and was able to significantly outperform the model-driven tools.

The primary goal of our project is to continue improving the work of Amorosino et al. [2022]. We observed two possible ways of doing this, The first is to try different model of NNs, the second is to improve the dataset. In particular, regarding the way to improve the dataset, we decide to try two approaches, one data-driven which aims to generating new plausible images from the dataset and one model-driven, that aims to use the 2.5D approach to extract more information from the dataset. Hence, we set three main sub-goals:

1. Enriching the dataset by generating synthetically distorted brains due to the fact that distorted brain scans are scarce across all available datasets. By having more deformed brains in the training set, the deep learning model's ability to label such samples will also improve. An additional requirement for this task is that the induced synthetic deformations have to be medically sensible (in line with the true malformation) and not random.

2. Identifying an appropriate neural network architecture and training a model capable of outperfoming the 3D U-Net model used by Amorosino et al. [2022] on the same tissue segmentation task. Apart from assessing the current state of the art biomedical imaging toolkit, nnUNet Isensee et al. [2018], previously unused for malformed brain segmentation, we also assess a 2.5D approach. In this case the dataset is enlarged because each image in the dataset is sliced through an axis, increasing the total number of images. Moreover, in this way we use 2D CNN reducing the total amount of parameters to optimize.

2.2 Data generation process

The data used in this challenge are anatomical T1-weighted (T1-w) brain magnetic resonance imaging (MRI) sequences. MRI is one of the most commonly used diagnostic techniques in neurology and neurosurgery. The main advantage of MRI is the fact that it is a non-invasive, non-ionizing diagnostic technique able of providing in great detail images of the internal structures of the body (i.e. the brain), which can be observed in all three planes: axial, sagittal and coronal. Hence it can be useful for research purpose i.e. in the neuroscience field.

An external magnetic field is employed to align the protons that are normally randomly oriented within the the tissues being examined. This alignment is next disrupted by introduction of an external radio frequency (RF) energy. The atomic nuclei return to their resting alignment through relaxation processes and in so doing emit RF energy – these signals are being measured and transformed into an image. Using the magnetic field gradient and by varying the sequence of RF pulses applied, different types of images are created. Repetition Time (TR) is the amount of time between successive pulse sequences applied to the same slice. Time to Echo (TE) is the time between the delivery of the RF pulse and the receipt of the echo signal.

The most common MRI sequences are T1-weighted (T1w) and T2-weighted (T2w) scans. In this project, only T1w brain scans are used. T1-weighted images are produced by using short TE and TR times. The contrast and brightness of the image are predominately determined by T1 properties of tissue – the time constant which determines the rate at which excited protons return to equilibrium (specific for each tissue) Itti et al. [2001].

2.3 Dataset description

The Distorted Brain Benchmark (DBB) dataset developed by Amorosino et al. [2022] is composed of more than 921 images of healthy subjects and 37 pathological subjects. However, in this work, we use a subset of DBB data belonging to the EMEDEA-PED [1] dataset. In total, there are 244 T1w MRI brain scans of pediatric individual, 207 of which belong to healthy individuals and 37 to patients with brain distortions. The properties of the dataset are shown in Table 1.

[1] https://emedea.it/medea/en/

The distorted 37 brain scans are aggregated into 4 different classes of pathologies: agenesis of corpus callosum (ACC, $n = 10$), posterior fossa malformations (PFM, $n = 10$), malformations of cortical development (MCDs, $n = 10$) and complex malformation lesions (HD, $n = 6$).

The data was retrieved from Brainlife.io platform[2] in the standard NifTI (Neuroimaging Informatics Technology Initiative) image format with the accompanying ground truth images and metadata. The ground truth images are 3D multi-label volumetric images composed out of integer values where each integer corresponds to the one of the six tissues of interest (plus the background encoded as 0). These ground truth images have the same dimensions as the T1w image they are referencing. The labeling of healthy and distorted brain images was performed as described in Amorosino et al. [2022]. The label nomenclature follows:

1. Background , voxel value 0
2. Cerebro-Spinal Fluid (CSF), voxel value 1
3. Gray Matter (GM), voxel value 2
4. White Matter (WM), voxel value 3
5. Deep Gray Matter (DGM), voxel value 4
6. Brain Stem (BS), voxel value 5
7. Cerebellum (CER), voxel value 6

Table 1: Basic dataset statisctics

	Gender		Age	
Type	M	F	Age range	Age mean
Healthy	108	99	2.33 – 21.76	13.03
Distorted	19	18	0.7 – 17.8	6.34

2.4 Description of challenge activities

In this subsection, a quick recap of the challenge activities (during the 10 weeks) follows.

In the first week, we had a meeting with APSS and FBK to understand the problem they were trying to solve and to discuss potential directions. First of all, we understood the scope of the work, what are the metrics of interest, and what are the main performance bottlenecks. Then, we understood that the main goal for our challenge was to improve the dice score w.r.t. the baseline model proposed in Amorosino et al. [2022]. In the brainstorming phase, many proposals could have led to an improvement in performance. Among all the proposals, two of them arose as the most promising: an improvement of the dataset, and an improvement of the model, as already explained in the introduction. Since both these ideas seemed promising, we decided to work on both ideas simultaneously, to maximize the probability of improving over the baseline results provided in Amorosino et al. [2022].

In the second week, we started exploring the data, and understanding the type of images that were being collected and their format. Moreover, we had to resize the scans to a common resolution. By speaking with the stakeholders, we understood that resizing all the scans to 256-voxels isotropic volumes was the optimal choice as it is the same resolution used in the reference paper. Thus, we started to study the training set and implement the code for both ideas.

[2] https://doi.org/10.25663/brainlife.pub.24

In the third week, we started preliminary experiments. We were able to produce some simple deformations of the brain using the Deformetrica package. Moreover, we saw that the 2D models were not easy to train because of the memory limitations in Colab. Thus, when interacting with the stakeholders, we understood two main things:

1. The deformations may not be based on random perturbations but they must be grounded in the medical domain

2. Different planes may lead to different insights for predicting the tissues.

During the fourth week, we focused on technical optimizations to be able to perform the training of the 2D models. We were able to run some epochs of training with 2D models using 256-pixels slices (of the whole volume) for one plane (sagittal plane). However, given the running time limits in colab, we could run just a few epochs, so the performance was not satisfactory at all. For what concern data augmentation, we encountered a bug in Deformetrica regarding the registration process. We decide to look for other software/toolkits, and finally, we opted for Advanced Normalization Tools (ANTSs), the state-of-the-art toolkit for image registration.

In the fifth week, following the advice of our mentor, we downsampled our volumes to 64×64 for testing our approach in a reasonable amount of time. By doing this, we were able to train three 2D networks, one for each plane, and present some results. In the meeting with the stakeholders, we understood that the domain metrics they were interested in are different from what we thought previously. They were not interested in the average dice score but, instead, in the dice score for each of the classes. Furthermore, we started to delve into the documentation of the ANTs toolkit to learn its basic usage of it, with a particular focus on two main functionality: ANTsRegistration, and ANTsApplyTransform.

In the sixth week, we learned that there is a well-defined color code for representing images in neuroimaging, thus we adopted the same color coding for our results. Moreover, concerning the 2D approach, we analyzed the classwise dice scores and understood that each of the planes had a significantly different impact on the performance. Thus, when speaking with our mentor, we were suggested to use an aggregation method to merge the three predictions into a single one, trying to get the best of each network. Moreover, our mentor recommended we increase the resolution of the input to 128×128 to increase the performance of the model we were training. Finally, we discovered some bugs in the reconstruction of the volumes from slices that were affecting performance. During this week we obtained even the first results from the registration process, showing the strength of the ANTs toolkit.

In the seventh week, we fixed the issues regarding the reconstruction of volumes. Moreover, speaking with the stakeholders we learned that some phenomena can happen with hydrocephalus that can have an impact on the way the images are represented. Thus, we have suggested a common workaround used in the field and we rebuilt the dataset from scratch by adopting this workaround. Finally, for our 2.5D model, we were suggested to try training on full-resolution images (i.e., 256×256). For the data augmentation side in the seventh week, we started automatizing the pipeline for the generation of what we define as Synthetically Malformed Brains, obtained using ANTs.

In the eightieth week, we started combining our 2D models into a 2.5D model. Moreover, regarding the 2D models, we attempted a training with 256×256 images but, again, it proved to be not feasible on Colab. We also performed a tuning of the hyperparameters. Furthermore, we prepared the dataset composed of the synthetically malformed brain and the healthy samples, and use it for training a 3D U-Net to perform tissue segmentation.

In the ninth week, we were suggested to use a 3D U-Net on 64-voxels isotropic volumes to have a fair comparison between our 2.5D model and a 3D model. However, this also proved to be not feasible, as it required a training time in the order of $300 - 450$ hours on Google Colab Pro. In the last week, we started the analysis of the results obtained by training the model using the augmented dataset.

The Miro board of the team is available on the following link.

3 Methods

In this section, we will explain the main methods used in this work. Firstly, we will describe how image registration works and how we used it to generate synthetically distorted brains. Then we will show the two different 3D neural networks implemented to tackle the task: the 3D U-Net and the nnUNet. Finally, we will explain how the 2.5D approach works and how we used it in this task.

3.1 Metrics of interest

The metric of interest for our stakeholders, in this challenge, is the dice score.

This score is defined as:

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

and it measures the similarity between the vectors X and Y. This, compared with a similar commonly used metrics in affine fields, Intersection over Union, pays less attention on the outliers in the prediction.

3.2 Image registration

To generate synthetically distorted brain we adopted Advanced Normalization Tools (ANTs), a state-of-the-art medical image registration and segmentation toolkit Avants et al. [2009]. Image registration is the process of estimating a geometrical transformation that aligns an image with another (image to image registration) and can be applied both for 2-D or 3-D images. Generally, the target image is called 'fixed image', while the one that moves towards the former is defined as 'moving image'. The procedure optimize a set of geometrical transformations $\mathbf{T}$ that relate the position of points in the moving image X_m with the corresponding points in the fixed images X_f. Figure 2 show this process for a single point, where the cyan point in the moving image (right) corresponds to the red point in the fixed image (left). Formally speaking we optimize a set of transformation to translate the X_m to the X_f: $T(X_m) = X_f$.

The different type of transformations depends on the purpose of the registration. If the intent is only to bring two images to overlap in the same space while preserving the differences, then a *linear registration* (or global) should be used. Instead, a *non-linear transformation* (or local) should be preferred if the aim is to minimize the difference between the images. Regardless of registration method, image registration algorithm requires the following three components:

1. A transformation model $\mathbf{T}$, which defines the type of mapping performed over the two images.
2. A metric expressing the similarity of the two images.
3. An optimization process that tend to maximize the similarity to improve $\mathbf{T}$

The algorithm, Figure 3, is composed by a sequence of steps in which firstly, the transformation $\mathbf{T}$ is applied over the moving image, and secondly the similarity between the fixed image and the output of the previous step is computed (based on the adopted similarity metric). Finally, the chosen optimizer computes a set of parameters to update the transformation. Those steps are repeated iteratively until a stopping criteria is meet (number of iterations or metric-based limits), giving as output the final learned $\mathbf{T}$ mapping moving to fixed image.

There are different types of possible transformation model, classified by the degree of freedom of the transformation between fixed and moving images.

The simplest method of registration is the translation registration, that as the name suggest, aligns the images by simply translating them. This is one of a broader type of registration called *Rigid registration*, that can handle translation and rotation only between image pairs. Rigid registration can be expanded to model even different scaling and shearing, and it is usually referred to as *affine registration* (rigid transformation is considered a special case of affine transformation). Considering the previous car example, performing a rigid registration between car M and car F allows only to adjust the different rotation of the images, while

(a) Fixed image (car F)

(b) Moving image (car M)

Figure 2: Example of image pairs for which a mapping from point(s) in B to A can be learn through image registration. In particular, it must be noticed that not only the cars' shape is different, but even the orientation.

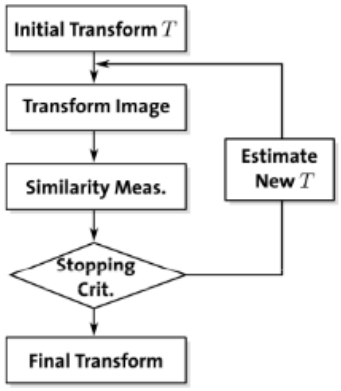


Figure 3: Main steps of a registration algorithm

(a) Car F - fixed image

(b) Car M - moving image

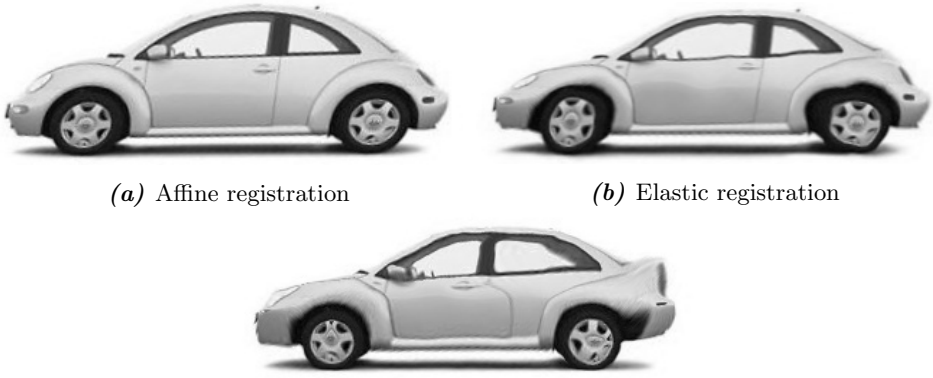
(c) Car M after rigid registration

(d) Car M after affine registration

Figure 4: Rigid/Affine registration example. Car F (b) and Car M (a) are shown before any registration is applied. Car F remains unchanged (fixed image), while car M (moving image) has been rotated by the transformation, showing now the same orientation of the former (c). No differences in cars' shape have been modelled.(d) shows the effects of affine registration, in which not only the rotation has been applied but even a scaling along the x-axis (the Car M appears slightly longer compare to the original image).

by applying an affine registration, the different length of the cars have been captured by the transformation, Figure 4.

Affine mapping is adequate when the difference between images is limited, but this is mostly not the case in the medical imaging field, where the physiological heterogeneity among individuals (inter-subject variability) accounts for structural changes in tissues and organs composition. On top of this, when congenital or acquired pathologies cause complex anatomical alterations, other registration methods must be taken into consideration to deal with such deformations. To catch those local deformations non-linear transformations must be used, among which the most common categories are: *Elastic* and *Diffeomorphic registrations*, with the latters being the state-of-the-art registration techniques in medical imaging in case of large deformation scenarios Figure 5. In general, concatenating registrations with increasing degrees of freedom (from a global alignment to a finer local one) of the same fixed-moving image pair leads to the best results in terms of how accurate is the mapping from moving to fixed image Avants et al. [2009].



(a) Affine registration

(b) Elastic registration

(c) Diffeomorphic registration

Figure 5: 'Effects of different image registration techniques. This example demonstrates the degree of deformation achievable using different registration methodologies, going from Affine (a) to elastic (b), and diffeomorphic registration (c).'

3.3 Augmented dataset generation via image registration

Considering the heterogeneity of the structural disruptions caused by the diseases, and the limited number of samples, a classic statistical model was not feasible. Moreover the induced synthetic deformations had to be medically plausible (in line with the true malformation) and not random. For this reason, a measure of the structural deformations occurring between healthy and diseased brain was necessary to generate synthetically distorted brain samples, composed by the T1w image and the associated tissue segmentation map.

We opted to compute the transformations $\mathbf{T}$ describing such alterations through image registration. As fixed image we used the distorted brain images, while healthy samples have been used as moving image. The pairing was done by age matching distorted-healthy brain images, in order to reduce as much as possible the physiological variability present in brains of different ages of development, and instead modelling only the variations caused by the disease. Pairing was done by finding the closest healthy age match which is not more than three categories away from the distorted being paired (for the age ranges we used the one provided by Amorosino et al. [2022]).

For each distorted-healthy pair, a concatenation of translational, rigid, affine and, diffeomorphic (SyN) Avants et al. [2008] registrations was applied, obtaining as output a transformation matrix and a deformation fields, which approximate the anatomical abnormalities between disease and healthy brains.

The synthetically distorted brain T1-image was finally created by applying the $\mathbf{T}$ transformation model and the deformation field to the healthy T1w-image. The same was applied over the healthy brain tissue segmentation map to obtain the corresponding one (ground truth) for the newly generated synthetically distorted brain. The steps for the generation of the synthetically distorted brain are summarized in Figure 6.

ANTs registration was used to generated 93 synthetically deformed brains by finding 3 matches (without resampling of healthy sample), if available, for each of the 37 distorted brains. In particular, the number

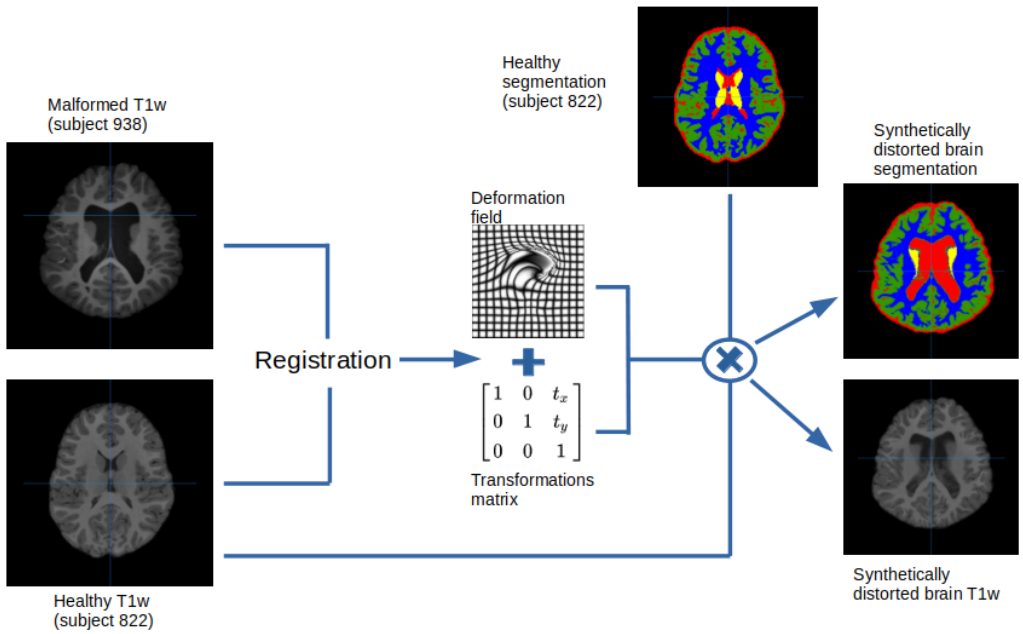


Figure 6: Generation workflow for synthetically distorted brains. For each distorted-healthy matched pair a concatenation of different registrations techniques was applied, obtaining in output the deformation field and transformation matrix describing the structural variations between the two. Those were then applied over the T1w image and tissue segmentation of the same healthy subject to acquire the corresponding for the synthetically distorted brain.

of iterations of each registrations were reduced with respect to what has been identified in the work of Amorosino et al. [2020] in order to prevent generating volumes which are too similar to the target volume. The generated brains are afterwards used as new training samples for different neural network architectures. Our assumption is that such data augmentation will result in anatomically sensible deformations which will enrich the initial dataset and improves the model's segmentation power over the real distorted samples.

The augmented training set is composed out of all healthy subject samples not used in the generation process, $n_h = 121$ and generated distorted samples $n_m = 86$ for a total of 207 training images. To prevent leaking test set information into the training data, we excluded all generated brains where the test set volumes have been used as the registration target. Hence, 86 out of 93 total generated.

As this is just a pilot study to evaluate the segmentation improvement when using synthetic deformations, we reserved only four true samples with the presence of congenital or acquired distortions of brain anatomy for evaluation of our models – one for each of the conditions (sample 932-PFM, 935-ACC, 940-MCDS, 956-HD).

Information about the samples used for training and testing is available on our project's **Github page**.

3.4 Data preprocessing

Before passing the samples to the network (both for training and testing), all images have been preprocessed as follows:

1. Apply a mask over the T1w image.
2. Normalize the voxel values by setting the mean at 100.
3. Due to the fact that the low-intensity CSF is often miss-classified as the background, we set the background value (everything outside the mask) at a fixed value of 2000.
4. Resample all T1w scans to $256 \times 256 \times 256$ isotropic images.

3.5 3D U-Net architecture

3D U-Net, described by Amorosino et al. [2020], Çiçek et al. [2016] is the currently most used whole-brain model, consisting of a contracting encoder part to analyze the whole image, and an expanding decoder to generate full-resolution segmentation, with bottleneck in the middle (both input and output are $256 \times 256 \times 256$ isotropic 3D images;Input images were resampled before training). The encoder part includes 4 block, each with two convolutional layers ($3 \times 3 \times 3$ kernel + ReLU) followed by MaxPooling ($2 \times 2\times$ filter, stride $2 \times 2 \times 2$) for downsampling. The bottleneck consists of 2 convolutional layers with 192 filter ($3 \times 3 \times 3$ kernel + ReLU) each, followed by a dropout layer (dropout rate: 0.15) to prevent overfitting. Finally, the decoder similarly to the encoder, is composed by 4 blocks, each with transposed convolution for upsampling, followed by skipping connections and two convolutions. Multiclass classification (6 tissue + 1 background) is obtained using as last a convolutional layer with $7 \times 1 \times 1$ filters, with a softmax activation at the end. The complete architecture is shown in fig. 7.

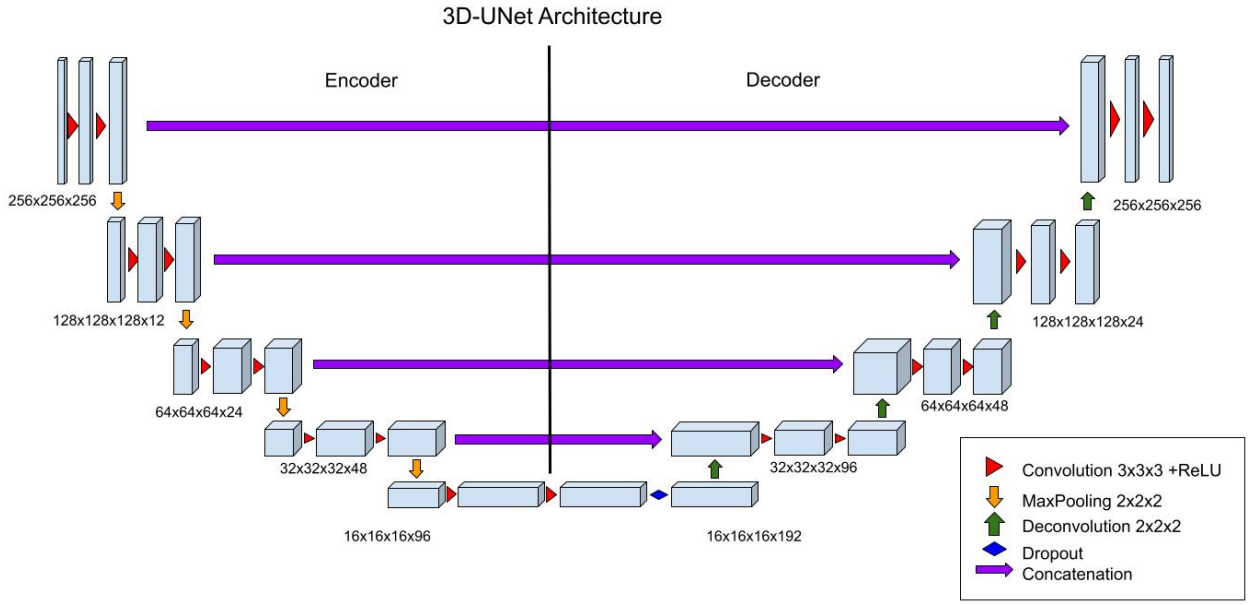


Figure 7: 3D U-Net architecture. The network is composed by three main components: the encoder, the bottleneck, and finally the decoder. The encoder is constituted by 4 block, each composed by two convolutional layer ($3 \times 3 \times 3$ kernel + ReLU), followed by a MaxPooling ($2 \times 2 \times 2$ filter, stride $2 \times 2 \times 2$) for downsampling. The bottleneck presents two convolutional layers with 192 filters ($3 \times 3 \times 3$ kernel + ReLU) each, with a final dropout layer (dropout rate: 0.15) to prevent overfitting. The decoder has the same structure of the encoder, but with deconvolutional layers for upsampling. A last convolutional layer with softmax activation returns the multiclass classification (6 tissue + 1 background).

3.6 NnUnet set of architectures

NnUnet, is not a brand new architecture but a set of self-configuring 3D and 2D architectures operatin on different resolutions. It is a deep learning-based segmentation method that automatically configures itself, including preprocessing, network architecture, training and post-processing for any new task in the medical imaing domain.

NnUNet's automated method configuration starts with the extraction of the dataset fingerprint and subsequent execution of heuristic rules. Based on this fingerprint the preprocessing, postprocessing and the hyperparameters of the following network configurations are determined. By default, nnUNet generates three different U-Net configurations: a two-dimensional (2D) U-Net a 3D U-Net that operates at full image resolution and a 3D U-Net cascade in which the first U-Net operates on downsampled images, and the second is trained to refine the segmentation maps created by the former at full resolution. After cross-validation,

nnUNet empirically chooses the best performing configuration or ensemble. Finally, nnUNet empirically opts for 'non-largest component suppression' as a post-processing step if performance gains are measured. We compared the parfomance of all possible configurations of the nnUNet in a 5-fold cross validation on the traning data. Since the nnUNet toolkit[3] has also the possibility to check the segmentation improvement of using an ensemble of these architectures in the CV process, we were able to identify the best configuration as the ensemble of the 2D U-Net and the 3D full resolution U-Net. Since all three axes are isotropic, the two trailing axes are used for slice extraction in the 2D U-Net. The kernel size for convolutions is $3 \times 3 \times 3$ and 3×3 for 3D U-Net and 2D U-Net, respectively.

3.7 2.5D Approach

The 3D approach described in Amorosino et al. [2022], while very powerful, requires a big amount of resources to be trained. To try to reduce the computational resources needed to train the segmentation models, we proposed a 2.5D approach to segmentation Avesta et al. [2022], Roy et al. [2022] A 2.5D approach makes use of 3 segmentation networks: one for each plane (sagittal, coronal, axial). When processing a 3D volume, we first slice it according to each plane. Then, we feed all the slices for a plane to the corresponding network and get the predicted segmentation (a volume). Finally we use the STAPLE Warfield et al. [2004] algorithm to combine the three segmentations, which performs a weighted voting.

While this approach may seem more costly, because of the fact that it uses three networks, it is important to note that the 3D U-Net makes use of 3D convolutional kernel, each of which contains $O(k^3)$ parameters. Instead, each of the three 2D networks uses only $O(k^2)$ parameters. Thus, this yields an advantage from the point of view of the computational cost for the training process. In fact, from Bienstock et al. [2018], we know that the computational cost of training a 2D CNN is $O(mw^{O(k^2)}/\epsilon)^{(n+m+N)})$, while the computational cost of training a 3D CNN is $O(mw^{O(k^3)}/\epsilon)^{(n+m+N)})$, where m is the number of inputs, n is the number of outputs, N is the total number of parameters, and k is the filter size.

Moreover, we also had another reason for choosing a 2.5D approach. In fact, the reason was not related to the complexity of the training process. Instead, it was mostly due to the small number of samples composing the training set (207 scans). When using a 3D approach, we could only use those 207 samples, which is quite small for a deep learning approach. By using 2D networks, instead, we could use (by downsampling to $64 \times 64 \times 64$ volumes) $64 * 209 = 13376$ images for each of them.

Finally, a last advantage of using 2D networks is that, given the highly diverse images that they are fed with (i.e. a slice from the initial part of the scan and a slice from the central part of the scan represent significantly diverse parts of the brain), they should be less "spatially biased" than a model that always sees data with no high diversity. This, in our hypothesis, should help in being less spatially biased in predicting the scans from the test set, where the spatial bias may be a disadvantage, since the patients' brains may be significantly different than healthy brains.

The architecture of the 2D U-Net used in our experiments is shown below.

```
__________________________________________________________________________________________
 Layer (type)                    Output Shape              Connected to
==========================================================================================
 inputs (InputLayer)             (None, 64, None, 1)       []
 conv_block (ConvBlock)          (None, 64, None, 64)      ['inputs[0][0]']
 max_pooling2d (MaxPooling2D)    (None, 32, None, 64)      ['conv_block[0][0]']
 conv_block_1 (ConvBlock)        (None, 32, None, 128)     ['max_pooling2d[0][0]']
 max_pooling2d_1 (MaxPooling2D)   (None, 16, None, 128)    ['conv_block_1[0][0]']
 conv_block_2 (ConvBlock)        (None, 16, None, 256)     ['max_pooling2d_1[0][0]']
 upconv_block (UpconvBlock)      (None, 32, None, 128)     ['conv_block_2[0][0]']
 crop_concat_block (CropConcatB   (None, 32, None, 256)    ['upconv_block[0][0]',
 lock)                                                      'conv_block_1[0][0]']
 conv_block_3 (ConvBlock)        (None, 32, None, 128)     ['crop_concat_block[0][0]']
```

[3] https://github.com/MIC-DKFZ/nnUNet

```
upconv_block_1 (UpconvBlock)    (None, 64, None, 64)    ['conv_block_3[0][0]']
crop_concat_block_1 (CropConca  (None, 64, None, 128) ['upconv_block_1[0][0]',
tBlock)                                                 'conv_block[0][0]']
conv_block_4 (ConvBlock)        (None, 64, None, 64)    ['crop_concat_block_1[0][0]']
conv2d_72 (Conv2D)              (None, 64, None, 7)     ['conv_block_4[0][0]']
activation_88 (Activation)      (None, 64, None, 7)     ['conv2d_72[0][0]']
outputs (Activation)            (None, 64, None, 7)     ['activation_88[0][0]']

==================================================================================================
Total trainable parameters: 1,862,087
--------------------------------------------------------------------------------------------------
```

3.8 Experimental setup

To test if the introduction of the synthetically distorted brains improved the segmentation, we trained with the augmented dataset a full resolution 3D U-Net model and the current state-of-the-art configuration: nnUNet, while on the same, downsampled images a 2.5D model was trained. In all three approaches we trained the networks using the augmented dataset described in the subsection 3.3 (121 healthy brain images, 86 synthethically malformed brains) with the same preprocessing steps descibed in 3.4. As the baseline, only 3D U-Net model is trained using only 207 healthy individuals.

3.8.1 3D U-Net training

The 3D U-Net model's training (with and without data augmentation) was performed on a FBK's machine with 94GB of RAM using an NVIDIA GeForce GTX 1080TI with 11Gb VRAM for 29 epochs i.e. until a plateau in dice score over the validation set was obtained (10% of the training data). Cross-entropy was used as the loss function while the learning rate was set at 10^{-4} with a single image randomly taken from training set for each batch. The model was implemented with TensorFlow 1.8.0.

3.8.2 NnUNet training

The first step was to identify the configuration of nnUNet best suited for this segmentation problem. Five-fold cross-validation, using the training data only, was performed in parallel on a 6-core Intel Xeon(R) CPU (3.6GH), 94GB of RAM and using an NVIDIA GeForce GTX 1080TI. After identifying the best configuration as the ensemble comprised of a 2D U-Net and a full resolution 3D, we trained both these models on the full training set on the same FBK's machine for 75 epoch i.e. until we observed a convergence in dice score over the validation set. The validation set in this case was 30 % of the data. For this task, a pre-implemented toolkit was retrieved from GitHub and used [4]. All other hyper-parameters were kept at default values.

3.8.3 2.5D Net training

The slicing (i.e., the transformation of the dataset from 3D to 2D) of the dataset was performed on a workstation with an Intel i9-7940X CPU (14 cores/24 threads @ 3.10GHz) and 62.5GB of RAM. The training, instead, was initially performed on Google Colab Pro with a Premium GPU but, after a little time, we were downgraded to a Standard GPU because we ran out of computational resources. The networks were implemented in TensorFlow v2.9.2 using an existing library[5]. The merging of the volumes with STAPLE[6] (along with the computation of the dice scores for the 2D networks and the 2.5D model) was performed on the same machine used for the slicing.

We trained each 2D model for 30 epochs, with a batch size of 16 images, using a learning rate of 10^{-4} using the Categorical CrossEntropy loss.

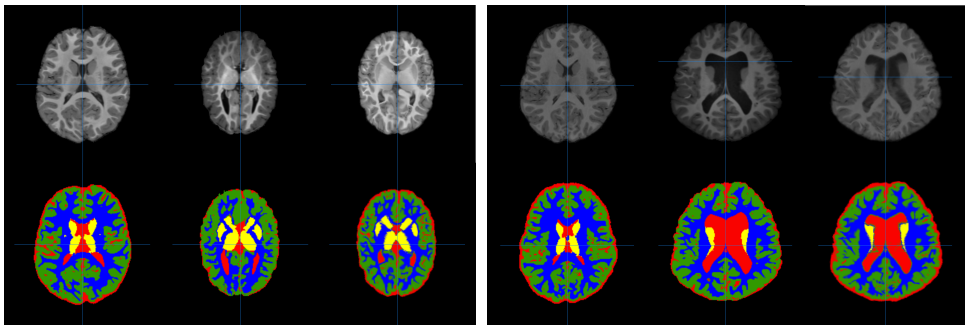
[4] https://github.com/MIC-DKFZ/nnUNet
[5] https://github.com/jakeret/unet
[6] from the SimpleITK package

4 Results

In this section, we describe the results obtained with the methods described in Section 3 and a comparison w.r.t. the baseline model.

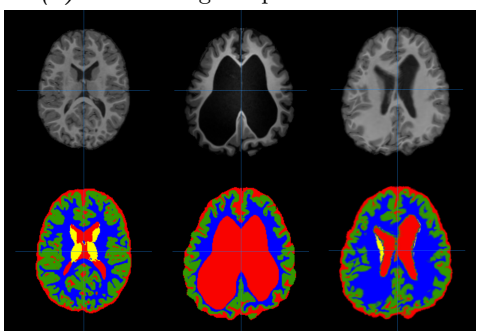
4.1 Synthetically distorted brains

Figure 8 shows the healthy, distorted and the synthetically distorted brains generated using the previous two (left to right). Generated ACC and PFM samples are able quite well to mimic their conditions, while in the case of HD, only partially. However, the inability to mimic anatomical changes in these distortions is not necessarily related to not improving the training process. We reduced the number of registration iterations on purpose to avoid having deformations too similar to the target image – obtaining a somewhat intermediate deformation stage between the healthy and the distorted brain is what we are aiming for. Additionally, we can use multiple healthy samples with the same distorted target brain – the naturally occurring variations in the healthy brains are what guarantees a different result.

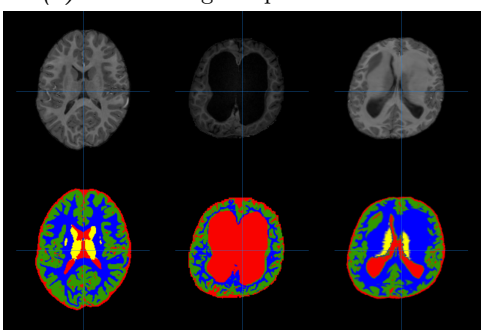


(a) ACC: using samples 872 and 934

(b) PFM: using samples 822 and 938



(c) HD: using samples 862 and 957



(d) HD: using samples 876 and 954

Figure 8: Genereated volumetric images (right) from a single deformed (middle) and a matched healthy brain (left). The second row represents the corresponding labels of the axial slice (red – CSF, green – GM, blue – WM, yellow – DGM), light blue – BS, pink – CER).

4.2 Data augmentation effects on segmentation results

All three approaches utilize the augmented dataset described in the subsection 3.3 (121 healthy brain images, 86 synthethically malformed brains) and the same preprocessing steps. As the baseline, only 3D U-Net results are reported using only 207 healthy individuals.

4.2.1 Full resolution 3D U-Net and ensemble nnUnet segmentation

Table 2 shows the baseline results of the 3D U-Net trained using healthy subject from the EMEDEA-PED dataset (207 individuals), as described in Amorosino et al. [2022].

Class	ACC	HD	MCDS	PFM
CSF	0.54	0.65	0.60	0.39
GM	0.77	0.38	0.82	0.77
WM	0.71	0.47	0.79	0.77
DGM	0.007	0.12	0.24	0.42
BS	0.45	0.24	0.50	0.68
CER	0.83	0.6	0.80	0.81

Table 2: Baseline dice score obtained with the 3D U-Net model trained on healthy subjects only with 29 epochs. The test set is composed by one subject for each condition (ACC, HD, MCDS, PFM), and accuracy expressed in terms of dice score is shown for each segmented tissue.

Instead, in Table 3 are presented the results of the same architecture, but trained with the augmented dataset. We can clearly see that in general the accuracy for all the tissues across different pathologies is increased.

Class	ACC	HD	MCDS	PFM	Mean gain w.r.t baseline
CSF	0.69	0.86	0.63	0.58	0.14
GM	0.82	0.71	0.81	0.84	0.1
WM	0.79	0.66	0.83	0.85	0.09
DGM	0.17	0.10	0.40	0.65	0.14
BS	0.71	0.29	0.82	0.81	0.19
CER	0.94	0.74	0.92	0.94	0.12
Mean gain w.r.t baseline	0.13	0.15	0.11	0.11	Global mean gain=0.13

Table 3: Dice score obtained with the 3D U-Net model trained on healthy subjects and synthetically distorted brain on 29 images. The last column and row denote the mean increase (or loss) of dice score across tissues and pathologies when using the augmented data w.r.t. using only healthy subjects.

Next we compared the results of the 3D U-Net with the ensemble configuration of nnUNet which has been identified as best in the 5-fold cross-validation procedure. This ensemble is composed of a 2D axial slice U-Net and a full resolution 3D U-Net. The results are shown in Table 4 only for the augmented dataset. Only two brain tissues show drastic improvement: deep gray matter (DGM) and the brain stem (BS).

Class	ACC	HD	MCDS	PFM	Mean gain w.r.t 3D U-Net
CSF	0.66	0.78	0.65	0.62	-0.01
GM	0.78	0.70	0.80	0.83	-0.01
WM	0.79	0.69	0.83	0.84	0.007
DGM	0.54	0.01	0.73	0.75	0.17
BS	0.80	0.64	0.85	0.82	0.12
CER	0.95	0.87	0.96	0.93	0.05
Mean gain w.r.t 3D U-Net	0.07	0.05	0.07	0.02	Global mean gain=0.06

Table 4: Dice score obtained with the nnUNet's 2D and 3D U-Net ensemble trained on healthy subjects and synthetically distorted brain. The last column and row denote the mean increase (or loss) of dice score across tissues and pathologies when using the ensemble configuration w.r.t. 3D U-Net (both on augmented data).

As the number of available samples is low, to draw a complete conclusion on improving the segmentation using our data augmentation protocol, a cross validation should be applied. However, due to the time limits of the challenge and the lengthy training time, a single sample for each condition was chosen. Even if preliminary, these results looks promising, showing that our data augmentation using synthetically malformed brains increased the accuracy of segmentation w.r.t the baseline model in all tissues and all pathologies. Segmentation dice score of each brain improved by 0.13 on average.
When comparing different configuration with the usage of augmented data the best result was obtained using the nnUNet's ensemble of 2D and 3D U-Nets which is yet another improvement. Furthermore, even for deep grey matter (DGM), known to be the most difficult tissue to correctly segment, we saw relevant improvement when both introducing the augmented data and improving the nerual net architecture.

4.2.2 Downsampled 2.5D Net segmentation

In this subsection we will show the performance of the individuals 2D networks and their combination using the STAPLE algorithm.

For this set of results, we used volumes downsampled to 64 voxels (isotropically). In fact, in order to adapt the task to our resources (i.e., Colab Pro), we could not use scans of 128 or 256 (i.e., the scan size used in Amorosino et al. [2022]) as they would require too much memory (RAM and/or GPU memory). We could solve this issue implementing a smarter data loader. However, this would increase training time even more, which we could not afford given the challenge timeline.

The results (in terms of dice scores) are shown in Tables 5 to 8. In these tables we can easily observe that STAPLE is able to (approximately) keep the performance of the best of the three networks in the given case. This means that, starting from 3 different networks, where each of which has different points of strength and points of weakness, can keep the performance of the best network discarding the others.

Finally, a very interesting point of this set of experiments is that these settings (2D + 2.5D) are the only ones that we were able to run on Colab Pro, as even training a 3D U-Net on 64 isotropic volumes would require from 90 to 135 hours of continuous training time. Instead, each 2D network took about 0.5 hours to train, resulting in significantly lower carbon emissions, lower RAM consumption, and smaller training time.

Of course, a fair comparison would require a comparison with a 3D U-Net trained on 64-voxels isotropic scans. However, given that the training time was not feasible on Colab Pro and that we could not run these experiments on the FBK hardware, we can only compare it to the 3D U-Net trained on 256-voxels isotropic scans. By comparing the results obtained with this approach to those of Table 3, we can observe that the 2.5D model underperforms w.r.t. the full resolution 3D U-Net. This may be due to the fact that, due to downsampling, the images lost some of the details that were necessary for the network to recognize some specific areas, e.g., Deep Gray Matter. However, in other classes, the performance are in the same order of magnitude (even though smaller) than those of the full-resolution 3D U-Net. This suggests that this approach, scaled to full scans (i.e. 256-voxels isotropic scans) may retain the performance of the 3D U-Net while requiring significantly less training effort, which in turns leads to faster iterations between models (e.g., for hyperparameter tuning) and better scaling to larger datasets.

Class	ACC	HD	MCDS	PFM
CSF	0.4069	0.7004	0.5169	0.4003
GM	0.1804	0.1317	0.2061	0.1788
WM	0.1882	0.2315	0.2955	0.2312
DGM	0.0000	0.0000	0.0000	0.0000
BS	0.0000	0.0000	0.0000	0.0000
CER	0.1841	0.1342	0.1148	0.1749

Table 5: Results obtained with the 2D model trained on the sagittal plane.

Class	ACC	HD	MCDS	PFM
CSF	0.6154	0.6219	0.7353	0.5928
GM	0.6494	0.3700	0.6501	0.5429
WM	0.6962	0.4400	0.6684	0.5956
DGM	0.0145	0.0338	0.0050	0.1360
BS	0.7310	0.1240	0.6055	0.6738
CER	0.9320	0.4155	0.9065	0.5862

Table 6: Results obtained with the 2D model trained on the coronal plane.

Class	ACC	HD	MCDS	PFM
CSF	0.4355	0.5514	0.6560	0.4753
GM	0.5050	0.2887	0.5047	0.4552
WM	0.6451	0.4525	0.6144	0.5796
DGM	0.0000	0.0020	0.0446	0.0573
BS	0.6984	0.3939	0.6322	0.5238
CER	0.9183	0.6898	0.9007	0.8367

Table 7: Results obtained with the 2D model trained on the axial plane.

Class	ACC	HD	MCDS	PFM
CSF	0.6276	0.6801	0.7462	0.6103
GM	0.6486	0.3945	0.6262	0.5845
WM	0.6968	0.4767	0.6606	0.6264
DGM	0.0145	0.0196	0.0269	0.1120
BS	0.7435	0.1569	0.6945	0.5718
CER	0.9282	0.6827	0.9094	0.8378

Table 8: Results obtained with the 2.5D model.

5 Discussion and future ideas

In this report, we presented the work that we have done in this competition in order to improve performance in distorted brain tissue segmentation.

Our strategy is three-fold: on the one hand, we prove it's possible to improve the segmentation performance w.r.t. the baseline model trained on only healthy brain images by creating synthetic images that replicate conditions in line with the pathologies of interest using our registration protocol. On the baseline architecture this lead to segmentation improvements across all pathologies and all tissues for all full resolution models.

Secondly, we identified a new configuration that further improves the results of the baseline architecture. By utilizing augmented data and an full resolution ensamble nnUNet model composed out of 2D U-Net and a 3D U-Net, we were able to further improve the segmentation of deep gray matter and the brain stem by 17% and 12%, respectively.

Finally, we reduce the computational cost of the training by proposing a 2.5D approach that reduces the training time of about 98.3%. However, due to the lack of time, we were only able to test the 2.5D approach in a downsampled scenario.

Future work includes: the scaling of the 2.5D approach to the full-resolution scenario; the creation of an ensemble of different models; and evaluating our approach on a larger dataset in a cross-validation mode.

5.1 Disclaimer

At the start of the competition, our plan was to use the computational resources provided by the challenge – Google Colab Pro. We came across multiple difficulties while using this working environment for a computationally heavy deep learning task:

1. The training process of deep neural networks used in medical imaging is extremely lengthy in general. The fact that Google Colab Pro does not allow closing the browser makes this task impossible when using the full resolution images.

2. The size of our full resolution dataset is 200GB and in each new session of Google Colab Pro it has to be uploaded to the disk again.

3. RAM limitations when using Google Colab Pro in high performance mode is 32GB.

4. All the available computational units (assigned to each account) become exhausted after around 15 hours of training.

It was useful to prototype a network with downsampled data and a lightweight model. However, after exhausting all computational units on our assigned account and in order to create a real-life applicable model we were forced to move our project on a high performance machine provided by FBK which allowed us perform full resolution off-line training and much longer training time without any resource constraints.

6 Data availability

To make our code available for further development, we published our files on the GitHub Repo for our project.

References

Gabriele Amorosino, Denis Peruzzo, Pietro Astolfi, Daniela Redaelli, Paolo Avesani, Filippo Arrigoni, and Emanuele Olivetti. Automatic tissue segmentation with deep learning in patients with congenital or acquired distortion of brain anatomy. In Seyed Mostafa Kia, Hassan Mohy-ud Din, Ahmed Abdulkadir, Cher Bass, Mohamad Habes, Jane Maryam Rondina, Chantal Tax, Hongzhi Wang, Thomas Wolfers, Saima Rathore, and Madhura Ingalhalikar, editors, *Machine Learning in Clinical Neuroimaging and Radiogenomics in Neuro-oncology*, pages 13–22, Cham, 2020. Springer International Publishing. ISBN 978-3-030-66843-3.

Gabriele Amorosino, Denis Peruzzo, Daniela Redaelli, Emanuele Olivetti, Filippo Arrigoni, and Paolo Avesani. Dbb-a distorted brain benchmark for automatic tissue segmentation in paediatric patients. *NeuroImage*, 260:119486, 2022.

B.B. Avants, C.L. Epstein, M. Grossman, and J.C. Gee. Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. *Medical Image Analysis*, 12(1):26–41, 2008. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2007.06.004. URL https://www.sciencedirect.com/science/article/pii/S1361841507000606. Special Issue on The Third International Workshop on Biomedical Image Registration – WBIR 2006.

Brian B Avants, Nick Tustison, Gang Song, et al. Advanced normalization tools (ants). *Insight j*, 2(365): 1–35, 2009.

Arman Avesta, Sajid Hossain, MingDe Lin, Mariam Aboian, Harlan M. Krumholz, and Sanjay Aneja. Comparing 3d, 2.5d, and 2d approaches to brain image segmentation. *medRxiv*, 2022. doi: 10.1101/2022.11.03.22281923. URL https://www.medrxiv.org/content/early/2022/11/04/2022.11.03.22281923.

Daniel Bienstock, Gonzalo Muñoz, and Sebastian Pokutta. Principled deep neural network training through linear programming, 2018. URL https://arxiv.org/abs/1810.03218.

Jacky Bolvin and Deborah Lancastle. Medical waiting periods: Imminence, emotions and coping. *Women's Health*, 6(1):59–69, 2010. doi: 10.2217/WHE.09.79. URL https://doi.org/10.2217/WHE.09.79.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation. In Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pages 424–432, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46723-8.

Fabian Isensee, Jens Petersen, Andre Klein, David Zimmerer, Paul F. Jaeger, Simon Kohl, Jakob Wasserthal, Gregor Koehler, Tobias Norajitra, Sebastian Wirkert, and Klaus H. Maier-Hein. nnu-net: Self-adapting framework for u-net-based medical image segmentation, 2018. URL https://arxiv.org/abs/1809.10486.

Laurent Itti, Linda Chang, and Thomas Ernst. Automatic scan prescription for brain mri. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 45(3):486–494, 2001.

David B. Portnoy. Waiting is the hardest part: Anticipating medical test results affects processing and recall of important information. *Social Science & Medicine*, 71(2):421–428, 2010. ISSN 0277-9536. doi: https://doi.org/10.1016/j.socscimed.2010.04.012. URL https://www.sciencedirect.com/science/article/pii/S0277953610003308.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

Saikat Roy, David Kügler, and Martin Reuter. Are 2.5d approaches superior to 3d deep networks in whole brain segmentation? In *Medical Imaging with Deep Learning*, 2022. URL https://openreview.net/forum?id=Ob62JPB_CDF.

Simon K Warfield, Kelly H Zou, and William M Wells. Simultaneous truth and performance level estimation (staple): an algorithm for the validation of image segmentation. *IEEE transactions on medical imaging*, 23(7):903—921, July 2004. ISSN 0278-0062. doi: 10.1109/tmi.2004.828354. URL https://europepmc.org/articles/PMC1283110.